

## Utilizing dynamic context semantics in smart behavior of informing cyber-physical systems

Li, Yongzhe

**DOI**

[10.4233/uuid:c4db06fd-30f6-419f-a05f-bf6fbb76a421](https://doi.org/10.4233/uuid:c4db06fd-30f6-419f-a05f-bf6fbb76a421)

**Publication date**

2019

**Document Version**

Final published version

**Citation (APA)**

Li, Y. (2019). *Utilizing dynamic context semantics in smart behavior of informing cyber-physical systems*. [Dissertation (TU Delft), Delft University of Technology]. <https://doi.org/10.4233/uuid:c4db06fd-30f6-419f-a05f-bf6fbb76a421>

**Important note**

To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

# **Utilizing dynamic context semantics in smart behavior of informing cyber-physical systems**

## **Dissertation**

For the purpose of obtaining the degree of doctor  
at Delft University of Technology  
by the authority of the Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen;  
Chair of the Board for Doctorates  
to be defended publicly on  
Tuesday, 8 October 2019 at 10:00 o'clock

by

**Yongzhe LI**

Master of Engineering in Materials Processing Engineering  
Harbin Institute of Technology, P. R. China  
born in Daqing, P. R. China

This dissertation has been approved by the  
promotor: Prof. dr. I. Horváth  
and the co-promotor: Dr. Z. Rusák

Composition of the doctoral committee:

Rector Magnificus	chairperson
Prof. dr. I. Horváth	Delft University of Technology, promotor
Dr. Z. Rusák	Delft University of Technology, co-promotor

Independent members:

Prof. dr. Y. Nagai	Japan Advanced Institute of Science and Technology, Japan
Prof. dr.-ing. I. Gräßler	University of Paderborn, Germany
Prof. dr. Ir. B. Tekinerdogan	Wageningen University, the Netherlands
Prof. dr. ir. arch. I.S. Sariyildiz	Delft University of Technology, the Netherlands
Prof. dr. H. de Ridder	Delft University of Technology, the Netherlands

Reserve member:

Prof. dr. P. Vink	Delft University of Technology, the Netherlands
-------------------	---

This Ph.D. research was funded by China Scholarship Council (CSC).

Utilizing dynamic context semantics in smart behavior of informing cyber-physical systems

Keywords: Informing cyber-physical systems; context information representation; dynamic context management; semantic inference; indoor fire evacuation application

Ph.D. thesis  
Delft University of Technology, Delft, the Netherlands  
ISBN: 978-94-6384-019-4  
Copyright © Yongzhe Li.  
All rights reserved

# Table of contents

<b>Chapter 1: Introduction</b>	<b>1</b>
<b>1.1 Background of research</b>	<b>1</b>
1.1.1. <i>Setting the stage</i>	1
1.1.2. <i>The paradigm of cyber-physical systems</i>	2
1.1.3. <i>Characteristics and application potentials of CPSs</i>	5
1.1.4. <i>Implementation approaches of CPSs</i>	7
<b>1.2 Towards next generation cyber-physical systems</b>	<b>8</b>
1.2.1. <i>Main enablers of evolution</i>	8
1.2.2. <i>On the way to smart cyber-physical systems and beyond</i>	10
1.2.3. <i>Open research issues concerning smart CPSs</i>	13
<b>1.3 Introducing the specific research domain</b>	<b>15</b>
1.3.1. <i>Context management and related issues</i>	15
1.3.2. <i>Towards context-aware computing</i>	16
1.3.3. <i>Processing of dynamic context information</i>	17
1.3.4. <i>Informing cyber-physical systems</i>	18
<b>1.4 Research objectives, challenges, and questions</b>	<b>20</b>
1.4.1. <i>Research objectives</i>	20
1.4.2. <i>Research challenges</i>	21
1.4.3. <i>Research questions</i>	22
<b>1.5 Methodological framing of the research approach</b>	<b>23</b>
<b>1.6 Structure of the thesis</b>	<b>26</b>
<b>1.7 Related own publications</b>	<b>27</b>
<b>1.8 References</b>	<b>28</b>
<b>Chapter 2: Aggregation of knowledge and exploring requirements about dynamic context information processing</b>	<b>35</b>
<b>2.1. Objectives and methodology of research cycle 1</b>	<b>35</b>
2.1.1. <i>Objectives of this research cycle</i>	35
2.1.2. <i>Methodology applied in this research cycle</i>	36
<b>2.2. Knowledge aggregation for computation of dynamic context information</b>	<b>36</b>
2.2.1. <i>Reasoning model of the survey</i>	36
2.2.2. <i>Approaches to representing dynamic context</i>	39
2.2.3. <i>Approaches to building awareness in dynamic context</i>	42
2.2.4. <i>Approaches to reasoning with dynamic context</i>	46
2.2.5. <i>Approaches to context-dependent messaging</i>	48
2.2.6. <i>Software enablers for dynamic context information processing</i>	53
2.2.7. <i>Major findings</i>	58
<b>2.3. Exploring requirements for dynamic context information processing</b>	<b>61</b>
2.3.1. <i>Objectives of requirement engineering</i>	61
2.3.2. <i>Exploring requirements based on smart system development</i>	63
2.3.3. <i>Exploring requirements for dynamic context management by I-CPSs</i>	64
2.3.4. <i>Exploring requirements based on real-life I-CPs applications</i>	67

2.3.5. <i>Exploring requirements based on a case study</i>	70
2.3.6 <i>Consistency checking of the aggregated requirements</i>	72
<b>2.4. Conclusions</b>	74
2.4.1. <i>Conclusions concerning the knowledge aggregation</i>	74
2.4.2. <i>Conclusions concerning the requirements exploration</i>	75
2.4.3. <i>Possible research and development oppprtuiities</i>	76
<b>2.5. References</b>	77

**Chapter 3: Conceptualization of computational mechanisms for dynamic context information processing** **97**

---

<b>3.1. Objectives and methodological framing of research cycle 2</b>	97
3.1.1. <i>Objectives of this research cycle</i>	97
3.1.2. <i>Methodology applied in this research cycle</i>	98
<b>3.2. Theoretical fundamentals</b>	98
3.2.1. <i>A semantically enriched model for managing context information</i>	98
3.2.2. <i>Inferring semantic knowledge from dynamic context</i>	102
3.2.3. <i>Interpretation of dynamic context with regard to one entity</i>	106
<b>3.3. A model for representing spatial, attributive and temporal context data</b>	109
3.3.1. <i>The spatial feature representation matrix</i>	109
3.3.2. <i>The context information reference cube</i>	111
<b>3.4. Specification of the functionalities of the computational mechanisms</b>	113
3.4.1. <i>Functionalities for dynamic context infoemation representation</i>	114
3.4.2. <i>Functionalities for inferring semantic knowledge from dynamic context</i>	114
3.4.3. <i>Functionalities for devising action-plans for people</i>	122
3.4.4. <i>Functionalities for messaging peiple about context and action plans</i>	125
<b>3.5. Devising the architecture and workflow of the computational mechanism</b>	127
3.5.1. <i>Architecture</i>	127
3.5.2. <i>Workflow</i>	128
3.5.3. <i>Discussion</i>	132
<b>3.6. Algorithm-level specification of the DCIP-M for indoor fire evacuation guiding</b>	133
3.6.1. <i>Algorithm-level specification of IRM</i>	133
3.6.2. <i>Algorithm-level specification of KIM</i>	135
3.6.3. <i>Algorithm-level specification of ADM</i>	137
3.6.4. <i>Algorithm-level specification of MCM</i>	139
<b>3.7. Concluding remarks</b>	140
<b>3.8. References</b>	141

**Chapter 4: Implementation of a prototype computational mechanism, and testing its functionality and performance** **145**

---

<b>4.1. Objectives and methodological approach of research cycle 3</b>	145
4.1.1. <i>Objectives and activities</i>	145
4.1.2. <i>Methodology applied in this research cycle</i>	146
<b>4.2. Fundamentals for the implementation</b>	146

4.2.1. <i>Digital representation of a concerned physical space</i>	146
4.2.2. <i>Implementation of a computational simulation of fire evacuation scenarios</i>	149
<b>4.3. Implementation of the modules</b>	152
4.3.1. <i>The information representation module</i>	152
4.3.2. <i>The context knowledge inferring module</i>	156
4.3.3. <i>The personal action-plan deriving module</i>	161
4.3.4. <i>The personalized message construction module</i>	168
<b>4.4. Implementation of the prototype</b>	172
4.4.1. <i>Procedural integration of the modules</i>	172
4.4.2. <i>Specification of the input and output couplings of the algorithms included in the prototype</i>	174
<b>4.5. Functionality testing of the prototype in the context of an IFEG System</b>	177
4.5.1. <i>Testing the information representation module</i>	178
4.5.2. <i>Testing the context knowledge inferring module</i>	182
4.5.3. <i>Testing the action-plan deriving module</i>	189
4.5.4. <i>Testing the message construction module</i>	193
4.5.5. <i>Testing the prototype as a whole</i>	198
<b>4.6. Performance testing of the prototype</b>	202
4.6.1. <i>Objectives of the simulation-based performance testing</i>	202
4.6.2. <i>Setting the targets for the performance testing</i>	203
4.6.3. <i>Testing the empirical performance of the prototype under varied loads of computation</i>	203
4.6.4. <i>Testing the empirical performance of the prototype under dynamic loads of computation in run-time</i>	208
<b>4.7. Reflections and conclusions</b>	210
4.7.1. <i>Reflection on the implementation work</i>	210
4.7.2. <i>Findings from the functionality testing</i>	211
4.7.3. <i>Findings from the performance testing</i>	213
4.7.4. <i>Reflections on some recognized limitations</i>	214
4.7.5. <i>Conclusions considering all findings</i>	215
<b>4.8. References</b>	217
<b>Chapter 5: Applicability validation of the computational mechanisms</b>	<b>219</b>
<hr/>	
<b>5.1. The objectives and approach of validation</b>	219
5.1.1. <i>The research objectives</i>	219
5.1.2. <i>The application leading the development of the mechanisms</i>	220
5.1.3. <i>The methodological approach to validation</i>	221
5.1.4. <i>An adapted validation square approach</i>	223
<b>5.2. The considered other applications</b>	226
5.2.1. <i>Road traffic management system</i>	226
5.2.2. <i>Home caretaking assistant system</i>	228
5.2.3. <i>Real-time football-play coaching system</i>	229
5.2.4. <i>Comparison of the concerned target applications</i>	231
<b>5.3. Analysis of the relevance of the theoretical structure of the DCIP-M</b>	233
5.3.1. <i>Relevance of the functionalities to the RTMS</i>	233

5.3.2. <i>Relevance of the functionalities to the HCAS</i>	236
5.3.3. <i>Relevance of the functionalities to the RFCS</i>	238
5.3.4. <i>Relevance of the workflow to the RTMS</i>	236
5.3.5. <i>Relevance of the workflow to the HCAS</i>	239
5.3.6. <i>Relevance of the workflow to the RFCS</i>	240
<b>5.4. Analysis of the suitability of the empirical structure of the computational mechanisms</b>	240
5.4.1. <i>Suitability of the algorithms to the RTMS</i>	243
5.4.2. <i>Suitability of the algorithms to the HCAS</i>	244
5.4.3. <i>Suitability of the algorithms to the RFCS</i>	245
5.4.4. <i>Suitability of the data constructs to the RTMS</i>	246
5.4.5. <i>Suitability of the data constructs to the HCAS</i>	246
5.4.6. <i>Suitability of the data constructs to the RFCS</i>	247
<b>5.5. Analysis of the empirical performance efficiency of the computational mechanisms</b>	247
5.5.1. <i>Efficiency of the algorithms in the RTMS application</i>	249
5.5.2. <i>Efficiency of the algorithms in the HCAS application</i>	250
5.5.3. <i>Efficiency of the algorithms in the RFCS application</i>	250
5.5.4. <i>Effects of the influencing factors on the performance in the RTMS application</i>	250
5.5.5. <i>Effects of the influencing factors on the performance in the HCAS application</i>	251
5.5.6. <i>Effects of the influencing factors on the performance in the RFCS application</i>	251
<b>5.6. Analysis of the theoretical performance sufficiency of the computational mechanisms</b>	252
5.6.1. <i>Revisiting the theoretical performance targets</i>	252
5.6.2. <i>Achievement of the overall performance targets in the RTMS application</i>	254
5.6.3. <i>Achievement of the overall performance targets in the HCAS application</i>	255
5.6.4. <i>Achievement of the overall performance targets in the RFCS application</i>	257
<b>5.7. Some reflections on the findings and approach of validation</b>	258
5.7.1. <i>Reflections on the approach of applicability validation</i>	258
5.7.2. <i>Conclusions</i>	259
5.7.3. <i>Future research opportunities</i>	260
<b>5.8. References</b>	261
<b>Chapter 6: Overall conclusions and recommendations</b>	<b>265</b>
<hr/>	
<b>6.1. Reflections on the work and the results</b>	265
6.1.1. <i>Research cycle 1</i>	265
6.1.2. <i>Research cycle 2</i>	269
6.1.3. <i>Research cycle 3</i>	272
6.1.4. <i>Research cycle 4</i>	274
<b>6.2. Propositions</b>	277
6.2.1. <i>Scientifically-based propositions</i>	277
6.2.2. <i>Socially flavored propositions</i>	281
6.2.3. <i>Self-reflective propositions</i>	281

<b>6.3. Possible future research work</b>	282
6.3.1. <i>Short-term follow up research</i>	282
6.3.2. <i>Long-term research opportunities</i>	282
<b>List of figures</b>	<b>285</b>
<hr/>	
<b>List of tables</b>	<b>289</b>
<hr/>	
<b>Abbreviations</b>	<b>291</b>
<hr/>	
<b>Summary</b>	<b>293</b>
<hr/>	
<b>Samenvatting</b>	<b>297</b>
<hr/>	
<b>Acknowledgements</b>	<b>303</b>
<hr/>	
<b>Biography</b>	<b>306</b>
<hr/>	





# Chapter 1

## Introduction

### 1.1. Background of research

#### 1.1.1. Setting the stage

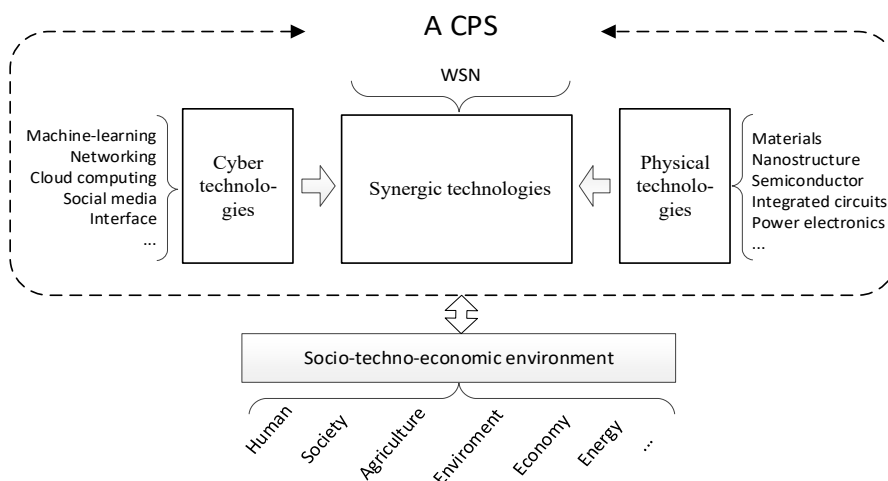
Everything is changing! This is not a new observation at all. However, it is a fact that computational handling of dynamically changing of contexts and the consideration of rapidly changing situations in awareness building, situated reasoning, and proactive adaptation of smart cyber-physical systems (CPSs) looks like still water. Less attention than it deserves was paid to these issues. Concerning the development of next-generation CPSs, they have just recently been recognized as important research phenomena that call for further intense studies. The main reason is that smart control (self-\* behavior) of many real-life processes requires quasi-real-time processing of context information. Though processing time-varied context information has been addressed in the literature, domain-independent solutions for reasoning about dynamic activity scenarios are scarce. Some related issues have been addressed in research focusing on safety-critical dynamic systems. The context was typically interpreted as a body of information dynamically created by a pattern of entities and relationships over a history of situations. Considering these, it is not an exaggeration to claim that explicit generation and utilization of dynamic context semantics as an enabler of the smart behavior of CPSs is a frontier endeavor.

Our overall objective was to make using dynamically changing context information in decision-making by application-specific smart CPSs possible. Towards this end, we needed purposeful representation and handling techniques for dynamically changing context information that allowed reducing the time of information input and computational processing, in harmony with critical short-time happenings. Our specific goals were (i) to provide an effective representation and a processing methodology for predefined kinds of dynamic context data, (ii) to enrich context information with derived semantics, (iii) to

address real-life applications that need reasoning and decision making for development of some sort of action plans, and eventually, (iv) to realize a dynamic context information processing mechanism (DCIP-M) as a kernel component of a reasoning platform for informing cyber-physical systems (I-CPSs). During the promotion research project, we used both formal (theoretical) knowledge and tacit (pragmatic) knowledge related to practical application cases. In addition to studying the state of the art and the theoretical progression in various aspects of the work, we used a so-called reference case to elicit empirical knowledge and to reflect on the properness and appropriateness of the ideas. The reference case was an indoor fire evacuation guiding application. This will be discussed extensively in the next Chapter.

### 1.1.2. The paradigm of cyber-physical systems

The notion of CPSs grew out from a number of existing disciplines such as embedded control systems, collaborative multi-agent systems, advanced robotics, artificial reasoning and learning, multi-scale biological systems, and socio-technical systems. CPSs are regarded as a model of future software-integrated systems that are able to interact with real-life processes and to deeply penetrate into them [1-2]. Understanding complicated CPSs means being aware of the logical, temporal and causal links established among (i) the observable inputs, (ii) the particular states of the system, and (iii) the observable outputs. It also means having mental models that describe, explain, and forecast the manifestation and behavior of the developed system in design time, implementation time, and operation time. The operations of CPSs are realized based on an integrated computation, communication, and control (3C) infrastructure, in which a large number of system resources, such as sensors,



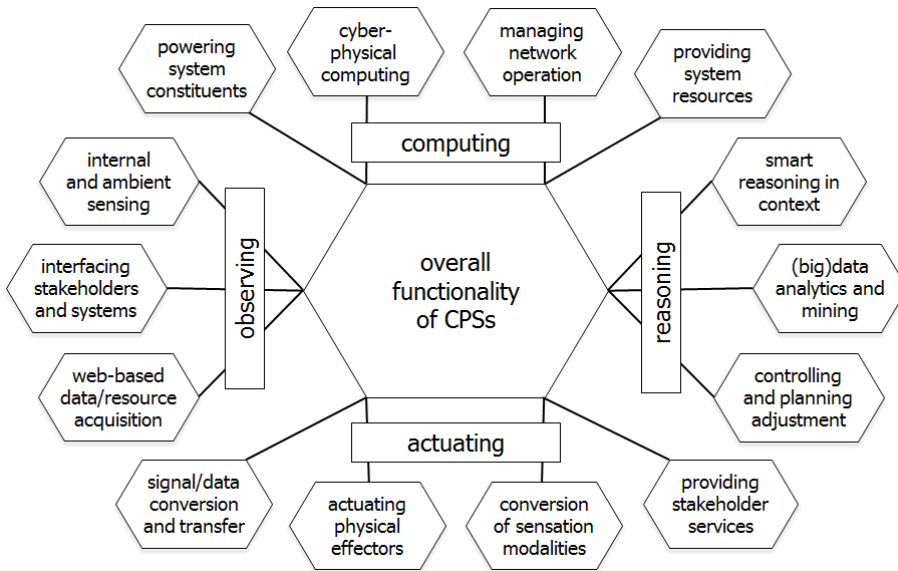
**Figure 1.1.** The generic enablers of developing CPSs

actuators, processors, data, humans, etc., are included [3]. Figure 1.1 illustrates the basic constituents of all CPSs. As it can be seen, CPSs integrate physical technologies with cyber technologies, and gradually involves emerging synergic technologies, which concomitantly reflect both physical and cyber characters.

Typically, a CPS involves one or more actor nodes, which provide specific transformative or informative functionalities and services as the output of their operation. The actor nodes (i) are interconnected with wired and/or wireless sensor networks (WSNs), (ii) interact with one or more particular socio-techno-economic environments, and (iii) often perform their functionalities and provide services in real-time [4]. Accordingly, the human-to-human interactions, the human-to-object interactions, and the object-to-object interactions in the physical world and in the virtual world may appear in enriched form in CPSs [5]. Unlike traditional sensor-controller-actuator systems, CPSs use synergic operations to optimize the utilization of sensors, controllers and actuators [6]. For instance, the physical world and the cyber world can be associated by integrating a multitude of sensor sub-systems and actuator sub-systems at different scales and levels under an intelligent decision-making system. In this way, a better system-level performance of the operations can be achieved.

CPSs are also differentiated by their broad functionalities, which are partly internal (sustaining system operation), partly external (providing transformative services). Usual transformative services are such as (i) changing material and energy states and flows, (ii) availing analog physical and digital computational resources, e.g. Internet resources, (iii) providing informative services, i.e. changing information states and flows, or (iv) all the three concurrently. CPSs implement cyber-physical computing that includes obtaining the information (needed for their regulatory control or anticipating control) from data acquired from physical and/or virtual processes in real-time or non-real-time. Cyber-physical computing also means that CPSs gradually abandon the traditional (explicit algorithm-based) computing and move towards a less predetermined (implicit algorithm-based) computing, or to a purposeful combination of the above two. The shift towards using explicit algorithms in their operations is facilitated by the recent results of artificial intelligence research and development.

As far as the internal operational and external servicing functionality are concerned, four major groups of system functions can be identified: (i) sensing and observing functions, (ii) computing and management functions, (iii) reasoning and planning functions, and (iv) actuating and effecting functions [7]. The specific system functions belonging to these groups are shown in Figure 1.2. The variety of the operational and servicing functions results in an overall functional heterogeneity of CPSs that is reflected in their complex architectures [8]. The complexity of CPS is seen both as a property and as a relation. Complexity as a system property has been characterized by structural, operational,



**Figure 1.2.** Typical articulation of the functionality of CPSs (after [7])

interactional, etc. measures, whereas complexity as a relation has been articulated as cybernetic, social, cognitive, etc. interrelationships between (i) the operational scenario of the system and the embedding environment, (ii) the related human stakeholders, and (iii) the concerned segment of the society.

Though remarkable results have been achieved in the last decade, research and development of CPSs is still a road-paving effort, i.e. an undeveloped area of research inquiry and system engineering, reflecting many farthestmost limits of knowledge or engineering achievements. The definition proposed by [9] considers the development trends as well as the most important system-level characteristics. According to this definition, CPSs includes: A CPS (i) manifests in a synergistic system of systems arrangement, (ii) includes numerous functionally networked actor nodes, (iii) realizes many sensing->reasoning->learning->adapting loops, (iv) provides tailored services and avails resources dynamically, and are characterized by: (v) deep penetration into real-life physical & social processes, (vi) data and patterns driven cyber-physical computing, (vii) ability to exploit growing level of system intelligence, and (viii) potentials for applications in human, societal, and industrial contexts. CPSs are becoming strongly multi-disciplinary, consequently, their research and development need non-reductionist research approaches including cognitive, sociotechnical and creativity issues. The typical functional articulation of CPSs shown in Figure 1.2 may be implemented in an infinite number of variations.

### **1.1.3. Characteristics and application potentials of CPSs**

The major operational attributes of CPSs are considered as timeliness, distributed, reliability, fault-tolerance, security, scalability, and autonomy [7]. As Horváth, I. & Gerritsen, B. discussed, most of the attributes are based on the synergic usage of sensors, controllers, and actuators [8]. This is because CPSs assemble together intelligent and advanced information processing technologies to monitor, handle and control the continuous dynamics of engineered physical systems [2]. These capabilities enable CPSs to timely adapt to the continuously changing and heterogeneous context in accordance with high-level goals [9]. CPSs have the potential of offering more effective and efficient solutions. This is the reason why CPSs attracted enough attention in many application domains in the past several years [10]. Nowadays, the overwhelming majority of CPSs appear as systems of systems (CPSoSs) [11]. A document of the EU Commission argued that the distinguishing characteristics of CPSoSs such as (i) physical size, (ii) geographic distribution, (iii) distributed control and management, (iii) partial autonomy, (iv) dynamic reconfiguration, (v) continuous evolution, and (vi) possibility of emerging behaviors [12]. Complexification is a well observable trend in the development of CPSs, which involves functional, architectural, behavioral, maintenance, etc. complexities.

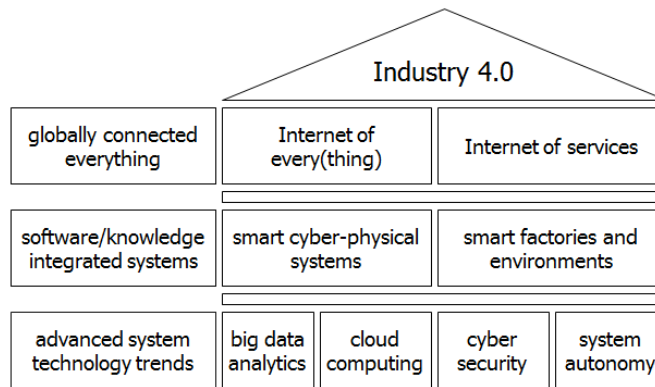
Composability and compositionality are the two major concerns of the systematic development of CPSs [13]. Composability is a system property, which is based on the assumption that the components' properties do not change by virtue of interactions with other components [14-15]. This assumption originates in the philosophy of reductionism, which highlights the consistent behavior of the component when it cooperates with other components to build a whole system. On the contrary, compositionality is originated from holism and assumes that system level properties can be derived only from the consideration of the overall purpose and behavior of a particular system and these can be decomposed into component properties if, and only if system holism is not hurt. On the other hand, composability is also about the capacity of decomposition of the system level functionalities and properties to lower level ones. It focuses on the maintenance of consistency between the system level properties and the divided properties (component properties). It assumes that the system level properties can be calculated by considering the interplay of the components/subsystems properties. As mentioned above, the compositionality issue is becoming a critical issue in the case of knowledge-integrated systems of the near future [16].

CPSs play an important role both in the implementation of transformative systems, provisioning systems, and informing systems for the industry, and well as in human and social aspects (such as clinical treatments, home care, safety installation, personal transportation, etc.). Including the Industrial Internet of Things as infrastructure, CPSs are regarded as the key enablers of the Industry 4.0

initiative [17][18]. The significance of this initiative is that it initially commenced as a national one, and now it is rapidly proliferating all around the world and provides a kind of framework of thinking about the industrial future. The embedment of CPSs in the framework of Industry 4.0 is shown in Figure 1.3. Implementation of production plans and manufacturing companies as a tightly coupled CPSoSs is facilitated by the results of digitization and informatization introduced in these industries [19]. Cyber-physical manufacturing and/or production systems (i) leverage real-time massive data sets throughout the entire production cycle, (ii) support aggregation, distribution, and utilization of both formal and tacit knowledge as a productive asset, (iii) help optimize physical and virtual processes for peak efficiency, and (iv) significantly contribute to the implementation of the doctrine of circular product realization. On the ground of reality, their implementation requires an extensive synthesis of novel, still developing technologies (as indicated in Figure 1.3), and the involvement of the entire supply and delivery chains.

The wide spectrum of information streams in cyber-physical production systems is closely monitored from all related aspects and is synchronized between the physical equipment of the factory floor and the cyber resources of the computational space. The networked machines can directly exchange and evaluate data and knowledge, communicate information about their operational state and that of the implemented processes, and thereby perform more flexibly, collaboratively, resiliently and efficiently [21-22]. Lee, J. et alia proposed a unified 5-level architecture, which is useful to guide the implementation of manufacturing systems as interoperable CPSs [23]. The so-called "5C architecture" identifies functions and attributes on five hierarchically arranged layers that are referred to as (i) smart connection level, (ii) data-to-information conversion level, (iii) cyber assets level, (iv) cognition level, and (v) configuration level. This model also implies hierarchical relationships among the constituent systems of the CPSoS, such as (i) condition-based monitoring subsystems, (ii) prognostics and health management subsystems, (iii) cyber-physical actor subsystems, (iv) decision support subsystems, and (v) resilient control subsystems. As for the interfacing process between the physical part and the cyber part of the CPSoS, and enabler of prognostics and health management, the cyber-twin concept emerged.

In the overall framework of Industry 4.0, as well as in the context of common social and human servicing contexts, next-generation CPSs are foreseen as systems equipped with some level of intelligence and referred to as smart CPSs or intelligent CPSs. Interestingly, there have been different proposals concerning the adjective 'smart' in front of the CPSs acronym. For instance, Chun, I. et alia proposed SMART-CPS as Self-MANaged Reliable system developmenT method for Cyber-Physical Systems [24]. Another interpretation counts on the gradual,



**Figure 1.3.** Synthesis of system technologies towards Industry 4.0 (after [20])

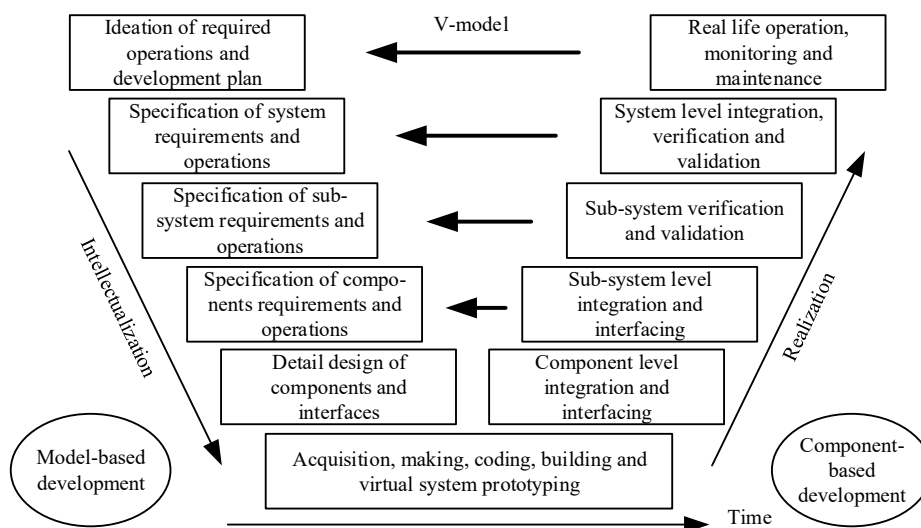
but steady increase of the cognitive capabilities, which lead to highly autonomous systems using reasoning and self-adaptation technologies.

#### 1.1.4. Implementation approaches of CPSs

As far as implementation processes of CPSs are concerned, their conceptualization, architecting, detailing, and integration are typically guided by the standardized V-model-based methodologies (Figure 1.4). The V-model (short reference to Validation & Verification strategy model) identifies three stages of development: (i) the specifications (or intellectualization) phase (represented by the left tail of Figure 1.3.), (ii) the development phase (represented by the bottom of the "V", and (iii) the realization and testing phase' (represented by the right tail). The V-model is sometimes regarded as a variant of the traditional linear (waterfall) development and implementation approach. In a different view, it establishes internal assessment and feedback loops (i.e. creates verification and validation relationships between each stage of development and the corresponding stages of testing). Originally intended for software development processes, the V-model involves model-based development and component-based implementation [25]. The implied developmental process is balanced and relies on the results of the verification done in the previous steps before advancing procedurally.

The V-model entails collaboration of designers, developers, and testers, who are supposed to work according to both high-level process documents and low-level process documents, respectively. A recognized disadvantage of the V-model is that it is very rigid procedurally and the least flexible methodologically. For instance, if any need emerges in the development phase for changing the requirements or component solutions, the test documents need to be updated accordingly. Furthermore, the systematic process and extensive documentation frequently prevent their application to short term and low scale projects. These





**Figure 1.4.** The V-model of realization of CPSs as complex systems

explain why the V-model is under continuous maintenance and updates – the latest advancements are summarized by Graessler, I. [26] and by Mathur, S., & Malik, S. [27]. The recent enhancements go beyond the development of mechatronic systems and extend to CPSs and all interdisciplinary technical systems. For instance, the V-model has been adapted to support interdisciplinary product engineering processes [28]. These adaptations are often referred to as enhanced V-model (eV-model). One advantage of the eV-model-based approaches is that they can be used for hardware, software, and cyberware design equally well (i.e. lend themselves to HW-SW-CW co-design). It has also been recognized that even the eV-model has limitations with regard to the development of knowledge-integrated smart systems, which require compositional design approaches.

## 1.2. Towards next generation cyber-physical systems

### 1.2.1. Main enablers of evolution

By studying complex natural, social and technical systems, systems science significantly contributed to the development of the trans-disciplinary domain of CPSs [29]. It offers knowledge assets and methodologies include systems dynamics modeling, agent-based modeling, micro-simulation, and big data techniques. CPSs development requires an integration of design thinking and system thinking [30]. Current systems science offers a large number of theories, formal and experiential knowledge, working principles, framing methodologies, and generalized best practices know-how for a rational development,

implementation and utilization of application-specific CPSs [31]. As a consequence of applying dedicated systems in transportation, clinical practice, logistics automation, mission-critical tasks, etc. the knowledge about optimal installations in the surrounding environment is gradually increasing. However, there seems to be a huge knowledge gap concerning the design and engineering principles of realizing next-generation CPSs. It is not completely known how to design for long-term self-learning, self-adaptation, and self-evolution, not to mention self-reproduction, of CPSs. At the time of compiling this dissertation, there are no tested design methodologies available that could provide guidance for designing CPSs for semi-autonomous or fully-autonomous operation. This is of importance since next-generation CPSs are envisioned to be a horizontally and vertically heterogeneous system of systems, having some level of reproductive intelligence.

As discussed by Horváth, I. & Gerritsen, B., the choice of enabling technologies for CPSs is extremely broad [7]. The most general classification is to sort them into the genres of hardware, software and cyberware technology, having many various types included. However, this classification is getting more and more difficult as the technology trends move towards integrated and multidisciplinary, therefore heterogeneous, technologies. One typical manifestation for this is the genre of synergic technologies, which intrinsically includes two of the abovementioned genres of technologies or even all of them. Information, computation, communication, and control technologies play a crucial role in the implementation of any CPSs [32]. In the area of hardware both analog and digital technologies are typically present, together with wired or wireless sensor and sensor network technologies, actuator and effector technologies, and data storage technologies. Implementation of the physical components of CPSs such as sensors, actuators, transducers, and transponders, for instance, creates a bridge to advanced materials technologies. In our days, quantum dots, carbon nanotubes, molecular switches, molecular motors, MEMS, etc. are often used as actuators or elements of actuators in CPSs together with functionally supplemented materials and multifunctional materials. The recently emerged advanced macro- micro-, and nano-robotics technologies, and sub-micro scale electromechanical technologies have also opened up new ways of realizing effectors.

Model-based design, component-based implementation, platform-based integration, and simulation-based verification and validation have become standards in the methodology of CPS development [19]. Cloud-based implementation of the computational functionality is becoming a de facto standard, independent of the fields of application. In the context of smart CPSs artificial intelligence technologies, such as awareness building, pattern recognition, massive data analytics, reasoning, learning, and planning technologies are indispensable. In addition to these technologies, which constitute

the components, modules, and sub-systems of CPSoSs, there are numerous development technologies. These are dedicated to specific phases of the life cycle of CPSs, such as conceptualization technologies, functional specification, architecting, system process modeling, logical and control design, operation design and simulation, interaction modeling, verification and validation, implementation, and middle-of-the-life data aggregation and analytics technologies. These technologies typically manifest in commercialized or proprietary software tools or packages, which are mushrooming in the present time.

CPSs are progressively becoming part of the socio-technical fabric of society [33]. They are becoming not only more intellectualized but also more socialized and personalized [34]. This is needed for the reason that they are supposed to be (i) installed everywhere (home, office, mobility, healthcare, entertainment, etc. environments), (ii) used by everyone (individuals, special groups, social networks, cultures, populations, etc.), and (iii) serve for many purposes dependably, efficiently, safely, economically, and context-sensitively on a 24/7 basis. On the one hand, social-cyber-physical systems have to work according to expectations of humans, communities, and society, and, on the other hand, under the constraints and conditions imposed by the embedding (often dynamically changing) environment. As socialization is concerned, there are efforts to interpret social-cyber-physical systems (or cyber-physical-social systems) as socio-technical systems [30]. In this case, the system control but also self-evolution is caused by social factors and social effects of the environment [31]. The latest conceptual shift in thinking is that not humans should be put in the operational loop of CPSs, but the newly developed systems should be placed in and harmonized with the natural culture and/or daily activities of individuals and communities. This raises the need for cognitive engineering as well as for a reasonably high level of self-adaptation by CPSs.

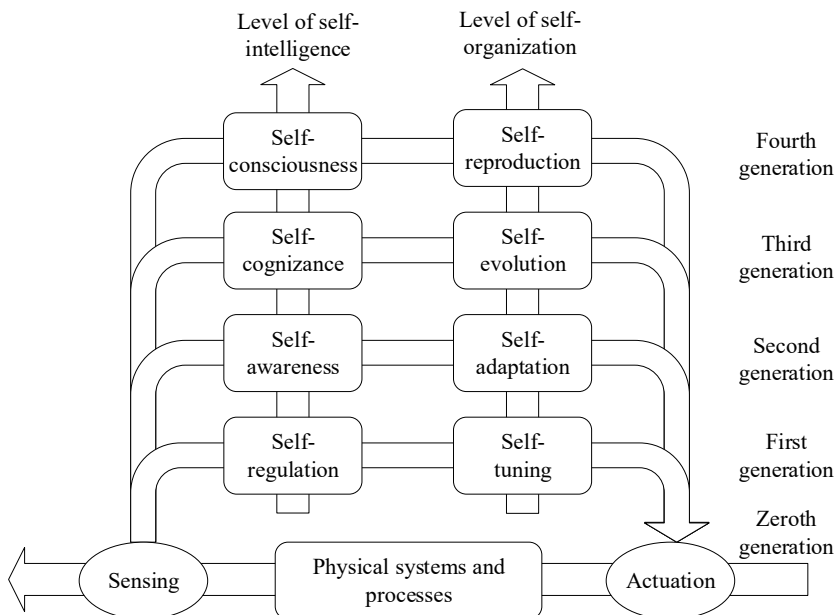
### **1.2.2. On the way to smart cyber-physical systems and beyond**

With the development of advanced computation, communication, and control technologies, it has become possible for CPSs to implement smart system behaviors. Although there is no accepted definition on what kind of system can be named as smartness, the smartness of a system was impressed as autonomous operations designed to achieve some system objectives in a general sense, which was envisioned from a replica of human intelligence [35]. The smartness of a system is performed for many purposes, including (i) describing and analyzing a situation, (ii) making decisions based on aggregated data in a predictive or adaptive manner, and (iii) actuating the decisions or controlling the execution the decisions through a proper manner [36]. In this way, CPSs of many application

domains are able to achieve stability, reliability, robustness, and efficiency [2] [37].

Actually, smart CPSs should know what they are doing and be able to do more than they have been designed for [38]. Fully-fledged system intelligence will probably be based on ‘artificial consciousness’ that will be a computational equivalent of human consciousness [39]. It means that smart CPSs should achieve a certain level of self-intelligence and self-organization. As claimed by Horváth et al. in [16], the ultimate form of self-intelligence (consciousness) and self-organization (evolutionary reproduction) is not all-or-none properties. Instead, they are deemed to be advancing and graded to varying degrees. Enabled by this, the functional and structural organization capabilities of CPSs will go beyond progressive self-organization of individual systems, and will ultimately extend to a collective reproduction of multiple functionality-orientated systems (a community of lower-complexity systems, as described by the system-of-systems notion of system theory). Actually, the specific properties in each of the degrees form an understanding of five generations (5G) of smart CPSs, as shown in Figure 1.5. The differences among the generations can be highlighted as follows.

The zeroth generation includes look-alike engineering systems and partial implementations of CPSs. Typical 0G-CPSs are such as embedded hardware/software systems, software-integrated plant systems, Internet of things systems, complicated production systems, medical monitoring systems, machine

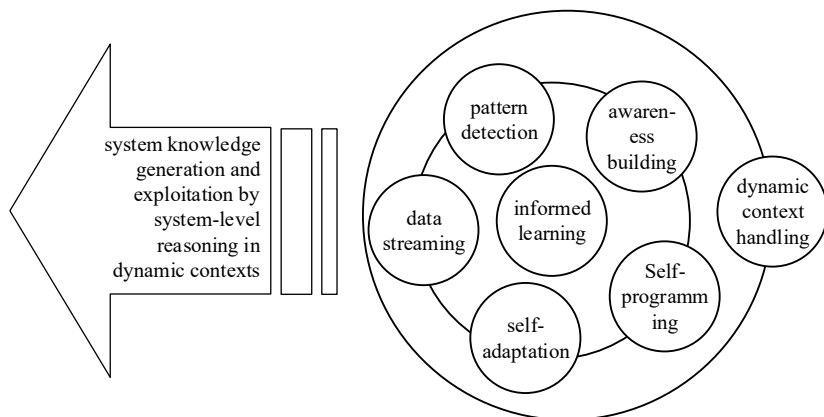


**Figure 1.5.** Subsequent generations of smart CPSs [40]

assembly robots in the automobile industry, and so forth [41]. They are functionally and architecturally closed systems and do not lend themselves to any run-time variation. They typically consist of one or more traditional ‘plant-type’ subsystems or monolithic artifacts, such as advanced robots, that form the physical subsystems. They are controlled by pre-defined (pre-programmed) closed-loop control and optimization subsystems.

The distinguishing paradigmatic features of 1G-CPSs are self-regulation and self-tuning. Feedback-based self-regulation and self-tuning represent the lowest level of proactive and deliberate system smartness and adaptability, respectively. Typically, 1G-CPSs are closed systems and are often referred to as software integrated systems. These systems include algorithms and software components that collect sensor data and react on them by issuing control signals via actuators to the physical effector components. However, the control system is neither adaptive nor predictive. Though 1G-CPSs were usually designed to operate as independent systems, nowadays many of them are conceptualized as networked (interlinked) system(s) of systems.

The second generation of CPSs (2G-CPSs) should implement anticipating smart behaviors through self-awareness and self-adaptation features. Having these capabilities is made possible by the recent trend in massive data analytics and semantic reasoning (Figure 1.6). The concept of self-awareness comes from the research domain of psychology. In 1972, self-awareness was defined as a capability that allows an object focuses on itself by evaluating and comparing its current behaviors to its internal standards and values [42]. Recently, self-awareness concepts have been employed to inspire new approaches for engineering computing systems [43]. The reason is that in many applications, the newly designed computing systems are required to operate on non-linear, complex and heterogeneous environment, in which the predefined principles and strategies can hardly be applied. Self-awareness enables computing systems to



**Figure 1.6.** Trends of the evolution of data-driven self-control (after [44])

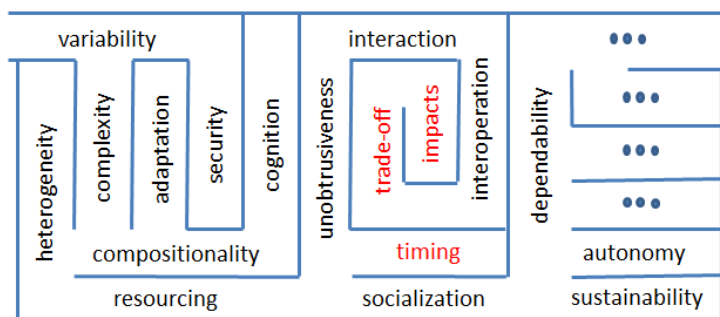
evaluate its own performance and learn from the performed operations. In this way, optimal operations could be realized to adapt to complex dynamic environments [45]. Self-adaptation is an outcome of the optimized operations in order to meet pre-set or possible (emerging) system objectives, in which certain reasoning and decision-making activities should be taken [46].

The distinguishing paradigmatic features of 3G-CPSs are self-cognizance and self-evolution. While system awareness was interpreted as the potential of a system to build a world model effectively in a given situation, cognizance is supposed to enable the development of multiple (but a restricted number of) models of the external world from various perspectives. Thus, self-cognizance is referred to as the capability to capture and assess what is happening in a given local word and to propose multiple models from various perspectives. The ability of evolving ('evolvability') of a CPS is defined as the potential ability (enthalpy or internal energy) to evolve from one stable system configuration to another, and from one multi-functionality state to another in response to changes in the requirements, goals, environment and the system itself, by using system excess and modularity [47]. Self-evolution would provide an opportunity for repurposing the system objectives and the concepts of operations in harmony with technological, social and environmental changes.

The distinguishing paradigmatic features of 4G-CPSs are self-consciousness and self-replication. These are direct manifestations of system intelligence. As conceived, system consciousness would be a fully featured replica of human consciousness, which is global and decontextualized. Actually, this demands a comprehensive and deep implementation of system intelligence (including traditional capabilities such as machine perception, situation awareness, computer vision, machine learning, etc., but also many new ones). Computational system consciousness necessitates the potential of deriving and/or maintaining large (but still infinite) number of computer-internal world models run time, even in cross-context perspectives. This is deemed to be a principal difference at comparing 3G-CPSs with 4G-CPSs. Only the fourth generation of CPSs is supposed to achieve self-consciousness and self-reproduction in the form of creating a system of systems.

### **1.2.3. Open research issues concerning smart CPSs**

Currently, there are many challenges concerning smart CPSs, which emerge either (i) from a methodological aspect (as difficulty associated with enabling and coordinating activities such as design inception and integration of constituents) or (ii) from an epistemological aspect (as a recognizable or observable extensive shortage of design knowledge and concepts due to immaturity of research in the given context). They together have been interpreted as the labyrinth of challenges (Figure 1.7).



**Figure 1.7.** The ‘labyrinth of challenges’ that smart CPSs developers are facing (after [44])

Based on the introduced properties with regard to different generations of smart CPSs, it can be seen that the achievement of full consciousness and evolutionary reproduction in computing systems is only an idea. Much research effort is needed to increase the level of self-intelligence and self-organization. Most of the existing computing systems are able to fulfill the requirements with regard to 0G-CPSs and 1G-CPSs. However, only some primary research progress has been achieved with regard to self-awareness and self-adaptation, which are fundamental features of 2G-CPSs. Although 2G-CPSs typically use run-time acquired data in addition to the data stored in the system’s operation model (and control model) for system control, they make the first step towards opening the system boundaries from both architectural and operational aspects [40]. For instance, a 2G-CPS that has been developed for detecting and enhancing short-term engagement of stroke patients in rehabilitation exercises based on a smart learning mechanism, which is able to be refined based on the runtime aggregated information [48].

In order to implement smart behaviors, 2G-CPSs are required to build awareness of the actual states and changes in the operation/servicing environment and the parts of the system. In addition, they should explicitly capture and reason with the contexts of operation (self-awareness) and alter the operation/servicing of the concerned system according to the changes in the contexts (self-adaptation) (Figure 1.8) [1] [49]. Both of the above-mentioned capabilities need a sophisticated consideration on the working context [50]. However, processing context information and using it in decision making are not trivial problems [51]. The reason is that 2G-CPSs normally works in a dynamic environment and/or under strongly varying circumstances. The working context reveals many dynamic features. Therefore, 2G-CPSs are supposed to behave smartly in various application cases featuring time-varying contexts. The processing of dynamic context information is a research challenge and key to the success of ubiquitous

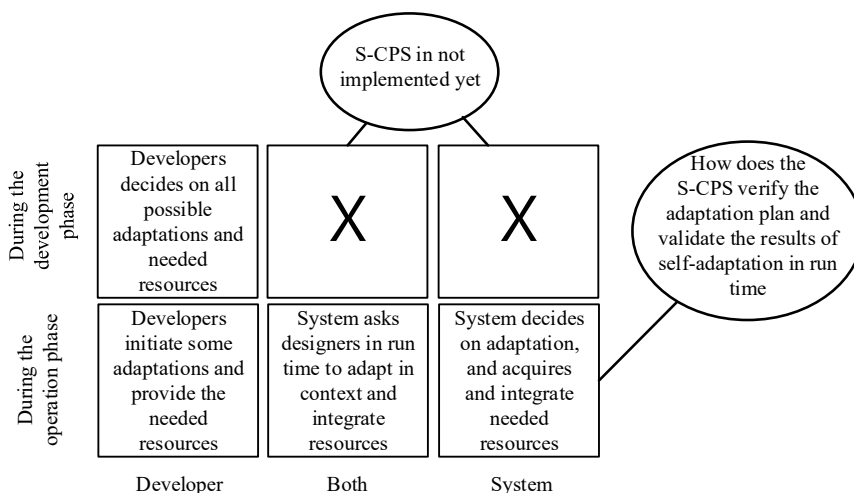
and pervasive computing [52][53]. Towards this end, the open research issues are how to acquire, manage and utilize dynamic context information and adapt system operations to the dynamic context.

## 1.3. Introducing the specific research domain

### 1.3.1. Context management and related issues

The term ‘context’ is variously used in the literature, and it has been defined either too general, or too specific, depending on the purpose [54]. There is no universally accepted definition of context. In its broadest interpretation, context is about any circumstance in which something happens. In the case of human beings, context is considered as a ‘state of the mind’ influencing interpretation and decision making. Technically, it refers to the setting of a thing or a process, i.e. a set of facts or circumstances that are associated with a situation. Context influences the outcome of a thing or a process that happens. In previous studies, context information has been defined as a set of information which influences the realizing of a certain objective with a particular manner and possibility.

By the developers of context-aware systems, context information has been defined as any information related to people, places or objects that are relevant for operations of the systems [55]. Other considerations of context information are ‘any information that can be used to characterize the situation of an entity [56]’ and ‘a set of (contextual) elements of the user’s environment that should be recognized by a computer [57]’. Although there is no universal definition of context information in computation, the basic principle of context-aware computing is to mimic the capability of human beings, which enables adaptive decision-making in changing circumstances. Many research outcomes have



**Figure 1.8.** Adaptation opportunities of smart CPSs (after [44])



approved that the performance of the systems can be improved if the context is adequately considered [58]. Even simple forms of context, such as time and identity, have been used in a number of meaningful ways [59]. For this reason, the capability which enables a system to aggregate, interpret, and use context information and automatically adapt its functionalities to the context is named as ‘context-awareness’ [60]. In real-life applications, the most commonly used specific aspects of context are location, emotion, nearby entities and changes to those entities, the time of day, season and temperature [61]-[63].

### **1.3.2. Towards context-aware computing**

A set of specific steps related to processing of context information was proposed in [64], including: (i) context acquisition, (ii) context aggregation, (iii) context consistency, (iv) context discovery, (v) context query, (vi) context adaptation, (vii) context reasoning, (viii) context quality indicators, and (ix) context integration. Technical investigations and implementations with regard to these aspects have been widely studied in recent years. Specific, though limited, attention was given to the operation in dynamic contexts, where the term ‘dynamic’ refers to procedural state changes, rather than to ontological ones. As mentioned above, the need for dynamic context information processing (DCIP) is derived from the rapid changing of the environment and the working conditions of 2G-CPSs. The dynamism of context should be focused on and investigated.

In general, dynamic context is considered as a set of instantaneous information within a given time span, which represents the changes in situations [65]. According to the frequency of change, context can be classified as: (i) static context, (ii) altering context, (iii) dynamic context, and (iv) emergent context [41]. Static context does not change as time elapses and has a constant influence on the thing or process it is associated with. A static context is, for example, the floor plan of a building. Altering context changes over a substantial period of time, in which its influence is sustained, e.g. the change of seasons in a year. Dynamic context is the result of situations with rapidly changing influence. As an example, traffic situations on highways can be mentioned. Emergent context changes in an unpredictable manner, hence its influence is uncertain. As an example, we may think of a sudden collapse of a bridge.

In addition, dynamics of context can be considered from an external aspect (e.g. the continuously changing context elements) and an internal aspect (e.g. perception, memory, and reasoning in the human mind) [67]. Both aspects require a sufficiently articulated representation of the context and a logical/semantic interpretation of the meanings and implications of the context changes. The representation (description) and the logical/semantic interpretation (inference) together form a basis for context prediction and decision-making.

### 1.3.3. Processing of dynamic context information

In addition to static context information (SCI), three types of dynamic context information (DCI) have been differentiated: (i) altering, (ii) varying, and (iii) emergent. These can be actively generated by (a network of) sensors or collected or extracted in other ways directly from the environment, or a rich representation of the environment. DCIP requires not only dynamic acquisition but also dynamic computation. This raises the need for modeling, representation, extraction, and processing (MREP) approaches that provide so-called real-time (i.e. task execution dependent) or near-zero time (i.e. optimized for a minimum possible time processing) possibilities. However, the overwhelming majority of the known MREP approaches have been developed without explicitly addressing these issues. On the other hand, truly dynamic MREP approaches are needed in the practice as we move towards real-time context monitoring services and managing incidental events, and towards dynamically evolving semantic models of context information. In contrast to fixed-context decision making, processing DCI has opened up the road to variable-context decision making, which is seen as an enabler for self-adaptive cyber-physical systems. DCIP in a time-optimized manner makes it possible to produce more useful computational services, and eventually, as proposed by Palau, M. et al [68], dynamic context-aware global information services.

As of today, the contours of a new discipline called context engineering (CE) can already be seen. CE puts information systems development into the position of a socio-technical phenomenon, and cultural and personal perspectives [69]. CE advises us on how to achieve quality context modeling and processing. These should be characterized by (i) comprehensiveness (sufficiency of the kinds and the amount of context information for a purpose), (ii) reduction of deviations (avoiding errors from imperfections of sensing, classification and reasoning), (iii) lessening ambiguities (avoiding models that do not formally define what to do with uncertainties of information), (iv) specification of dependence (description of associations with a sufficient number of quality parameters), and (v) using quality metrics (each context parameter/variable is to be described by one or more appropriate quality metrics).

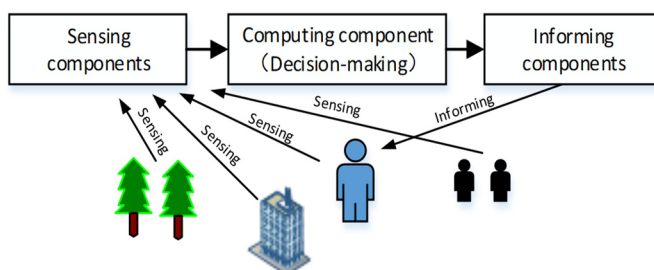
Beyond capturing and representation of SCI and DCI, CE also extends to inferring semantic knowledge from context information and reasoning with it under various circumstances. It is also addressing management of contextual relationships, which can be: (i) static associations (fixed relationships over the lifetime of an entity that capture it with a high confidence), (ii) varying associations (which are sensed as temporal associations and are not inserted directly into the context model straight from the sensor, but transformed into some level abstraction), and (iii) derived associations (which are obtained from one or more other associations using a derivation function - e.g. collection or

history associations.

A central topic of CE is context feature eliciting and context model manipulation, which are challenging tasks in the presence of DCI. Context features are interpreted as unique information patterns, which may change over time. Features-based context matching and mapping include computational activities such as (i) feature extraction, (ii) feature classification and labeling, and (iii) feature-implied inferring/reasoning. Dynamic situations are characterized by variable feature vectors. Other focal points of research in CE are the theory and programming of context learning mechanisms, which will have a significant role in future CPSs. The approaches currently used in context learning and reasoning are such as neural networks, Bayesian networks, and symbol clustering maps. However, these have serious constraints with regards to capturing time-dependent situational information, other than time series data.

### 1.3.4. Informing cyber-physical systems

Informing cyber-physical systems (I-CPSs) represent a specific cluster of CPSs, which implements a sensing-reasoning-actuation operation scenario, but their actuation functions are dedicated to informing people or systems based on messaging (Figure 1.9). In other words, the objective of I-CPSs is to provide services for stakeholders in the form of customized action plans and timely-refreshed information, guidance, and/or instructions. What actuation means in this branch of CPSs is generating individualized messages, forwarding them to specific clients, and observing and monitoring their reactions. As reported in the literature, typical examples are mobile tourist information systems [70], context-aware navigation systems [71], home security systems [72], driving coaching systems [73], and evacuation management systems [74]. In I-CPSs, the internal and external actuation typically takes the form of (i) deriving or synthesizing application-specific information, (ii) distribution and visualization of informative and instructive messages, and performing (usually real-time and dislocated) interactions between the users and the systems. Driven by the belief that I-CPSs will proliferate already in the near future, we concentrated on this



**Figure 1.9.** Basic constituents of an informing CPS

kind of systems in the promotion research.

The main operations of the computing components of I-CPSs may include: (i) situation awareness generating operations, which process various types of context information, (ii) inferring operations, which are used to build awareness of the varying circumstances, (iii) reasoning operations, which are used to generate solutions for users to act, and (iv) messaging operations, which are used to construct the messages to be communicated. In order to arrive at proper messages (e.g. for guidance), the knowledge construction operations (i.e. inferring and reasoning) of I-CPSs should consider the dynamic contexts of the users. This is important since the situation of the users of I-CPS services may change (e.g. they may dynamically change their location and states), but the environment (i.e. the surroundings of the users) may also change rapidly as well [75]. It means that processing of dynamic context information is a necessity, but it is also a typical challenge for I-CPSs.

Smart operations of I-CPSs concern providing informational services for applications and stakeholders. The range of possible informational services is rather broad (including, e.g. customized action plans, timely-refreshed information, or context-sensitive guidance). Actually, the variety of application opportunities for these systems is constrained only by the imagination of the system designers and by the economics of implementing them for various applications. Usually, the control functionality of I-CPS is extended with data analytics functionality, which is based on multiplexed sensor nodes and pervasive sensor networks, and information modality transformers and message generators. The servicing activities of I-CPSs include various messaging functions such as: (i) selecting informing modality, (ii) constructing personalized messages, and (iii) distributing messages to stakeholders.

I-CPSs are supposed to execute messaging operations according to the actual situational context that gives the reference for the interpretation of messages and completing actions. Towards this end, two types of messages are normally used in I-CPSs, namely, instructive messages and informative messages. Instructive messages are used to inform stakeholders about “what they should do”. Instructive messages may manifest as personal recommendations, situated solutions, or action guidance. Informative messages indicate “what the situation is” and “what the stakeholder should be aware of”. These messages are intended to increase the situational awareness of the stakeholder (or in other words, “of what is happening around the stakeholder”). Both types of messages are to be based on a factual description and ‘understanding’ of the situation or the circumstances that are relevant to individual stakeholders. Message generation has both syntactic and semantic aspects. The former is related to information engineering, while the latter is associated with language processing. This gives an interdisciplinary flavor to messaging in I-CPSs.

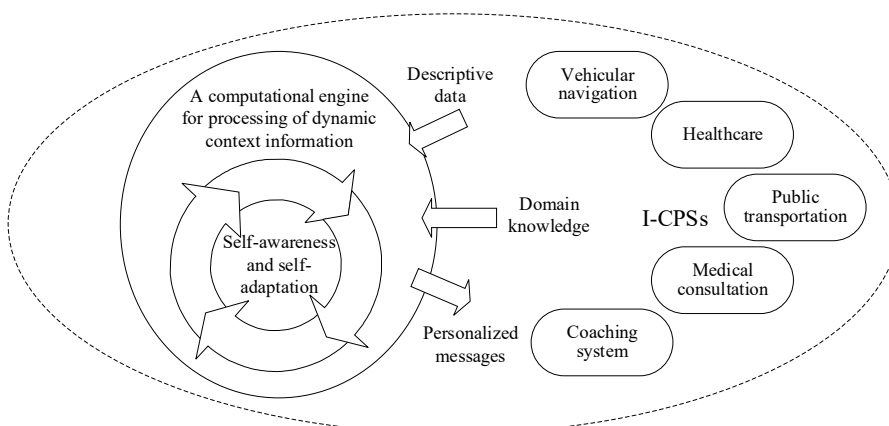
The ultimate objective of DCIP for I-CPSs is to support the decision-making process of the stakeholders in cases such as hazardous events (evacuation scenarios) or mission-critical applications. Therefore, the messages constructed for stakeholders should be sensitive and tailored to dynamically changing contexts. They may include descriptive information about the situation the stakeholders are in (or might be troubled with), and instructive information to command or to assist their actions. Although many computational message construction mechanisms were developed, most of them consider only static context information, e.g. weather, temperature, daytime or permanent things of a particular location, so as discussed by [76], [77] and [78].

## 1.4. Research objectives, challenges, and questions

### 1.4.1. Research objectives

As briefly introduced above, the main objective of the completed research project was to develop a computational engine for processing dynamic context information, which enables self-awareness and self-adaptation for I-CPS applications. As shown in Figure 1.10, the designed computational mechanism makes use of both domain knowledge and descriptive data aggregated from I-CPS applications, while provides personalized messages for the applications to perform informing actions. This project focuses on the development of approaches to DCIP, rather than presenting particular solutions for a given I-CPS. Accordingly, the computational engine should work in an application-independent fashion.

As mentioned previously, the main operations of I-CPSs are inferring, reasoning and messaging. The self-awareness feature has been realized through various



**Figure 1.10.** The focus of the research

inferring operations, while the self-adaptation feature is achieved by means of various reasoning operations. The messaging operations consider the results of both to generate personalized messages. Therefore, the main objective was decomposed into several interrelated sub-objectives and the details of the sub-objectives are explained below. The dynamic context information processing mechanisms (DCIP-M) is a purposeful set of computational algorithms and associated data constructs.

As a first step of the development of the algorithms of the DCIP-M on a prototype level, a comprehensive set of general requirements and technological requirements were needed. Dominantly, the general requirements were deduced from pertinent theories, but some of them were formulated based on the generalization of the user needs of various existing I-CPS applications. The computational and technological requirements were derived and generalized through the investigation of a particular real-life case. These together were utilized as the basis for the generation of the theory and the concepts underpinning the DCIP-M (discussed in detail in the following chapters).

The developed computational mechanism was supposed to have abilities for building some level of awareness based on dynamic context processing. Various inferring operations were conceptualized to extract information about the to-be-recognized changes in states and situations. They were used at the identification of the implications of the recognized changes of the contextual entities. The software prototyping part of the research project included the tailoring of the reasoning (making decisions on the actions that the stakeholders were supposed to do) and the messaging according to the interpretation of the situational changes. In addition, an inevitable part of the research problem was the confirmation of the underpinning theories and validation of the results. In the lack of opportunity and resources for the realization of true experimental studies, this latter had to be completed based on computational simulations and critical systems thinking. Nevertheless, a real-life interrogation was included as a complementing knowledge inquiry method.

### **1.4.2. Research challenges**

The main research challenge arose from the conceptualization of the addressed research problem – based on which it had both information science and engineering research flavors. Starting out of the above formulated objective, the major tasks were formulated as follows: (i) obtaining insight in the essence of DCI and exploring the state of the art in dynamic context information management, in particular in I-CPSs, (ii) development of a conceptual framework, which captures entities, relationships, attributes, and changes in space and time, (iii) transferring the conceptual framework to a multi-functional computational mechanism that can be used as a kernel of a reasoning platform for specific

I-CPSs, (iv) using evacuation of a building in fire as a source of empirical knowledge and as a practical case study throughout the completed research, and (v) conceptualization, design, implementation and testing the performance and applicability of the proposed computational mechanism. Our work was also challenged by the idea of generalizing the proposed multi-functional computational mechanism to be able to handle dynamically changing spatial and attributive context information in other target application cases, such as protection in disaster, crowd management, and medical rehabilitation.

### 1.4.3. Research questions

Our guiding research question has been formulated so as:

*In what way can semantic information be obtained from DCI and how can it be utilized in certain elements of smart behavior (such as situation awareness building, situated reasoning, decision making, and action planning) of I-CPSs?*

Based on a decomposition of the research problems, four groups of working research questions (WRQs) have been identified. The WRQs concerning knowledge aggregation and building a knowledge platform for the follow-up research were as follows:

- (i) What is the current state of the art in research and development of DCIP?
- (ii) What are the general and technical requirements for a computational mechanism for multi-functional DCIP from the perspective of smart I-CPSs?
- (iii) What are the recognized limitations of the existing DCIP solutions with regard to enhancing system smartness?

Concerning the ideation, conceptualization, and specification of a sophisticated computational solution, the WRQs have been formulated in the following forms:

- (i) What representation can be used for describing context data in a cohesive manner?
- (ii) What functionalities is the pursued computational mechanism supposed to provide?
- (iii) What way can the needed different functionalities be integrated to form a computation mechanism (context processing engine)?
- (iv) What way can the fulfillment of the explored requirements by the proposed functionality and architecture be tested and approved?

With regards to the detailing, implementation, and testing of the DCIP-M, we addressed the following WRQs:

- (i) What algorithms and data structures are needed for the realization of the design functionalities?
- (ii) What way can the functional modules and the whole of DCIP-M be brought to a software level realization?
- (iii) What way can the functionalities and performance of the implemented DCIP-M prototype be validated?

Finally, concerning the practical utilization and potential adaptation of the proposed DCIP-M, we intended to find answers to the following WRQs:

- (i) What relevant application cases, other than the reference application case, can be considered for the application of the DCIP-M?
- (ii) Based on what methodological approach can the applicability of the DCIP-M be tested?
- (iii) What transferable knowledge can be obtained based on the applicability testing of the computational mechanism?

The guiding and the working research questions were operationalized through the development of a research model (RM) (by which we defined the scope of research) and research design (RD) (by which we defined the conduct of research).

## **1.5. Methodological framing of the research approach**

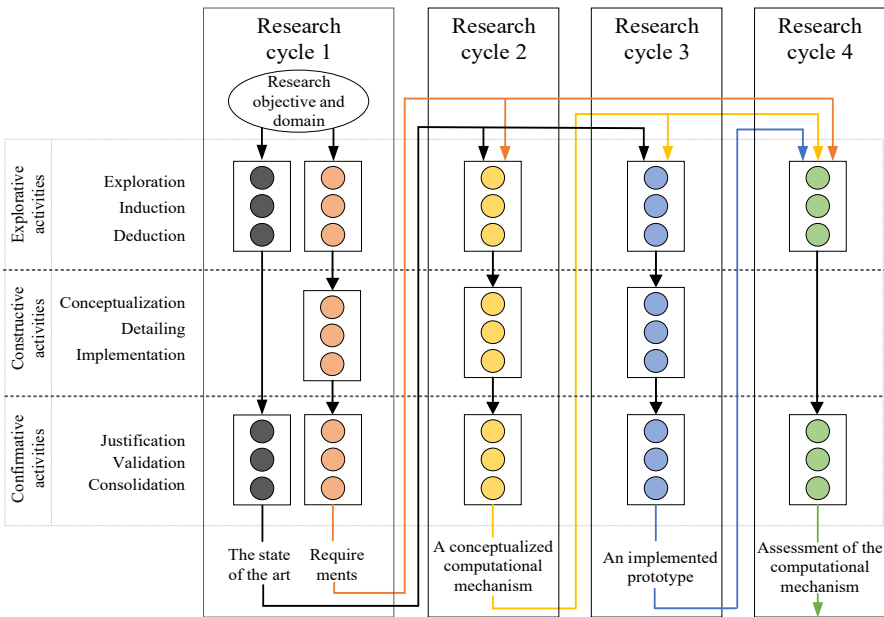
At the beginning of the four-year promotion research, a Topic and Work Specification was developed. This summarized the initial plans concerning the contents and the procedures of doing research. From a methodological point of view, an obvious challenge was to bring the scope of research and the procedure of research into harmony. In order to facilitate it, as briefed above, both a detailed research model (RM) and a research design (RD) have been elaborated. These were elaborated and refined following the guidelines proposed in a conference paper [62]. The research model clarified the boundaries of the domain of interest (the so-called local world) and the scientific and system engineering foci of the promotion research. The RM also involved a specification of the addressed research problems, the considered research phenomena, and the major research concepts that have been used in the conceptualization and specification of the concrete research and software development activities. The research model served as an information basis for formulating the overall (guiding) research questions as well as the specific (working) research questions for the whole study.

The research design has been elaborated with a view to the research constructs, the research variables, the research processes, the research methods, and the research instrumentation. The RD extended to all research cycles and specified



their internal procedural structure/ As far as the overall research process is concerned, it has been divided into a stream of four research cycles. The general nature of the research was what is typical in information system research and engineering. The methodological framing of the research project was done accordingly and it helped transfer the theoretical framework of the DCIP-M into a testable implementation. The research cycles, their internal design, and the logical flow of research knowledge are shown in Figure 1.11.

The overall objective of the first research cycle was knowledge aggregation concerning DCIP. There were three related sub-objectives also specified: (i) refinement of the research objectives, (ii) critical investigation a utilization of the state-of-the-art knowledge, and (iii) aggregation of requirements for the pursued dynamic context management mechanism. The cycle has been decomposed into an explorative part, a constructive part, and a confirmative part, in which various methods were used. The constructive part focused on the building of a robust knowledge platform and a conceptual reasoning framework. Though the compilation of the knowledge and the list of requirements have got a constructive flavor, this research cycle was methodologically framed as a research in design context (RiDC) process. The design context was the development of the DCIP-M for the indoor fire evacuation guiding application. The specific needs for dynamic context information processing were explored based on surveying existing I-CPS applications.



**Figure 1.11.** Methodological framing of the promotion research

The general requirements for I-CPSs and the specific application needs found were used as a starting point of generating technical (general) requirements. In addition, a real-life application case (indoor fire evacuation guiding) was analyzed based simulated scenarios and practical problems in various execution scenarios were investigated in order to extend the set of specific requirements with additional ones concerning self-awareness and self-adaptation in I-CPSs. At the end of the exploratory phase of the research cycle, the generated requirements were clustered and their consistency was checked. The validation of the set of requirements was done by critical system thinking and feasibility analysis.

Research cycle 2 focused on the ideation and technical conceptualization of a multi-component reasoning mechanism (DCIP-M). This research cycle was methodologically framed as design inclusive research. The explorative part focused on the exploration of enabling knowledge and technologies, as well as on ideation and ranking the ideas. At the end of the explorative phase, the fundamental concepts related to DCIP were specified. One input for the constructive part of this research cycle was the general knowledge explored and the list of requirements synthesized in Research cycle 1. The constructive part concentrated on the establishment of a comprehensive conceptual model of the DCIP-M. Towards this end, various methods for DCI were considered and the implications of context changes were investigated from an information engineering point of view.

Based on the foundational ideas, a comprehensive concept of a computational mechanism for ‘building situation awareness’ in dynamic context was developed. In addition, the issues of inferring semantic information based on the changes in syntactic context information representation and utilization of the obtained semantic context information in action planning were investigated. In this research cycle, computational approaches to context-dependent reasoning about action plans and context-sensitive message generation were also elaborated. To validate the feasibility of the proposed components of the DCIP-M, a plan for a software prototype implementation was developed.

The third research cycle was dedicated to the functional and architectural implementation of the components (modules) of the DCIP-M and the various algorithms in executable forms. A scientific challenge was associated with the establishment of a self-adaptation behavior of the reasoning module of the overall DCIP-M, which was necessary due to the effects of the dynamic context changes on the decision-making process executed by the DCIP-M. Also, this research cycle was framed methodologically as a design inclusive research cycle. The explorative phase of this research cycle collected information concerning the prototype level implementation of the modules and the algorithms, and about the logical and computational techniques of context-dependent decision making. The constructive phase of this research cycle targeted the software level

implementation of all modules and algorithms of the DCIP-M. The confirmative part of the research cycle focused on the functional and performance testing of all critical algorithms and the investigation of their proper interoperation and results. In the simulative studies, scenarios generated based on the assumed real-life application (i.e. the reference case of indoor fire evacuation guiding) was used.

In the fourth research cycle, applicability validation of the DCIP-M was considered. Methodologically, this research cycle was completed as a practice driven research. The major issue was to demonstrate and evaluate the applicability of the developed DCIP-M in application cases different from the reference case. Four aspects of applicability, namely (i) theoretical structural relevance, (ii) empirical structural suitability, (iii) empirical performance efficiency, and (iv) theoretical performance sufficiency were considered at the application of the adapted validation square approach. In the applicability investigation, three differing real-life application cases have been considered: (i) road traffic management, (ii) home care-taking assistant, and (iii) coaching football training.

## **1.6. Structure of the thesis**

Based on the methodological framing, concrete research actions in the four research cycles were specified and proposed in each of the following chapters. In Chapter 2, a literature study was conducted. The limitations of existing solutions were deduced by comparing the needs of 2G-CPSs and the offerings from existing solutions. The findings from the literature were summarized, in which a knowledge platform was built for the rest research cycles. In addition, both system-related requirements and application-related requirements were explored. The consistency of the generated requirements was sorted was checked. A set of requirements were specified according to the limitations observed from the knowledge aggregation.

In Chapter 3, theoretical fundamentals with regard to dynamic contexts were proposed, including (i) a semantically enriched conceptual model for understanding of varied types of contextual phenomena, (ii) a context knowledge pyramid, which is the basis for inferring semantic knowledge from context changes, and (iii) an approach to interpreting the dynamic context with regard to one entity. Based on the fundamental concepts, computational mechanisms for representing dynamic context information, building awareness in a dynamic context, devising personal action-plan, constructing personalized messages were carried out. Algorithms needed for achieving the functionalities were specified.

Then, a prototype of the proposed computational mechanisms was implemented. Details with regard to the implementation were presented in Chapter 4. The implemented modules were tested individually in themselves and together in a

specific application case: indoor fire evacuation guiding (IFEG). A simulation of an assumed real-life scenario was implemented, in which the behaviors of people were specified. Based on the simulated scenario, the functionality of the implemented algorithms and data constructs was validated. In addition, this research cycle also includes a performance testing. The performance of computation was quantitatively evaluated concerning computational time spent by the implemented functions and algorithms under different kinds of loads.

In Chapter 5, the applicability of the proposed computational mechanisms was validated concerning the above-mentioned three application cases. To this end, an adapted validation square approach was proposed. Various applicability indicators were defined and used as a basis to form applicability profiles of the developed multi-component computational mechanism (as well as of the methodology implied by this) were generated.

In Chapter 6, we summarized the whole of the completed work and answered the main research question of the project: 'How to process dynamic context information to support self-awareness and self-adaptation?' We reflect on the research activities conducted in the research cycles as well as on their outcomes, based on which our conclusions and propositions were drawn.

## **1.7. Related own publications**

1. Li, Y., Song, Y., Horváth, I., Opiyo, E., Zhang, G., Xiong, J. (2014). Development of a framework for information acquisition and processing in cyber-physical systems. In: Proceedings of the TMCE 2014. pp. 53-65.
2. Horváth, I., Li, Y., Rusák, Z., van der Vegte, W.F., Zhang, G. (2016). Dynamic spatial context computation for time-varying process scenarios. In: Proceedings of the ASME 2016 IDETC-CIEC, pp. 1-16.
3. Li, Y., Horváth, I., Rusák, Z., van der Vegte, W., Zhang, G. (2016). Conceptualization of a real-time information processing platform for context-aware informing cyber-physical systems. In: Proceedings of the TMCE 2016. pp. 53-65.
4. Horváth, I., Rusák, Z. Li, Y. (2017). Order beyond chaos: Introducing the notion of generation to characterize the continuously evolving implementations of cyber-physical systems. In: Proceedings of the ASME 2017/ IDETC-CIEC, pp. 1-12.
5. Horváth, I., Li, Y., Rusák, Z., van der Vegte, W.F., Zhang, G. (2017). Dynamic computation of time-varying spatial contexts. *Journal of Computing and Information Science in Engineering*. 17(1), pp. 1-12.
6. Li, Y., Horváth, I., Rusák, Z. (2018). Constructing personalized messages for

informing cyber-physical systems based on dynamic context information processing, In: Proceedings of the TMCE 2018. pp. 1-18.

7. Li, Y., Horváth, I., Rusák, Z. (2019). Personalized messaging based on dynamic context assessment: Application in an informing cyber-physical system, *Journal of Integrated Design and Process Science*, pp. 1-21. (In press)
8. Li, Y., Horváth, I. (2019). Fundamentals and contributions to validation of constructive computational methodologies: a focused survey. 7th International Forum on Industrial Design, May. 17-19, Luoyang, China, pp. 1-6.
9. Li, Y., Horváth, I., Rusák, Z. (2019). Building awareness in dynamic context for the second generation cyber-physical systems, *Future Generation Computer Systems*. (revised paper submitted)
10. Li, Y., Horváth, I., Rusák, Z. (2019). A systematic applicability validation of a dynamic context information processing-based reasoning and messaging mechanism, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. (under review)

## 1.8. References

- [1] Lee, E.A. (2008). Cyber physical systems: Design challenges, In: *Proceedings of the 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing*, pp. 363-369.
- [2] Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010). Cyber-physical systems: The Next Computing Revolution, In: *Proceedings of the 47th Design Automation Conference*, ACM/IEEE Anaheim, CA, pp. 731-736.
- [3] Ning, H., Liu, H., Ma, J., Yang, L.T., & Huang, R. (2016). Cybermatics: Cyber-physical-social-thinking hyperspace based science and technology, *Future Generation Computer Systems*, 56, pp. 504-522.
- [4] Schoitsch, E. (2010). Cyber-physical systems (CPS): The "systems-of-systems" challenges, *Schriften Informatik*, 32, pp. 163-176.
- [5] Wu, F.-J., Kao, Y.-F., & Tseng, Y.-C. (2011). From wireless sensor networks towards cyber physical systems, *Pervasive and Mobile Computing*, 7, pp. 397-413.
- [6] Lee, E.A. (2008). Cyber physical systems: Design challenges, In: *Proceedings of the 11th IEEE Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, pp. 363-369.

- [7] Horváth, I., & Gerritsen, B. (2012). Cyber-physical systems: Concepts, technologies and implementation principles, in: *Proceedings of the Tools and Methods of Competitive Engineering Symposium*, Karlsruhe, Germany, pp. 19-36.
- [8] Dong, J.T., Xiao, T.Y., & Zhang, L.X. (2012). A prototype architecture for assembly-oriented cyber-physical systems. In: *Proceedings of the AsiaSim 2012 Conference*, pp. 199-204.
- [9] Horváth, I., Rusák, Z., & Li, Y. (2017). Order beyond chaos: Introducing the notion of generation to characterize the continuously evolving implementations of cyber-physical systems. In: *Proceedings of the ASME 2017 International Design Engineering Technical Conferences*, Cleveland, Ohio, USA, pp. 1-14.
- [10] Gurgen, L., Gunalp, O., Benazzouz, Y., & Gallissot, M. (2013). Self-aware cyber-physical systems and applications in smart buildings and cities. In: *Proceedings of the Conference on Design, Automation and Test in Europe*, EDA Consortium, Grenoble, France, pp. 1149-1154.
- [11] Jamshidi, M. (2008). *Systems of systems engineering: Principles and applications*. CRC press.
- [12] Engell, S. (2014). Cyber-physical systems of systems: Definition and core research and innovation areas, Working Paper of the Support Action CPSoS, pp. 1-11.
- [13] Horváth, I., Tepjit, S., & Rusák, Z. (2018). Compositional engineering frameworks for development of smart cyber-physical systems: A critical survey of the current state of progression. In: *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 1-14.
- [14] Gössler, G., & Sifakis, J. (2005). Composition for component-based modeling. *Science of Computer Programming*, 55(1-3), pp. 161-183.
- [15] Zhou, P., Zuo, D., Hou, K. M., & Zhang, Z. (2017). A decentralized compositional framework for dependable decision process in self-managed cyber physical systems. *Sensors*, 17(11), pp. 2580-2613.
- [16] Tripakis, S. (2016). Compositionality in the science of system design. *Proceedings of the IEEE*, 104(5), pp. 960-972.
- [17] Mosterman, P.J., & Zander, J. (2016). Industry 4.0 as a cyber-physical system study. *Software & Systems Modeling*, 15(1), pp. 17-29.
- [18] Jazdi, N. (2014). Cyber physical systems in the context of Industry 4.0. In: *Proceedings of the International Conference on Automation, Quality and*

*Testing, Robotics*, pp. 1-4.

- [19] Wan, J., Cai, H., & Zhou, K. (2015). Industrie 4.0: Enabling technologies. In: *Proceedings of International Conference on Intelligent Computing and Internet of Things*, pp. 135-140.
- [20] Forbes, H., & Schaefer, D. (2017). Social product development: The democratization of design, manufacture and innovation, *Procedia CIRP*, 60, pp. 404-409.
- [21] Herterich, M.M., Uebernicketel, F., & Brenner, W. (2015). The impact of cyber-physical systems on industrial services in manufacturing. *Procedia CIRP*, 30, pp. 323-328.
- [22] Li, L. (2018). China's manufacturing locus in 2025: With a comparison of "Made-in-China 2025" and "Industry 4.0". *Technological Forecasting and Social Change*, 135, pp. 66-74.
- [23] Lee, J. Bagheri, B., & Kao, H.A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3, pp. 18-23.
- [24] Chun, I. Lee, H. Kim, W., & Lee, E. (2014). SMART-CPS: Self-managed reliable system development method for cyber-physical systems. *International Information Institute, Tokyo, Information*, 7(3), pp. 1025-1030.
- [25] Balaji, S., & Murugaiyan, M.S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, 2(1), pp. 26-30.
- [26] Graessler, I. (2017). A new V-Model for interdisciplinary product engineering. In: *Proceedings of the 59th Ilmenau Scientific Colloquium on Engineering for a Changing World*, Technische Universität Ilmenau, 59, pp. 1-6.
- [27] Mathur, S., & Malik, S. (2010). Advancements in the V-Model. *International Journal of Computer Applications*, 1(12), pp. 30-35.
- [28] Graessler, I., Hentze, J., & Bruckmann, T. (2018). V-models for interdisciplinary systems engineering. In: *Proceedings of the DESIGN 2018 15th International Design Conference*, pp. 747-756.
- [29] Klötzer, C., Weißenborn, J., & Pflaum, A. (2017). The evolution of cyber-physical systems as a driving force behind digital transformation. In: *Proceedings of 19th Conference on Business Informatics*, 2, pp. 5-14.
- [30] Horváth, I. (2014). What the design theory of social-cyber-physical systems must describe, explain and predict? In: *An Anthology of Theories and Models of Design*. Springer. pp. 99-120.

- 
- [31] Horváth, I. (2012). Beyond advanced mechatronics: New design challenges of Social-Cyber-Physical systems. In: *Proceedings of the ACCM Workshop on Mechatronic Design*. pp. 1-20.
- [32] Baheti, R., & Gill, H. (2011). Cyber-physical systems. *The impact of control technology*, 12(1), pp. 161-166.
- [33] Horváth, I. (2017). A method for systematic elaboration of research phenomena in design research. In: *Proceedings of the 21st International Conference on Engineering Design*, Vancouver, pp. 1-10.
- [34] Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, 6, pp. 1-10.
- [35] Park, K.J., Zheng, R., & Liu, X. (2012). Cyber-physical systems: Milestones and research challenges, *Computer Communications*, 36, pp. 1-7.
- [36] Akhras, G. (2000). Smart materials and smart systems for the future, *Canadian Military Journal*, 1, pp. 25-31.
- [37] Huang, L. (2017). Intelligence of smart systems: Model, bounds, and algorithms, *IEEE/ACM Transactions on Networking*, 25, pp. 2960-2973.
- [38] Kim, K.D., & Kumar, P.R. (2012). Cyber-physical systems: A perspective at the centennial, *Proceedings of IEEE*, 100, pp. 1287-1308.
- [39] Brachman, R.J. (2002). Systems that know what they're doing, *IEEE Intelligent Systems*, 17, pp. 67-71.
- [40] Khan, Z., Ali, A.B., & Riaz, Z. (2014). Computational intelligence for decision support in cyber-physical systems (Springer), *Studies in Computational Intelligence*, pp. 540.
- [41] Penas, O., Plateaux, R., Patalano, S., & Hammadi, M. (2017). Multi-scale approach from mechatronic to Cyber-Physical Systems for the design of manufacturing systems, *Computers in Industry*, 86, pp. 52-69.
- [42] Duval, S., & Wicklund, R.A. (1972). A theory of objective self-awareness, Academic Press.
- [43] Lewis, P.R., Chandra, A., & Glette, K. (2016). Self-awareness and self-expression: Inspiration from psychology. In: *Self-aware Computing Systems: An Engineering Approach*, Springer International Publishing, Cham, pp. 9-21.
- [44] Horváth, I. (2019). Design challenges of smart cyber-physical systems. ICONN Awardee Keynote Lecture at IcoRD '19, January 9, Bangalore, India.
- [45] Agne, A., Happe, M., Lösch, A., Plessl, C., & Platzner, M. (2014). Self-awareness as a model for designing and operating heterogeneous



- multicores, *ACM Transactions on Reconfigurable Technology and Systems*, 7, pp. 1-18.
- [46] Black, W., Haghi, P., & Ariyur, K. (2014). Adaptive systems: History, techniques, problems, and perspectives, *Systems*, 2, pp. 606-660.
- [47] Juzgado, P.B., & Bonnema, G.M. (2008). On the origin of evolvable systems; Evolvability or extinction. In: *Proceedings of the TMCE 2008*, pp. 1-12.
- [48] Li, C., Rusák, Z., Horváth, I., & Ji, L. (2016). Validation of the reasoning of an entry-level cyber-physical stroke rehabilitation system equipped with engagement enhancing capabilities, *Engineering Applications of Artificial Intelligence*, 56, pp. 185-199.
- [49] Smirnov, A., Sandkuhl, K., Shilov, N., & Telsya, N. (2015). Context variation for service self-contextualization in cyber-physical systems. In: *Lecture Notes in Business Information Processing*, Springer International Publishing, pp. 309-320.
- [50] Kim, K.D., & Kumar, P.R. (2013). An overview and some challenges in cyber-physical systems, *Journal of the Indian Institute of Science*, 93(3), pp. 341-351.
- [51] Buchholz, T., Küpper, A., & Schiffers, M. (2003). Quality of context information: What it is and why we need it. In: *Proceedings of the 10th HP-OVUA Workshop*, pp. 1-14.
- [52] Kaemarungsi, K., & Krishnamurthy, P. (2012). Analysis of WLAN's received signal strength indication for indoor location fingerprinting, *Pervasive and Mobile Computing*, 8, pp. 292-316.
- [53] Weiser, M. (1993). Some computer science issues in ubiquitous computing, *Communication of ACM*, 36, pp. 75-84.
- [54] Lewis, P.R., Chandra, A., & Glette, K. (2016). Self-awareness and self-expression: Inspiration from psychology. In: *Self-aware Computing Systems: An Engineering Approach*, Springer International Publishing, Cham, pp. 9-21.
- [55] Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., & Steggles, P. (1999). Towards a better understanding of context and context-awareness, *Lecture Notes on Computer Science*, pp. 304-307.
- [56] Dey, A.K. (2001). Understanding and using context, *Personal and ubiquitous computing*, 5(1), pp. 4-7.
- [57] Brown, P.J. (1996). The stick-e document: A framework for creating context-aware applications. In: *Proceedings of the Sixth International Conference on Electronic Publishing*, pp. 259-272.

- [58] Berardinelli, L., Cortellessa, V., & Di Marco, A. (2010). Performance modeling and analysis of context-aware mobile software systems. In: *Fundamental Approaches to Software Engineering*, Springer, Berlin Heidelberg, pp. 353-367.
- [59] Hong, J.I., & Landay, J.A. (2001). An infrastructure approach to context-aware computing, *Human-Computer Interaction*, 16, pp. 287-303.
- [60] Brown, P.J., Bovey, J.D., & Chen, X. (1997). Context-aware applications: From the laboratory to the marketplace, *IEEE Personal Communications*, 4(5), pp. 58-64.
- [61] Brown, P.J., Bovey, J.D., & Chen, X. (1997). Context-aware applications: From the laboratory to the marketplace, *IEEE Personal Communications*, 4(5), pp. 58-64.
- [62] Schilit, B.N., & Theimer, M.M. (1994). Disseminating active map information to mobile hosts, *IEEE Network*, 8, pp. 22-32.
- [63] Ryan, N., Pascoe, J., & Morse, D. (1997). Enhanced reality fieldwork: The context aware archaeological assistant. In: *Proceedings of the 25th Anniversary Conference Computer Applications and Quantitative Methods in Archaeology*, University of Birmingham, pp. 269-274.
- [64] Anagnostopoulos, C. B., Tsounis, A., & Hadjiefthymiades, S. (2007). Context awareness in mobile computing environments. *Wireless Personal Communications*, 42(3), pp. 445-464.
- [65] Li, Y., Song, Y., & Horváth, I. (2014). Development of a framework for information acquisition and processing in cyber-physical systems. In: *Proceedings of the 10th International Symposium on Tools and Methods of Competitive Engineering*, Budapest, pp. 1255-1270.
- [66] Horváth, I., Li, Y., Rusák, Z., van der Vegte, W.F., & Zhang, G. (2016). Dynamic spatial context computation for time-varying process scenarios, *Journal of Computing & Information Science in Engineering*, 17, pp. 1-12.
- [67] Kokinov, B. (1995). A dynamic approach to context modeling. In: *Proceedings of the IJCAI-95 Workshop on Modeling Context in Knowledge Representation and Reasoning*, Laforia, pp. 199-209.
- [68] Palau, M., Gómez-Sebastià, I., Ceccaroni, L., Vázquez-Salceda, J., & Nieves, J.C. (2010). A framework for the development and maintenance of adaptive, dynamic, context-aware information services. In: *Proceedings of the International Conference on Agents and Artificial Intelligence*. Springer, Berlin, Heidelberg, pp. 227-239.
- [69] Roque, L., Almeida, A., & Figueiredo, A.D. (2004). Context engineering: An

- IS development research agenda. In: *Proceedings of the ECIS 2004 Conference*, pp. 1-15.
- [70] Osborn, W., & Hinze, A. (2014). TIP-tree: A spatial index for traversing locations in context-aware mobile access to digital libraries, *Pervasive and Mobile Computing*, 15, pp. 26-47.
- [71] Saeedi, S., Moussa, A., & El-Sheimy, N. (2014). Context-aware personal navigation using embedded sensor fusion in smartphones, *Sensors*, 14, pp. 5742-5767.
- [72] Jose, A.C., & Malekian, R. (2015). Smart home automation security: A literature review, *Smartcr*, 5, pp. 6-10.
- [73] Heyes, D., Daun, T.J., Zimmermann, A., & Lienkamp, M. (2015). The Virtual Driving Coach - Design and preliminary testing of a predictive eco-driving assistance system for heavy-duty vehicles, *European Transport Research Review*, 7, pp. 1-13.
- [74] Ibrahim, A.M., Venkat, I., Subramanian, K.G., Khader, A.T., & Wilde, P.D. (2016). Intelligent evacuation management systems: A review, *ACM Transactions on Intelligent Systems & Technology*, 7(3), 36.
- [75] Gallacher, S., Papadopoulou, E., Abu-Shaaban, Y., Taylor, N.K., & Williams, M.H. (2014). Dynamic context-aware personalisation in a pervasive environment, *Pervasive and Mobile Computing*, 10, pp. 120-137.
- [76] Wu, B.F., Yao, S. J., Hou, L.W., Chang, P.J., Tseng, W.J., Huang, C.W., Chen, Y.S., & Yang, P.Y. (2016). Intelligent shopping assistant system. In: *Proceedings of the 2016 International Automatic Control Conference*, pp. 236-241.
- [77] Lehsten, P., Bader, S., & Tavangarian, D. (2014). Casa-context-aware service access. *Journal of Integrated Design and Process Science*, 18(1), pp. 21-38.
- [78] Braunhofer, M., Elahi, M., Ricci, F., & Schievenin, T. (2013). Context-aware points of interest suggestion with dynamic weather data management. In: *Information and Communication Technologies in Tourism*, pp. 87-100.

# Chapter 2

## **Aggregation of knowledge and exploring requirements about dynamic context information processing**

### **2.1. Objectives and methodology of research cycle 1**

#### **2.1.1. Objectives of this research cycle**

The overall objective of the first research cycle was knowledge aggregation concerning DCIP. This research cycle pursued a robust underpinning knowledge platform and explored requirements for dynamic context information processing. There were three related sub-objectives also specified: (i) refinement of the objectives, (ii) critical exploration of the knowledge with regard to the state-of-the-art of DCIP, and (iii) aggregation of requirements for the pursued dynamic context management mechanism. On the one hand, the purpose of the knowledge aggregation work is to build up a knowledge platform for understanding the essence of dynamic context information, summarizing the existing approaches to DCIP, identifying limitations of existing solutions and developing the research vision for building our own approach. The targeted resource of knowledge is considered as the published literature where the technical details of relevant computational mechanisms are proposed. On the other hand, the purpose of the requirement exploration work is to specify both general and technical requirements for DCIP, which form a basis for conceptualizing a new computational solution. To this end, general requirements were explored through a literature study, while the technical requirements were found based on a case study. Therefore, a real-life application case (indoor fire evacuation guiding) was selected and practical problems in various execution scenarios were investigated for exploration of requirements.

### **2.1.2. Methodology applied in this research cycle**

Driven by the research objective, the research conducted in this research cycle mainly focus on two topics: knowledge aggregation and requirement exploration. To aggregate knowledge, the methodological framing employed is the research in design context (RiDC). According to the principles of RiDC, the research activity includes two main phases, namely an explorative phase and a confirmative phase. In the explorative phase, the focused topic was decomposed into several interrelated sub-domains of interest. Knowledge related to the states of each of the sub-domains as aggregated and findings from the aggregations were discussed, respectively. Then, a synthesis of the findings was conducted, based on which a summary of the generated knowledge was obtained. In the confirmative phase, the findings from the knowledge aggregation were analyzed considering the fulfillment of the characteristics of smart cyber-physical systems, based on which the limitations of existing approaches can be known. After this, possible research and development opportunities of DCIP towards self-awareness and self-adaptation were identified.

Considering the objective to explore design requirements, the research is methodologically framed as design inclusive research. There are three phases included in the research cycle, namely (i) an explorative phase, (ii) a constructive phase, and (iii) a confirmative phase. The specific needs for DCIP were explored by surveying existing I-CPS applications and smart systems. The explored specific needs were used as a starting point of generating innovative (general) requirements. In addition, a real-life application case (indoor fire evacuation guiding) was selected and practical problems in various execution scenarios were investigated through a demonstrative simulation. By comparing the cases regarding (i) fire evacuation without personalized informing and (ii) fire evacuation with context information processing and personalized informing, technical requirements regarding DCIP were generated, which extend the set of specific requirements with additional ones concerning self-awareness and self-adaptation in I-CPSs. At the end of the research work, the generated requirements were clustered and their consistency was checked. The validation of the set of requirements was done by critical system thinking and feasibility analysis.

## **2.2. Knowledge aggregation for computation of dynamic context information**

### **2.2.1. Reasoning model of the survey**

Many survey papers dealing with specific topics related to the context information processing have been published. For instance, Chen and Kotz surveyed the

research related to context information processing in mobile computing applications, which is one of the application domains of CPSs [1]. Baldauf et al. generalized the common architectural elements of systems, which are able to process context information [2]. Hong et al. classified the state-of-the-art of context information processing by using a classification framework [3]. Different from other similar survey works, this survey work focuses on the dynamism of context in computation.

Related to the knowledge aggregation, a web-based survey was completed, supported by a keyword tree as a reasoning model. In I-CPSs, the computational operations may include (i) building awareness, (ii) computational reasoning and (iii) messaging operations. As a basis for these three essential operations, a proper information presentation mechanism is normally required to translate the acquired heterogeneous context data into a computer-processible way. This is referred to as the representation of dynamic context information. This operation should be done before the above-mentioned three essential computational operations. In addition, to enable implementing and utilizing these operations in real-life application, various software enablers are proposed, which integrate some of the operations in a cohesive manner. The software enablers achieved a high-level synthesis of DCIP and support the development of smart CPSs. For this reason, the existing software enablers were included in our review work.

The reasoning model for the literature study is shown in Figure 2.1. The main focus area of this research project was decomposed into five interrelated sub-domains of interest. It was used to identify and to correlate sub-domains of interest from the main focus and the relations of the sub-domains. A detailed explanation of the sub-domains is given as follows.

- **Representation of dynamic context information**

The representation considers various approaches to describe the dynamic context in computation. This is the computational fundamental of DCIP. According to the ways of representation reported in the literature, this sub-domain is further decomposed into three categories, (i) the methods of acquiring dynamic context information, (ii) the model-based representations of dynamic context, and (iii) the data-driven representation of dynamic context.

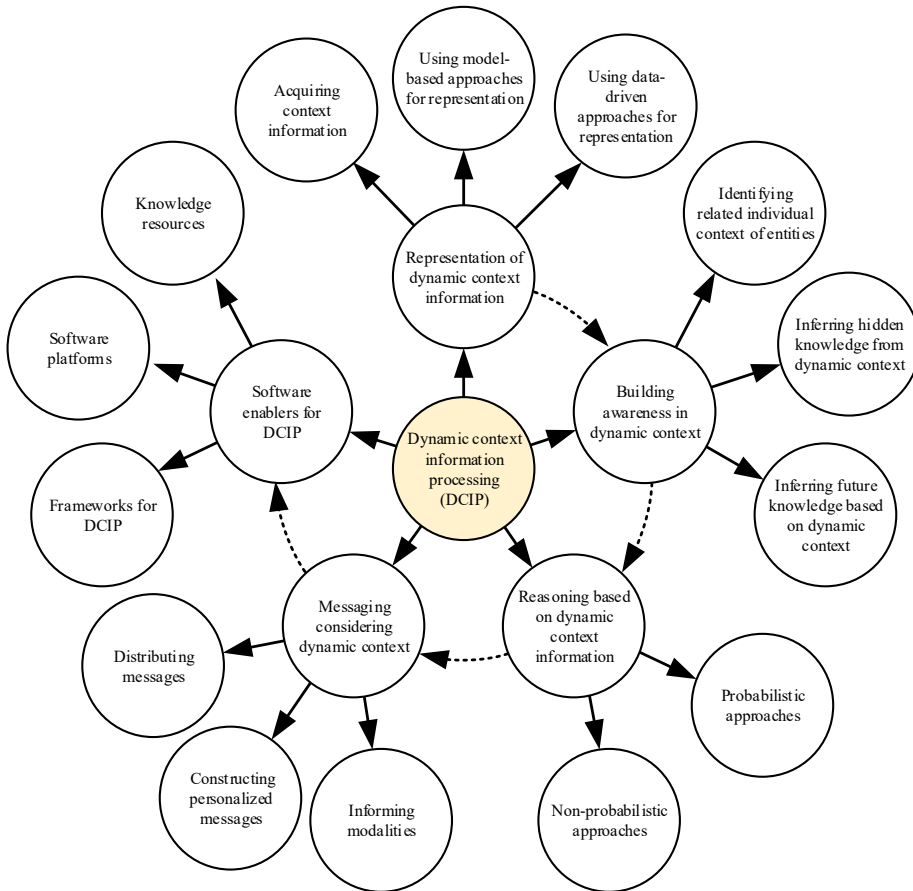
- **Building awareness in dynamic context**

Due to the dynamic nature of context, the relevance of any piece of information to a given entity might be changing in time-varying scenarios. For instance, when a person enters a building, the outdoor temperature, as a piece of information may not be relevant to the person. For this reason, identification of the related individual context of entities is the first aspect we considered in this sub-domain. In addition, to be aware of the dynamic context and interpret the meaning of

dynamic context, various inferring operations are required and semantic knowledge should be generated based on the context changes. To achieve this, another two sub-domains are specified: inferring intrinsic (or even) hidden context knowledge (e.g. uncertainty, probability, implication) and future knowledge (e.g., near-future context) from a particular representation of dynamic context.

- **Reasoning based on dynamic context information**

To distinguish the approaches applied in inferring operations, the reasoning operations are specified as decision-making operations, based on which solutions to the main objective of the concerned system are generated. In the case of I-CPSs, the solutions are expected to be given and executed by the stakeholders. The related literature work reported on approaches to reasoning with dynamic context, considering both probabilistic reasoning, and non-probabilistic reasoning.



**Figure 2.1.** Reasoning model for the literature study

- **Messaging operations considering dynamic context of stakeholders**

In I-CPSs, messaging operations may include (i) to transfer the generated solutions to various messages written in neutral language, (ii) to distribute the generated messages through proper method to enhance the communication with devices and (iii) to present the messages to the concerned stakeholders with proper informing modalities to enhance the informing to stakeholders. The survey work addressed on these three aspects.

- **Software enablers for DCIP**

The last interest domain was focused on the DCIP from viewpoint of integrated computation. Some of the above-mentioned functions may be integrated as software enablers, which are reusable for multiple applications. Thus, we reviewed existing software enablers that employ certain computational mechanisms for DCIP to see how the functions are integrated together. According to the various types of software enablers, we considered (i) computational frameworks, (ii) software platforms, and (iii) reusable knowledge resources.

## **2.2.2. Approaches to representing dynamic context**

### **2.2.2.1. Acquiring dynamic context information**

Context information acquisition is employed to sense, retrieval, and query context information for computations[4]. Context-aware functions will not properly fulfill if the descriptive information about the real world is not correctly obtained [5]. Generally, there are three ways to acquire context information, including (i) sensing, (ii) deriving from existing models or context information carriers, and (iii) manual input from system users [6]. As indicated in the literature, major concerns of context information acquisition are (i) responsibility for initializing context acquisition, (ii) frequency of acquisition to be performed, (iii) the source where context information comes from, (iv) the type of sensor used for collecting context information and (v) the process how context information is acquired [7]. In case of dynamic context, new requirements and challenges emerge out and should be considered for the design of context-aware applications.

Many types of information have been recognized as the context in the literature, such as location, temperature, presence, motion, pressure, illumination, etc. Among those, time and space are considered as the most fundamental aspects [8]. To depict the dynamic context, time should not be processed as an independent parameter, such as time of day. Instead, time should be used to enable the calculation of contextual changes and to infer the temporal relations of contextual facts. It means that every piece of acquired dynamic contextual data has the attribute of timeliness and should be treated according to the time it is representing. In addition, since dynamic context may change rapidly, the behavior



of acquiring contextual data may be challenged, e.g. frequency of acquisition. In the literature, some solutions addressing these challenges have been proposed, such as acquiring dynamic contextual data with an adaptive sampling rate of sensors [9] and managing the delay of sensing [10]. More research effort should be paid to focus on the responsibility, frequency, and fusing of multi-source dynamic context data.

Data acquired from physical sensors without any further interpretation or management can be meaningless, trivial, vulnerable to small changes, or uncertain [11]. When target contextual data is aggregated, interpretation of the data to form a primary understanding of dynamic context is an important step. Dynamic context needs to be interpreted based on the data aggregated at multiple sampling time points, which may result in a large volume of streaming data [12]. Accordingly, more computational time will be consumed and the timeliness (temporal validity) of acquired information is challenged [13]. Furthermore, to acquire proper dynamic context information, the relation among pieces of information is the basis. However, poor performance or unexpected behavior may be experienced for all kinds of reasons following the deployment of a sensor network [14]. For example, an I-CPS may experience the sudden death of sensor nodes, unsatisfactory implementation of application logic, topology changes and mutated network conditions. The quality of acquired information (e.g. accuracy, completeness and reliability, etc.) has a great influence on the associated processing procedures [15]. Towards this, many solutions have been reported in the literature to enhance the quality of information, e.g. accuracy [16], fault-tolerance [17], adaptive sensing strategy [18], etc.

#### **2.2.2.2. Model-based representation of dynamic context**

Reasoning in context assumes a proper representation of context information. One straightforward approach is developing a context model and applying the method of model-based information structuring. Though a formal context model allows the development of effective computational algorithms, construction of models is challenging in the cases of dynamic context and emergent context. Most of the context information models have been proposed for handling static context and altering context information. The essence and capabilities of these context information models have been discussed in various survey papers [3] [19] [2]. As the findings of these studies show, the models proposed for processing static and/or altering context information, typically cannot be adopted to represent and process dynamic context information. Although some research projects, such as those presented in [20]-[22], considered time as an aspect of context, physical time is usually regarded as an independent parameter of context, and the issues of time-dependent computation of contextual changes are not addressed sufficiently yet. Certain event-based models were also proposed, which consider dynamic

context as changes of information, e.g. the models discussed in [23] and [24]. This type of models is suitable for rule-based reasoning and can hardly be used to investigate hidden context or predict future contexts.

Many models and mechanisms are reported that capture and reason with the rapid changes of context. Yin et al. designed a temporal context-aware mixture model (TCAM) to capture the related context information according to the preferences of people at a given point in time [25]. Using context modeling language (CML), Henriksen and Indulska proposed an approach, which supports the dynamic creation, activation and deactivation of contextual elements and associations of the elements in the context model [26]. Mohan and Singh proposed a context ontology model, which captures rapidly changing context information and incorporates it into a knowledge base [27]. Another example is iConAwa, in which contextual changes were modeled using web ontology language (OWL) and rule-based reasoning was performed over the context ontologies [28]. Although these approaches have been successfully applied in applications requiring handling of dynamic context, contextual states and changes are modeled as independent variables. Actually, the acquired contextual states and changes are interrelated and should be synthetically integrated into a more flexible and application-independent model.

### **2.2.2.3. Data-driven representation of dynamic context**

While two decades ago, the attention in context information engineering was given to the formal model or at least predefined frameworks, the last decade witnessed a move towards more lean and flexible forms of representation of (dynamic) context information. Earlier, applications typically used modeling tools and toolkits, together with conceptual frameworks, entity and attribute libraries, and ontologies to perform context acquisition, preprocessing, storing, and reasoning tasks. The emergence of big data analytics and pattern-based reasoning created new opportunities for context information management. The main differences are as follows: (i) extraction of information from context data happens at run time, (ii) tentative (probabilistic) patterns, rather than crisp model architectures are considered, (iii) Internet proliferated as a rich source of context information, (iv) context information management has become distributed, and (v) data-driven applications have been developed instead of model-based ones. For instance, Chen, G. et al. proposed a data-centric infrastructure design to support context-aware applications [29]. A dynamic data-driven approach to extract context from sensor data for context-aware pattern recognition was discussed in [30]. By developing the VisContext framework, Chandrashekar, M., & Lee, Y. intended to implement context learning from large scale multimedia data. This approach includes (i) the discovery of contextual (ii) the association of terms and images, (iii) the visualization of image context, and (iv) the classification of

images based on contexts. The issue of making temporal abstraction based on a knowledge-based framework and higher-level concepts from time-stamped data in a context-sensitive manner was addressed in [31].

Lu et al. hierarchically divided the global physical environment into several physical spaces (domains and smart spaces) [32]. A ubiquitous gateway (U-gate) was deployed in each smart space for distributed context information and service management. The distributed toolkits support common features required by context-aware applications, namely, to capture, access and store context information independently. As an inter-space context discovery platform, Orion employed a peer-to-peer architecture to connect basic units called context gateways (CGs), which operated in a distributed manner [33] [34]. CGs are able to manage local context information and to communicate with each other through a global area network. Moreover, the basic infrastructure of context management can be designed considering the servers deployed in the physical environment, which can be a room, an entire building, or a regular area in the outdoor environment [35]. Each room server was considered as a proxy for service discovery and was used to deliver context information concerning the user to a ubiquitous fashionable computer, which is a wearable device.

In line with the advancement of ubiquitous computing technologies, the need for context-aware operation of ubiquitous devices became salient. Various computational methods have been proposed for context-dependent ubiquitous data management [36]. The interrelated issues of context selection and context reasoning were addressed in different projects [37] [38]. Context awareness has come to the limelight in association with self-managing systems [39]. The issue of verification and run time validation of context representations have been addressed in [40] and [41]. A lightweight, energy-efficient, distributed and adaptive event reasoning model was proposed, which can be used on various IoT devices (sensors/actuators) in multiple-context perspective under uncertainty. The devices are supposed to locally, but collaboratively process contextual data, and infer and reason about the appearance of a specific phenomenon (event) in the environment [42]. A context service framework, which includes a context acquisition layer, a context management layer, an event management layer, and a context access layer was developed to support context-aware computing in complicated application requirements [43].

### **2.2.3. Approaches to building awareness in dynamic context**

#### **2.2.3.1. Determining the individual context of entities**

Context information is generally recognized as relevant information associated with an entity or a process [44]. Starting out with this interpretation, the basic

requirement for context representation is to aggregate relevant information elements concerning, e.g. entities, attributes, and relations, to represent the context. In many context-aware systems with unchanging system objectives, the relevance of dynamic context elements is more or less predefined, e.g. as in the work presented in [45] and [46]. However, the representation of dynamic context information becomes a challenging issue when the operational objective of a system dynamically changes during run-time, since the relevance of the context information elements changes as well. To solve this problem, Taconet et al. presented a context representation mechanism that is able to update itself during run-time in order to address new application requirements [47]. Jaroucheh et al. developed a framework, called CANDEL, which dynamically combines various contextual ontologies related to the tasks completed by the given system [48]. In the project called 'Daidalos', preferences of the users were considered as an aspect of the user context, and were refined and updated by a machine-learning mechanism whenever (i) the task of the system changed, and (ii) the user's context changed [49].

In many research projects, dynamic context representation was derived by evaluating the relevant contextual information elements in real-time. Wan et al. proposed a solution that recognizes individual context by identifying and evaluating the situations within a given distance from a reference location [50]. An evolving context schema was presented by Forsström and Kanter, which represents the context in the form of progression and evolution [51]. The relevance of context information was calculated based on a context proximity function. Wolf et al. proposed a dynamic context provisioning mechanism, which is able to judge the relevance of any piece of information with regard to the user's context in a real-time manner [52]. Euzenat et al. proposed a context representation mechanism using semantic web language. It supports the dynamic integration of relevant information resources when the environment of a person is changed (e.g., a person enters into a room) [53]. Zhu et al. proposed a reasoning mechanism, which validates a particular set of values to represent the context of people at any point in time [54].

It can be seen from the results of the related research work that representing dynamic context is still in a relatively early stage. The two main ideas of the proposed solutions are: (i) predefining a set of aspects and values for representing different contexts that the system might face, and (ii) selecting a set of aspects and values at a given point in time to represent a particular context when certain conditions are detected. In general, the currently existing dynamic context representations are still in the lack of flexibility and generality. They construct and prioritize information based on spatial relationships of the entities without considering the temporal and attributive relationships and their implications.

### 2.2.3.2. Inferring hidden knowledge from dynamic context

The dynamic context is supposed to change continuously [55]. The changes of the context information can be used to infer additional (hidden) knowledge, which cannot be obtained directly from sensor readings or warehoused data. This is a way to infer high-level context from raw contexts [56]. Many papers in the literature interpret context dynamics based on shifting temporal contexts and use it to infer uncertain context. Reportedly, there are two different sources of uncertainty concerning contexts: (i) aleatoric uncertainty, and (ii) epistemic uncertainty [57]. Aleatoric uncertainty is normally referred to as randomness, while epistemic uncertainty is considered to be caused by a lack of knowledge. Therefore, aleatoric uncertainty typically characterizes emergent contexts, whereas epistemic uncertainty characterizes dynamic context. The reason for the latter claim is that epistemic uncertainty is associated with inferring based on temporal relations of contextual data or information. Bobek and Nalepa proposed a mechanism for reasoning out missing or ambiguous data using time-based operators, and a rule-based reasoning engine [58] [59]. In spite of inconsistent context data, the solution proposed by Zhang et al. is able to infer from context information by using the extended evidence theory [60]. Ranganathan et al. applied a Bayesian network and fuzzy logic to infer knowledge under sensing uncertainty by relying on the probability of facts under dynamically changing circumstances [61].

Papers dealing with inferring other types of hidden knowledge based on context dynamics can also be found in the literature. Chahuara et al. modeled temporal relations of the recognized events by chaining property of OWL to represent the ‘after’ and ‘before’ relations of events [62]. A branch of research concentrates on inferring dynamic information concerning the cognitive actions of people, e.g. preferences and emotion. The goal is to explore hidden pieces of knowledge concerning their dynamic behaviors. As a representative example of the results in this branch of research, Kwon and Kim proposed a mechanism to infer personal preferences, which combines an analytic hierarchy processing and a case-based reasoning engine [63]. Liu applied Gaussian process to infer temporal preference of people for personalized recommendation [64]. A machine-learning approach was used to infer the emotional condition of people based on their observed behavior in [65].

As a takeaway, we claim that inferring hidden knowledge from dynamic contexts is not a fully explored and understood issue yet. At the same time, it is recognized that inferring makes it possible for a system to build awareness of ‘what is happening in context’, ‘what is the meaning of context’ and ‘what the implications of a specific context are’. Inferring explores and makes the implications of a particular context tangible and provides input for system level reasoning and decision making. Eventually, the inferred semantic knowledge

enables a system to devise an adaptation strategy for itself, which best matches the operational objectives in specific dynamic contexts. However, this latter affordance has received only rather limited attention in system engineering and cognitive engineering research so far.

### **2.2.3.3. Inferring future knowledge based on context changes**

Inferring future context knowledge can be done by applying one or another approach of predictive computation. However, the efforts in this direction are still in a premature stage. The literature presents two somewhat resembling concepts, which are called: (i) context-aware prediction, and (ii) context prediction, respectively. The interest of the former research is to predict content information in context, e.g. future locations of users [66], whereas the interest of the later is in predicting the future of context, e.g. available devices [67]. Prediction based on dynamic contexts just recently appeared in the literature. Liu et al. proposed a unified model, called dynamic metric embedding with temporal and spatial factors (DME-TS) to predict user's interest, which integrates three aspects of future context: (i) temporal non-uniformness, (ii) temporal sequencing, and (iii) spatial influence [68]. Mu et al. used a neural network to predict the probability of human errors based on scoring the common performance conditions of operators in time-varying context [69]. To support the development of autonomous vehicles, Neogi et al. applied probabilistic models to predict pedestrians' behavior based on spatial context and the interactions of pedestrians with vehicles [70]. Ridhawi et al. applied the Dempster-Shafer theory to predict users' future locations based on personal context, i.e. interests of the users, scheduling constraints, goals, and tasks, and dynamic environmental context, e.g. change of rooms [71].

In order to become proactive and to act ahead of the actual change of the context or according to the change, future contexts have to be predicted [72]. Actually, inferring future context is regarded as the ultimate goal of building context awareness [73]. Context prediction can be done based on actual information (i.e. current context) or past information (i.e. historical context). Future context is normally reasoned out from a chunk of knowledge aggregated through activities such as context modeling, context representation, and hidden-context inferring [74]. Both model-based approaches (such as Markov chains, Bayesian networks, neural networks, and Kalman-filter) and non-model-based approaches (such as middleware solutions, and general frameworks) were proposed for context prediction in the literature [75]. The existing approaches to context prediction mainly address data level prediction (static context) and event level prediction (altering context). It is fair to mention that these approaches are suitable for predicting the context only in applications in which every single person can be considered as an isolated entity (computational object) and the influence caused

by other objects and/or other dynamic situations can be neglected. Characteristic examples are such as recommender systems [76] and smart home systems [77].

### **2.2.4. Approaches to reasoning with dynamic context**

Context-dependent reasoning needs a complex problem-solving mechanism to enable the I-CPSs to make decisions according to the structured context models. In context-aware applications, dynamic contexts are often imprecise and incomplete due to unreliable connectivity, user mobility and resource constraints [60]. Due to the uncertain nature of the physical world and the imprecise measurements taken on it, some applications might take the uncertainty of context information into the account of context-dependent reasoning. Therefore, the existing context-dependent reasoning approaches can be classified into two categories: non-probability-based approaches and probability-based approaches, by considering if the approaches are able to reason on the uncertainty of context information.

#### **2.2.4.1. Probabilistic reasoning**

Bayesian network is able to exert qualitative and quantitative description in order to analysis the probability of uncertainty happening over time [78]. Wang et al. [79] proposed Bayesian networks in the complex situation assessment of intelligent traffic incident management (TIM) system. The forward propagation algorithm of Bayesian network was used to assess the traffic incident according to dynamic context information collected by real-time multi-source sensors. The backward propagation algorithm can analyze the cause of traffic incidents and predict the happening of a follow-up incident. In the context sharing systems, high-level contexts, such as user activities and emotions, can be recognized by Bayesian network models to handle uncertainty in mobile environment according to user's dynamic location or movement [80][81]. The effectiveness of these approaches depends on explicit Bayesian network models with appropriate given probabilities of elements.

However, a general model, which towards multiple applications, is hard to develop. Dempster-Shafer theory [82], a mathematical theory of evidence, is commonly used as a tool to propagate uncertainty values and consequently provide an indication of the interferences [83]. In Smart Grid, an evidence-based approach that supports automated decision making is able to detect anomaly [84]. The evidence-based decision-making is able to infer the true system states and make decisions without defining an explicit model of the system, just based on some evidence (measurements).

Compared to the Bayesian approaches, the evidence-based approach does not require all the hypotheses, so it is more appropriate to handle the uncertainty

problem [85]. The main drawback is the lack of semantics in context representation [60]. Hidden Markov Model (HMM) is a suitable mathematical model to address runtime prediction, such as predicting routines of elderly people in order to preheat the bed before the elderly people go to sleep [86]. The HMM enables the system to compute based on context information regarding high-level abstractions [87]. In the HMM, explicit durations for observation is predefined. If the state transition of the HMM fails to be captured, there might cause the missing of situations. Fuzzy logic is another tool to make use of imprecise context information. The fuzzy logic-based approach can determine and reason the current situation of a user by considering the historical and current locations of the user [88]. Fuzzy logic is useful for defining the conceptual states of a primitive context to enable human-like reasoning. However, precise interpretation of a modifier is application specific [87].

#### **2.2.4.2. Non-probabilistic reasoning**

The simplest non-probability-based approach used in context-dependent reasoning is the rule-based approach. System states, parameters or a particular information value are predefined as rules. When one of the rules is observed, corresponding reactions will be triggered depending on a tree-based retrieve engine. The rule-based approach is able to resolve simple problems, in which limited situations are handled [89][90][91]. Event-based reasoning is similar to rule-based reasoning. The main difference is that the event-based reasoning contains an inference engine, which infers events based on pieces of aggregated information. Predefined operations are executed automatically upon the identification of situational events, such as location information of users and nearby people [92]. Rules-based reasoning is easily understood and widely used. However, it is difficult to handle dynamic, ambiguous, and imperfect context information.

Another typical approach is case-based reasoning (CBR), which stores cases in a general knowledge repository that offers operations, such as reasoning and retrieval. A ‘pattern recognition-conflict resolution-action’ cycle is repeated until the solution is reached, or no applicable case can be found in the case base [93]. In this way, past experiences (cases) can be used to generate similar cases and can be adapted to suggest new solutions [94] [95]. As presented in [96], case-based reasoning adopts a static system structure, which may raise complication at adaptation to dynamic context information. A dual-CBR approach has been presented in [97] for recommending music depending on the listener’s context. Time-based reasoning approach can solve problems in model-based system engineering procedures [98]. The occurrence time of all system states is calculated and arranged into a time-based tree. This approach is designed based on exact time-based predictions of future system states. This approach is not able



to resolve issues regarding complex numerical calculation, e.g. controlling the speed of a car.

Many implementations of problem-solving approaches use vectors to represent the dynamic spatial context. In a dynamic vehicle routing approach, traffic condition of any location is sensed and represented by a vector, which includes four aspects of context: location, traffic flow, traffic density and speed of vehicle [99]. The travel times of a car to the destination through different routes can be calculated based on the vectors. Zhou et al. proposed an adaptive traffic light control scheme that adjusts the sequences and length of green lights in multiple intersections based on the real-time traffic situations [100]. The dynamic context was encapsulated into diverse vectors, including traffic volume, waiting time, number of stops, and traffic flow. The main drawback of the vector-oriented approach is that when the dimensions of the context vector increases, the complexity of computation will exponential growth. In object-oriented reasoning approaches, individual context information is constructed as parameters of a profile. Reasoning algorithms demonstrate on the changes of the parameters according to inference rules, such as the potential field of safety around vehicles [50] and personal quality of service (QoS) files [45]. Object-oriented approaches are suitable for applications that require individual optimal solutions. They can hardly be used in applications that require solutions for the purpose of group optimization.

## **2.2.5. Approaches to context-dependent messaging**

### **2.2.5.1. Generating personalized messages based on dynamic context**

Message generation is a subfield of natural language generation (NLG). Based on the knowledge of computational linguistics and artificial intelligence, it synthesizes texts in natural languages to satisfy certain communication requirements [101]. Context-dependent generation of messages should consider the context of users in order to adapt to their needs, which is normally addressed from two aspects: user preferences and user environment [102]. Towards this end, different approaches have been suggested for message generation in the literature. A typical example is the shopping assistant system (ISAS) that is able to follow and provide instant messages to users according to their shopping list, the arrangement of the shopping mall and the monitored environmental information in real-time [103]. Reported in [104], the context-aware recommender systems (CARS) provides information to users concerning weather conditions or the nearby surroundings depending on the locations of the users. In this line of systems, we should mention the context-aware tourist informing system (CATIS), which provides information to the users based on their individual location, speed

of motion, the direction of traveling, personal references, device type and the time of day [105]. In addition, a location-based recommendation system was proposed by [106] that generates a list of locations for the user to visit and supports social networking of the user. The above-mentioned approaches were articulated with the assumption of using template-based generators. Most of the template-based generators reported in the literature are domain-dependent. The main drawback is the need to create, maintain and update templates for the use in multiple applications.

In addition, using model-driven approaches to the generation of context-dependent messages attracts much research attention in recent years [107]. The major research challenges are (i) interpretation and utilization of the implication of the detected contextual elements in the generation of message contents and (ii) generation of human-like sentences that imply the communication context. To address the former challenge, [108] proposed a gated attention mechanism to self-adaptively and selectively use news context for predicting the next word in the generation of news comments based on the relevance between comments and news. [109] proposed a context-aware approach for the generation of natural language texts, which encodes the contexts into a continuous semantic representation and decodes the semantic representation into text sequences by using recurrent neural networks. To generate human-like sentences, [110] applied a context-aware long short-term memory (LSTM) network model for sentence generation in question answering systems. It was a data-driven approach concerning many aspects of the static context, including question to be answered, semantic values to be addressed in the response, and the dialogue act type during the interaction. Most of the existing model-driven approaches rely on textual context in message generation, which are suitable for correcting the semantics of messages to a grammatically correct sentence. In addition, an artificial neural network is normally selected as a tool for text generation in this type of systems, which may be challenged when multiple aspects of user context are to be taken into consideration simultaneously, not to mention the case when their cardinality increases and the contextual relevance changes in various situations.

In the above-mentioned and many other applications, static context information is normally utilized in message generation. Static context information can be represented by using a relatively fixed information model, and its meaning, implication, and relevance to the users could be predefined. As a contrast, not much research related to the generation of personalized messages based on dynamic context information has been conducted so far due to the insufficient information model for understanding and representing dynamic context information. In addition, compared to the model-driven approaches, template-based approaches are more suitable for real-time messages generation

about the dynamic context of users. Message templates provide dependent structures, which reduces the time needed for message generation and increases the relevance and specificity of the message content. The text embedded in messages and the situations happening around users could be bridged through predefining various messages components used to describe the situations.

### **2.2.5.2. Distributing messages according to dynamic context**

In I-CPSs, the generated messages have to be delivered to informing devices for the purpose of personalized informing. The context of the informing devices (e.g., possible low bandwidth of communications) may affect the sending of messages and thus, may have an effect on the performance of informing. For this reason, [111] proposed a context-aware messaging system, which is able to redirect the incoming e-mails or telephone calls according to the schedule and location of the users and the media available for them. Knox et al. proposed a context-aware message forwarding platform, which is able to send certain incoming e-mails to users based on their changing situations and shifting priorities[112] [113]. The context of the user (e.g. the daily routine, activity) is derived by tracking their locations and monitoring their (next) daily schedule. [114] proposed a context-aware architecture to capture context information of users and to control multimedia channels (e.g. unicast, multicast or broadcast channels) for message delivery. The proposed architecture supports efficient and sophisticated content sharing within mobile communities. In order to optimize the delivery of personalized session contents to multiple mobile users based on their context information, a software architecture was developed, which has the capability of performing required adaptations on the session, transport, and network levels of interoperation, triggered by context changes such as events, locations or a deterioration of network condition [115].

Similar to the context-aware Machine-to-Machine (M2M) communication technologies [116], I-CPSs have to acquire context information from data sensors and information receivers and utilize it to enhance communication. Considering the situations in which distributed nodes, cluster heads, or base stations might move in the physical space, Costa and Miao developed an algorithm to dynamically calculate the transmitting power of each node to save energy [117]. Depending on the existence of cluster head nearby and the dynamic distances from the node to the cluster head and to the base station, each distributed node may communicate with the cluster head or may send information to the base station directly. This approach may also work in a situation, in which the energy-limited mobile cluster head is trying to inform users through the distributed mobile nodes. Their framework did not support applications based on real-time communications, such as surveillance. Similarly, in the case study of ahSN, agents are able to move closer to a user's device to reduce communication

cost [118]. The processed context information includes communication load, link speed and physical distance to the user's agent. In some other cases, distributed nodes need to turn off communication tunnels periodically, in order to save energy consumption. However, this may result in a situation that the nodes are unable to be informed. To solve this problem, Context-Aware Mobile Agent Network (CA-MAN) [119] was proposed, in which agents are able to transfer information to active agents based on the presence information of the agents which indicates the willingness or the ability of being informed by the devices.

Another objective of context-dependent informing is to ensure the security of the information packages [120]. This concern should be demonstrated in the I-CPSs since the distributed computing of the I-CPSs requires a dispersed security model. However, the conventional centralized IT network, which is protected by a firewall, is challenged by the need [121]. Qian and Moayeri proposed a framework to ensure information security in vehicular ad-hoc networks (VANETs), in which rapid and frequent changes in network topology were considered [122]. According to our survey work, the implementation and verification aspects of this research direction have not been demonstrated in the literature. Existing message distribution strategies generated based on static context information cannot satisfy the requirements for hazard-intense I-CPS applications properly, where personal context is heterogeneous, unstructured and may change rapidly. A sophisticated solution for handling dynamic context of users is needed, based on which adaptive and customized distribution of personalized messages can be realized.

### **2.2.5.3. Informing modalities in the case of dynamic context**

At designing informing systems, the possible limited attention of the users should be addressed. In this context, the principles of the modality of the constructed personal messages should be considered to enhance communication with users. Giving attention to this, the efficiency of informing can be increased. The results reported in the literature have shown that context-aware solutions can enhance the communication efficiency between the informing systems and the users [123].

In informing systems, the communication modalities can be either of human-to-human types, e.g. face-to-face, voice-only, linked teletypes, and interactive handwriting [124], or of machine-to-human types, e.g. graphical modality, voice modality, or textual modality [125]. Thus, informing systems might be required to select proper modalities for message delivering, in order to enhance the deployment of the informing services. To achieve this, the context of the user is of an important influencing factor and has been addressed in the literature. For instance, [126] proposed a context-aware system, which allows users to access ubiquitous web services, through a suitable modality. In their work, context information was defined as a combination of the situational context of the

user, his environment, and his computing system. An assistive service provision architecture for providing assistive services to dependent people (elderly and people with disabilities) proposed in [127], which process information regarding user context, environmental context, and the status of the end-user terminal. In addition, a context-aware service provisioning mechanism was reported in [128], which allows informing systems to adapt the interaction modalities according to the working context such as user profiles, device profiles, software profiles, and environment topology.

To promote the informing process, the human is a crucial factor in I-CPSs. As information recipients, human passively receives messages about the result of reasoning from the I-CPSs. The informing strategy should be tailored according to the personal context, e.g. location of the users. For example, if the system detects that a user does not take his/her mobile phones at hand by comparing the locations of the user and the mobile phone, it is not possible to inform the user through the mobile phone. The willingness or ability of a user to be informed is represented as presence information [120], which can be derived from situational information or activity of a user, such as the location and preferences [129]. Based on this, Plesa and Logrippo proposed a Belief-Desire-Intention (BDI) agent architecture for real-time informing systems [130]. This architecture includes a personal communication manager, which is able to consider the desires, preference and the presence of a user at handling calls. Another application with a similar architecture is the Mercury [131], which allows a person to initiate a conversation using any available device. Mercury uses a session initiation protocol to manage communication sessions and exploits dynamic user context to proactively route and migrate calls.

Some other research focused on developing proper informing methods to inform the users. For example, a cloud-based architecture for context-dependent informing was used to aggregate users' presence information, manage context information and provide remote informing solutions for the users [132]. Based on this architecture, users can be informed according to the context. When the temperature in a fridge of a truck exceeds a certain threshold, the notification can be structured in voice messages to a smartphone of a user rather than in a textual form to the user's computer, if the user presence status is offline. In addition, an adaptive and proactive access control approach for building smart infrastructures was used to deal with criticalities [133]. This approach is able to inform people to take required actions according to the particular context of the people. For example, on an oil rig, if the fire is detected in the control room in the meanwhile of rescuing a people with a heart attack, the fire control should prioritize over the medical emergency. Then, the people near the control room will be informed about the changes in their work from rescuing the patient to extinguishing the fire.

An important observation is that the human-to-human modalities are rarely

considered in the existing informing systems. This type of modalities should be given more research attention since it is a possible solution for communicating with users who are unable to be informed by the system directly. For instance, in hazard-intense applications, the I-CPS system may request a stakeholder (who can be informed by using her/his operational device), to warn another stakeholder about a dangerous situation (who is unable to receive messages from the system due to the lack of a mobile communication device).

## **2.2.6. Software enablers for dynamic context information processing**

### **2.2.6.1. Computational frameworks**

The information processing frameworks concentrate on an application-independent design by providing a basic structure for a certain class of applications and ways of customization for specific needs [134]. Some scientists employed a central context manipulator in their frameworks to deal with dynamic context information in simple forms. ParcTabs [135] is accepted as the earliest attempt on the development of a framework for dynamic spatial context information processing. All the spatial relations, such as containment, location, path, and distance are managed in an active map, which is a typical central context manipulator. Another example is the framework used in the Stick-E Notes system [136] [137]. It contains a context manager class to respond to the notifications sent by the stick-e notes.

The situation fencing, which is a situation-aware framework, is able to handle both macro context and personal context by using “if...then...” rules [138]. This framework is effective to process dynamic context information in simple forms, such as key-value pairs by simple reasoning mechanisms. The functionalities of the central context manipulator can be decomposed into a layered structure. Lun and Cheng [139] proposed a system model, which consists of four layers: context acquisition, context processing, context selection, and context application. Similarly, the CAA [140] contains acquisition layer, context aggregation layer, context reasoner layer, and application layer. In addition, JCAF [141] is a Java-based context-awareness infrastructure containing context service tier and context client tier. The layered structure makes the framework easy to be applied in specific applications. However, it only generates one context-dependent strategy for adaptation in each working cycle.

In addition, the central context manipulator can also be made by functional components. Widget-based framework is a general tool employed for context-aware services [44] [142]. WildCAT is an extensible Java framework, in which context information is processed through two complimentary interfaces: synchronous requests (pull mode) and asynchronous notifications (push mode)

[143]. In CAFCLA, different types of context information are processed in different servers including the activity server, the database server and the file server [144]. Korpipaa proposed a framework for distributed context-aware computing in an event-based manner [145]. With a hierarchical structure, the framework includes four main functional entities: context manager, resource servers, context recognition services, and applications. Kim et al developed an application framework based on partially ordered knowledge sharing for loosely coupled systems [146]. The framework consists of cyber-hosts, cyber-engines, cyber-nodes, and cyber-applications. Processing dynamic context information in multiple components at the same time is able to generate multiple strategies for multiple purposes according to the context.

Several proposals have focused on designing a proper middleware to process dynamic context information. Sun and Zhang [147] proposed a service-oriented middleware which is able to provide run-time adaptation support for CPSs to deal with the dynamic spatial context in a mobile environment. The service-oriented architecture facilitates system flexibility[148]. SOCAM is a middleware used for handling ontology-based context models for mobile applications[149]. It employs a hierarchical architecture to process both domain-specific ontologies and generalized ontologies. However, it does not support service adaptation in a dynamic context. SATWARE is presented as a multi-level semantics-based middleware [150] [151]. The SATRuntime, a layer in the middleware, contains a repository of operators which can be implemented as mobile agents (software entities) to dynamically change the execution environment. Context-aware middleware is a flexible solution to provide support for achieving complex tasks in the dynamic context at multiple levels, for instance, at network, device and environment levels.

### **2.2.6.2. Reusable software platforms**

Supporting the development of context-aware I-CPSs with an embeddable platform is not a brand-new idea. Development of generalized hardware, software, and cyberware for reusable platforms is one of the major research challenges associated with the implementation of CPSs [152] [153]. Actually, many platforms have been proposed, which include simple and efficient architectures to support the rapid development of I-CPSs that are able to handle context information.

Several platforms have been designed to support the development of context-aware applications based on centralized architectures. In these platforms, a centralized component (e.g. cloud server) manages context information aggregated from distributed nodes and provides solutions for all kinds of operations. For instance, the smart healthcare platform, introduced in [154], employs a cloud server to maintain the information collected from carry-on

wallets, which aggregate personal information and bio-signals of patients. If an unexpected pattern of information is detected by the wallets, appropriate actions will be triggered by the cloud server. In other cloud-based platforms, such as the context-aware platform [155] and the RaaS platform [132], functional components share data with the cloud server for context management and processing. The main drawback of the cloud-based platforms is that they cannot solve complex problems.

The context broker architecture (CoBrA) is an agent-based architecture for supporting the development of context-aware applications [156]. Central to this architecture is an agent called context broker that maintains dynamic context from heterogeneous sources, such as sensors and devices and share with computing entities (context-aware agents). It is specially used to develop applications for smart spaces (e.g. living rooms, corporate offices, and meeting rooms) that provide pervasive computing services to users [157]. Among others, Gaia is a middleware infrastructure that can be used to prototype context-aware I-CPSs used in ubiquitous computing environments [158] [159]. Gaia offers resource management and user-oriented interfaces to support the seamless interaction between mobile users, and their physical and digital environment.

In addition, a platform can be built in a distributed form and manner. For instance, the collaborative and distributed search platform presented in [160], and the mobile networking platform proposed in [161], consist of a group of mobile devices with positioning and communication capabilities. Each mobile device is able to find, assign, and refine routes considering speed changes, and dynamically connect with and leave other devices. As an example, the cooperative mobile platform, discussed in [162], offers device sharing and application migration functionalities, which enable applications to operate dynamically in changing contexts. Considering distributed platforms in the development facilitate modularization of I-CPSs. However, the lack of consistency among distributed components may prevent the systems to generate globally optimal solutions.

Furthermore, there are some other platforms that apply hybrid architecture, based on which context management and solution development can be made in both centralized and distributed manners. For instance, robots in the assistive robot platform presented in [163], are able to aggregate information of elderly people in a distributed way. On the other hand, the used cloud server can control the robots and extend their adaptation and learning capabilities. In general, hybrid architectures enable platforms to process dynamic context information and to perform dynamic behavior in context by either a centralized processing or a distributed processing. However, dynamism in context-dependent information processing has not been exemplified or clearly demonstrated in the literature. It seems that the recently developed platforms, such as the event information management (EvIM) platform for real-time complex event discovery [164], and



the citizen sensing & actuation platform [165], have not been designed for dynamic operations and adapting themselves to dynamic contexts. Therefore, there is still a need for an information processing platform which is able to change sensing, computing and actuating behaviors in context.

Table 2.1. provides an overview of the existing platforms that enables DCIP. It can be observed that most of the existing platforms support cyberware interfaces, by which cyberware entities can be integrated into the platforms. This offers opportunities for processing context information from multiple resources. In addition, cloud-based external integration is becoming the mainstream of research to develop platforms. The reason is that the cloud server is capable to aggregate context information from multiple scales regarding time and space. Thirdly, the smart healthcare platform is the only platform that supports hardware integration. We suppose the main reason is that most CPSs employ wireless communication technologies to integrate distributed components. Therefore, there is no need to design physical interfaces in the platforms. EvIM is the only platform that is able to support all of three context-dependent operations. The reason is that event awareness and solution development are allocated to independent components in the EvIM. This can reduce the complexity and improve the scalability of the platform to handle different types of context-dependent operations.

In terms of the existing platforms, we observed an immature stage and variety of intended applications. The existing platform solutions are developed for domain-specific systems. For instance, CoBrA and Gaia platforms provide ubiquitous computing environments (intelligent spaces and active spaces, respectively) in which the applications are deployed. They do not include reusable components to support general utilization. Furthermore, existing platforms process only static or altering context information, or use a relative constant procedure to respond on every change of the environmental context (e.g. using ‘if...then...’ to making decisions). However, it remains a research challenge what the changes mean to system operations and how adaptive system operations can be made based on the DCI.

### **2.2.6.3. Reusable knowledge resources**

As a tool enabling the semantic representation of context, ontologies are commonly used in context-aware applications [166]. For this reason, many ontology repositories have been implemented and can be reusable by application practitioners [167]. OWL [168] is the current standard description language (standardized by the W3C consortium) used for expressing ontologies, which is an extension of the Resource Description Framework (RDF) [169]. Many ontology vocabularies were constructed based on OWL for the use of dynamic context management. A typical example is a meta-context ontology model called MCOnt, which is able to represent dynamic and uncertain context with regard to

**Table 2.1.** An overview of existing context information processing platforms

Software platforms	Internal integration		Service -driven	External integration		Hardware interface	Software interface	Cyberware interface
	Data-driven	Event-driven		Cloud	WL AN			
Mobile networking platform [161]	✓					✓		✓
Collaborative and distributed search platform [160]		✓				✓		✓
Cooperative mobile platform [162]		✓				✓		✓
Assistive robot platform [163]	✓				✓		✓	✓
Smart healthcare platform [154]	✓				✓	✓	✓	✓
RaaS platform [132]			✓		✓			✓
Citizen sensing & actuation platform [165]		✓						✓
EvIM [164]		✓			✓		✓	✓
The cloud-based platform [155]	✓				✓			✓

user, computing actuator, environment and service [170]. Another example is the ontology named as CAMEnto, which considers 5Ws as context: who, when, what, where and why [171].

In addition, the ontology generated in [172] enables the evolution of ontologies representing dynamic operating contexts. Another type of knowledge resource is the logic rules that are used for context-based reasoning. In the case of dynamic context, first-order logic models are most commonly used in the literature since

they allow implementing complex rules involving the logic behind contextual changes. Typical first-order logic rules applied in context-reasoning infrastructures, are SOCAM [173], Gaia [174], Semantic Space [175], Smart Office [176] etc. Based on the first-order logic model, other complex rules can be developed, such as XTT2 [177], HMR language [178], rules with five logic predicates [179].

Many context-aware applications employ case-based reasoning (CBR), in which various cases used for context reasoning are defined in a general format and stored in a database (named as a case base) [180]. In [181] a trace-based reasoning approach was proposed, which is a variant of CBR. A trace is considered as a case that reflects the historical observation of a given situation. A trace base was developed and managed by a trace-based management sub-system. In addition, the cases defined by Zimmermann enable the generation of recommendations for users in a mobile environment [182], while the cases defined in [183] support real-time analysis of the evolution of a dynamic situation. Other types of usable knowledge resources for dynamic context information processing may include (i) patterns for context recognition, e.g. [30] and [184], (ii) evidence models for context reasoning, e.g. [185], and (iii) profile trees for contextual personalization, e.g. [186] and [187].

## **2.2.7. Major findings**

### **2.2.7.1. Regarding dynamic context representation**

It can be concluded based on the surveyed literature that the need for modeling dynamic context and inferring from dynamic context representation has been explicitly formulated and many related research issues have already been addressed. However, the research efforts that address complex cases are still limited, as well as the range of results. The current situation is caused by two facts. On the one hand, and most importantly, dynamic context is a high-level concept that integrates heterogeneous context information from multiple sources. Temporal context is incorporated in various spatial, attributive and emotional contexts in order to enable time-sensitive modeling and representation of complex scenarios with regard to entities and processes. This requires a mechanism that cohesively handles various types of context data in an application-independent way and interprets dynamic context on a high abstraction level - as high-level context information elements beyond states (static context) and/or changes (altering context). The computational benefit of doing this is the capability to reason out interactions of situations, e.g. causality, dependency, which is the basis of predicting future dynamic context.

On the other hand, dynamic context management entails the need to represent and capture not only the value changes of the various context variables (e.g. changes

of temperature in time), but also the change of relevance of the aspects in consideration (e.g. if the variable ‘room temperature’ is relevant to the user’s context). The relevance of context information elements can be predefined for a particular application in certain cases. It can also be tailored to a particular usage. Typically, it is suitable for application-driven and non-reusable context processing solutions. From the point of view of representing and inferring with dynamic context, the relevance of the context information elements should be computed in real-time. The computation is challenging since it has to be completed for each involved entity considering their direct or indirect interactions and the interplay of the dynamic contexts. Exploration of the implications of the changes needs some sort of integration of the context information elements as well as some sort of abstraction that facilitate ‘distillation of the meaning’ of the implications. The computational mechanism proposed in this paper has been developed with the aim of addressing the above two aspects.

#### **2.2.7.2. Regarding dynamic context reasoning**

According to our survey, a majority of context-aware applications use non-probability-based approaches. The first reason is that the non-probability-based approaches employ simple computation mechanisms which are easy to be implemented and tested in real-life applications. The second reason is that most context-aware applications focus on historical information aggregated during the past. The predicted information about future changes, which is normally represented as probability-based variables, is not taken into consideration. In addition, non-probability-based approaches are competent for reasoning in a certain context, such as certain changes of distance between two entities. However, they lack to deal with uncertainty and randomness. Probability-based approaches make use of uncertainty of context information from a probability perspective. This might provide a solution to deal with heterogeneous context by calculating the happening probability of an event. In addition, probability-based approaches can be an effective tool to deal with human dynamics. Based on the historical behaviors of users, the user intentions can be analyzed and the follow-up actions can be predicted.

#### **2.2.7.3. Regarding context-dependent messaging**

Although the need for context-aware capabilities is recognized in various application fields, the phenomenon of contextualized communication between informing systems and human people has only been superficially addressed so far. Many white spots can still be found in the field of I-CPSs, in particular in the subfield of smart (self-aware and self-adaptive) I-CPSs. Proposals and solutions for message generation using natural language and messaging in dynamic contexts are also scarce. As hinted at above, the overwhelming majority of the

existing computational mechanisms consider static context information only. The progress in terms of reasoning with dynamically changing context information in real-time is still limited. With regards to the people to be informed, processing dynamic context information is restricted to location changes or daytime changes. However, personal context modeling should include not only the specific personal information of the target person but also information about the state and activities of other relevant entities and the surroundings.

Context information is normally considered as the whole of descriptive attributes of the people and is stored as various profiles in the existing systems having adaptive interaction modality. Notwithstanding, this type of systems has limited capabilities to deal with situations in real-life scenarios, when the actual context does not accord with the context information stored in the profiles, e.g. a person uses the communication device of another person. Several similar cases can be conceived, in which dynamically changing situations, rather than steady-state situations are to be dealt with. Therefore, researchers need to provide adequate theoretical fundamentals and computational methodologies for processing dynamic context information. This issue should be addressed not only in research but should also be considered in the development of upcoming systems, which are supposed to adapt themselves to the changes which appear in varied forms in real-life applications. The issue raised by a possible low communication bandwidth requires using a prioritizing function in these future systems. These systems may treat every person differently, for instance, in the case of interacting with a great number of people in emergency situations.

#### **2.2.7.4. Regarding software enablers**

Context-aware I-CPS applications are featured by cross-domain cooperation of sensors, heterogeneous information flow, and intelligent decision/actuation. Differing from other types of CPSs, the actuation activities of context-aware I-CPSs are performed in the cognitive domain of users. It means that the activities of users before informing and reactions of users after informing should be detected and considered in the computation. Another important feature of context-aware I-CPS application is the personalization of service. Each user involved should be given instructions developed according to the dynamic context of the user. These two special features require the I-CPSs to handle DCI smartly in order to provide dynamic control information for the system components and the users. The smart handling of DCI refers to a dynamic context management mechanism, which should be designed to support dynamic system operations, e.g. making optimal decisions for each user, reasoning of personalized informing plans, and customizing instructive messages. In addition, the realization of dynamic system operations requires a situational evaluation mechanism, based on which the system is able to judge what the current situation is and to decide on

which operation is needed for the current situation. However, related dynamic context processing techniques are just in a booting up phase, and the functions required by context-aware I-CPSs are not available or partially available in existing platform solutions.

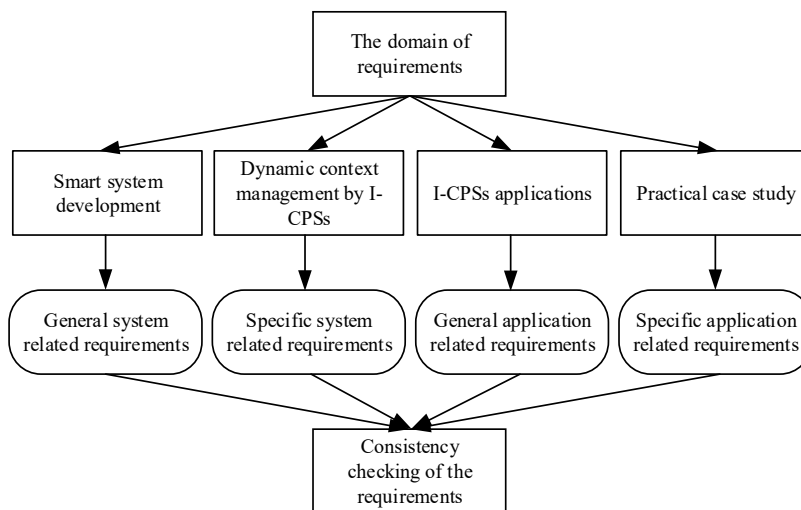
To support the development of context-aware I-CPSs, there is a need for a novel platform. The kernel of the platform should include essential elements of dynamic context processing. On the one hand, the kernel should be application-independent. On the other hand, it should contain a composition of the functions required by context-aware I-CPSs. Therefore, whenever an I-CPS should be developed, this kind of facility can be customized for the use of that particular context of the target I-CPS application. In addition to the kernel, the platform needs additional components for different tasks, e.g. getting the information from the sensor network or sending messages to the users. The platform is required to dynamically manage the system components according to context of the target application. Therefore, we believe context-aware I-CPS applications require an adaptive and smart platform to support the rapid development of them. Adaptability calls for application independent design, while smartness requires intelligent cyber-physical computing of DCI.

## **2.3. Exploring requirements for dynamic context information processing**

### **2.3.1. Objectives of requirement engineering**

The principles of requirements engineering laid down by Pohl, K. [188] were applied and the process of requirements engineering was conducted by following the principles proposed by Finkelstein, A., & Savigni, A. [189]. Its primary objective was to collect technical requirements from multiple sources and to derive a consistent list of requirements for the conceptualization of the target reasoning mechanism. The research design for requirement exploration and synthesis is shown in Figure 2.2. The major domains of requirement were specified as system-related requirements and application-related requirements. In this context, four general sources of requirements were considered:

- (i) *General system-related requirements (GSR)*, which are implied by the theory and implementation practice of smart (cyber-physical) system development. To support the development of concepts for smart system design, certain characteristics should be realized. Towards this end, our objective was to aggregate requirements about smart features of CPSs.
- (ii) *Specific system-related requirements (SSR)*, which formulate expectations with regard to the computational implementation of dynamic context management by I-CPSs. As explained above, a major characteristic of



**Figure 2.2.** The process flow of requirement exploration, synthesis, and processing

I-CPSs is context management with the purpose of informing people and requesting them to do certain activities. A critical analysis of the characteristics required for dynamic context management was used to explore and formulate the most relevant technical requirements.

- (iii) *General application-related requirements (GAR)* originating in multiple I-CPS applications. Particular system operations should be articulated to support the implementation of I-CPS applications. This aspect tries to explore the requirements with regard to using smart system operations in multiple application contexts based on a review of proliferating I-CPS applications.
- (iv) *Specific application-related requirements (SAR)*, which are related to a reference application case. To obtain sufficient insight in and empirical experiences concerning the role of DCIP in I-CPSs, a specific reference application case was selected and elaborated on. Requirements were explored based on a critical analysis with regard to this concrete application case, discussed in detail later in the Dissertation.

Consideration of both system-related and application-related requirements makes it possible to tailor DCIP to the concrete needs of the computational mechanisms and the application cases of I-CPSs. Accordingly, consistency of the requirements is checked and additional requirements are generated by integrating multiple requirements. These obtained requirements were considered in the

conceptualization, implementation, and validation of the computational mechanism. Details with regard to the requirement exploration are given in the following sections.

### **2.3.2. Exploring requirements based on smart system development**

Like machines in the early stages, which were intended to amplify manpower to improve manual productivity, smart systems are also required to enable human capabilities in the cognitive domains [190]. This ultimate objective is, however, can be achieved only by a long-term development and innovation. The current objective is to develop systems that can make decisions based on context awareness and deliver better performance targets and results based on their aggregated knowledge [191]. Smart systems are proliferating and built up a legacy in many fields of application. According to the overall functionality of CPSs (discussed in Section 1.1.2) and the characteristics specified for various generations of smart CPSs, requirements for smart CPS development may be an infinite number of variations (e.g. hardware and software requirements for computing, communicating and controlling). For the need of the thesis, we only focus on the general system-related requirements with regard to DCIP that support self-awareness and self-adaptation (attributes of 2G-CPSs).

Self-awareness built up in a computation process is one of the bases for achieving high-level autonomous behavior [192]. In cognitive engineering, self-awareness has been defined as “the ability of knowing about knowing” [193] or “the capacity to become the object of one’s own attention” [194]. Agarwal et al. argued that “the constraints of computation in self-aware systems should not be preprogrammed or predefined at design time” [195]. Recently, Kounev et al. proposed that “self-aware computing should (i) learn models capturing knowledge about themselves and their environment on an ongoing basis, and (ii) reason using the models enabling them to act based on their knowledge and reasoning in accordance with higher-level goals” [196]. Although different interpretations have been proposed, the general goal of self-awareness is to generate knowledge about (i) phenomena internal to oneself (e.g. system goals or behaviors), and (ii) phenomena external to oneself (e.g. relation to one’s environment) [197] [198]. Self-awareness is fundamental for self-adaptation. A system is self-adaptive if it is able to adjust its behavior in response to their perception of the environment and the system itself [199]. The self-adaptation process can be triggered by either (i) changes in the higher-level goals, or (ii) changes in the system itself or in its run-time model, or (iii) changes in the environment [200]. Recently, a conceptual model for self-adaptation has been proposed, which considers the interactions among four basic elements: environment, managed systems, adaptation goals, and managing system [201].



The basic concepts concerning self-awareness and self-adaptation are discussed in the literature, which also informs about the fact that the DCIP is employed by smart systems in order: (i) to facilitate knowledge aggregation with regard to operational context, and (ii) to enable implementation of various levels of self-control and self-organization. Considering these, the general system-related requirements (GSR) for DCIP have been specified as follows.

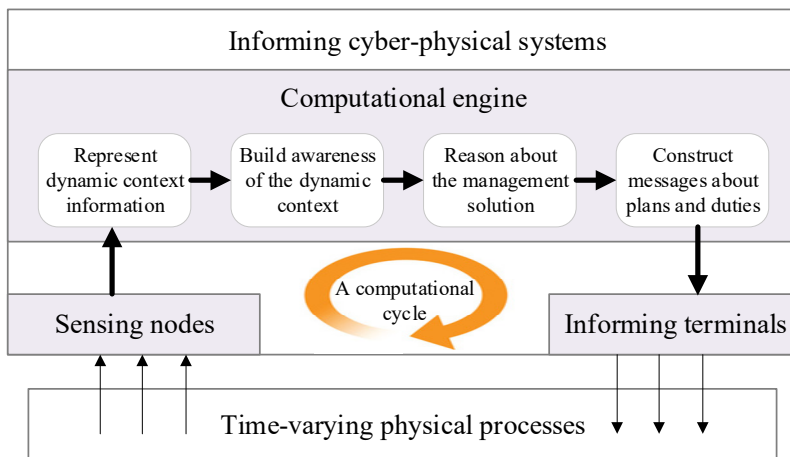
- GSR 1: DCIP should be based on a computationally robust and, at the same time, situation- and application- adaptive theoretical framework.
- GSR 2: The DCIP-enabled system should harmonize the data received from a large number of (dynamically activated) sensors.
- GSR 3: Semantic knowledge of a high abstraction level should be inferred from contextual changes by using DCIP to facilitate awareness building.
- GSR 4: The DCIP mechanism should forecast changes and trends of the situations and not only describe observable situations.
- GSR 5: I-CPSs need a strategy development mechanism that converts the results of situation evaluation into action strategies for people to perform.
- GSR 6: The computational mechanism for DCIP should be linearly computable (that is, a linear increase in the number of the concerned entities and relations should not result in an exponential increase in the computation time of the context).
- GSR 7: The computation mechanism of DCIP is supposed to require a low amount of computational resources (e.g. CPU, memory) to facilitate its usage in multiple resource-constrained applications.
- GSR 8: The adaptation responsiveness should be fast enough in particular in cases where the context changes frequently (i.e., the delay in processing contextual data should be low).
- GSR 9: The adaptation realized by the computation mechanism of DCIP should involve run-time planning verification and validation of the developed strategies/solutions.

### **2.3.3. Exploring requirements for dynamic context management by I-CPSs**

According to the general architecture of I-CPSs discussed in Section 1.2.4, the reasoning functionality of I-CPS is extended with data analytics functionality, which is based on multiplexed sensor nodes and pervasive sensor networks, and

with information modality transformers and message generators. There are four main operational steps served as a computation engine for processing relevant dynamic context information, as shown in Figure 2.3. To enable the computation, dynamic context information should be computationally represented in the first step. It implies that time-dependent descriptive data aggregated by sensing nodes should be properly constructed to enable the execution of functions belonging to the following computational steps. Then, computational awareness with regard to the dynamic context is to be built in the second step. Semantic knowledge hidden from dynamic context should be derived. The knowledge obtained in the second step is the basis for generating strategies and personal solutions, which are dedicated for the purpose of context information management. In the last step, the computational engine constructs messages which are about to read by message recipients. The personal context and the results of reasoning are considered at constructing the messages. Finally, the generated messages about personal plans and duties are supposed to be sent to the informing terminals and to be read and reacted upon by people.

These four operational steps work in a sequential manner (from aggregated sensor data to personalized messages), which forms a computational cycle. After the messages have been sent to the message recipients, the computational engine should analyze the advancement of the situations after informing and generate knowledge with regard to the situational changes in the building awareness function of the new computational cycle. The result of analysis enables the computational engine to be aware of the factors (or situations) that hinder the realization of certain management goal (e.g. some message recipients disobey the



**Figure 2.3.** The general operational flow for dynamic context information management by I-CPSs

messages), and to adapt to the factors by generating new strategies and plans in the new computational cycle. The specific system-related requirements (SSR) with regard to these four operational steps are listed as follows:

### **Requirements for dynamic context information representation**

- SSR 1: The part of the overall mechanism dedicated to dynamic information representation should aggregate the largest set of information elements regarding entities, attributes, and relations unless specified otherwise.
- SSR 2: The context information model should enable cohesive handling of multiple types of context data (e.g. spatial, attributive, and temporal).
- SSR 3: The context information model should be able to update itself during run-time (i.e., when entity states or relations change).
- SSR 4: The context information model should be constructed and tested in the shortest possible time frame (i.e. in quasi-real-time).

### **Requirements for building awareness in dynamic context**

- SSR 5: The meaning of dynamic context should be specified on different abstraction levels in order to support building awareness by interpretation of dynamic context semantics.
- SSR 6: The computation mechanism of DCIP should assess the influence of given contextual phenomena (i.e. situations) and interpret their implications in a time-dependent manner.
- SSR 7: The computation mechanism of DCIP should evaluate not only the already acquired dynamic context information, but should also predict and assess the near-future of dynamic context information.
- SSR 8: The relevance of a contextual phenomenon to the concerned people should be evaluated and, based on this, personal context should be specified in order to enable generating personalized services.
- SSR 9: Individual people should be prioritized according to the implication (impact) of the evaluated individual context and this prioritization should form a basis for optimizing the context management solutions.
- SSR 10: The obedience of people to the given instructions should be critically judged considering the attitude of people to neglect or disobey the received instructions.

### **Requirements for reasoning about situated action plan solutions**

- SSR 11: Situated solutions for people management should be generated based on the results of an evaluation of the impact of context and they should be adapted to the changes of context (e.g. attributes of entities).
- SSR 12: Personalized informing/guidance and situated action plans solutions for people should be generated based on the actual contexts of individuals.
- SSR 13: The situated action plans solutions should consider optimization not only on the individual level but also on the group level.
- SSR 14: The dynamic context management should proactively deal with dangerous situations (i.e., to prevent the formation of a dangerous situation, or to prevent a dangerous situation getting worse).
- SSR 15: Decisions concerning the instructions included in a situated personal action plan should be made based on the dangerousness or safeness of the circumstance of each person.

#### **Requirements for messaging people about action plans and duties**

- SSR 16: The construction of personalized messages should be made by a flexible approach that adapts the content of messages to the time-varying context of people.
- SSR 17: The relevance of contextual phenomena (i.e. situations) should be taken as the basis of informing people under any circumstance and at any time.
- SSR 18: Sufficient information describing the phenomenon (at least, location, attributes, and time of happening) should be included in the constructed message when the individuals are informed about contextual phenomena.
- SSR 19: Prioritized messaging services should be provided for the people according to the order of being in a higher level of danger.
- SSR 20: The rhetoric used in message construction should properly reflect the seriousness of the computed personal danger.

#### **2.3.4. Exploring requirements based on real-life I-CPS applications**

As indicated by the system-related requirements, I-CPSs are supposed to execute messaging operations according to the actual situational context that gives the reference for the interpretation of messages and completing actions. The smart operations of I-CPSs are focused on providing context-dependent information

services for applications and people. The possible range of informational services is rather broad (including, e.g. customized action plans, timely-refreshed information, or context-sensitive guidance). Actually, the variety of application opportunities for these systems is constrained only by the imagination of the system designers and by the economics of implementing them for particular applications. To generate transferable knowledge with regard to the requirements of DCIP for multiple I-CPS applications, their common needs should be investigated.

This section summarizes a number of typical I-CPS applications, in which information generation, time-sensitive messaging, and multi-point communication abilities of I-CPSs are utilized. Based on a critical analysis, general application-related requirements (GAR) are to be generated. Table 2.2 shows the I-CPS applications that are surveyed from the literature. The system objective-induced information is given to users, and the processed context information of the systems are compared. In the table, ‘St.’ refers to as static context information, while ‘Dy.’ refers to as dynamic context information. ‘S.’, ‘A.’, ‘T.’ represent for spatial, attributive and temporal context information, respectively.

It can be seen from the surveyed literature that, most I-CPS applications are designed with the objective to provide personalized information to users based on static spatial context information (the location of the user is the most frequently used type of information in decision-making and message generating). In addition, several systems considered dynamic context information in computation. Among them, only the location changes of users are processed, whereas the time-dependent changes with regard to the entity relations (e.g. distance) have rarely been involved. Furthermore, most of the I-CPS applications make decisions based on explicit context information (e.g. time of day, location), while implicit information (e.g. human intention, imagination) was not mentioned. Last but not least, for most of the applications, the relevance between processed context information and the target user is certain and predefined. There is no sufficient computational approach in the literature that enables run-time calculation of the relevance between context phenomena (e.g. a situation) and the target user.

Based on the summarized attributes of I-CPS applications, general application-related requirements (GAR) have been identified as follows:

- GAR 1: To enable a tight interaction with the physical processes, I-CPS applications should aggregate sufficient amount of descriptive context data.
- GAR 2: I-CPS applications should inform users according to their changing needs as the context or objective varies and the needs should be interpreted, modeled, updated and predicted during run-time.

- GAR 3: I-CPS applications need cohesive handling of context information from multiple sources which are related to spatial, attributive and temporal contexts.
- GAR 4: To enhance the quality of informing, implicit context information (e.g. intention, attention) concerning the people should be taken into consideration at generating personalized services.

**Table 2.2.** Typical I-CPS applications and the computed context information

Ref.	Name of system	Objective of the system	Information given to users	Type of context information						
				St.	Dy.	S.	A.	T.	Other	
[202]	Transportation information system	Travel planning in case of delays and service disruptions	The situation of buses and stations	√		√		√		
[203] [204]	Tourist information system	Provide tourist information to travelers	Location, past viewing history, and information in a digital library	√	√	√		√		
[205]	Tour guide system	Support tour guide	A map of labs and information of all projects	√		√			√	
[206]	Personal navigation systems	Vision-aided navigation for walking people	Route	√		√				
[207]	Activity recognition system	Recognition of user activities	Activity that the user is performing	√		√				
[208]	Healthcare monitoring system	Report information of patients to the physician	Information related to activities being executed by patients	√		√				Usage of artifacts
[45]	Road navigation system	Provide traffic information to users	Road information and the route to a position	√	√	√	√	√		Quality of service
[209]	Shop planning system	Help customers define shopping plans	A list of goods to buy	√		√	√			Network condition
[210]	Driving assistance system	Support the driving of users	Provide information about the traffic and the driver	√	√	√	√	√		Driver's gaze
[77]	Smart home	Health monitoring in home environment	Report activities and status of residents to professionals	√	√	√	√	√		

- GAR 5: In the case of daily life applications, the personalized messages generated by I-CPSs should be concise, representative and relevant to the user's actual context.
- GAR 6: I-CPS applications need model-free or data-driven operations in order to be able to adapt to varying circumstances.
- GAR 7: In I-CPS applications, the processed and communicated data should be confidentially protected according to the GDPR standard.
- GAR 8: To provide real-time services, I-CPS applications should be implemented with near-zero-time system operation and with slight non-linear up-scaling.
- GAR 9: Operations of I-CPS applications should not only adapt to the contextual changes of users but should also adapt to the internal changes of the system.

### **2.3.5. Exploring requirements based on a case study**

As discussed in the previous section, we have considered many practical cases which can be supported by DCIP. From the set of considered application cases, we selected the case of building fire evacuation and used it as a demonstrative case for exploring the specific application-related technical requirements.

The newest report of the National Fire Protection Association (NFPA) reveals that from 2012 to 2016 there were an estimated 355,400 reported home structure fires every year in the United States, resulting in an annual average of 2,560 civilian fire deaths and 11700 civilian fire injuries [211]. The existing solutions for the indoor fire evacuation are (i) fire alarm systems, (ii) enabling facilitates (e.g. sky-bridges, escape scuttle, etc.) and (iii) fire rescue by firemen [212]. Most of the existing solutions make use of the predefined and constructed facilities to guide people to escape through predefined paths, such as signs and maps. However, the guided people may be unfamiliar with the structure of the burning building and individuals may get panic when the fire alarm is warning unexpectedly [213]. This might lead to extreme situations such as congestion and inhaling toxic air imperceptibly.

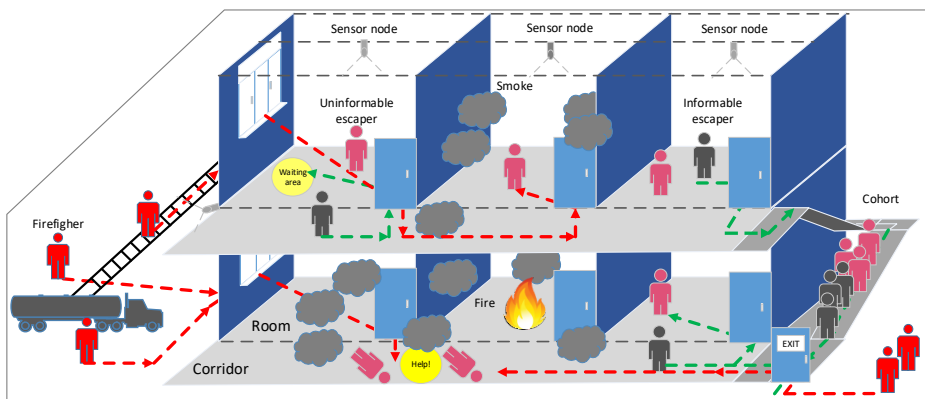
A typical scenario of indoor fire evacuation is shown in Figure 2.4. In this type of emergency cases, the major problems are: (i) the optimal escape route is not always the shortest path out of a building since it may be occupied by fire or people jams, and (ii) some people may not recognize the proliferating fire due to some physiological reasons (e.g. may be deaf) or due to being heavily submerged in amusement activities (e.g., listening music). Being tagged in a burning part of a building, people should be given personalized guidance regarding (i) what relevant situations (e.g. fire, smoke, people jams, people in danger) are happening

in the building, and (ii) what actions should be performed when the relevant situations are taken into consideration (e.g. feasible escape paths and rescue paths).

The above-mentioned problems cast a light on the dynamic context of people, which is changing over time (e.g. the formation and changes of people jams and fire). It implies that the personalized context should be run-time evaluated and considered in the computation. In the literature, several smart emergency evacuation systems have been proposed in recent years [214], which makes use of the context of people to generate adaptive solutions. However, the development of personalized services according to the real-time aggregated dynamic context of people has not been sufficiently addressed so far. Therefore, adequate processing of dynamic context information is generally required.

Therefore, an indoor fire evacuation guiding system, which is a typical I-CPS, should consider the following specific application-related requirements (SAR):

- SAR 1: The IFEG system should aggregate time-dependent descriptive data with regard to the states of entities (e.g. people, exits, fire) and their relations.
- SAR 2: Dynamic context of people (escapers) should be efficiently represented and the personal context should be processed in a real-time manner.
- SAR 3: People (escapers) should obtain information about the danger and the optimal escape route/solution.
- SAR 4: The IFEG system should make decisions about individual action plans and duties of people and should evaluate the consequences of



**Figure 2.4.** Demonstrative image of the indoor fire evacuation guiding application case that served as a reference



the action plans and duties in real-time.

- SAR 5: The IFEG system should use the most appropriate and probable informing modality (e.g. sound messages, textual messages, etc.) to inform the people.
- SAR 6: The IFEG system should adapt to the obedience of involved people and consider that they may disobey the instruction given to them.
- SAR 7: The generated personalized action plans or duties should be optimized considering multiple factors such as (i) people should be kept away from dangerous situations as much as possible, (ii) all people should be evacuated in the shortest time, and (iii) every person should be guided through the nearest evacuation path.
- SAR 8: The IFEG system should be capable to rank dangerous situations (e.g. fire, people jams) according to their actual impact on people and to handle multiple interrelated situations at the same time.
- SAR 9: The IFEG system should be able to treat people differently in evacuation circumstances (e.g. first to inform people being in a dangerous situation) and to create a priority list based on the individual danger level of the individuals.
- SAR 10: The IFEG system should be able to generate personalized messages within an allowed reaction time and to avoid general information by strongly restricting the instruction and information given to people to their actual context.
- SAR 11: The IFEG system should be able to predict the future of situations and states of entities and to generate both long-term and short-term escape solutions.
- SAR 12: The IFEG system should consider future context as a reference for generating the management strategy.
- SAR 13: People in dangerous situations should be given messages reflecting a style of being in urgency.

### **2.3.6. Consistency checking of the aggregated requirements**

The above-discussed requirements are obtained from multiple sources. Their consistency should be checked since the system-related requirements may not be needed by, or maybe even conflict to the requirements derived from the practical needs of I-CPS applications. Accordingly, for every obtained requirement, it was examined if the requirement accords with any other requirement (s) or not. The

result of the consistency checking is shown in Figure 2.5. It can be seen that most of the explored system-related requirements are consistent with the application-related requirements. It means that (i) smart system features (i.e. self-awareness and self-adaptation) are generally in the need of I-CPS applications, and (ii) dynamic context information management is one of the required features of the IFEG system.

Specifically, there are several inconsistent requirements. In GSR 7, low computational resource consumption is required for using a developed DCIP mechanism in multiple applications, while GAR 1 claims that sufficient descriptive context data are needed to enable a tight interaction between the

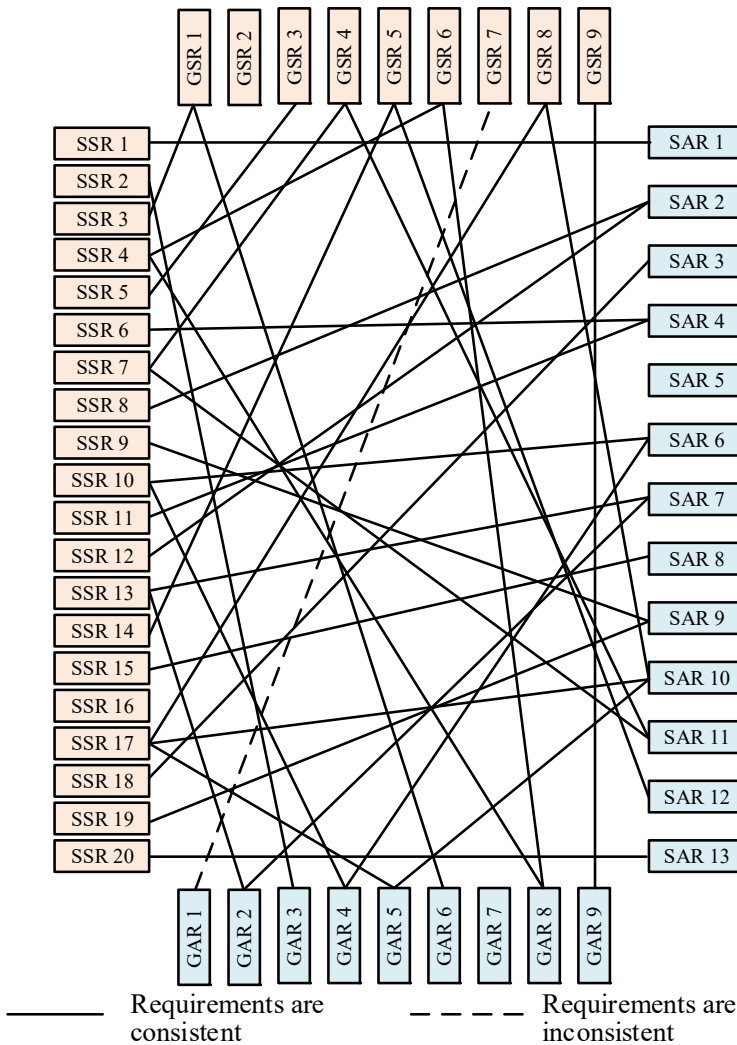


Figure 2.5. Consistency checking of the aggregated requirements

I-CPS applications and the associated physical processes, which may increase the amount of data for processing. The reason for the inconsistency can be explained as follows: Context information considers entity states and their relations, forming a huge pool of data to be processed at a given moment. When the context changes rapidly, the more frequently the data aggregated, the better the dynamic context can be represented. However, the increased amount of data needs an additional computation time for processing, which may cause significant delays between the represented context and the context in the physical world. Accordingly, the implication of the inconsistency can be formulated as two additional technical requirements (ATR).

- ATR 1: The issue of a trade-off with regard to the amount of aggregated contextual data (e.g. sampling rate) should be taken into consideration in order to optimize the performance of the dynamic context information processing.
- ATR 2: The scheme of dynamic context information representation should be able to capture the physical context information and the cyberspace context information in a synchronized manner and to use both in decision-making.

In addition, there are several requirements that are irrelevant (neither consistent nor conflicting) to the rest requirements. For instance, protecting the private information of users (GAR 7) is not considered from a smart system point of view. This indicates that the method applied for requirements exploration from multiple inter-related sources is necessary. All the obtained requirements should be considered in the conceptualization, implementation, and validation of a DCIP mechanism.

## **2.4. Conclusions**

### **2.4.1. Conclusions concerning the knowledge aggregation**

By using informing CPSs, the hazard in critical events and situations can be reduced. An opportunity for this is providing context-dependent personalized messages for people who are involved. By informing people to perform certain activities, certain goals for dynamic context information management can be realized. As disclosed by our literature review, context information is always used as any information related to people and environment. There is no clear-cut boundary to define what context really is in general and what the meaning of it in computation is. In the case of dynamic context, both entity states and the associated relations may change rapidly. The challenge in this regard is that a proper dynamic context information management implies the need for a proper judgment of the relevance between any piece of information and the concerned

entity. This has to be made prior to computation as well as in the process of computation.

In addition, most of the existing context modeling and reasoning approaches or frameworks focus on describing context either on data level or on information (event) level. Consequently, they lack the capability of handling the knowledge associated with a dynamic context that addresses not only the capture of situational changes but also the interpretation of the interrelationships and implications of the changes. Actually, the rapid changes of context should be tracked for optimal control of real-life processes. As demonstrated in the state of the art in the field of automated and context-sensitive messaging, dynamically changing situations of people should be dealt with in order to increase the quality of informing. However, the existing solutions only consider static context information of people and they can hardly be applied to process the heterogeneous, unstructured and dynamic context of people.

Putting together everything, our conclusion is that current research with regard to dynamic context information processing (DCIP) is still in its infancy, not to mention using DCIP in I-CPSs to realize dynamic context representation and management. Due to the proliferation of smart I-CPS applications in recent publications, an adaptive and smart computational mechanism (served as a software tool) is needed to support the development and implementation of such I-CPSs. Circumventing these limitations is in the center of our research inquiry and development efforts.

#### **2.4.2. Conclusions concerning the requirements exploration**

In this chapter, we explored the technical requirements for DCIP from four main sources, including (i) the features of smart (i.e. self-aware and self-adaptive) systems, (ii) necessary operational steps taken by I-CPSs for the purpose of dynamic context information management, (iii) general needs of I-CPSs applications for providing personalized messages to users, and (iv) the specific needs of an indoor fire evacuation guiding application case. All the concerned domains contribute to the requirements for DCIP. The consistency of the obtained multi-source requirements has been checked. Additional requirements have been identified based on the results of consistency checking.

Based on the work, it is concluded that processing of dynamic context information is a necessary step towards the realization of smart behavior (self-awareness and self-adaptation) of I-CPSs. As an important character of a smart system, dynamic context information management raises the need of effective computational mechanisms (i) to represent the aggregated descriptive context data, (ii) to infer the hidden knowledge (implication of a phenomenon)

from a time-varying process, (iii) to use the learnt knowledge in a dedicated decision-making operation, and (iv) to construct situation-dependent messages to enhance personalized informing.

From the application point of view, the discussed indoor fire evacuation guiding system (IFECS) is a typical I-CPS. It highlights the necessity of using self-aware and self-adaptive operations to manipulate the time-varying context of entities, which are associated with the states of and relations to other entities. The requirements aggregated from the IFECS are more specific than the requirements with regard to I-CPSs. This implies that the computational mechanisms developed in this work can be first validated considering the fire evacuation application, as a reference. Then, the functions, algorithms and data constructs contained in the mechanisms are to be generalized with multiple I-CPS applications.

### **2.4.3. Possible research and development opportunities**

It is evidential that there are many possible research and development opportunities. For this reason, only the most exigent ones are addressed below:

- (i) *Computational frameworks for dynamic context information representation and management.* The core issue of dynamic context representation and management is capturing the rapidly and simultaneously happening changes in the surround information concerning mutually connected and/or isolated entities. This makes the decision-making in dynamically changing and high complexity processes difficult. Therefore, it seems to be necessary to consider and develop a framework which enables designers to build decision-making components into I-CPSs in a somewhat systematic, rather than just in an intuitive manner.
- (ii) *Effective approaches to handling states and relations of entities during run-time.* For the purpose of managing time-varying contexts, quasi-real-time computation should be achieved. However, when the number of the involved entities increases, the number of relations among them may squarely or even exponentially increase. This poses a challenge for real-time processing of the currently known approaches to DCIP.
- (iii) *Inferring semantic knowledge about dynamic context.* The fundamental challenge is to take out and interpret the meaning (and the actual implications) of contextual changes. As discussed in the related literature, the concept of dynamic context of entities has been defined either too general or too specific. For the purpose of inferring contextual knowledge, dynamic context should be purposefully and clearly defined. Knowledge associated with dynamic contexts, such as the relevance of the informational elements of context to entities, the actual influence of context on entities, and the implications of context on entities, should be specified in an application-independent manner.

These concepts play a crucial role with regard to inferring.

- (iv) *Personalized and real-time multi-target informing in emergency applications.* Hazard-intense applications of cyber-physical systems (CPSs), such as the evacuation of a building on fire, require personalized informing on the basis of a real-time assessment of individual context. A typical challenge is that informing and instructive messages must be concurrently and context-sensitively generated for and forwarded to multiple individuals. Towards this end, the actual context of the concerned individuals should be parallel computed. Computational mechanisms, such as crowd management strategy, person-oriented instruction edition, and message scheduling and communication are to be parallelized.
- (v) *Development of computational platforms for a family of I-CPSs.* It has been revealed that certain families of I-CPSs have common characteristics and requirements with regard to DCIP. This lends itself to a novel research challenge. The developed reasoning mechanisms can be decontextualized to make them capable to provide dynamic special context computation and reasoning in an application semantics independent manner. An important step towards this is to design and implement a (multi-)functional kernel that can provide computational services according to specific DCIP requests.

## 2.5. References

- [1] Chen, G., Kotz, D. (2000). A survey of context-aware mobile computing research. *Dartmouth Computer Science Technical Report*. TR2000-381.
- [2] Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4), pp. 263-277.
- [3] Hong, J.Y., Suh, E.H., & Kim, S.J. (2009). Context-aware systems: A literature review and classification. *Expert Systems with Applications*. 36(4), pp. 8509-8522.
- [4] Yamabe, T., Takagi, A., & Nakajima, T. (2005). Citron: A context information acquisition framework for personal devices. In: *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 489-495.
- [5] Temdee, P., & Prasad, R. (2018). Elements of context awareness. In: *Context-Aware Communication and Computing: Applications for Smart Environment*. Springer International Publishing, pp. 33-64.
- [6] Alegre, U., Augusto, J.C., & Clark, T. (2016). Engineering context-aware systems and applications: A survey. *Journal of Systems and Software*. 117,

pp. 55-83.

- [7] Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *Communications Surveys & Tutorials*, IEEE, 16(1), pp. 414-454.
- [8] He, Y., Li, L., He, K., & Chen, X. (2009). A contextual information acquisition approach based on semantics and mashup technology. In: *Cloud Computing*. Springer Berlin Heidelberg, pp. 510-518.
- [9] Nguyen, N., & Khan, M.M.H. (2013). Context aware data acquisition framework for dynamic data driven applications systems. In: *Proceedings of the Military Communications Conference*, pp. 334-341.
- [10] Mainland, G., Parkes, D.C., & Welsh M. (2005). Decentralized, adaptive resource allocation for sensor networks. In: *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation*, 2, pp. 315-328.
- [11] Ye, J., Coyle, L., Dobsonm S., & Nixon, P. (2007). Using situation lattices to model and reason about context. In: *Proceedings of the 4th International Workshop on Modeling and Reasoning in Context*, pp. 1-12.
- [12] Nguyen, N., & Khan, M.M.H. (2015). A closed-loop context aware data acquisition and resource allocation framework for dynamic data driven applications systems (DDDAS) on the cloud. *Journal of Systems and Software*, 109, pp. 88-105.
- [13] Li, X., Eckert, M., Martinez J.F., & Rubio, G. (2015). Context aware middleware architectures: survey and challenges. *Sensors* (Basel, Switzerland) 15(8), pp. 20570-20607.
- [14] Serpen, G., Li, J.K., & Liu, L.Q. (2013). AI-WSN: Adaptive and intelligent wireless sensor network. *Procedia Computer Science*, 20, pp. 406-413.
- [15] Sachidananda, V., Khelil, A., & Suri, N. (2010). Quality of information in wireless sensor networks: A survey. In: *Proceedings of ICIQ*, pp. 1-15.
- [16] Chou, C.T., Rana, R., & Hu, W. (2009). Energy efficient information collection in wireless sensor networks using adaptive compressive sensing. In: *Proceedings of the 34th Conference on Local Computer Networks*, Zurich, IEEE, pp. 443-450.
- [17] Balakrishnan, D., & Nayak, A. (2009). A context dissemination scheme for adaptive and fault-tolerant services in heterogeneous environments. In: *Proceedings of the Global Information Infrastructure Symposium*, pp. 1-4.
- [18] Zhu, Y.M., Liu, Y.H., Ni, L.M., & Zhang, Z. (2007). Low-power distributed event detection in wireless sensor networks. In: *Proceedings of the 26th*

- International Conference on Computer Communications*, Anchorage, AK, IEEE, pp. 2401-2405.
- [19] Hoareau, C., & Satoh, I. (2009). Modeling and processing information for context-aware computing: A survey. *New Generation Computing*, 27(3), pp. 177-196.
- [20] Abowd, G.D., Atkeson, C.G., Brotherton, J., Enqvist, T., Gulley, P., & LeMon, J. (1998). Investigating the capture, integration and access problem of ubiquitous computing in an educational setting. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Los Angeles, USA, pp. 440-447.
- [21] Schmidt, A., Beigl, M., & Gellersen, H-W. (1999). There is more to context than location. *Computers & Graphics*, 23(6), pp. 893-901.
- [22] Dedes. M., Lewandowska. A., & Seitz. J. (2005). Definition and implementation of context information. In: *Proceedings of the 2nd Workshop on Positioning, Navigation and Communication & 1st Ultra-Wideband Expert Talk*, Hannover, Germany, pp. 63-68.
- [23] Schilit. B., Adams. N., & Want. R. (1994). Context-aware computing applications. In: *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, Xerox Corporation, Palo Alto Research Center. pp. 85-90.
- [24] Samulowitz. M., Michahelles. F., & Linnhoff-Popien. C. (2002). Capeus: An architecture for context-aware selection and execution of services. In: *Proceedings of the Conference on New Developments in Distributed Applications and Interoperable Systems*, 70, pp. 23-39.
- [25] Yin, H., Cui, B., Chen, L., Hu, Z., & Huang, Z. (2014). A temporal context-aware model for user behavior modeling in social media systems. In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, Snowbird, Utah, USA, pp. 1543-1554.
- [26] Henriksen, K., & Indulska, J. (2006). Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing* 2(1), pp. 37-64.
- [27] Mohan, P., & Singh, M. (2015). Ontological approach for context aware modeling and reasoning in sensor network. *International Journal of Computer Technology and Applications*, 6(2), pp. 244-248.
- [28] Yılmaz, Ö., & Erdur, R.C. (2012). iConAwa-An intelligent context-aware system. *Expert Systems with Applications*, 39(3), pp. 2907-2918.
- [29] Chen, G., Li, M., & Kotz, D. (2008). Data-centric middleware for



- context-aware pervasive computing. *Pervasive and Mobile Computing*, 4(2), pp. 216-253.
- [30] Phoha, S., Virani, N., Chattopadhyay, P., Sarkar, S., Smith, B., & Ray, A. (2014). Context-aware dynamic data-driven pattern classification. *Procedia Computer Science*, 29, pp. 1324-1333.
- [31] Chandrashekar, M., & Lee, Y. (2016). Visual context learning with big data analytics. In: *Proceedings of the 16th International Conference on Data Mining Workshops*, IEEE, pp. 600-607.
- [32] Jun-Hong, L., Chiung-Ying, W., & Ren-Hung, H. (2009). An open framework for distributed context management in ubiquitous environment. In: *Proceedings of the Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, pp. 88-93.
- [33] Chin, C.Y., Zhang, D., & Gurusamy, M. (2005). Orion: P2P-based inter-space context discovery platform. In: *Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pp. 490-493.
- [34] Zhang, D., Chin, C.Y., & Gurusamy, M. (2005). Supporting context-aware mobile service adaptation with scalable context discovery platform. In: *Proceedings of the 61st Vehicular Technology Conference*, 5, pp. 2859-2863.
- [35] SangKwon, M., Kim, J., & Daeyeon, P. (2007). USD Protocol: Ubiquitous service discovery protocol on infrastructure-based architecture for ubiquitous fashionable computer. In: *Proceedings of the International Conference on Multimedia and Ubiquitous Engineering*, pp. 779-784.
- [36] Van Bunningen, A.H., Feng, L., & Apers, P.M. (2005). Context for ubiquitous data management. In: *Proceedings of the International Workshop on Ubiquitous Data Management*, IEEE, pp. 17-24.
- [37] Guan, D., Yuan, W., Lee, S., & Lee, Y.K. (2007). Context selection and reasoning in ubiquitous computing. In: *Proceedings of the International Conference on Intelligent Pervasive Computing*, pp. 184-187.
- [38] Rizou, S., Häussermann, K., Dürr, F., Cipriani, N., & Rothermel, K. (2010). A system for distributed context reasoning. In: *Proceedings of the Sixth International Conference on Autonomic and Autonomous Systems*, pp. 84-89.
- [39] Chong, S.K., McCauley, I., Loke, S.W., & Krishnaswamy, S. (2007). Context-aware sensors and data muffling. In: *Proceeding of the Conference on Context Awareness for Self-managing Systems (Devices, Applications and Networks)*, pp. 103-117.

- [40] Padovitz, A., WaiLoke, S., Zaslavsky, A., & Burg, B. (2007). Verification of uncertain context based on a theory of context spaces. *International Journal of Pervasive Computing and Communications*, 3(1), pp. 30-56.
- [41] Hartmann, T., Fouquet, F., Nain, G., Morin, B., Klein, J., & Le Traon, Y. (2014). Reasoning at runtime using time-distorted contexts: A models@run.time-based approach. In: *Proceedings of the 26th International Conference on Software Engineering and Knowledge Engineering*. Knowledge Systems Institute, USA. pp. 586-591.
- [42] Anagnostopoulos, C., & Kolomvatsos, K. (2018). Predictive intelligence to the edge through approximate collaborative context reasoning. *Applied Intelligence*, 48(4), pp. 966-991.
- [43] Zhang, Q.C., & Zeng, G.P. (2012). Research on context service framework for supporting context-aware computing. *Computer Engineering and Design*. 33(5), pp. 2036-2041.
- [44] Dey, A.K., Abowd, G.D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2-4), pp. 97-166.
- [45] Wang, C-Y., Hwang, R-H., & Ting, C-K. (2011). UbiPaPaGo: Context-aware path planning. *Expert Systems with Applications*, 38(4), pp. 4150-4161.
- [46] Tang, K., Chen, S., & Khattak, A.J. (2018). Personalized travel time estimation for urban road networks: A tensor-based context-aware approach. *Expert Systems with Applications*, 103, pp. 118-132.
- [47] Taconet, C., Kazi-Aoul, Z., Zaier M., & Conan, D. (2009). CA3M: A runtime model and a middleware for dynamic context management. In: *Proceedings of the On the Move to Meaningful Internet Systems: OTM 2009*. Springer Berlin Heidelberg, pp. 513-530.
- [48] Jaroucheh, Z., Liu, X., & Smith, S. (2010). CANDEL: Product line based dynamic context management for pervasive applications. In: *Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, pp. 209-216.
- [49] Gallacher, S., Papadopoulou, E., Abu-Shaabn, Y., Taylor, N.K., & Williams, M.H. (2014). Dynamic context-aware personalisation in a pervasive environment. *Pervasive and Mobile Computing* 10, pp. 120-137.
- [50] Wan, J.F., Zhang, D.Q., Zhao, S.J., Yang, L.T., & Lloret, J. (2014). Context-aware vehicular cyber-physical systems with cloud support: Architecture, challenges, and solutions. *IEEE Communications Magazine*, 52(8), pp. 106-113.

- [51] Forsström, S., & Kanter, T. (2012). Enabling continuously evolving context information in mobile environments by utilizing ubiquitous sensors. In: *Proceedings of the Third International ICST Conference on Mobile Networks and Management*. Springer Berlin, Heidelberg, pp. 289-303.
- [52] Wolf, H., Herrmann, K., & Rothermel, K. (2009). Modeling dynamic context awareness for situated workflows. In: *Proceedings of the OTM 2009 Workshop On the Move to Meaningful Internet Systems*. Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 98-107.
- [53] Euzenat, J., Pierson, J., & Ramparany, F. (2008). Dynamic context management for pervasive applications. *The Knowledge Engineering Review*, 23(1), pp. 21-49.
- [54] Zhu HM, Stuart E., Siegel, & Michael D.(2004). Reasoning about temporal context using ontology and abductive constraint logic programming. In: *Proceedings of the Principles and Practice of Semantic Web Reasoning*, PPSWR 2004: St. Malo, France, pp. 90-101.
- [55] Maren, S., Phan, KL., & Liberzon, I. (2013). The contextual brain: implications for fear conditioning, extinction and psychopathology. *Nature Reviews Neuroscience*, 14(6), pp. 417-428.
- [56] Junior, W., Oliveira, E., Santos, A., & Dias, K. (2019). A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment. *Future Generation Computer Systems*, 90, pp. 503-520.
- [57] Senge, R., Bösner, S., Dembczyński, K., Haasenritter, J., Hirsch, O., Donner-Banzhoff, N., & Hüllermeier, E. (2014). Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Information Sciences*, 255, pp. 16-29.
- [58] Bobek, S., & Nalepa, GJ. (2017). Uncertain context data management in dynamic mobile environments. *Future Generation Computer Systems*, 66, pp. 110-124.
- [59] Bobek, S., & Nalepa, GJ. (2017). Uncertainty handling in rule-based mobile context-aware systems. *Pervasive and Mobile Computing*, 39, pp. 159-179.
- [60] Zhang, D., Guo, M., Zhou, J., Kang, D., & Cao, J. (2010). Context reasoning using extended evidence theory in pervasive computing environments. *Future Generation Computer Systems*, 26(2), pp. 207-216.
- [61] Ranganathan, A., Al-Muhtadi, J., & Campbell, RH. (2004). Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 3(2), pp. 62-70.

- [62] Chahuara, P., Portet, F., & Vacher, M. (2017). Context-aware decision making under uncertainty for voice-based control of smart home. *Expert Systems with Applications*, 75, pp. 63-79.
- [63] Kwon, O., & Kim, M. (2004). My message: Case-based reasoning and multicriteria decision making techniques for intelligent context-aware message filtering. *Expert Systems with Applications*, 27(3), pp. 467-480.
- [64] Liu, X. (2015). Modeling users' dynamic preference for personalized recommendation. In: *Proceedings of the 24th International Conference on Artificial Intelligence*, Buenos Aires, Argentina, pp. 1785-1791.
- [65] Nalepa, G.J., Kutt, K., & Bobek, S. (2018). Mobile platform for affective context-aware systems. *Future Generation Computer Systems*, 92, pp. 490-503.
- [66] Bar-David, R., & Last, M. (2015). Context-aware location prediction. In: *Proceedings of the 5th International Workshop on Big Data Analytics in the Social and Ubiquitous Context*, Springer-Verlag New York, pp. 165-185.
- [67] Mayrhofer, R. (2005). An architecture for context prediction. Schriften der Johannes-Kepler-Universität Linz: TraunerVerlag.
- [68] Liu, W., Wang, J., Sangaiah, A.K., & Yin, J. (2018). Dynamic metric embedding model for point-of-interest prediction. *Future Generation Computer Systems*, 83, pp. 183-192.
- [69] Mu, L., Xiao, B., Yuan, Z., & Li, D. (2015). The research of CREAM prediction analysis method based on BP neural network under dynamic context. In: *Proceedings of the 2015 First International Conference on Reliability Systems Engineering (ICRSE)*, pp. 1-6.
- [70] Neogi, S., Hoy, M., Chaoqun, W., & Dauwels, J. (2017). Context based pedestrian intention prediction using factored latent dynamic conditional random fields. In: *Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1-8.
- [71] Ridhawi, Y.A., Ridhawi, I.A., Karmouch, A., & Nayak, A. (2011). A context-aware and location prediction framework for dynamic environments. In: *Proceedings of the 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 172-179.
- [72] Rosa, J.H., Barbosa, JLV., & Ribeiro, GD. (2016). ORACON: An adaptive model for context prediction. *Expert Systems with Applications*, 45, pp. 56-70.
- [73] Mayrhofer, R. (2005). Context prediction based on context histories:

- Expected benefits, issues and current state-of-the-art. *Cognitive science research paper-University of Sussex CSRP*, 577: 31.
- [74] Burbey, I., & Martin, T. L. (2012). A survey on predicting personal mobility. *International Journal of Pervasive Computing and Communications*, 8(1), pp. 5-22.
- [75] Boytsov, A., & Zaslavsky, A. (2011). Context prediction in pervasive computing systems: Achievements and challenges. In: *Proceedings of the Supporting Real Time Decision-Making: The Role of Context in Decision Support on the Move*. Springer US, Boston, MA: pp. 35-63.
- [76] Adomavicius, G., & Tuzhilin, A. (2011). Context-aware recommender systems. In: *Recommender systems handbook*. Springer, pp. 217-253.
- [77] Minh, P., Yehenew, M., Do, H., & Sheng, W. (2018). Delivering home healthcare through a Cloud-based Smart Home Environment (CoSHE). *Future Generation Computer Systems*, 81, pp. 129-140.
- [78] Mihajlovic, V., & Petkovic, M. (2001). Dynamic Bayesian networks: A state of the art. University of Twente, Centre for Telematics and Information Technology, Enschede, pp. 1-37.
- [79] Wang, Y., Tan, G., Wang, Y., & Yin, Y. (2012). Perceptual control architecture for cyber-physical systems in traffic incident management. *Journal of Systems Architecture*, 58(10), pp. 398-411.
- [80] Park, HS., Oh, K., & Cho, SB. (2011). Bayesian network-based high-level context recognition for mobile context sharing in cyber-physical system. *International Journal of Distributed Sensor Networks*, 7(1), 650387.
- [81] Hong, JH., Yang, SI., & Cho, SB. (2010). ConaMSN: A context-aware messenger using dynamic Bayesian networks with wearable sensors. *Expert Systems with Applications*, 37(6), pp. 4680-4686.
- [82] Yager, RR. (1987). On the Dempster-Shafer framework and new combination rules. *Information Sciences*, 41(2), pp. 93-137.
- [83] Ye, J., Dobson, S., & McKeever, S. (2012). Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing*, 8(1), pp. 36-66.
- [84] Siaterlis, C., & Genge, B. (2011). Theory of evidence-based automated decision making in cyber-physical systems. In: *Proceedings of the IEEE International Conference on Smart Measurements for Future Grids*, pp. 107-112.
- [85] Liu, P., & Zhang, JA. (2008). Context-aware application infrastructure with reasoning mechanism based on Dempster-Shafer evidence theory. In:

*Vehicular Technology Conference*, pp. 2834-2838.

- [86] Chang, CK., Hsin-yi, J., Hua, M., & Oyama, K. (2009). Situ: A situation-theoretic approach to context-aware service evolution. *IEEE Transactions on Services Computing*, 2(3), pp. 261-275.
- [87] Dargie, W. (2007). The role of probabilistic schemes in multisensorcontext-awareness. In: *Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, 19-23 March, pp. 27-32.
- [88] Anagnostopoulos, CB., Ntarladimas, Y., & Hadjiefthymiades, S. (2007). Situational computing: An innovative architecture with imprecise reasoning. *Journal of Systems and Software*, 80(12), pp. 1993-2014.
- [89] Fok, C-L., Roman, G-C., & Lu, C. (2013). Adaptive service provisioning for enhanced energy efficiency and flexibility in wireless sensor networks. *Science of Computer Programming*, 78(2), pp. 195-217.
- [90] Jing, L., Zhou, Y., Cheng, Z., Yen, NY., & Park, JJ. (2013). Context-aware service roaming for heterogeneous embedded devices over cloud. *Journal of Systems Architecture*, 59(9), pp. 776-784.
- [91] Hussein, D., Soochang, P., Han, S.N., & Crespi, N. (2015). Dynamic social structure of things: A contextual approach in CPSS. *IEEE Internet Computing*, 19(3), pp. 12-20.
- [92] Magid, Y., Sharon, G., Arcushin, S., Ben-Harrush, I., & Rabinovich, E. (2010). Industry experience with the IBM active middleware technology (AMiT) complex event processing engine. In: *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems*, Cambridge, UK, 140.
- [93] Khan, N., Alegre, U., Kramer, D., & Augusto, JC. (2017). Is 'Context-Aware Reasoning = Case-Based Reasoning'? In: Cham, *Modeling and Using Context*. Springer International Publishing, pp. 418-431.
- [94] Kolodner, J.L. (1992). An introduction to case-based reasoning. *Artificial Intelligence Review*, 6(1), pp. 3-34.
- [95] Bergmann, R., Kolodner, J., & Plaza, E. (2005). Representation in case-based reasoning. *Knowledge Engineering Review*, 20(3), pp. 209-213.
- [96] Woskov, S.M., Grimaila, M.R., Mills, R.F., & Haas, M.W. (2011). Design considerations for a case-based reasoning engine for scenario-based cyber incident notification. In: *Proceedings of the IEEE Symposium on Computational Intelligence in Cyber Security*, Paris, 11-15 April, pp. 84-91.
- [97] Lee, J.S., & Lee, J. C. (2007). Context awareness by case-based reasoning

- in a music recommendation system, *Ubiquitous Computing Systems*, pp. 45-58.
- [98] Petnga, L., Austin, M. (2013). Ontologies of time and time-based reasoning for MBSE of cyber-physical systems. In: *Proceedings of the 2013 Conference on Systems Engineering Research*, Atlanta, GA, 16, pp. 403-412.
- [99] Ni, D., Liu, H., Ding, W., Xie, Y., Wang, H., Pishro-Nik, H., & Yu, Q. (2012). Cyber-physical integration to connect vehicles for transformed transportation safety and efficiency. In: *Proceedings of the Advanced Research in Applied Artificial Intelligence*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 88-94.
- [100] Binbin, Z., Jiannong, C., & Hejun, W.(2011). Adaptive Traffic Light Control of Multiple Intersections in WSN-Based ITS. In: *Proceedings of the IEEE 73rd Vehicular Technology Conference*, 15-18 May, pp. 1-5.
- [101] Zhang, L., & Sun, J-T. (2009). Text generation. In: *Encyclopedia of Database Systems*. Springer, Boston, pp. 3048-3051.
- [102] Dethlefs, N. (2014). Context-sensitive natural language generation: From knowledge-driven to data-driven techniques. *Language and Linguistics Compass*, 8(3), pp. 99-115.
- [103] Wu, BF., Yao, SJ., Hou, LW., Chang, PJ., Tseng, WJ., Huang, CW., Chen, YS., & Yang, PY. (2016). Intelligent shopping assistant system. In: *Proceedings of the 2016 International Automatic Control Conference (CACCS)*, 9-11 Nov, pp. 236-241.
- [104] Braunhofer, M., Elahi, M., Ricci, F., & Schievenin, T. (2013). Context-aware points of interest suggestion with dynamic weather data management. Springer International Publishing, pp. 87-100.
- [105] Pashtan, A., Blattler, R., Heusser, A., & Scheuermann, P. (2003). CATIS: A context-aware tourist information system. In: *Internet Measurement Conference*, 2003.
- [106] Bagci, H., & Karagoz, P. (2016). Context-aware location recommendation by using a random walk-based approach. *Knowledge and Information Systems*, 47(2), pp. 241-260.
- [107] Mellish, C., Scott, D., Cahill, L., Paiva, D., Evans, R., & Reape, M. (2006). A reference architecture for natural language generation systems. *Natural Language Engineering*, 12(1), pp. 1-34.
- [108] Zheng, HT., Wang, W., Chen, W., Chen, JY., & Sangaiah, AK. (2017). Automatic generation of news comments based on gated attention neural

- networks. *IEEE Access*, pp. 702-710.
- [109] Tang, J., Yang, Y., Carton, S., Zhang, M., & Mei, Q. (2016). Context-aware Natural Language Generation with Recurrent Neural Networks. arXiv:1611.09900
- [110] Zhou, H., Huang, M., & Zhu, Xi. (2016). Context-aware natural language generation for spoken dialogue systems. In: *Proceedings of the 26th International Conference on Computational Linguistics: COLING 2016, Technical Papers*, Osaka, Japan, pp. 2032-2041.
- [111] Nakanishi, Y., Tsuji, T., Ohyama, M., & Hakozaki, K. (2000). Context aware messaging service: A dynamical messaging delivery using location information and schedule information. *Personal Ubiquitous Computing*, 4(4), pp. 221-224.
- [112] Knox, S.SR., Coyle, L., Clear, A., Dobson, S., Quigley, A., & Nixon, P. (2008). Scatterbox: Context-aware message management. *Revue d'Intelligence Artificielle*, 22(5), pp. 549-568.
- [113] Knox, SC., Adrian, K., Shannon, Ross., Coyle, Lorcan., Dobson, Simon., Quigley, Aaron., & Nixon, Paddy. (2007). Towards scatterbox: a context-aware message forwarding platform. In: *Proceedings of the Fourth International Workshop on Modeling and Reasoning in Context in conjunction with Context '07*, pp. 1-12.
- [114] Pinto, FC., Videira, A., Carapeto, N., & Dinis, M. (2012). Context-aware multimedia distribution for multiparty communications. In: *Proceedings of the 6th International ICST Conference on Mobile Multimedia Communications: MOBIMEDIA 2010*, Springer Berlin Heidelberg, pp. 88-101.
- [115] Coutinho, N., Sargento, S., Condeixa, T., Janneteau, C., Simoes, J., Christophorou, C., Pinto, FC., Neto, A., & Carapeto, N. (2010). Context-aware multiparty services delivery: Evaluation and experimentation. In: *Proceedings of the 2010 Future Network & Mobile Summit*, pp. 1-12.
- [116] Maracic, H., Miskovic, T., Kusek, M., & Lovrek, I. (2014). Context-aware multi-agent system in machine-to-machine communication. In: *Proceedings of the 18th Annual Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, Gdynia, Poland, pp. 241-250.
- [117] Costa, JM., & Miao, GW. (2014). Context-aware machine-to-machine communications. In: *Proceedings of the IEEE Conference on Computer Communications Workshops*, Toronto, pp. 730-735.



- [118] Lovrek, I. (2012). Context awareness in mobile software agent network. *Tehničkeznosti*, 15, pp. 7-28.
- [119] Kusek, M., Lovrek, I., & Maracic, H. (2013). Rich presence information in agent-based machine-to-machine communication. *Procedia Computer Science*, 22, pp. 321-329.
- [120] Jin, B. (2013). An introduction: Context-aware computing for secure message transmission system in M2M. *Smart Computing Review*, 3(6), pp. 416-424.
- [121] Cha, I., Shah, Y., Schmidt, AU., Leicher, A., & Meyerstein, MV. (2009). Trust in M2M communication. *IEEE Vehicular Technology Magazine*, 4(3), pp. 69-75.
- [122] Qian, Y., & Moayeri, N. (2008). Design of secure and application-oriented VANETs. *IEEE 67th Vehicular Technology Conference*, Singapore, pp. 2794-2799.
- [123] Jones, S., & O'Neill, E. (2009). Context-aware messaging: How personal, spatial and temporal constraints affect text-based communication. In: *Proceedings of the 8th International Conference on Mobile and Ubiquitous Multimedia*, Cambridge, United Kingdom, 8.
- [124] Ochsman, RB., & Chapanis, A. (1974). The effects of 10 communication modes on the behavior of teams during co-operative problem-solving. *International Journal of Man-Machine Studies*, 6(5), pp. 579-619.
- [125] Cohen, PR., & Oviatt, SL. (1995). The role of voice input for human-machine communication. *Proceedings of the National Academy of Sciences*, 92(22), pp. 9921-9927.
- [126] Zaguia, A., Hina, MD., Tadj, C., & Ramdane-Cherif, A. (2010). Interaction Context-Aware Modalities and Multimodal Fusion for Accessing Web Services. *Ubiquitous Computing and Communication Journal*, 5(4), pp. 1-15.
- [127] Ghorbel, M., Mokhtari, M., & Renouard, S. (2006). A distributed approach for assistive service provision in pervasive environment. In: *Proceedings of the 4th International Workshop on Wireless mobile applications and services on WLAN hotspots*, Los Angeles, CA, USA, pp. 91-100.
- [128] Gouin-Vallerand, C., Abdulrazak, B., Giroux, S., & Dey, AK. (2013). A context-aware service provision system for smart environments based on the user interaction modalities. *Journal of Ambient Intelligence & Smart Environments*, 5(1), pp. 47-64
- [129] Gaham, M., Bouzouia, B., & Achour, N. (2015). Human-in-the-loop

- cyber-physical production systems control (HiLCP2sC): A multi-objective interactive framework proposal. In: *Service Orientation in Holonic and Multi-agent Manufacturing*, Springer International Publishing, pp. 315-325.
- [130] Plesa, R., & Logrippo, L. (2007). An agent-based architecture for context-aware communication. In: *Proceedings of 21st International Conference on Advanced Networking and Applications Workshops*, Niagara Falls, Ont., pp. 133-138.
- [131] Lei, H., & Ranganathan, A. (2004). Context-aware unified communication. In: *Proceedings of the 2004 IEEE International Conference on Mobile Data Management*, pp. 176-186.
- [132] Chihani, B., Bertin, E., & Crespi, N. (2012). Enhancing M2M communication with cloud-based context management. In: *Proceedings of the 6th International Conference on Next Generation Mobile Applications, Services and Technologies*, Paris, pp. 36-41.
- [133] Venkatasubramanian, KK., Mukherjee, T., Gupta, SKS. (2014). CAAC: An adaptive and proactive access control approach for Eemergencies in smart infrastructures. *ACM Transactions on Autonomous and Adaptive Systems*, 8(4), pp. 1-18.
- [134] Hong, JI., & Landay, JA. (2001). An infrastructure approach to context-aware computing. *Human-Computer Interaction*, 16(2), pp. 287-303.
- [135] Schilit BN. (1995). A context-aware system architecture for mobile distributed computing. Columbia University.
- [136] Brown, PJ. (1996). The stick-e document: A framework for creating context-aware applications. In: *Proceedings of the Electronic Publishing' 96*, pp. 259-272.
- [137] Pascoe, J. (1997). The stick-e note architecture: extending the interface beyond the user. In: *Proceedings of the 2nd International Conference on Intelligent user interfaces*, Orlando, Florida, USA, pp. 261-264.
- [138] Pongpaichet, S., Singh, VK., Jain, R., & Pentland, A. (2013). Situation fencing: Making geo-fencing personal and dynamic. In: *Proceedings of the 1st ACM International Workshop on Personal Data Meets Distributed Multimedia*, Barcelona, Spain.
- [139] Lun, YL., & Cheng, LL. (2011). The research on the model of the context-aware for reliable sensing and explanation in cyber-physical system. *Procedia Engineering*, 15, pp. 1753-1757.
- [140] Mamo, B., & Ejigu, D. (2014). A generic layered architecture for context

- aware applications. *Procedia Computer Science*, 34, pp. 619-624.
- [141] Bardram, J. (2005). The Java Context Awareness Framework (JCAF): A service infrastructure and programming framework for context-aware applications. In: *Proceedings of the Third international conference on Pervasive Computing*, pp. 98-115.
- [142] Buthpitiya, S., Luqman, F., Griss, M., Bo, X., & Dey, A.K. (2012). Hermes -- A context-aware application development framework and toolkit for the mobile environment. In: *Proceedings of the 26th International Conference on Advanced Information Networking and Applications Workshops*, Fukuoka, 26-29 March, pp. 663-670.
- [143] David, P-C., & Ledoux, T. (2005). WildCAT: A generic framework for context-aware applications. In: *Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing*, Grenoble, France, pp. 1-7.
- [144] García, Ó., Alonso, R., Tapia, D., García, E., De la Prieta, F., & de Luis, A. (2012). CAFCLA: A conceptual framework to develop collaborative context-aware learning activities. In: *Proceedings of the Workshop on Learning Technology for Education in Cloud*, pp. 11-21.
- [145] Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H., & Malm, E.J. (2003). Managing context information in mobile devices. *IEEE Pervasive Computing*, 2(3), pp. 42-51.
- [146] Minyoung, K., Stehr, MO., Jinwoo, K., & Soonhoi, H. (2010). An application framework for loosely coupled networked cyber-physical systems. In: *Proceedings of the IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing*, Hong Kong, pp. 144-153.
- [147] Sun, J., & Zhang, Y. (2014). A middleware for highly dynamic distribution in CPS environment. In: *Proceedings of the 3rd International Conference on Context-Aware Systems and Applications*, Dubai, United Arab Emirates, pp. 169-172.
- [148] Palau, M., Gómez-Sebastià, I., Ceccaroni, L., Vázquez-Salceda, J., & Nieves, J.C. (2010). A framework for the development and maintenance of adaptive, dynamic, context-aware information services. In: *International Conference on Agents and Artificial Intelligence*, Springer, pp. 227-239.
- [149] Gu, T., Pung, H.K., & Zhang, D.Q. (2004). A middleware for building context aware mobile services. In: *Proceedings of IEEE Vehicular Technology Conference*, Milan, Italy, pp. 2656-2660.
- [150] Mehrotra, S., Venkatasubramanian, N., Stehr, MO., & Talcott, C. (2012). Pervasive sensing and monitoring for situational awareness. In: *Handbook*

*on Securing Cyber-Physical Critical Infrastructure*. pp. 505-535.

- [151] Hore, B., Jafarpour, H., Jain, R., Shengyue, J., Massaguer, D., Mehrotra, S., Venkatasubramanian, N., & Westermann, U. (2007). Design and implementation of a middleware for sentient spaces. In: *Proceedings of the Intelligence and Security Informatics*, New Brunswick, NJ, pp. 137-144.
- [152] Park, KJ., Zheng, R., & Liu, X. (2012). Cyber-physical systems: Milestones and research challenges. *Computer Communications*, 36(1), pp. 1-7.
- [153] Sanislav, T., & Miclea, L. (2012). Cyber-physical systems-concept, challenges and research areas. *Journal of Control Engineering and Applied Informatics*, 14(2), pp. 28-33.
- [154] Bhatti, AM., & Masud, M. (2014). Context aware intelligent wallet for healthcare. *International Journal of Computer Applications*, 96(15), pp. 31-36.
- [155] Martín, D., Lamsfus, C., & Alzua-Sorzabal, A. (2016). A cloud-based platform to develop context-aware mobile applications by domain experts. *Computer Standards & Interfaces*, 44, pp. 177-184.
- [156] Chen, H., Finin, T., & Joshi, A. (2003). An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, 18(3), pp. 197-207.
- [157] Chen, H., Finin, T., Anupam, J., Kagal, L., Perich, F., & Dipanjan, C. (2004). Intelligent agents meet the semantic web in smart spaces. *IEEE Internet Computing*, 8(6), pp. 69-79.
- [158] Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., & Nahrstedt, K. (2002). A middleware infrastructure for active spaces. *IEEE Pervasive Computing*, 1(4), pp. 74-83.
- [159] Roman, M., & Campbell, R. H. (2000). Gaia: enabling active spaces. In *Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*, pp. 229-234.
- [160] Shen, C.Y., Yang, D.N., & Chen, M.S. (2011). Collaborative and distributed search system with mobile devices. *IEEE Transactions on Mobile Computing*, 11(10), pp. 1478-1493.
- [161] Wan, J.F., Suo, H., Yan, H.H., & Liu, J.Q. (2011). A general test platform for cyber-physical systems: unmanned vehicle with wireless sensor network navigation. *Procedia Engineering* 24, pp. 123-127.
- [162] Seshasayee, B. (2008). Middleware-based services for virtual cooperative mobile platforms. *Georgia Institute of Technology*, Atlanta, GA, USA.

- [163] Oniga, S. (2013). ICT tools for smart homes and assisted living for elders. In: *Proceedings of Advances on Wireless Sensor Networks*, pp. 41-46.
- [164] Dao, M.S., Pongpaichet, S., Jalali, L., Kim, K., Jain, R., & Zettsu, K. (2014). A real-time complex event discovery platform for cyber-physical-social systems. In: *Proceedings of the International Conference on Multimedia Retrieval*, Glasgow, United Kingdom, pp. 201-209.
- [165] Crowley, DN., Curry, E., & Breslin, JG. (2014). From citizen sensing to citizen actuation. In: *Proceedings of the International Conference on Digital Ecosystems and Technologies (DEST)*, 24-26 July, pp. 108-113.
- [166] Wang, X.H., Zhang, D.Q., Gu, T., & Pung, H.K. (2004). Ontology based context modeling and reasoning using OWL. In: *IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 14-17 March, pp. 18-22.
- [167] Chebba, A., Bouabana-Tebibel, T., & Rubin, SH. (2015). Context in ontology for knowledge representation. In: *Advanced Computational Methods for Knowledge Engineering*. Springer International Publishing Cham, pp. 311-320.
- [168] McGuinness, D.L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, 10.
- [169] Beckett, D., & McBride, B. (2004). RDF/XML syntax specification (revised). *W3C recommendation*, pp. 7-8.
- [170] Lu, ZJ., Li, GY., & Pan, Y. (2016). A method of meta-context ontology modeling and uncertainty reasoning in SWOT. In: *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 13-15 Oct, pp. 128-135.
- [171] Aguilar, J., Jerez, M., & Rodriguez, T. (2018). CAMEnto: Context awareness meta ontology modeling. *Applied Computing and Informatics*, 14(2), pp. 202-213.
- [172] Agelli, M., & Colucci, F. (2006). An ontology-based content management system for a dynamic operating context: Issues and prototype evaluation. In: *ECKM 2006: The 7th European Conference on Knowledge Management*, pp. 1-9.
- [173] Gu, T., Pung, HK., & Zhang, DQ. (2004). A middleware for building context-aware mobile services. In: *2004 IEEE 59th Vehicular Technology Conference*, 17-19 May, pp. 2656-2660.
- [174] Ranganathan, A., & Campbell, R. H. (2003). An infrastructure for

- context-awareness based on first order logic. *Personal and Ubiquitous Computing*, 7(6), pp. 353-364.
- [175] Wang, X., Dong, J. S., Chin, C. Y., Hettiarachchi, S., & Zhang, D. (2004). Semantic space: An infrastructure for smart spaces. *IEEE Pervasive computing*, 3(3), pp. 32-39.
- [176] Miraoui, M., El-etriby, S., Zaid, AA., & Tadj, C. (2016). A logic-based context modeling and context-aware services adaptation for a smart office. *International Journal of Advanced Studies in Computers, Science and Engineering*, 5(11), pp. 1-6.
- [177] Nalepa, G. J., LIGEZA, A., & Kaczor, K. (2011). Formalization and modeling of rules using the XTT2 method. *International Journal on Artificial Intelligence Tools*, 20(6), pp. 1107-1125.
- [178] Nalepa, G.J., & Bobek, S. (2014). Rule-based solution for context-aware reasoning on mobile devices. *Computer Science and Information Systems*, 11(1), pp. 171-193.
- [179] Miraoui, M., & Hina, M.D. (2018). A logic based context modeling and context-aware services adaptation for mobile phones. In: *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, pp. 153-158.
- [180] Pla, A., Coll, J., Mordvaniuk, N., & López, B. (2014). Context-aware case-based reasoning. In, *Mining Intelligence and Knowledge Exploration*. Springer International Publishing, pp. 229-238.
- [181] Cordier, A., Mascret, B., & Mille, A. (2010). A dynamic case-based reasoning for contextual reuse of experience. In: *Provenance-Awareness in Case-Based Reasoning Workshop*. ICCBR, Alessandria, Italy, pp. 69-78.
- [182] Zimmermann, A. (2003). Context-awareness in user modelling: Requirements analysis for a case-based reasoning application. In: *International Conference on Case-Based Reasoning*, Springer, pp. 718-732.
- [183] Zouhair, A., Amami, B., Boukachour, H., Person, P., & Bertelle, C. (2012). Intelligent tutoring systems based on the multi-agent systems (ITS-MAS): The dynamic and incremental case based reasoning (DICBR) Paradigm and the inverse longest common sub-sequence (ILCSS) in the CEHL. *International Journal of Computer Science Issues (IJCSI)*, 9(6), pp. 112-121.
- [184] Liu, F., Deng, D., & Li, P. (2017). Dynamic context-aware event recognition based on Markov logic networks. *Sensors*, 17(3), pp. 1-14.
- [185] Lee, H., Choi, JS., & Elmasri, R. (2011). A dynamic context reasoning

- based on evidential fusion networks in home-based care. *Sensor Fusion-Foundation and Applications*. pp. 1-26.
- [186] Pitoura, E., Stefanidis, K., & Vassiliadis, P. (2011). Contextual database preferences. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 34(2), pp. 19-26.
- [187] Miele, A., Quintarelli, E., & Tanca, L. (2009). A methodology for preference-based personalization of contextual data. In: *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, Saint Petersburg, Russia, pp. 287-298.
- [188] Pohl, K. (2010). Requirements engineering: Fundamentals, principles, and techniques. Springer Publishing Company, Inc. Berlin, Heidenberg.
- [189] Finkelstein, A., & Savigni, A. (2001). A framework for requirements engineering for context-aware services. In: *First International Workshop From Software Requirements to Architectures (STRAW 01) 23rd International Conference on Software Engineering*, pp. 1-6.
- [190] Maeda, J. (1994). Development and application of the SMART system. In: *Automation and Robotics in Construction XI*. Elsevier Amsterdam, pp. 457-464.
- [191] Wiesner, S., Gorltdt, C., Soeken, M., Thoben, K. D., & Drechsler, R. (2014). Requirements engineering for cyber-physical systems. In: *IFIP International Conference on Advances in Production Management Systems*, pp. 281-288.
- [192] Lewis, P. R., Chandra, A., Parsons, S., Robinson, E., Glette, K., Bahsoon, R., & Yao, X. (2011). A survey of self-awareness and its application in computing systems. In: *2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops*, pp. 102-107.
- [193] Metcalfe, J., & Shimamura, A.P. (1994). Metacognition: Knowing about knowing. MIT press.
- [194] Morin, A. (2006). Levels of consciousness and self-awareness: A comparison and integration of various neurocognitive views. *Consciousness and Cognition*, 15(2), pp. 358-371.
- [195] Agarwal, A., Miller, J., Eastep, J., Wentziuff, D., & Kasture, H. (2009). Self-aware computing. Massachusetts Inst of Tech, Cambridge.
- [196] Kounev, S., Lewis, P., Bellman, K. L., Bencomo, N., Camara, J., Diaconescu, A., & Inverardi, P. (2017). The notion of self-aware computing. In: *Self-Aware Computing Systems*: pp. 3-16.
- [197] Duval, S., & Wicklund, R. A. (1972). A theory of objective self awareness.

Oxford, England: Academic Press.

- [198] Goukens, C., Dewitte, S., & Warlop, L. (2009). Me, myself, and my choices: The influence of private self-awareness on choice. *Journal of Marketing Research*, 46(5), pp. 682-692.
- [199] de Lemos, R., Giese, H., Bencomo, N., Cukic, B., Müller, H., & Weyns, D. (2009). Software engineering for self-adaptive systems: A research road map. *Software Engineering for Self-Adaptive Systems*, pp. 1-26.
- [200] Maggio, M., Abdelzaher, T., Esterle, L., Giese, H., Kephart, J. O., Mengshoel, O. J., & Wolter, K. (2017). Self-adaptation for individual self-aware computing systems. In: *Self-Aware Computing Systems*, pp. 375-399.
- [201] Weyns, D. (2017). Software engineering of self-adaptive systems: an organised tour and future challenges. *Handbook of Software Engineering*, pp. 1-43.
- [202] Bruglieri, M., Bruschi, F., Colorni, A., Luè, A., Nocerino, R., & Rana, V. (2015). A real-time information system for public transport in case of delays and service disruptions. *Transportation Research Procedia*, 10, pp. 493-502.
- [203] Osborn, W., & Hinze, A. (2014). TIP-tree: A spatial index for traversing locations in context-aware mobile access to digital libraries. *Pervasive and Mobile Computing*, 15, pp. 26-47.
- [204] Cheverst, K., Mitchell, K., & Davies, N. (2002). The role of adaptive hypermedia in a context-aware tourist GUIDE. *Communications of the ACM*, 45(5), pp. 47-51.
- [205] Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., & Pinkerton, M. (1997). Cyberguide: A mobile context-aware tour guide. *Wireless Networks*, 3(5), pp. 421-433.
- [206] Saeedi, S., Moussa, A., & El-Sheimy, N. (2014). Context-aware personal navigation using embedded sensor fusion in smartphones. *Sensors*, 14(4), pp. 5742-5767.
- [207] Riboni, D., & Bettini, C. (2011). COSAR: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3), pp. 271-289.
- [208] Tentori, M., & Favela, J. (2008). Activity-aware computing for healthcare. *IEEE Pervasive Computing*, 7(2), pp. 51-57.
- [209] D’Aniello, G., Gaeta, M., Loia, V., & Orciuoli, F. (2015). An ami-based software architecture enabling evolutionary computation in blended



- commerce: the shopping plan application. *Mobile Information Systems*, pp. 1-19.
- [210] Sun, J., Wu, Z-h., & Pan, G. (2009). Context-aware smart car: From model to prototype. *Journal of Zhejiang University-SCIENCE A*, 10(7), pp. 1049-1059.
- [211] National Fire Protection Association. (2019/02/11).
- [212] Ronchi, E., & Nilsson, D. (2013). Fire evacuation in high-rise buildings: a review of human behaviour and modelling research. *Fire Science Reviews*, 2(1), 7.
- [213] Gerges, M., Mayouf, M., Rumley, P., & Moore, D. (2017). Human behaviour under fire situations in high-rise residential building. *International Journal of Building Pathology and Adaptation* 35(1), pp. 90-106.
- [214] Ibrahim, A.M., Venkat, I., Subramanian, K.G., Khader, A.T., & Wilde, P.D. (2016). Intelligent evacuation management systems: A review. *ACM Transactions on Intelligent Systems & Technology*, 7(3), pp. 1-27.

# Chapter 3

## Conceptualization of computational mechanisms for dynamic context information processing

### 3.1. Objectives and methodological framing of research cycle 2

#### 3.1.1. Objectives of this research cycle

In the previous research cycle, the state-of-the-art theoretical and methodological understanding of DCIP was discussed, together with the development of computational solutions. It has been revealed that: (i) DCIP is a complex and somewhat underdetermined activity that is influenced by the purpose of using DCI as well as by the environment in which it is elicited, (ii) semantic knowledge implied from dynamic context is crucial for realizing smart behaviors (e.g. self-awareness and self-adaptation), (iii) the implications of contextual changes are not sufficiently considered in reasoning, and (iv) there is no sufficiently sophisticated computational tools available to support representing and utilizing dynamic context semantics. Based on the requirements generated, this chapter aims at conceptualizing computational mechanisms that enable DCIP.

There have been four specific objectives stated for the part of the research presented in this chapter: (i) establishing the theoretical fundamentals for DCIP, including the formal definition of a semantically enriched conceptual model, (ii) specification of the functionalities of a multi-component reasoning mechanism (DCIP-M), (iii) deriving the algorithms for computational processing and arranging them in a computable architecture, and (iv) testing the feasibility of the conceptualized computational mechanism. The feasibility checking of the algorithms considered what (i) were available in the literature or commercialized, (ii) could be implemented as adaptations of existing ones, and (iii) had to be designed and implemented from scratch. The sub-chapters will present the results

in this order.

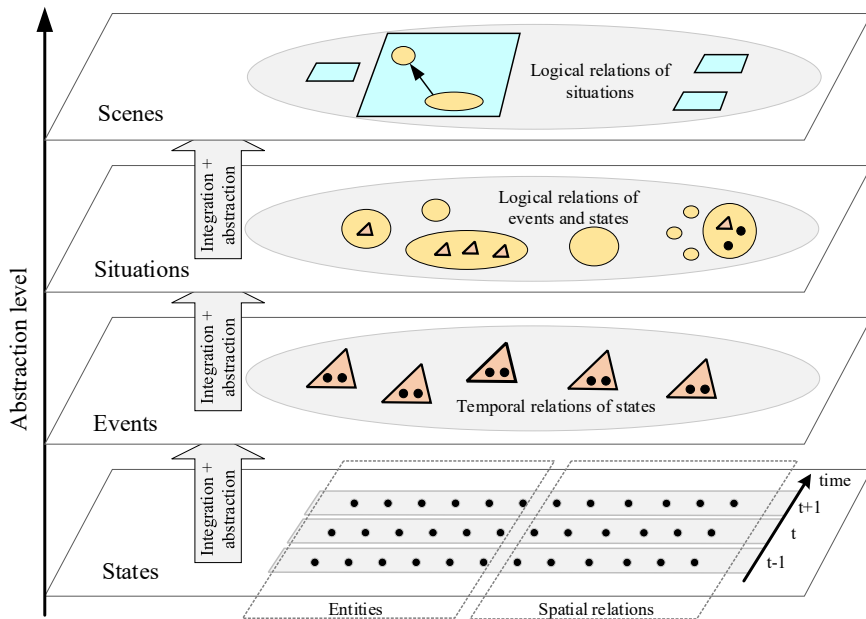
### **3.1.2. Methodology applied in this research cycle**

Research cycle 2 focused on the ideation and technical conceptualization of the DCIP-M. Therefore, this research cycle was methodologically framed as design inclusive research (DIR). The explorative part focused on the exploration of enabling knowledge and technologies. At the end of the explorative phase, the fundamental concepts related to the DCIP were specified. One input for the constructive part of this research cycle was the knowledge explored and the list of requirements synthesized in Research cycle 1. The constructive part concentrated on the establishment of a comprehensive conceptual model of the DCIP-M. Towards this end, various methods for DCIP were considered and the implications of context changes were investigated from an information engineering point of view. Based on the foundational ideas, a comprehensive concept of a computational mechanism for ‘building situation awareness’ in dynamic context was developed. In addition, the issues of inferring semantic information based on the changes in syntactic context information representation and utilization of the obtained semantic context information in action planning were investigated. In this research cycle, computational approaches to context-dependent reasoning about action plans and context-sensitive message generation were also elaborated. To validate the feasibility of the proposed components of the DCIP-M, a plan for a software prototype implementation was developed.

## **3.2. Theoretical fundamentals**

### **3.2.1. A semantically enriched model for managing context information**

To support awareness building by 2G-CPSs and to facilitate their functional, architectural and performance adaptations, the actual context of operation and servicing (i) should be captured at least in quasi-real time, (ii) should be represented with sufficient articulation, and (iii) should be schematized in a way that facilitates effective computation. From an epistemological point of view, the elements of intelligence related to context may be conveyed in four different forms, namely as (i) bulk of data/signals, (ii) pieces of information, (iii) bodies of knowledge, and (iv) principles of meta-knowledge. As it is known from the literature of information engineering, these reflect a transition from syntactic contents to semantic contents. The syntax to semantics transition (SST) makes the elements of intelligence dependent on each other and, arranges them virtually in a four-layer architecture shown in Figure 3.1. The enabler of the transition towards a higher level of semantics is a synthesis of meanings, which includes concurrent integration and abstraction with regards to each of the abovementioned elements



**Figure 3.1.** Progression of understanding context on different abstraction levels

of intelligence. This layered architecture was the starting point of our thinking about a sophisticated and effective approach to dynamic context management.

The four layers of intelligence have been regarded in our research as four different abstraction levels of capturing dynamic context. Thus, we assigned contextual meanings to these levels considering the essence of SST. As can be seen in Figure 3.1, these contextual meanings are: (i) states, (ii) events, (iii) situations, and (iv) scenes involved in the context. The states of contextual entities and their relations (CERs) are derived based on descriptive input data/signals. The events are interpreted as changes of states of CERs. The situations are interpreted as the momentary interplay of events and states. The scenes are derived as the coexistence of interrelated and/or independent situations, respectively. It has to be noted that the above interpretation of the notion of ‘scene’ is not in line with other uses of the term in the literature, e.g. as used in the papers [1]-[4].

What has been recognized as an affordance of the introduced model is the

potential to represent the implications of the interaction of CERs, states, events, and situations. More specifically, the interactions of states allow inferring emergent and/or developing CERs and the interactions of events allow inferring emergent and/or developing states and CERs. Likewise, the interactions of situations allow inferring emergent and/or developing events, states, and CERs. The potential internal (i.e. within the boundaries of a given context representation) inference gives an additional dimension to computational context management. Context processing can be supported not only with a contextual meanings-based context representation but also by inferring the implications of states, events, situations, and scenes. By doing so, the described model provides a low-fidelity replica of the human awareness building process, which not only provides a mental representation of context but also facilitates immediate inferring based on this representation. We consider it as a distinguishing scientific novelty of our proposal and approach. A computational advantage is that the introduced contextual meanings-based context management makes it possible to represent each of the four above-mentioned types of contexts, namely: (i) static context, (ii) altering context, (iii) dynamic context, and (iv) emergent context.

In this first phase of our research, we have considered only spatial, attributive and temporal (SAT) data/signals in the representation of context. This is however just a technical issue and constraint, rather than a theoretical or methodological limitation. In the context of the reported research, it means that only SAT data/signals have been used for the representation of CERs. In the presence of this constraint, the state of a CER is described by a set of spatial and attributive data that exist in a real-life process at a given point in time. This can express a spatial relation between two entities (or their manifestations in space). States can be captured based on the spatial and attributive data of an entity at a given time. For example, typical states are: (i) the color of an entity, (ii) the location of an entity at a given point in time, and (iii) the distance between two entities at a given point in time. Every state has a discrete meaning that facilitates understanding of that particular state in association with a process. An event represents changes of states in time captured in a local system of reference of finite extent. Computationally, an event is derived by computing the differences of two subsequent states of the same set of CERs at different points in time (i.e. as a temporal relation). Thus, the concept of the event allows deriving some semantic meaning from the computed changes. Typical examples of events are: (i) change of the location of an entity (i.e. an entity is moving in the space), (ii) change of a spatial relation (distance) between two entities (i.e. the entities are approaching each other), (iii) disappearance of the distance between two entities (i.e. the two entities have collided).

A situation is generated by the aggregation of a series of events and/or states appearing at a given point in time. Computationally, it is realized by: (i)

combining/integrating events and states that are logically related, and (ii) abstracting the meaning of combined and/or integrated events and states towards a higher-level understanding. For instance, a typical situation is a traffic jam at a given point in time, which can computationally be generated by the aggregation of (i) the locations of cars (states), (ii) the distances among cars (states), (iii) the changes of locations of cars (events), and (iv) the changes of distances among cars (events). As briefly noted above, a scene describes what interrelated and/or independent situations coexist and what kind of interactions are among them, respectively. By combining interrelated situations information about the circumstances can be obtained. Changing the relationships of situations allows generating various possible (feasible) scenes and inferring about the implications of the situational changes. This needs abstraction but, in turn, it provides a higher level of semantic interpretation of the situational changes. A simple combination of situations into a scene involves the consideration of the logical and semantic relationships of at least two situations. A compound combination of situations raises a combinatorial issue, in which the temporal precedencies or logical importance of the situations, or both, should be taken into consideration in order to arrive at some principle of ordering.

If done this way, then the scenes can capture a multitude of logical and semantic relations of multiple situations, such as causality, similarity, transitivity, terminate influence, etc. Consequently, scenes are regarded as a low-fidelity computational replica of awareness since they carry intelligence about different situations and offer the possibility of inferring about the implications of different interrelationships among different situations. The other side of the coin is that this computationally replicated (mimicked) awareness makes it possible not only to know what is in a contextual scene (what is the context) but also to infer about what can be expected when this contextual scene prevails. This latter plays a crucial role in reasoning with the dynamic context in the process of making a decision and solving a practical problem by a 2G-CPS. The interrelations of situations lend themselves to deriving semantically-rich procedural knowledge about the time-varying process (i.e. interpretation and understanding of the causalities and correspondences of procedural happenings).

One issue needs to be revisited here. Namely, the interpretation of the implications of the contextual meanings introduced on the different abstraction levels. First of all, static context can be inferred from states on the first layer of the reasoning model (layer of states), since states define context entities and spatial relations among them in a discretized manner. On the second layer (events), however, both static contexts and altering contexts can be interpreted based on the consideration of the nature of the events. The change between two states designates altering context, while the change itself can be understood as a state inferred from one or more events. For instance, the speed of a car

(computationally represented as an entity ID) is inferred from the change of its locations. On the other hand, the speed of this entity at a given point in time means static context.

On the third layer (situations), static, altering and dynamic context can be obtained equally well. For instance, the size of a traffic jam caused by cars at a given point in time is a typical example of a static context. Nevertheless, any change related to the traffic jam (e.g. in its size) between two points in time represents an altering context. Furthermore, changes in the state of the traffic jam (e.g. its density) over a period of time represent a typical dynamic context. On the fourth layer (scenes), emergent context can be derived from the interrelated situations, e.g. uncertainty, probability, complexity. For instance, uncertainty can be inferred based on the fact that specific relations of two situations cannot be completely foreseen, e.g. how much a traffic jam slows down the motion of a car in the jam.

### **3.2.2. Inferring semantic knowledge from dynamic context**

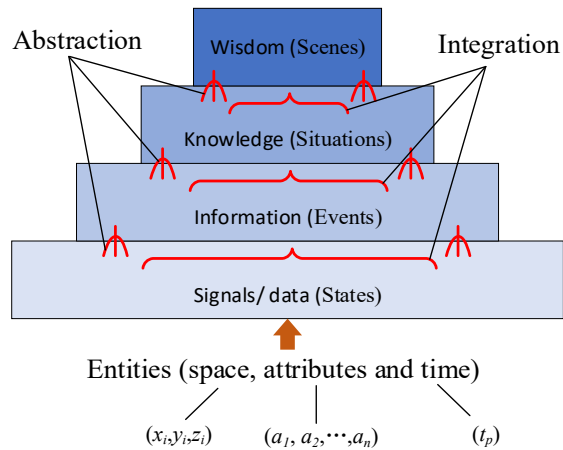
Computational inferring semantic knowledge is a wicked problem of knowledge engineering for two reasons. On the one hand, it needs a conversion of a low-level representation of information into a higher level representation, which typically suffers from information deficit. On the other hand, understanding semantics (meaning) assumes having consciousness, purpose, and surrounding, which are concepts strongly related to human beings. The issues and challenges of computational tangibility and situation dependence of semantic conversion have been addressed in research for a rather long time. Therefore, there is a wide range of approaches as well as applications studied. For instance, a standard approach is to infer knowledge based on large semantic networks [5]. In combination with information filtering, the contents of the nodes and the structure of the network are used in the inferring process.

The concept of semantic roles (encode semantic links between a verb and its arguments) and annotated spatial knowledge statements were used by researchers to infer about whether something is located or not located somewhere [6]. A system was proposed to generate semantic graphs as a representation of the meaning of a text based on textual inference (assuming that the meaning of one text can be inferred from the meaning of another and from background knowledge) [7]. In the mentioned work, three basic principles were exploited: (i) aggregation of data (syntactic elements), (ii) integration of syntactic elements, and (iii) abstraction towards semantics (meaning). Actually, these three principles are utilized in the generation of the so-called knowledge pyramid that represents the process of converting data, through information and knowledge, into decisions [8]. Ultimately, this interpretation allows getting to intelligence, which is seen as specific actionable knowledge needed to make a specific decision in a specific

context [9]. This concept was reused in our research to make the inference of the semantic implication of syntactic DCI structures possible. There are substantive differences with regards to the interpretation and influences of the knowledge pyramid [10]. We used it only as an analogy of representation of 'what-is' type of epistemic elements, without considering 'how-to' type elements. The constructed dynamic context knowledge pyramid is shown in Figure 3.2.

Based on the explanations given in Section 3.2.1 above concerning the theoretical considerations underpinning the proposed model, it can be concluded that the progression of context building from a syntactical level to a semantic level is analogous to the progression on the so-called 'knowledge ladder' or 'knowledge pyramid' [8]. These models also identify data, information, knowledge, and wisdom as intelligence elements of growing semantic meaning and problem-solving enablers. The benefit of using this conceptualization in the case of the proposed dynamic context management mechanism is that: (i) the different abstraction levels can be handled with dedicated computational algorithms, and that: (ii) opportunities are provided for various forms of computational inferring.

The mechanism for inferring semantic knowledge from dynamic context is presented in Figure 3.3. We will explain the working of the mechanism based on an exemplified dynamic process, in which the movement of three entities (i.e.  $e_1, e_2, e_3$ .) at three given points in time (i.e.  $t, t'$  and  $t''$ ) are considered. The processed data concerning the entities and their relations describe the momentary 'states' of the related process (e.g. the attributes or the location of an entity at a given point in time). The actual variations of the physical process are inferred from the temporal relations of the momentary states. For the purpose of computation, the variations of the states of entities and relations are captured as 'events'. An event (EV) may mean changes, among others, in the location, attributes, and relations of entities during a given time increment. Multiple interacting states and events form a 'situation' (SI), which is one higher level of computational



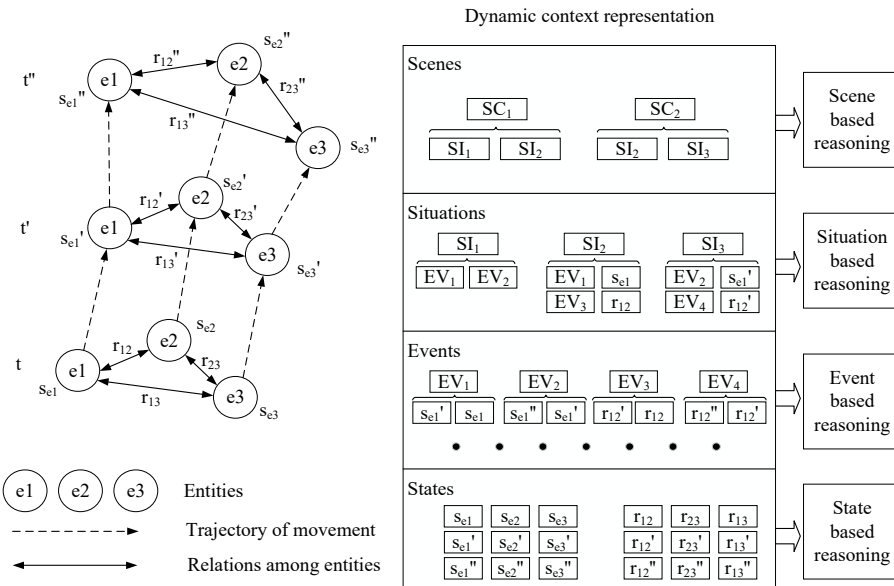
**Figure 3.2.** The dynamic context knowledge pyramid



abstraction. The concept of the situation was introduced to be able to describe a phenomenon, which happens in a duration of time (possibly, over multiple computational time increments). A situation can be inferred by integrating and abstracting information about multiple states and/or events according to certain predefined rules. For instance, a people jam can be defined based on the states and/or events related to several people who are present at a location. An arrangement of situations has been captured by the concept of ‘scene’ (SC). This is the theoretical basis of reaching from syntactic data (information) to semantic constructs (knowledge) in the proposed mechanism.

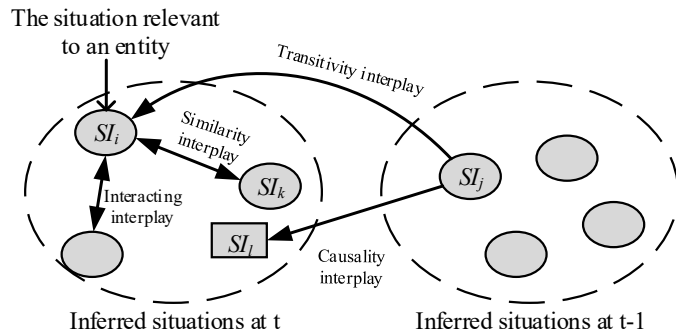
Interplays of situations may be one of the multiple types. There are one-way interplays, i.e. causality and transitivity, and dual-way interplays, i.e. similarity and interacting. By investigating the various interplays of situations we can determine which relations the concerned situations are. This is explained in Figure 3.4, in which  $SI_i$  is the situation relevant to a concerned entity (i.e. in the LW of the entity). The (situational) interplays related to  $SI_i$  are noted. For instance, there might be an interplay between  $SI_i$  and  $SI_k$ , which are two situations happening at the same point in physical time, e.g., at  $t$ . The interplay can be noted as ‘ $SI_k$  is similar to  $SI_i$ ’. It means that if  $SI_i$  is detected, then another situation,  $SI_k$  whose attributes are similar to  $SI_i$  is about to take place somewhere.

In the first step of the inference mechanism, the transitivity interplay between two situations with the same type, but happening at different time points was



**Figure 3.3.** The mechanism for inferring semantic knowledge from dynamic context

considered, as shown in Figure 3.4. If a situation (e.g.,  $SI_i$ ) is transferred from another situation (e.g.,  $SI_j$ ), the difference between the considered two situations indicates the change of  $SI_i$ .



**Figure 3.4.** The interplays of situations

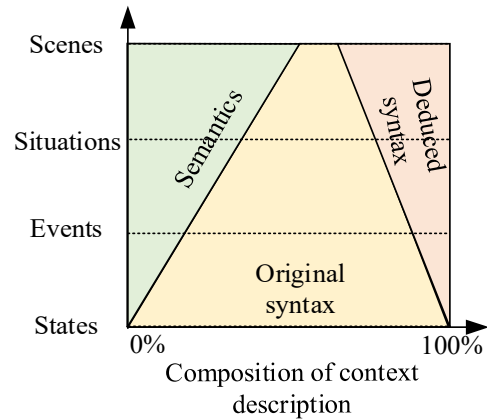
Particularly, the considered two situations in transitivity interplay should be the same type. The change between two different types of situations is meaningless since the attributes of them might be different. The type of interplay, which is created between two types of situations, is called ‘causality interplay’ ( $SI_j$  and  $SI_l$ ). For instance, a traffic jam might be caused by the malfunction of a car. Although the difference between the traffic jams and the malfunction of a car is meaningless, the knowledge obtained from the procedural sequence is important for predicting future context.

There might be multiple situations with the same type happenings at a point in time, e.g. multiple traffic jams at  $t$ . The change of a particular situation (e.g. a traffic jam at  $t$ ) should be obtained by considering two situations (the traffic jam at  $t$  and another traffic jam at  $t - 1$ ) that are related to each other. It means that the concerned two situations should contain some common elements (e.g. they both contain several particular cars). In turn, the common elements can be used as a measurement to figure out the related two situations at different points in time. In our work, this was considered as the basis for inferring transitivity interplay between two situations.

It can be seen from the proposed inferring mechanism that dynamic context representation is a compositional process that assumes four abstraction (semantics) levels, which imply four computational layers. In the respective layers, computational integration of data, information, and knowledge takes place, which is complemented by logical/semantic abstractions. Eventually, this dualism makes it possible to provide a sophisticated and comprehensive representation, and to infer about the implications of the identified contextual means. On the layer of states, context is only syntactically described and processed. On the other layers, however, the context representation is completed with various derived elements of semantical intelligence, which are generated by the inferring algorithms of the proposed dynamic context management mechanism.

Regulated by the underlying pyramid, the original syntactic context representation is gradually appended by semantic intelligence elements derived on the subsequent layers, as shown in Figure 3.5. The contents of these derived intelligence elements are determined by the integrations and abstractions that are done computationally on a given layer and on the underlying layers. Thus, as the level of abstraction increases over the layers, both the amount of the syntactic intelligence elements and the amount of the deducible semantic intelligence elements increase. The inferred semantics enables capturing various contextual meanings and their interrelationships through either some natural or artificial language expressions, or mathematical constructs.

Eventually, the proposed conceptual framework enables: (i) inferring descriptive knowledge from time-varying artifacts and processes, and (ii) deriving predictive (prescriptive) knowledge about future circumstances of the concerned artifacts and processes. Contextual inferring is done on each abstraction level based on the respective representations by the related inferring algorithms. Future states can be predicted based on the states and the events (the changes of states), and knowing situations allows inferring events, states, and descriptive data and relationships. From the perspective of the development work, an advantage is that resembling (or even similar) computational operations can be implemented and used to predict events on the situation layer and to predict situations on the scene layer. On the other hand, the description of dynamic context on the different abstraction levels requires a coherent and consistent overall representation of multiple different contents and compositions.



**Figure 3.5.** The composition of context description on different abstraction levels

### 3.2.3. Interpretation of dynamic context with regard to one entity

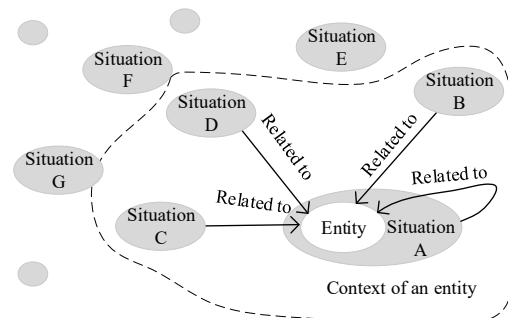
Our investigations explored that high-potential dynamic context management cannot be implemented without considering situations and their interactions. Therefore, we took situations as a starting point of interpretation of the dynamic context and as a basic representational element of a dynamic context management mechanism. In our conceptualization, the highest-level manifestation of context is formed by a set of interrelated situations pertinent to an entity, as shown in Figure

3.6. Though an infinite number of distinct situations can in principle be identified in real life, only a finite number of them can be taken into consideration in the representation of the varying circumstances of any entity. Accordingly, investigation of dynamic context assumes that: (i) the set of entities, (ii) the type of entities, (iii) the set of relevant relations, and (iv) the types of relations are specified. The type of relations can be both direct relations and indirect (chained multiple direct) relations.

Furthermore, explicit and implicit context elements are distinguished. For instance, space, time and attributes of entities, and the relations among them are explicit elements of context representation, while human intention and operation objective are typically implicit elements of context representation. Until now, we have dealt only with explicit elements of context representation in our research. The reason is the inherent interpretational uncertainty associated with implicit contexts and the complex computation that needs the application of artificial intelligence techniques. For example, a traffic jam that a car is involved in can be represented by explicit context elements. However, the intention of the drivers, who are approaching and getting involved in the jam, to slow down or not, is an implicit element of the context, which is difficult to process computationally.

The dynamism of context can be interpreted from two aspects. On the one hand, the relevance of any situation to a concerned entity may change as time elapses. For instance, if a traffic jam is far away from a moving car, then the relevance of the jam to the car is relatively low. In addition to the spatial-temporal relations between the situation and the entity, the objective and activity of the entity are also important elements of the dynamic context. For instance, if the driver of the above car together with many others decides to leave the highway and continues driving on the main road, then this may influence the formation of a jam over time. Furthermore, the change of situations may influence (increase and reduce) the relevance of some other situations. For instance, malfunctioning of a car involved in the jam lends itself to a situation, which influences the situation in which the car is already involved. This new situation may be more relevant to the concerned car, than to the traffic jam.

The above explanation casts light on the fact that the relevance of a situation to a concerned entity is a crucial issue with regards to the representation of and inferring from the entity's individual

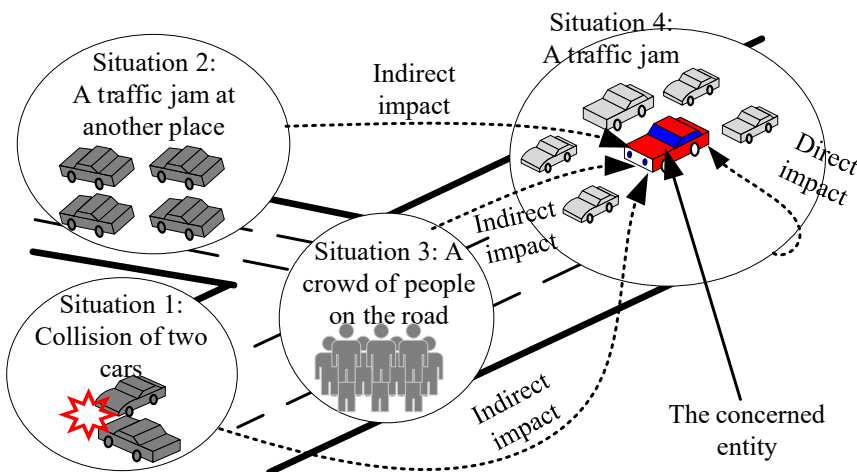


**Figure 3.6.** Interpretation of the dynamic context of an entity

context. This is however much less crucial in the case of static or altering contexts since, in these cases, the influence of situations does not change over time, or just changes periodically. The relevance of states or events can often be specified in advance in specific applications. However, the relevance of a situation, as a contextual element, is contingent. Typically, it should be determined and specified in the process of dynamic context computation.

In order to support the evaluation of the relevance of a situation, we have considered the impacts of changing situations on a particular entity. By doing so, we took into account that the existence of a particular situation influences the decision-making process in terms of that entity. The possible impacts of changing situations have been classified as (i) direct impact, and (ii) indirect impact. These types of impacts are illustrated in the case shown in Figure 3.7. In this case, four situations are supposed to influence the path planning of the concerned entity. As a direct impact, we considered that the concerned entity is a part of the situation at the given place and time (Situation 4 in Figure 3.7). For instance, the traffic jam in Situation 4 has a direct impact on the concerned car, which is in the jam (forces the car to slow down).

As an indirect impact, the other three situations were simultaneously considered (Situations 1, 2 and 3 in Figure 3.7). These situations actually reflect the circumstances created by other entities (e.g. humans, cars and objects). It is not for the disfavor of generality to assume that these situations normally happen at any other place and/or at any other point in time. For example, the collision of the two cars shown in Figure 3.7 could happen five minutes ago (Situation 1), and a crowd of people may come together in front of the traffic jam (Situation 2), and a traffic jam may be formed at another place in the near future (Situation 3). These individually and together have an indirect impact on the concerned entity. (Note



**Figure 3.7.** The impact of situations on a concerned entity

that implicit context elements are intentionally not mentioned in Figure 3.7 related to the dynamic context management).

The situations with different types of impacts should be quantitatively evaluated in order to be able to judge which situation is relevant and which one is not. For instance, the impact of Situation 1 on the ‘concerned entity’ shown in Figure 3.7 may be small because there is a chance that Situation 1 will disappear by the time when the entity arrives at the location where the collision took place. Quantitative evaluation of impacts of situations should facilitate finding out which situation is more relevant than the others. In this way, context management solutions can be generated by focusing on the most relevant situation(s) of individual entities. This ideation is the basis for generating personalized messages in case of I-CPSs.

### **3.3. A model for representing spatial, attributive and temporal context data**

According to the context knowledge pyramid, dynamic context management requires a sufficiently articulated representation of the context and a logical/semantic interpretation of the meanings and implications of the context changes. The representation (description) and the logical/semantic interpretation (inference) together form a basis for context reasoning and decision-making. To facilitate dynamic context representation, this sub-chapter introduces a novel data construct, namely context information reference (CIR-) cube. It integrates (represents and arranges) spatial and attributive data, as well as temporal (timing) data, which describe the states regarding entity attributes and relations, and form a basis for inferring semantic knowledge about events, situations, and scenes. The basic element of a CIR-cube is a spatial feature representation (SFR-) matrix. A particular arrangement of multiple SFR-matrices forms a CIR-cube. In order to explain these fundamental concepts and enabler, the technical details about SFR-matrix and CIR-cube are given below in this subchapter.

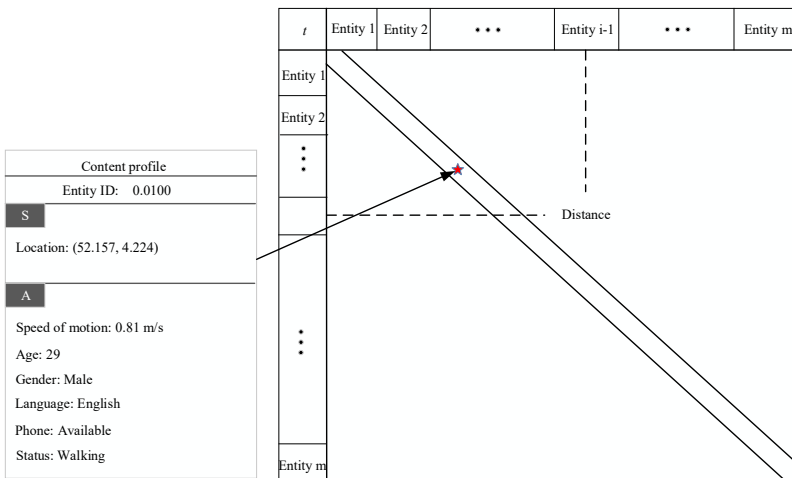
#### **3.3.1. The spatial feature representation matrix**

According to the context knowledge pyramid, the change of spatial relationships between any two entities, as an example of a relationship, should be considered. This calls for a structured representation of spatial and attributive information of entities and spatial relationships. As the conceptual framework of modeling dynamically changing contexts and data structuring, we considered a ‘attributes  $\leftarrow$  entities  $\leftrightarrow$  relationships  $\rightarrow$  attributes’ pattern. Based on this pattern, we conceptualized a two-dimensional spatial feature representation matrix (SFR-matrix) for a structured representation of spatial entities, attributes, and relations. The SFR-matrix is set out by rows and columns that arrange the entities in a structured manner.

The principal arrangement of an SFR-matrix is shown in Figure 3.8. It is a rectangular array of size  $(m+1)*(m+1)$ , where  $m$  is the total number of entities captured and  $i$  and  $j$  are the running indices. The organization of the contents of the matrix is as follows. The first row, as well as the first column, contains the identifiers of the various entities included. The elements of the first column are generated by a transposition of the first row. The cells  $(i \geq 2)$  in the main diagonal  $(i = j)$  store all entity-reflective information, while the cells outside the main diagonal  $(i \neq j)$  store the connectivity information.

The cells in the main diagonal capture the so-called ‘content profile’ of the entity. For a person, this includes sensed attributes (e.g. location, type) and calculated attributes (e.g. speed of motion, status) and profiled attributes (e.g. identification number, age). For an exit, it might include sensed attributes (e.g. if the door is locked or not), calculated attributes (e.g. when the exit will be occupied by fire) and profiled attributes (e.g. identification number and the throughput capability (the number of people that can pass through per minute)) as well. Actually, these ‘content profiles’ include multiple one-dimensional arrays that carry the data about the spatial and attributive features of the entities.

As shown in Figure 3.8, the upper and lower triangular sub-matrices represent the distances among the concerned entities. The upper triangular matrix includes the length of the route from entity $_{i-1}$  to entity $_{j-1}$  (to be done by entity $_{i-1}$ ), while the lower triangular matrix indicates the length of the route to be done by entity $_{j-1}$ . In some applications, these two values might be different. For example, the length of the route that is followed by a passenger to a taxi might be different to the length of the route taken by the taxi to reach the passenger since the passenger can walk through a shortcut path.

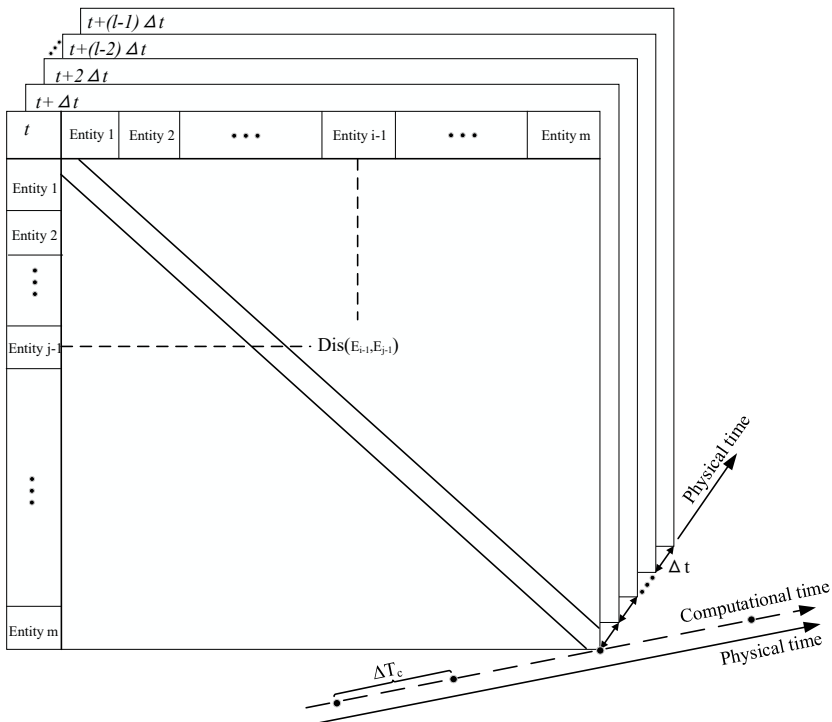


**Figure 3.8.** The spatial feature representation matrix (SFR-Matrix)

In the SFR-matrix, an invariable and a variable part can be differentiated. The invariable part is filled in as default of initial values, which describe the unchanged (static) context. Some preliminary computations can be done before the actual dynamic context computation starts, such as the distance between two exits in a building. In terms of the variable part, the momentarily distances between the reference (central) points of any pairs of entities can be computed based on their location data. The distances are expressed as scalar values in the relevant cells of the triangular sub-matrix.

### 3.3.2. The context information reference cube

The SFR-matrix is used to represent the spatial and attributive information of entities and relationships among them at a given time point. In order to handle temporal context information, multiple SFR-matrices related to a series of time points were combined to form a cube-like data structure (Figure 3.9). Computational processing of the values contained by the SFR-matrix involves updating and re-computing the contents of all of the concerned cells subsequently at given points in time. The sequence of the multiple re-computable SFR-matrices is computational equivalent in the CIR-cube with regards to spatial and attributive context information. The time increment between two successive SFR-matrices in



**Figure 3.9.** The context information reference cube (CIR-cube)



a CIR-cube is  $\Delta t$ . Accordingly, a CIR-cube represents SAT context information within  $\Delta t_{pt} = (l - 1)\Delta t$ , where:  $l$  is an integral variable, which indicates the number of SFR-matrices in a CIR-cube.

Computational processing of the values contained in the CIR-cube involves updating and re-computing the contents of the concerned cells at every subsequent period of time. The sequence of these computational loops forms a (virtual) computational time axis, which associates the processing actions completed according to the CIR-cube with physical time. The time increment between the computations of two successive computational loops is  $\Delta T_c$ , which means that the content of a CIR-cube is updated at each and every  $\Delta T_c$ . For this reason, the CIR-cube intertwines the (continuous) physical time dimension and the computational time dimension. Distinguishing these two temporal dimensions is important for proper computation of dynamic context. The reason is that the set of information, which is computed at a computational time  $T_1$  to describe a situation at a point in physical time, may differ from the set of information computed at another computational time, e.g.  $T_1 + \Delta T_c$  to describe the same.

Therefore, the size of a CIR-cube depends on the captured information as well as on the needed re-computations. First, the ‘width’ of the CIR-cube depends on how many entities are included in the SFR-matrices within a period of time. For instance, if an entity is not anymore present in a given scenario (e.g. a person successfully escaped, an exit became unusable, a fire has been extinguished, etc.), then it is labeled as inactive in the corresponding cell of the main diagonal, and excluded from any further computation. Contrarily, if an entity becomes involved or re-activated in a given situation, its label is changed or set accordingly in the main diagonal. However, the widths of the CIR-cubes generated at different points in computational time may be different, since the number of concerned entities may change. Second, the ‘length’ of the CIR-cube is also changeable. This depends on the total number of SFR-matrixes needed to describe the whole sequence of the time-varying process as well as on the allowed sampling time increments between every two successive SFR-matrixes for sufficient representation of a time-varying process. Thus, CIR-cube is a dynamic model by nature.

The proposed dynamic structure provides many benefits for modeling and efficient computation of SAT contextual variations. It enables modeling the dynamics of situations caused by joining or leaving by entities or, alternatively, the attributes of entities (e.g. type) may also change over a period of time. In addition, the made-to-order dimension (length) of the CIR-cube in temporal domain enables modeling of situational events with different durations of time. Furthermore, the possibility of using different sampling time increments, and different lengths of CIR-cubes, supports an ‘event-orientated’ discretization of computational time and quasi-real time processing of DCI.

The contents of all cells in a CIR-cube are concurrently used in the dynamic context computation process. All relative distances can be easily computed and stored this way. Actually, it is used by a computational scheme for logical/semantic inferring the implications of dynamic contexts. To represent a cell in a CIR-cube and to support the manipulation of the contents of the CIR-cube, the following symbolic variable has been defined:  $CIRC(i, j, t, T)$ . This variable refers to the cell that is located in the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column of the CIR-cube built at the computational time  $T$  and represents the SAT data at the physical time  $t$ , and  $i, j > 2$ . Some typical examples are listed below and are interpreted as follows:

- $CIRC(i, i, t, T)$  refers to the content profile of  $Entity_{i-1}$  at a given point in physical time  $t$ , which is calculated at the computational time  $T$ .
- $CIRC(i, j, t, T)$  refers to the distance between two entities,  $Entity_{i-1}$  and  $Entity_{j-1}$  at a given point in physical time  $t$ , which is calculated at the computational time  $T$ .
- $CIRC(i, j, \Delta t, T)$  refers to the change of the distance between two entities between two points in physical time, which is calculated at the computational time  $T$ .
- $CIRC(i, j, t, \Delta T)$  refers to the difference between two computed distances at two computational time points (e.g.  $T_1$  and  $T_1 + \Delta T$ ).

The elementary cells of the CIR-cube capture and represent the explicit states and events regarding the entities and their spatial relations. As discussed in the part dealing with the theoretical fundamentals, a situation is a composition of states and/or events. Situations happening in a time-varying scenario can be inferred based on checking if the states and/or events that define the situation do happen or not. Accordingly, a situation can be represented by a combination of the symbolic variables with predefined conditions. In this way, the inference of situations can be realized. A typical example of rule-based reasoning can be: IF ‘a man is running at a point in time  $t'$  (a state) AND ‘the man is an elder’ (a state), THEN ‘the speed of the man is below 1 m/s at  $t'$  (a situation). In specific applications, the rules defining the situations should be specified by application designers.

### **3.4. Specification of the functionalities of the computational mechanisms**

The main function of the computational mechanism is to enable dynamic context management in the case of I-CPS applications based on information modeling, inferring, reasoning and messaging operations. The target functionalities, which are derived by the decomposition of the main function and should be fulfilled by

the prototype system, are shown in Figure 3.10. In the process of decomposition, the requirements identified in Chapter 2 have been considered. A detailed explanation with regard to the functionalities is presented below.

### **3.4.1. Functionalities for dynamic context information representation**

As discussed in Sub-chapter 3.3, the CIR-cube integrates data regarding (i) the attributes of concerned entities, (ii) the spatial relations among the entities, and (iii) the changes about the attributes and the spatial relations. One of the main functionalities is to construct a CIR-cube for the purpose of DCI representation. Since the DCIP-mechanisms may serve as a computational platform, the descriptive context data aggregated by sensors belonging to I-CPSs should be firstly acquired by the platform. The acquisition may involve multiple types of pre-processing operations, e.g. data filtering, data fusion, etc.

When the SAT data with regard to the attributes and relations of a set of concerned entities are ready, a SFR-matrix should be constructed following several computational steps: (i) creating a SFR-matrix by filling the first column and first row of the SFR-matrix with the identifications of the entities, (ii) calculating required attributes of the concerned entities to enable the creation of content profiles, and (iii) calculating the distances among the entities based on their attributes and the layout of a given physical space.

For a given point in physical time, an SFR-matrix is generated. When a time-varying process is concerned, multiple SFR-matrices might be generated and arranged according to the physical time they represent for. In this way, a CIR-cube is formed in one computational session. It's worth mentioning that the configuration parameters of a CIR-cube, e.g. the number of SFR-matrices, and the increment interval between two successive SFR-matrices should be specified when constructing a CIR-cube. In our work, we assume that these parameters will be given by the served I-CPS to facilitate computation. For this reason, functionalities related to the determination of the configuration parameters were not included.

### **3.4.2. Functionalities for inferring semantic knowledge from dynamic context**

As discussed by the context knowledge pyramid, the represented states and events should be used to generate semantic knowledge from dynamic context by inferring situations and calculating their implications. Towards this end, four interrelated sub-level functionalities are included: (i) interpret current dynamic context, (ii) predict near-future dynamic context, (iii) analyze predicted context, and (iv) analyze the individual context of entities.

### 3.4.2.1. Interpret current dynamic context

A specification of the inference mechanism by using the CIR-cube is shown in Figure 3.11. It can be seen that situations are inferred based on the actual contents of the CIR-cube and several predefined rules. Situations happening in a

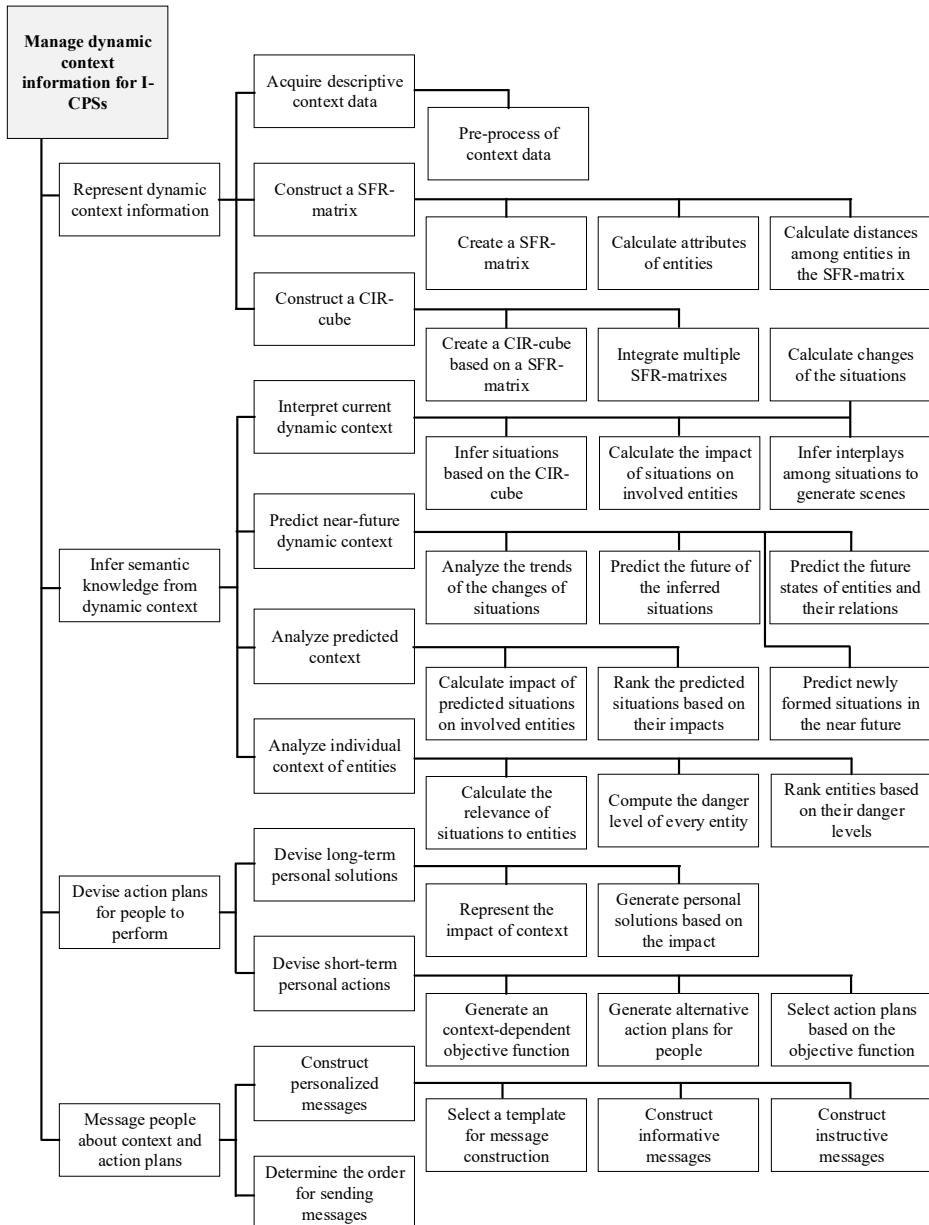


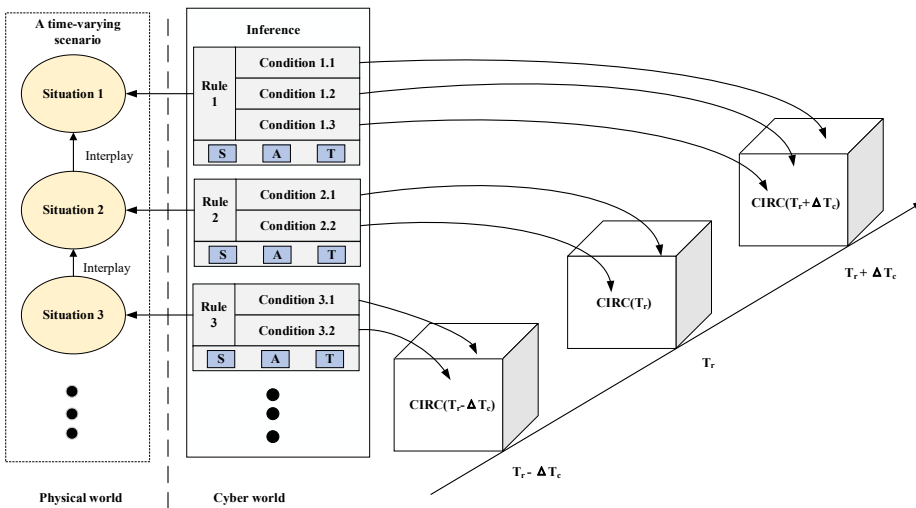
Figure 3.10. Conceptual design of the functionalities of the DCIP-mechanism

time-varying scenario are inferred by checking if the states and/or events that define the situation do happen or not. Therefore, the inference of a situation can be realized using a set of reusable rules, which combines symbolic variables (introduced in Section 3.3.2) with predefined conditions. Typical examples of reusable rules are given in Table 3-1. In a rule, it can be seen that conditions that define a concerned situation should be clearly specified. Every condition had a Boolean value and logic operators (i.e. AND, OR) that were used to combine individual conditions. Actually, the rules can be included in a library and reused in a purposeful combination in various applications in which similar situations are to be handled. For instance, the situation described in the second row of Table 3.1 can be referred to either as the status of motion of two people in a fire evacuation scenario or as the status of motion of two cars in a traffic jam scenario.

The impact of the situations on involved entities is to be calculated based on the actual influence of the situations on the attributes of the entities. For instance, the impact of a traffic jam can be determined as (i) reducing the speed of motion of the involved cars and (ii) increasing the fuel consumption. The impact of situations is normally heterogeneous and qualitatively represented. To support the calculation and enable comparison of the impact of different situations, a new concept, the impact indicator (II) was applied to quantitatively evaluate the impact of a situation, which is defined as follows:

$$II = f(A_{si}, \Delta A_{en}) \tag{3-1}$$

where:  $A_{si}$  refers to the attributes of the situation and  $\Delta A_{en}$  refers to the change



**Figure 3.11.** Inferring situations and scenes from the CIR-cubes generated in multiple computational sessions

of attributes of the involved entity. The indicator  $II$  expresses the extent of the impact a situation makes on the involved person (in the value range of  $-1 \leq II \leq 1$ ). If the value  $II = -1$ , then it means that the concerned situation has an extremely negative impact. Likewise,  $II = 1$  refers to an extremely positive impact, and  $II = 0$  means that the concerned situation has no impact on the involved people at all.

The value of  $II$  of a situation can be either predefined or calculated based on the attributes of the situation and the people. For instance, in the case of an indoor fire, the  $II$  of a people jam with 10 people can be predefined as -0.1, the  $II$  of a people jam with 20 people can be predefined as -0.2, while that for the fire can be predefined as -1. It is worth mentioning that the impact of a situation is a relative concept and that the  $II$  of a situation is a relative value. They should be determined by the application designers according to the actual influence of the situation on the fulfillment of the system objective. It means that the qualitatively evaluated impact should be projected to the value range according to the specific requirements of the concerned application. In the above-mentioned example case, the people jam has a negative  $II$ . (It can reduce the speed of motion of the involved people and, by doing so, it goes against the objective of the system – i.e. evacuating all people in shortest possible time). The  $II$  value of the fire is set to -1 since people may lose their life when involved in fire.

Another important functionality is to infer interplays among situations to form scenes. As discussed in Section 3.2.2 the different types of interplays can support dynamic context computations with regard to different purposes. The transitivity interplay enables calculation of situational changes by comparing the difference between any two concerned situations. The causality interplay can be applied for inferring knowledge about procedural consequences of a varying process. In addition, both similarity and inactivity interplays can be used to infer context when a deficiency of related descriptive data happens (e.g. due to sudden damage

**Table 3.1.** Examples of rules used for inferring situations

Rules used for inferring situations	Situations
IF ‘(abs (CIRC( $i, j, t, T$ )) < $d_{th}$ )’, THEN ‘Close_to{ $e_{i-1}, e_{j-1}, t, T, Loc$ }’	$e_{i-1}$ is close to $e_{j-1}$ at t and at $Loc$ .
IF ‘(CIRC( $i, j, \Delta t, T$ ) < 0) &((CIRC( $i, i, t, T$ ).speed > $s_{th}$ ))’, THEN ‘Move_towards{ $e_{i-1}, e_{j-1}, t, T, Loc$ }’	$e_{i-1}$ moves towards $e_{j-1}$ at t and at $Loc$ .
IF ‘(abs (CIRC( $i, j, t, T$ )) < $d_{th}$ )&(CIRC( $i, j, \Delta t, T$ ) < 0) &((CIRC( $i, i, \Delta t, T$ ).speed < 0), THEN ‘Blockage{ $e_{i-1}, e_{j-1}, t, T, Loc$ }’	$e_{i-1}$ slows down due to the blockage of $e_{j-1}$ at t and at $Loc$ .

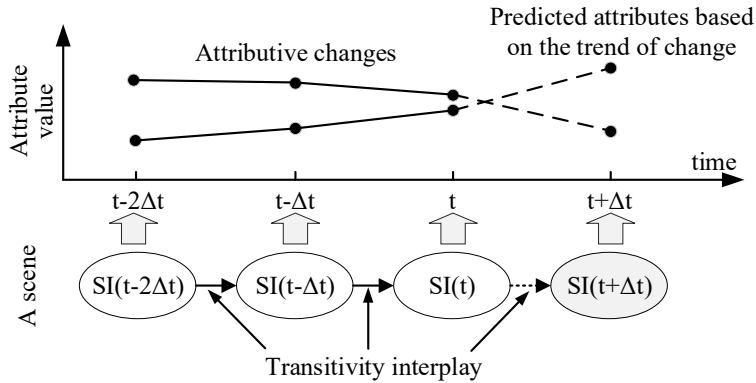
of related sensors). Although all interplays can be considered as a basis for conducting reasoning and decision-making operations, the transitivity and causality interplays are discussed in the following part of this paper, whereas the similarity and interactivity interplays are not considered. Because we assume that all the required data can be fully aggregated and they are reliable.

The changes implied by the inferred interplays of situations (i.e. the transitivity interplay) are calculated to form a basis of semantic interpretation. For a transitivity interplay, the change is defined as the spatial and attributive differences between the two considered situations. For example, the change in the number of vehicles trapped in a traffic jam can be an attribute describing the change of the traffic jam. When the changes of a series of situations over a given period of time are calculated, knowledge about the process is obtained, which indicate the trend of the changes. Actually, there might be multiple aspects with regard to a change of a situation. Each aspect of the change can be interpreted as knowledge describing the situation and the context. The obtained changes can be represented by either an implicit model (e.g. neural network) or an explicit model (e.g. relational expression model), which are used for semantic interpretation.

#### **3.4.2.2. Predict near-future dynamic context**

Due to the reason that the dynamic context of an entity may change rapidly, its impact on the entity changes as well. Consequently, the entity may behave differently, which further results in a change with regard to the relation between the dynamic context and the concerned entity. Therefore, we argue that it is hardly possible to reliably predict a far-future of dynamic context. For this reason, the prediction of the near-future of dynamic context is considered in the proposed DCIP-mechanisms. To achieve this, the mechanism should (i) predict the future of existing situations, and (ii) check if any new situations will be formed in the near future. The former objective should be realized by the processing of the existing situations, while the latter objective should be achieved based on integration and abstraction operations with regard to the future states of concerned entities and relations. Accordingly, four sub-functionalities have been considered.

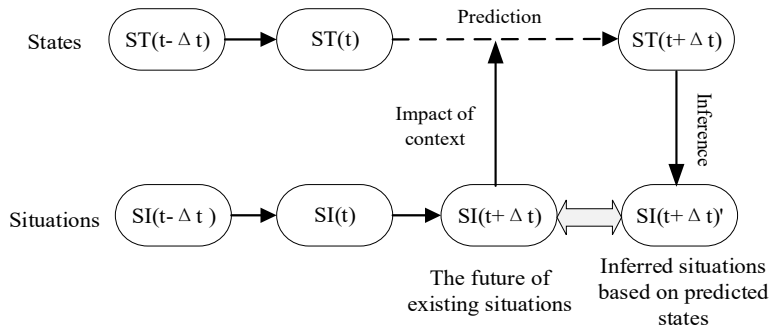
The prediction of the near-future of existing situations is realized through an analysis of the trends regarding the attributive changes of a series of situations in a scene. As shown in Figure 3.12, the situations inferred at three subsequent points in time (i.e.  $SI(t)$ ,  $SI(t-\Delta t)$ ,  $SI(t-2\Delta t)$ ) are used to predict a near-future situation ( $SI(t+\Delta t)$ ). The inferred and the predicted situations are connected by some transitivity interplays. This means that they are actually defined as the same type of situation. Accordingly, several attributes can be selected to describe the changes among the situations, such as location, speed of motion, fuel consumption, etc. The attributes of the near-future situation can be predicted based on the trend of the changes. This can be realized using statistics, machine



**Figure 3.12.** Predicting the future of inferred situations based on the trend of attributive changes

learning or pattern recognition techniques, e.g. looking at the correlation among the changes.

In addition, to predict newly formed situations, the future context should be handled on the state level. It means that the near-future states (of attributes and relations) of entities should be predicted based on which events are to be generated. In this way, newly formed situations are inferred using the predicted states and calculated events. As shown in Figure 3.13, the situation  $SI(t+\Delta t)$  is predicted according to the principle presented above. In order to derive the near-future states of entities (i.e.  $ST(t+\Delta t)$ ), both the historical changes of the states and the impact of context (relevant situations) should be taken into consideration. For instance, the future location of a car (which is involved in a traffic jam) depends on the changes of its locations in the past, as well as on the impact of the traffic jam. Then, based on the  $SI(t+\Delta t)$ , events can be calculated and situations can be inferred (i.e.  $SI(t+\Delta t)'$ ).



**Figure 3.13.** Predicting the future states of entities concerning the impact of situations



Particularly, the inferred situations based on predicted states,  $ST(t+\Delta t)'$  and the predicted future of existing situations,  $ST(t+\Delta t)$ , may contain common elements. For instance, an actual situation might be processed by both approaches. To avoid possible conflicts, we only consider the difference between  $ST(t+\Delta t)'$  and  $ST(t+\Delta t)$ . It means that the situations included in the  $ST(t+\Delta t)'$ , but not included in the  $ST(t+\Delta t)$  are considered as the newly formed situations in the near future.

### 3.4.2.3. Analyze predicted context

When future situations are predicted, the impact of the predicted situations on involved entities should be calculated. The applied principle is as the same as that proposed in sub-section 3.4.2.1. In addition, according to the requirements, the predicted situations should be ranked into a priority list, which is considered as a basis for management of the situations, such as promote the formation of positive situations and prevent the happening of negative situations.

### 3.4.2.4. Analyze individual context of entities

As discussed in Section 3.2.3, the relevance between situations and entities should be calculated, which is the basis to generate the individual context of entities. Should the impact indicator  $II$  of all situations be specified, the relevance indicator,  $RI_{ij}$ , of a situation,  $s_i$ , can be calculated for each and every entity,  $\varepsilon_j$ , using Equation (3-2):

$$RI_{ij} = \left| \left(1 - \frac{\Delta t_{ij}}{\Delta t_{max}}\right) \left(1 - \frac{d_{ij}}{d_{max}}\right) II_i \right| \quad (3-2)$$

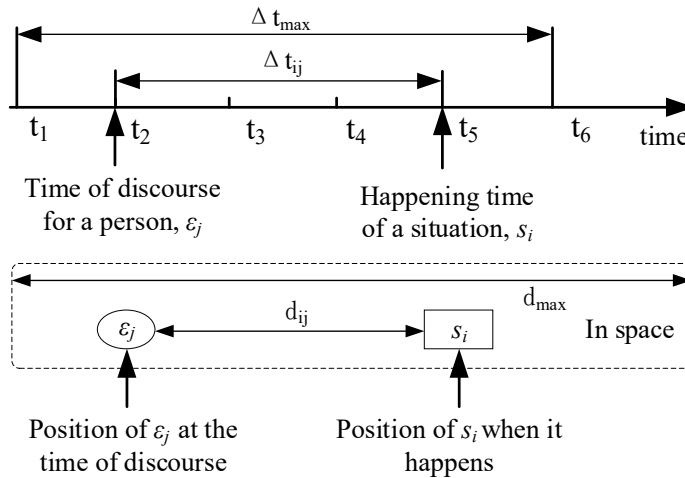
where:  $\Delta t_{ij}$  is the time difference between the point in time when  $s_i$  happens and the time of discourse for a concerned entity,  $\varepsilon_j$ , and  $d_{ij}$  is the distance between the location where  $s_i$  happens and the location of  $\varepsilon_j$  at the time of discourse. The principle for calculating  $\Delta t_{ij}$  and  $d_{ij}$  is shown in Figure 3.14. The values of  $\Delta t_{max}$  and  $d_{max}$  are used to define the boundary of condition with regard to both time and space, forming a local area where the calculation is taken. It means if  $\Delta t_{ij} \geq \Delta t_{max}$  or  $d_{ij} \geq d_{max}$ , then the value of the relevance of a particular situation to the entity should be 0. The values of  $\Delta t_{max}$  and  $d_{max}$  may be different in various applications. Both of them should be specified depending on the practical requirements. For instance, the value of  $d_{max}$  may be several kilometers in the case of a traffic management system, while it may be a few hundred meters in the case of a fire evacuation system. The values of  $\Delta t_{ij}$  and  $d_{ij}$  are set to 0 when an entity is involved in a situation. If this applies, then the values of  $RI_{ij}$  and  $II_i$  are equal. What it in general means is that the relevance of a situation should be handled as a variable, which depends only on its actual impact.

The calculated value of  $RI$  may have various implications and may enable multiple computational functionalities. On the one hand, it shows the importance of situations to various entities. For instance, if a situation is characterized by a high  $RI$  value with regard to a given entity, then that situation is of higher importance for the entity. Relying on this interpretation, particular individual contexts can be specified by integrating situations whose  $RI$ s are higher than a predefined threshold. On the other hand, the danger levels concerning the various entities can also be evaluated using the calculated value of  $RI$ . If a situation having a negative impact (i.e.  $RI < 0$ ) is strongly relevant to an entity, then this entity is more likely involved in some dangerous circumstances. To evaluate the overall level of danger, all situations that have negative impacts should be taken into consideration. The sum of the relevance indicators ( $SRI$ ) of all negative situations related to an entity,  $\varepsilon_j$ , can be calculated by Equation (3-3):

$$SRI_j = \sum_{i=1}^n RI_{ij} \quad (3-3)$$

where:  $RI_{ij}$  is the indicator of the relevance of a situation,  $s_i$ , to an entity,  $\varepsilon_j$ . Assuming that, in total, there are  $n$  negative situations associated with the entity at a given point in time.

Having the calculated values of  $SRI$ s, the danger level of entities can be assessed. Actually, the assessment can be realized using different thresholds, which are determined according to the requirements of applications. Another benefit of using the calculated  $SRI$ s is that the addressed entities can be ranked based on the



**Figure 3.14.** Calculation of the spatial and temporal relations between situations and entities

calculated SRI and included in a priority list accordingly. Using this priority list, the computational mechanism is able to (i) specify the order for providing personalized services (e.g. order of informing people), and (ii) achieve certain prioritization in the management of situations.

### **3.4.3. Functionalities for devising action-plans for people**

Dynamic context of entities considers spatial-temporal changes of situations related to the entities. It implies that dynamic context management is realized by informing people to perform certain actions, which may proactively change related situations. In I-CPS applications, both long-term personal solutions and short-term personal actions should be devised for people to perform, which jointly form personalized action plans. This is because of the fact that I-CPSs normally need to achieve certain goals depending on specific application-oriented requirements. For instance, a vehicular navigation system should guide cars on a highway to reach certain locations (e.g. destination). In the meanwhile, time-dependent elementary actions performed by people, such as increase or decrease the speed of motion, can be generated by I-CPSs to achieve sub-objectives. For instance, a moving car on a highway might be informed to change the traffic lane in order to give the space for the following ambulance.

Long-term personal solutions and short-term personal actions are interrelated. On one hand, a long-term solution developed for a person is the basis to generate alternative short-term actions. On the other hand, a determined personal action should contribute to the realization of the ultimate goal associated with the long-term solution. Therefore, by informing people about the developed action plans (which consider both long-term solutions and short-term actions) I-CPSs are able to manipulate the context of the informed people themselves, but also the context of other people. The related sub-functionalities are explained below.

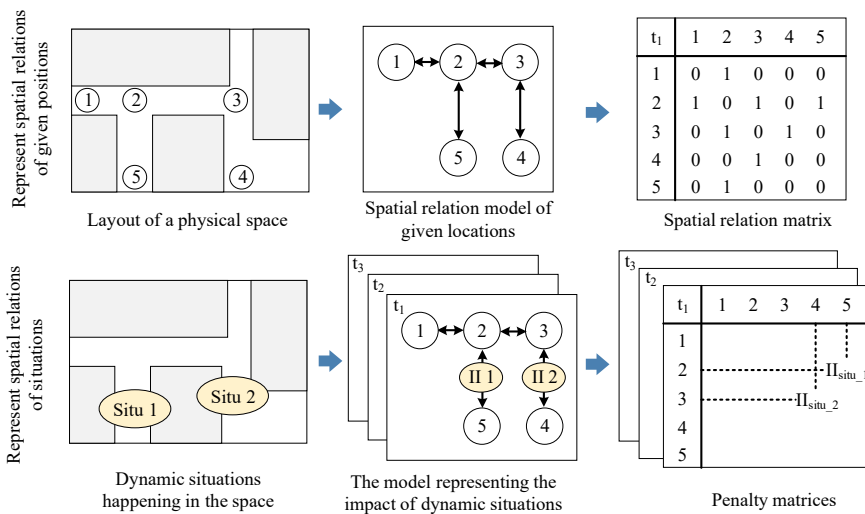
#### **3.4.3.1. Devise long-term solutions**

The goal of an I-CPS with regard to context management was achieved by informing people about the devised long-term solutions. To this end, the long-term solutions to be taken by people should depend on the impact of changing contexts. This implies that (i) the impact of context should be computationally represented, and then (ii) considered in an articulated reasoning or decision-making process. Therefore, two sub-functionalities can be decomposed, namely (i) representing the impact of context, and (ii) generating personal solutions based on the represented impact.

According to our assumptions, the consideration of contextual impact should integrate the impacts of interplaying situations. As discussed in sub-section 3.4.2.1, the impact of a situation can be quantitatively assessed by its impact

indicator (II), which is calculated based on the attributes of situations and attributive changes of involved entities. Therefore, the model dedicated for representing the contextual impact should consider (i) spatial-temporal information of related situations and (ii) time-dependently calculated IIs of situations. For this reason, we proposed a representation mechanism, which is shown in Figure 3.15.

For a given physical space, its layout can be digitally represented by a relation model (e.g. a graph model), based on which the spatial relations among several predefined locations (e.g. landmarks) can be known. Actually, the spatial relation model can be used as a basis to indicate how the inferred situations are spatially related to each other. For instance, the location of ‘situation 1’ and ‘situation 2’ in Figure 3.15 can be represented by considering the locations of several predefined locations, (i.e. ‘Positions 1-5’). Then, the calculated II of the situations can be assigned to the referred locations in the model, such as to the edges ‘2-5’ and ‘3-4’. If the spatial relations of the given positions are represented in a matrix, a penalty matrix can be built by assigning the IIs of concerned situations to the corresponding positions in the spatial relation matrix, as shown in Figure 3.15. In addition, since the II of situations are calculated in a time-dependent manner (concerning the actual impact of situations at given points in time), representation of the impact of dynamic situations over a period of time can be achieved by generating multiple time-dependently calculated penalty matrices. The integrated penalty matrices show how the concerned situations vary within the given space and the period of time.



**Figure 3.15.** Representing the impact of situations based on the spatial relation model of a given physical space

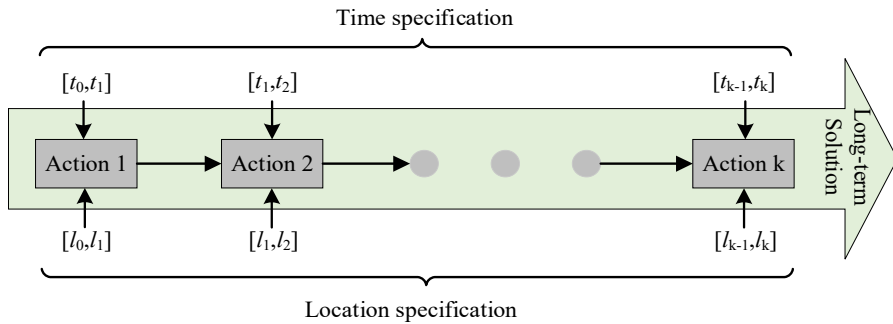
Based on the represented model, long-term personal solutions can be generated through application-specified decision-making or reasoning algorithms. For instance, planning a proper path within the concerned physical space can be an alternative long-term personal solution. This could be realized using a path-finding algorithm to plan an optimized route under multiple criteria (e.g. nearest route, shortest time, lowest II, etc.).

### **3.4.3.2. Devise short-term actions**

Normally, an action is defined as one of several possible activities that can be performed by a concerned person, such as ‘move to a location’ and ‘wait for 30 seconds’. An action plan is a composition of a series of actions arranged into a particular order, as shown in Figure. 3.16. Both an action plan and the elementary actions contained in the action plan have clear objectives. It assumes that the concerned person takes the actions one by one according to the order. The elementary actions cumulatively contribute to the action plan. In this way, the devised long-term personal solution is achieved when the last action is finished.

The actions to be taken by people should depend on changing situations. It implies that the action plan should not only contain details about what the people should do step by step but also include the specified time points and locations in the actions. Because, both the specified time points and locations provide a basis for evaluating the change of context if the generated action plan is performed by the concerned person. Based on these, three aspects should be specified in an action plan, including (i) what actions should be taken by the concerned people, (ii) what time should the action be completed, and (iii) where should the action be taken place.

Therefore, in addition to the specified actions and the order of the actions, every action and action plan should consider a period of time, during which the action is taking place. It implies that the concerned person should start to perform the action at a point in time (e.g. start to move towards a location at  $t_0$  in Figure 3.16), and compete for the action at another one (e.g. reach the target location at  $t_1$ ). Since the actions are placed in an order, the completion time of an action is the starting time of the next action followed. The total time consumed for completing the action plan is considered as the completion time of the last action. In addition, every action is labeled with spatial information regarding where the action is taken place. Similar to the time information, the specification of the spatial information should consider two aspects: the location where an action is started (e.g.  $l_0$  in Figure 3.16) and the location where an action is completed (e.g.  $l_1$ ). The distinction between the concerned two locations of action implies the distance to be taken by the person. Accordingly, the sum of the distances of the elementary actions is considered as the total distance of the action plan.



**Figure 3.16.** An action plan for a person to perform

The concept of action plan satisfies the proposed requirement with regard to dynamic context management. The specification of SAT information of people at given points in time enables evaluation of an assumed context formed by the people and their relations. This facilitates assessing if any contextual change is good or not. Accordingly, the best contextual change can be selected from a set of alternatives to achieve an effect of proactive handling of dynamic context. Actually, the selected action plans can be easily translated into personalized messages and used to inform people about the activities they should perform. To this end, three sub-functionalities can be decomposed, including (i) generating a context-dependent objective function based on the II of situations, (ii) generating alternative action plans based on the devised long-term action plans, and (iii) selecting the best action plan based on the generated objective function.

#### 3.4.4. Functionalities for messaging people about context and action plans

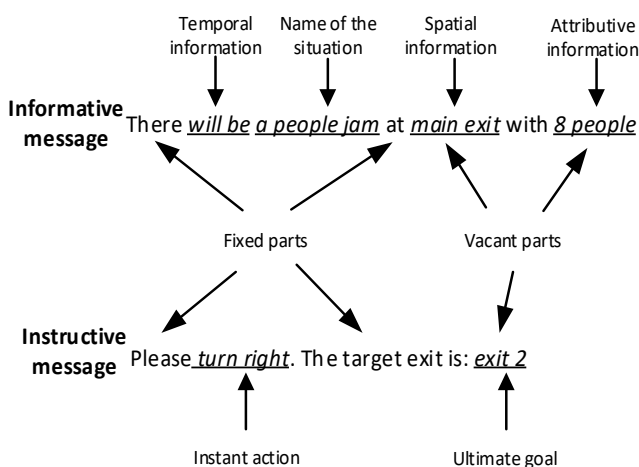
When the semantic knowledge hidden from dynamic context is inferred and action plans for people are determined, personalized messages can be generated and used to inform people (e.g. people who can be informed) about their personal context and devised activities. To achieve the main function, the messaging mechanism should: (i) construct personalized messages, and (ii) send the constructed messages according to a specified order.

The template-based approach was considered for message construction since it provides a simple and fast way for real-time generation of natural language texts under different conditions. To enable this, as a first step, a proper message template should be selected. The assessed value of the personal danger (in sub-section 3.4.2.4) can be considered as a basis for the selection. Then, having the selected template, both informative and instructive messages can be generated, which are the rest functionalities of the messaging mechanism. Informative messages indicate “what the person should be aware of”, while instructive

messages are used to inform people about “what they should do”. The former may manifest as personal recommendations, situated solutions, or action guidance, whereas the latter is used to achieve certain context management objectives. Therefore, to generate informative messages, the SAT information related to the most relevant situation of a person is used to generate message components required by a selected message template. In addition, the derived personal action plans should be converted into various message components at constructing instructive messages. In the end, the constructed informative and instructive message parts are integrated to form the personalized messages communicated to the target people.

To use the template-based approach, various message templates should be articulated and stored in a warehouse. A typical example of a message template is shown in Figure 3.17. The principles for designing message templates are as follows: Every message template includes fixed parts and variable (vacant) parts. The generated message components should be placed into vacant places according to the type of information they represent. The fixed parts serve as linking words among the vacant parts (e.g. prepositions and modal particles), making a sentence readable and expressing some rhetorical styles.

In addition, the message template for constructing informative messages includes some places showing the name and the SAT information of the concerned situation. In order to facilitate the construction of instructive messages, both the instance actions (the first action in an action plan) the person should do and the ultimate goal of the personal solutions to be achieved (about the solution) should



be transited to various messages components, and placed at the vacant parts of the template.

As discussed in the sub-section 3.4.2.4, the calculated SII of entities can be used to assess the danger level of people. It can also be used as a reference for ranking people into a priority list.

**Figure 3.17.** Construction of personalized messages based on a message template

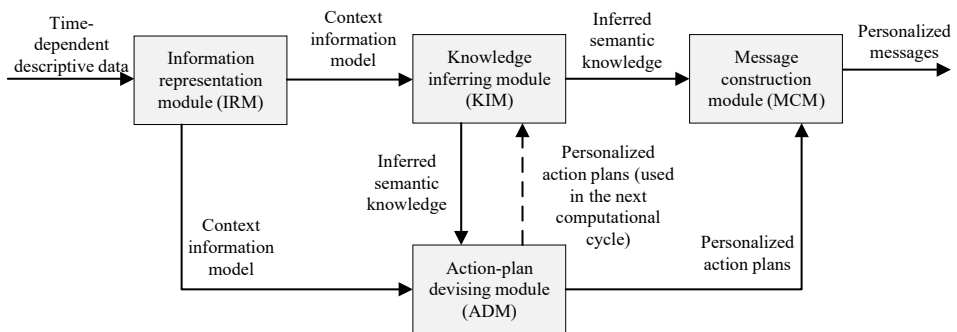
According to the list, the order for sending the constructed messages to the personal devices (e.g. mobile phone) of the concerned people can be specified. This way, certain prioritization can be achieved in the management of distribution as a whole.

### 3.5. Devising the architecture and workflow of the computational mechanism

#### 3.5.1. Architecture

The above-introduced functionalities can be conceived as a multi-module computational platform, which can provide real-time dynamic context computation services for I-CPSs. The overall architecture of this platform is shown in Figure 3.18. The computational engine of the platform contains four modules, namely information representation module (IRM), knowledge inferring module (KIM), action-plan devising module (ADM) and message construction module (MCM). Accordingly, the computational engine includes multiple algorithms for (i) representing DCI about the states and relations of concerned entities, (ii) inferring semantic knowledge from dynamic context to enable awareness building, (iii) devising personalized action plans for involved people through context management to achieve certain objectives, and (iv) constructing personalized messages for the involved people based on the inferred semantic knowledge, their communication opportunities, and the recurrently generated action plans. These four modules are procedurally interconnected and eventually manifest in an integrated computational engine.

The computational engine makes use of the time-dependent descriptive data aggregated from a time-varying process, and generates personalized messages for the purpose of real-time guidance and informing. The specific objective of the



**Figure 3.18.** The generic framework of a multi-module computational platform for I-CPSs



IRM is to reformulate the descriptive contextual data to a constructed CIR-cube, which facilitates modeling SAT contextual information about the states and relations of physical entities and the associated changes. The framework of the CIR-cube was developed as a foundational and representational model, which supports context inferring and reasoning functionalities.

In the KIM, the actual variations of the physical process are inferred from the temporal relations of the momentary states, which are represented by the cells of the CIR-cube. For the purpose of computation, the variations of the states related to entities are captured as ‘*events*’. Multiple interacting states and events form a ‘*situation*’. A situation can be inferred by integrating and abstracting information about multiple states and/or events according to certain predefined rules. Then, an arrangement of situations has been captured by a ‘*scene*’. This is the theoretical basis of reaching from syntactic data (information) to semantic constructs (knowledge) in the KIM. The associated semantic knowledge, e.g. future context, prioritization and danger level of entities, generated in the KIM formed a basis for the computation of the ADM and the MCM.

Based on the inferred situations and interpreted semantic knowledge associated with the situations, personalized ‘*action plans*’ are generated for the people by the ADM. As discussed, each action plan contains a series of actions that are supposed to be performed by informed persons. Generation of the personalized action plans considers (i) the knowledge generated on the basis of interplaying situations, and (ii) the capabilities of concerned persons, which means that both the semantic knowledge and the CIR-cube are needed by the ADM. In addition, the devised action-plans may also be used by the KIM to generate semantic knowledge. For instance, predicting the obedience of informable people can be realized by comparing the action-plans generated for them in the previous computational cycle and the actual action they have performed.

The specific objective of the MCM is to make use of the information encompassed by the ‘inferred situations’ and the developed action plans to construct personalized messages that support the informative and instructive communications to the involved people. According to the description with regard to the functionality of the MCM, the construction of personalized messages should make use of both inferred semantic knowledge and devised action-plans. A detailed specification of the workflow of the computational mechanism on the scheme-level is given in the following section.

### **3.5.2. Workflow**

The workflow of the DCIP-M is shown in Figure 3.19. As the kernel of a computational platform, the execution of the DCIP-M should be controlled by the I-CPSs (if it is needed to compute). After control commands (e.g. a computation

request) given by the I-CPSs are confirmed, the IRM acquires the time-dependent descriptive context data with regard to entities and uses the data to construct CIR-cubes. Actually, the construction of a CIR-cube may involve multiple computational cycles. Based on the input data at a given point in time (e.g.  $t_0$ ), an empty SFR-matrix and the associated content profiles can be created.

Then, the distances among the considered entities are to be calculated based on the locations of entities and the layout of the concerned physical space. The completed SFR-matrix is considered to be the first frame of a CIR-cube. After the descriptive context data regarding another point in time is acquired (e.g.  $t_0 + \Delta t$ ), a new SFR-matrix should be generated. Due to the reason that some information (e.g. distance between two entities) to be filled in the newly created SFR-matrix may not change within the time duration between  $t_0$  and  $t_0 + \Delta t$ , it is reasonable to focus on the changed information only, while the unchanged information can be loaded from the CIR-cube. Thus, the IRM should generate a list of entities whose location have changed within the time duration, and compute distances among entities based on the list and compute the speed of motion of the entities. Then, the changed distances, unchanged distances and the calculated speeds are placed into the corresponding positions in the SFR-matrix. The completed SFR-matrix is then integrated into the CIR-cube. After this is done, the IRM judges the sufficiency of the information contained in the CIR-cube according to some conditions (e.g. the number of SFR-matrices included in a CIR-cube). If it is approved, further computations can be conducted. Otherwise, a new SFR-matrix should be generated by IRM (e.g. using the data at  $t_0 + 2\Delta t$ ).

In the KIM, situations are inferred (i.e. situations at  $t_0 + 2\Delta t$ ) based on the CIR-cube and their impact indicators on involved entities are individually calculated. According to our proposed concepts, scenes cannot be obtained unless situations happening at two successive points in time are known. It means that a new computational cycle is needed, in which the contextual data at  $t_0 + 3\Delta t$  should be included in the CIR-cube. And, situations at  $t_0 + 3\Delta t$  can be inferred. The two sets of situations are used to form scenes, based on which the changes (e.g. location, attributes) of the concerned situations can be known. In order to predict the future of existing situations, the trend of situational changes should be obtained, which is calculated based on two successive changes. It means that the descriptive data at  $t_0 + 4\Delta t$  should be used in the CIR-cube as a basis to infer one more set of situations (i.e. situations at  $t_0 + 4\Delta t$ ). In this way, the trend of situational changes can be calculated using the situations inferred at  $t_0 + 2\Delta t$ ,  $t_0 + 3\Delta t$  and  $t_0 + 4\Delta t$ . Then, the future of existing situations and the locations of entities can be predicted based on the above-mentioned functions in the sub-section 3.4.2.2. Based on the predicted locations of entities, the distances among the entities can be calculated and a new SFR-matrix can be generated, which is used to infer newly created situations in the near future.

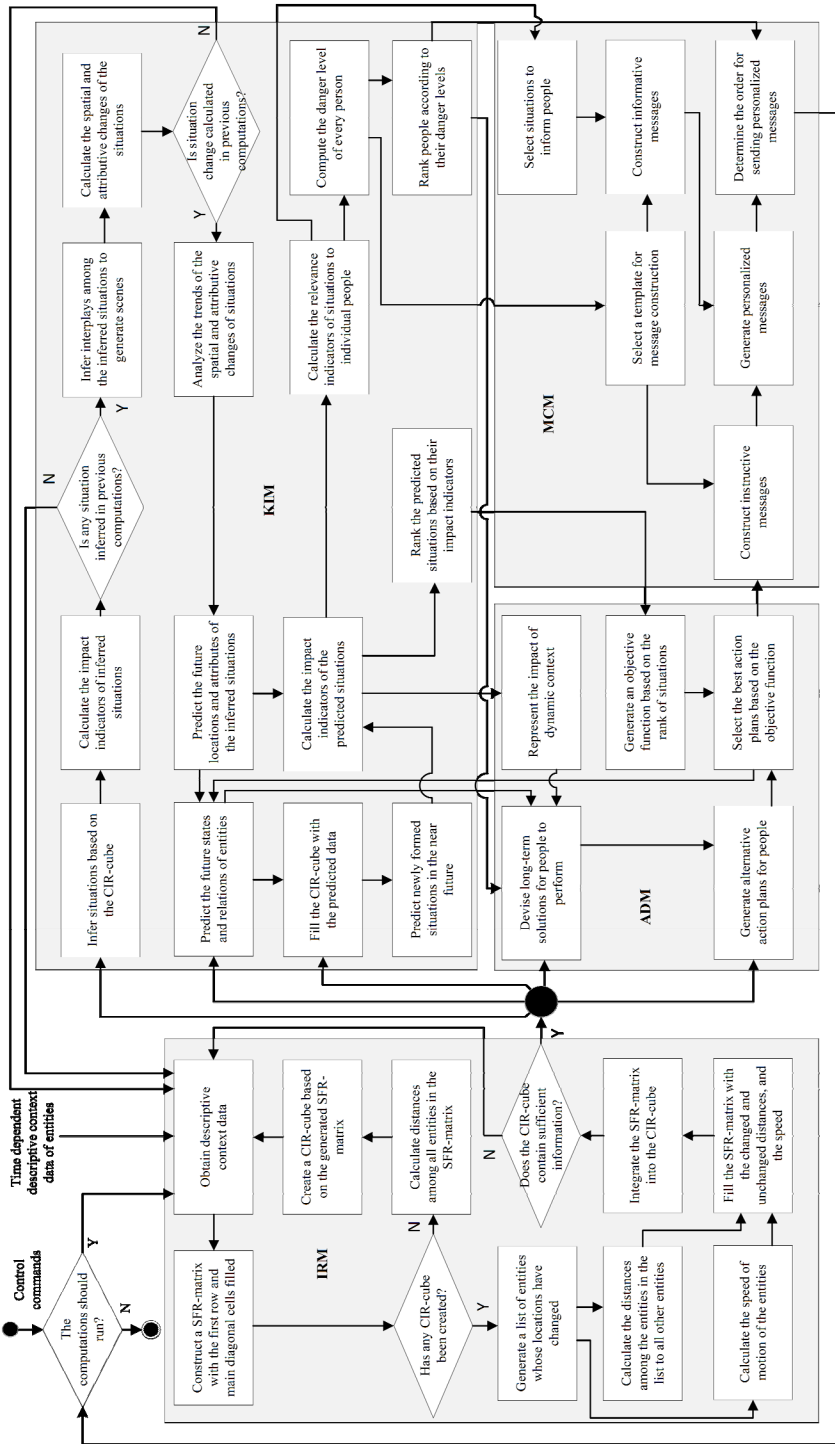


Figure 3.19. Integration of the functionalities of the computational mechanisms

To evaluate the impact of predicted situations, their II should be calculated based on Equation (3-1). The calculated II provide a basis for treating the predicted situations towards different functional needs. On one hand, the predicted situations can be arranged into a priority list based on their II. The result could be used by the ADM in generating personalized solutions and actions for context management. On the other hand, Equation (3-2) can be applied in the calculation of the relevance indicators between predicted situations and individual people. Accordingly, the danger level of people can be assessed, and people could be ranked into a priority list based on the assessed danger level. Based on the scheme-level specification, the inferred semantic knowledge from dynamic context by the KIM include: (i) the predicted future of entities, (ii) the predicted near-future situations, (iii) a priority list of predicted situations, (iv) the impact of predicted situations, (v) the relevance between situations and people, (vi) assessed danger level of people, and (vii) a priority list of people.

Some of the inferred semantic knowledge is used in the KIM for devising personalized action-plans. As the first step, the impact of dynamic context should be represented based on a dedicated spatial relation model, which encapsulates the locations and the II of the predicted situations. Then, long-term personal solutions are to be reasoned out depending on the (i) predicted attributes of people (e.g. obedience of people), (ii) the represented impact model, (iii) the basic information of people (contained in the CIR-cube), and (iv) the priority list of people. Based on the long-term solutions, alternative action-plans can be generated for every people. In addition, due to the fact that different alternative actions taken by people may result in different contextual changes, the best action plans should be selected from the alternatives. In the KIM, the selection is realized using a time-dependently constructed objective function, in which the II of the situations is used.

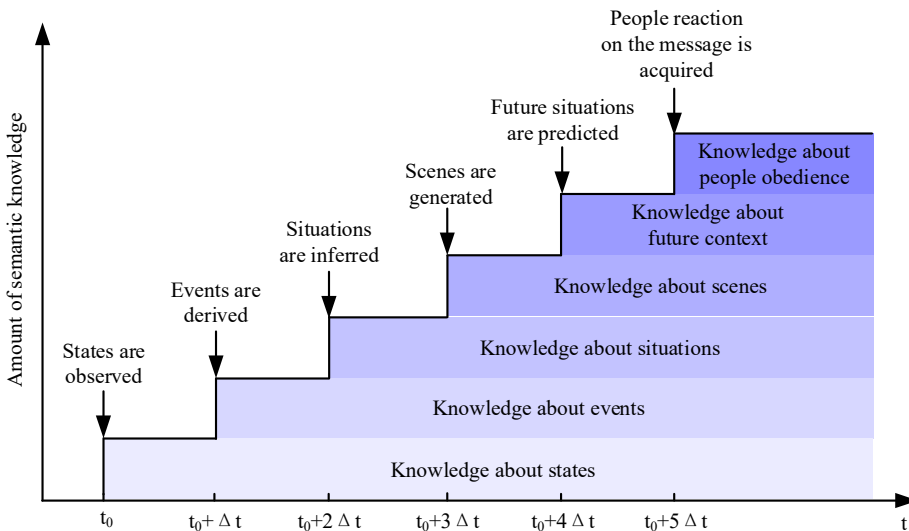
Based on the determined personal action plan and the inferred semantic knowledge, the MCM is able to generate personalized messages. To this end, a message template should be firstly selected from a set of alternations according to certain principles (e.g. based on the danger level of people). The selected message template can be used for constructing both informative messages and instructive messages. Particularly, to construct informative messages, situations should be selected based on the calculated relevance indicators (e.g. the most relevant situation to individual people). Instructive messages are generated considering the devised personal action plans. After this, both informative messages and instructive messages are combined to form personalized messages. Then, the order for sending personalized messages is determined based on the priority list of people.

### 3.5.3. Discussion

It can be seen from the scheme-level specification of the DCIP-M that all the decomposed functionalities in the Sub-chapter 3.4 are considered. Actually, the elementary functions of every module are not linearly arranged but are integrated according to particular algorithmic purposes. In the IRM, the inner-operations make it possible to concentrate on the distances changed among the concerned entities in computation. When the number of entities concerned by the IRM increases, their spatial relations may squarely grow. By doing so, the IRM does not need to re-calculate the unchanged spatial relations. This is important for real-time processing of DCI since the computation time for constructing an SFR-matrix (CIR-cube) can be decreased. Especially, in I-CPS applications that most of the entities are motionless or move slowly, e.g. car-parking management applications, this kind of design could significantly reduce the reaction time of systems in handling contextual changes.

In addition, it can be seen from the articulated workflow of the system that semantic knowledge with regard to dynamic context is generated based on continuous monitoring of a varying process over a given period of time ( using the descriptive context data aggregated at multiple time points). System awareness is built through multiple computational cycles. As shown in Figure 3.20, the amount of semantic knowledge gradually increases as the number of computational cycle increases after the computational engine is controlled to compute.

At  $t_0$ , states with regard to entities and relations are observed, which means that



**Figure 3.20.** Obtaining semantic knowledge using multiple computational cycles

the system can only obtain knowledge about states. When contextual data at  $t_0 + \Delta t$  is known, the system could compare the states, derive events, and generate knowledge about events. In the third computational cycle (at  $t_0 + 2\Delta t$ ), situations can be inferred and situation-related knowledge is able to be acquired. By inferring interplays among situations, scenes and associated knowledge could be known in the fourth computational cycle. At this moment ( $t_0 + 3\Delta t$ ), the generated scenes only contain two situations, and only one situational change could be identified. This is the reason why the fifth computational cycle (at  $t_0 + 4\Delta t$ ) is needed to compute two successive changes for the purpose of dynamic context prediction. According to the analyzed requirements, a proactive kind of context management should be achieved by the system.

Thus, personalized action-plans could be devised based on the predicted context, and then be delivered to the associated people for execution. In the next computational cycle ( $t_0 + 5\Delta t$ ), the knowledge about the reaction of informed people to the given action-plan could be obtained. This is the basis for analyzing the obedience of people, which enables I-CPSs to adapt to the cases that people disobeying happen.

The interoperations among the above-mentioned functions of the modules lend themselves to different computational mechanisms. On the other hand, each of the functions may require interrelated algorithms to support the realization. Specification of the algorithms is discussed in the next sub-chapter. For the purpose of simplification, all the functions to be realized by the DCIP-M are listed in Table 3.2. The numbering of functions will be used as a reference in the discussion with regard to the algorithms specification, implementation and the testing of the functions.

### **3.6. Algorithm-level specification of the DCIP-M for indoor fire evacuation guiding**

To implement and test the proposed DCIP-M, algorithms used to realize the required functions are specified in this subchapter. For this purpose, the reference application case, indoor fire evacuation management is concerned. All specified algorithms are dedicated to this reference application. A detailed explanation is given as follows.

#### **3.6.1. Algorithm-level specification of IRM**

The algorithms required for realizing the functions belonging to the IRM are listed in Table 3.3. There are 8 algorithms in total, and some of them are reused in multiple functions. To obtain descriptive context data, the location of people in a building should be estimated. Because, in the practice, the existing technologies applied for indoor positioning always produce certain observation errors [11].

**Table 3.2.** Functions to be realized by the DCIP-M

No.	Functions of the computational mechanisms
F1.1	Obtain descriptive context data
F1.2	Construct an SFR-matrix with the first row and diagonal cells filled
F1.3	Calculate distances among all entities in the SFR-matrix
F1.4	Create a CIR-cube based on the SFR-matrix
F1.5	Generate a list of entities whose locations have changed
F1.6	Calculate the distances among entities in a list and all other entities
F1.7	Calculate the speed of motion of the entities
F1.8	Fill the SFR-matrix with the changed and unchanged distances, and the speed
F1.9	Integrate the SFR-matrix into the CIR-cube
F2.1	Infer situations based on the CIR-cube
F2.2	Calculate the impact indicators of the inferred situations
F2.3	Infer interplays among the inferred situations to generate scenes
F2.4	Calculate the spatial and attributive changes of the situations
F2.5	Analyze the trends of the spatial and attributive changes of situations
F2.6	Predict the future locations and attributes of the inferred situations
F2.7	Predict the future states and relations of entities
F2.8	Fill the CIR-cube with the predicted data
F2.9	Predict newly formed situations in the near future
F2.10	Calculate the impact indicators of the predicted situations
F2.11	Rank the predicted situations based on their impact indicators
F2.12	Calculate the relevance indicator of all situations to individual people
F2.13	Assess the danger level of every person
F2.14	Rank people according to their danger levels
F3.1	Represent the impact of dynamic context
F3.2	Devise long-term solutions for people to perform
F3.3	Generate alternative action-plans for individual people
F3.4	Generate an objective function based on the rank of situations
F3.5	Select the best action plans based on the generated objective function
F4.1	Select situations to inform people
F4.2	Select a template for message construction
F4.3	Construct informative messages
F4.4	Construct instructive messages
F4.5	Generate personalized messages
F4.6	Specify the order for sending personalized messages

It may influence the accuracy of context representation and computation (e.g. distance calculation). Thus, A1 is used to increase the accuracy of the acquired locations. The objective of F1.2 is to calculate the attributes of entities, and then create an empty SFR-matrix and content profile for the entities based on the identification, attributes, and locations. In the case of IFEG, three types of entities are considered, including people, exits and fire. The sub-zone (e.g. room number) of entities (people and fire) should be judged by A2 as an attribute since it is the basis for the calculation of distance among the entities. The rest algorithms in the IRM are quite simple, a detailed explanation of them will be given in Chapter 4.

### 3.6.2. Algorithm-level specification of KIM

The algorithms required for realizing the functions belonging to the KIM are listed in Table 3.4. One of the objectives of the IFEG system is to guide people in order to reduce the number of those who are involved in people jams as much as possible. It has to be seen that jams formed by people not only obstruct the motion of involved people but also increase the risk of danger to the people. Therefore, being aware of the time-varying people jams is crucial for the IFEG system. In F2.1, we considered three types of situations: (i) floating people jam, which was formed by a group of people moving together (at arbitrary location), (ii) anchored people jam, which is formed by an accumulation of people at a fixed location, e.g. in front of an exit, and (iii) the proliferation of fire, which is represented by its boundary. Accordingly, in F2.2, both impact indicators of people jams and the fire should be calculated.

In F2.3, transitivity and causality interplays are focused on, as demonstrations to show the effectiveness of the concept of scenes. The principle for inferring these interplays has been presented in Section 3.2.2. Another important purpose of using the transitivity interplay is that it enables calculation of the changes in

**Table 3.3.** Specified algorithms for the IRM for the IFEG application

Func.	Required algorithms
F1.1	A1: Estimate the location of people
F1.2	A2: Determine the sub-zone of entities A3: Create a SFR-matrix containing entity's ID, location, sub-zone and type
F1.3	A4: Calculate the distance between two entities
F1.4	A5: Add a SFR-matrix to the CIR-cube
F1.5	A6: List the ID of entities whose location change is bigger than a threshold
F1.6	A4: (See the description above)
F1.7	A7: Calculate the speed of motion of entities
F1.8	A8: Fill the distances and speed of entities into a SFR-matrix
F1.9	A5: (See the description above)



situations. According to the exemplified situations, algorithms in F2.4 considered (i) the changes of location and speed of motion of people jams, and (ii) the location change of fire.

The reason is that these types of information facilitate prediction of future context. To represent the change of the situations in F2.5, the regression models with

**Table 3.4.** Specified algorithms for the KIM for the IFEG application

Func.	Required algorithms
F2.1	A9: Infer floating people jams A10: Infer anchored people jams A11: Infer fire situations
F2.2	A12: Calculate impact indicators of people jams and fire
F2.3	A13: Infer transitivity interplay among the inferred situations A14: Infer causality interplays among the inferred situations
F2.4	A15: Calculate the location change of people jams A16: Calculate the change of speed of motion of people jams A17: Calculate the location change of fire situations
F2.5	A18: Generate regression models representing the location changes of people jams A19: Generate regression models representing the speed changes of people jams A20: Calculate the coverage increment of fire
F2.6	A21: Predict the future locations of floating people jams A22: Predict the future speed of motion of people jams A23: Predict the future coverage of fire
F2.7	A24: Predict near-future locations of people A4: (See the description above) A25: Predict the obedience of people to the given instructions
F2.8	A3: (See the description above) A7: (See the description above) A8: (See the description above) A5: (See the description above)
F2.9	A9: (See the description above) A10: (See the description above)
F2.10	A12: (See the description above)
F2.11	A26: Rank situations according to their impact indicators
F2.12	A27: Calculate the relevance indicators of all situations to people
F2.13	A28: Compute the danger level of people
F2.14	A29: Rank people according to the computed danger level indicators

regard to the location changes and speed changes of people jams should be generated, while the coverage increments of the boundary of fire are to be calculated. The results will be used in the algorithms of F2.6 to predict the future locations, speed of motions of people jams, and future coverage of the fire. In this way, the future of existing situations can be known.

Algorithms in F2.7 are dedicated to predicting the near-future states of entities. According to the specific requirements of the IFEG application, the future locations of people, the distances among people and the obedience of informable people to given instructions should be calculated. To include the future information in the CIR-cube, A3, A7, A8, and A5 should be used again in F2.8, while A9 and A10 are reused in F2.9 to infer people jams in the future based on the newly updated CIR-cube. When the future situations are obtained, F2.10 calculates the impact indicators of the situations based on A12. The calculated impact indicator enables (i) ranking the predicted situations in A26, (ii) calculating the relevance indicators of all situations in A27, (iii) computing the danger level of people in A28, and (iv) ranking people in A29.

### 3.6.3. Algorithm-level specification of ADM

The specified algorithms for ADM are listed in Table 3.5. To represent the impact of context for reasoning, many models have been proposed in the literature, such as the context spaces model reported in [12], semantic web model in [13], and the object role modeling used in [14]. Actually, the approach used for representation is strongly dependent on the objective and approach of context reasoning. In case of IFEG application, the personal long-term solution could be specified as the escape route that to be followed by people, while short-term actions can be

**Table 3.5.** Specified algorithms for the ADM for the IFEG application

Func.	Required algorithms
F3.1	A30: Generate penalty matrices to represent the impact of context
F3.2	A31: Calculate the lengths of optimal paths among spatial landmarks A32: Calculate the optimal paths of individual people to exits A33: Determine potential informants and help-recipients A34: Revise the paths of help-givers to enable help-recipients be informed
F3.3	A35: Calculate the positions that will be reached by informable people on their path A36: Generate alternative actions that could be performed by informable people
F3.4	A37: Calculate the weights of situations for optimization A38: Generate an objective function based on the weights of situations
F3.5	A39: Select the best action plans based on the generated objective function

considered as the possible actions to be taken by people when they are following the escape route. For this reason, a path-planning algorithm is in the need for generating personalized long-term solutions.

To facilitate path-planning based on the impact of situations, a penalty matrix approach should be generated. The initial content of the main cells of the penalty matrix is set to '0'. Thus, to jointly represent the locations and impacts of the situations at a given moment, the IIs of concerned situations are given to the represented positions in the penalty matrix. It's worth mentioning that a penalty matrix can be either symmetric or asymmetric, depending on if the actual impact of the situations on the motion of people is directional or not. In case of a fire evacuation, all people are assumed to move towards exits. For this reason, we will use a symmetric penalty matrix only.

The escape solution generated for a person may be one of the multiple paths. In A31, the optimal paths among the assigned SLMs are determined, and the lengths of the paths are calculated. This algorithm is used to facilitate the path-finding in the concerned physical space, which will be achieved by A32. In addition, in order to inform people who are unable to receive messages directly from the system, potential informants and help-recipients are determined in A33. Then, some of the potential informants will be selected as help-givers and their escape paths will be changed to enable the help-recipients to be informed. This should be achieved by a certain optimization approach applied in A34.

Due to the reason that dynamic context management assumes that people may behave differently, every behavior and every combination of behaviors performed by multiple people may result in different consequences. Theoretically, the combinatorial optimization for devising personal action-plans is similar to the Travelling Salesman Problem, which has been proved to be a non-deterministic polynomial-time hardness (NP-hard) problem [15]. Heuristic type of approximation algorithms can be applied to solve this problem [16].

After personal escape paths are determined, the algorithms specified for F3.3, F3.4 and F3.5 are used for generating personal action plans. Due to the fact that there are multiple elementary actions in an action plan, and every action may be one of the multiple possible alternatives, optimizing all actions in an action plan is a complex issue, which may not be realized in real-time information processing. Therefore, we only focus on the first action in an action plan, as a demonstration of the proposed concepts. To this end, A35 is used to calculate the next positions that will be reached by all informable people (who are controllable), if they will follow the escape paths given to them. For a concerned person, A36 aims to generate alternative actions, which are different from the position calculated by A35 and could be done by the concerned person.

Since different actions taken by people may result in different contexts, the

generated alternative actions are evaluated, and the best actions are selected in F3.5. The basis of the evaluation is an objective function, which is constructed depending on the impact of situations. To achieve this, A37 calculates the weights of concerned situations for optimization. The principle is that the bigger the impact of a situation (represented by  $\Pi$ ) is, the bigger the associated weights will be. In addition, according to the requirements of the fire evacuation, people should be kept away from dangerous situations as much as possible. The calculated weights will be used for the construction of an objective function by A38, based on which the above-mentioned requirement is satisfied.

### 3.6.4. Algorithm-level specification of MCM

The needed algorithms for the MCM are presented in Table 3.6. In the case of indoor fire, people may not have enough time to read a long message. Thus, the most important situation of individual people is considered to be contained in informative messages. This is the reason why the situation with the largest relevance indicator (RI) is selected in A40. To enable message construction, a message template should be selected using A41. The assessed value of the personal danger level (DL) is to be considered as a basis for the selection. For instance, when a concerned person is in a danger, a message template with an emergency-style could be used for sentence building.

A42 and A43 are employed for generating various message components, which form the contents of the messages. To achieve this, a large pool of the components of the informative and instructive messages should be defined. The information related to the most relevant situation should be used at constructing an informative message, whereas the derived personal action plans should be converted into various message components at constructing instructive messages. In the end, the constructed informative and instructive message parts are integrated to form the personalized messages communicated to the target people by A44. The order of sending personalized messages is decided upon based on the generated priority list of the people.

**Table 3.6.** Required algorithms for the MCM for the IFEG application

Func.	Specified algorithms
F4.1	A40: Select situations with the largest RI of informable people
F4.2	A41: Select message templates based on the DLs of informable people
F4.3	A42: Convert the descriptive information of the situation to messages components
F4.4	A43: Convert the personal action plans to messages components
F4.5	A44: Integrate informative message and instructive message
F4.6	A45: Determine the order for sending the messages

### 3.7. Concluding remarks

In this chapter, computational mechanisms for DCIP are proposed, which strive after dynamic context management by I-CPSs. First of all, a framework for reasoning about contexts from different levels of abstraction was proposed. The framework considers a logical/semantic progression in the interpretation of a time-varying process from (i) states, through (ii) events and (iii) situations, to (iv) scenes. It was investigated that dynamic context should be represented by a set of situations that are relevant to people. Accordingly, the specific objective of building awareness in dynamic context is to generate semantic knowledge about situations and the interplays among the situations in real-time.

To facilitate the implementation of the proposed concepts and operations, a sophisticated data structure called CIR-cube was designed. The CIR-cube captures not only spatial and attributive relationships among entities, but also the relationships between the changes in the spatial and attributive domain and in the temporal domain. The CIR-cube is a dynamic construct by nature, which supports the dynamic representation of data with regard to entities and their relations according to various computational requirements (e.g. sampling time, allowed computational time). Based on the CIR-cube, situations can be inferred out by using articulated reusable rules, which integrate the contents of the CIR-cube with predefined conditions.

The inferring mechanism also includes an evaluation algorithm that elaborates the impact of situations on involved entities, which is a quantitative approach for judging the relevance of situations to entities. In this way, the dynamic context of an entity is represented by integration of relevant situations to the entity over a given period of time. The proposed ‘impact indicator’ and ‘relevance indicator’ enable real-time assessment of the situations that are relevant to people. In this way, the most relevant situation to individual people can be selected and the content of personalized messages can be determined based on the descriptive information of the selected situation. Another important consideration is that the proposed knowledge inferring mechanism is able to calculate the personal danger level of individual people. This enables I-CPSs to provide personalized services according to the dangerousness of people in emergency situations.

The proposed action-plan devising mechanism generates both long-term solutions and short-term actions for people to perform, which achieves certain objectives for dynamic context management. The generation of alternative personal action-plans considers the attributes of people, while the selection of the best action-plan depends on the impact of the predicted situations in the near future. In this way, the action-plan devising mechanism supports proactive optimizing situations that are about to happen. Management of dynamic context has been realized.

In the personalized messaging mechanism, context-dependent informative and instructive messages are considered to be given to involved people. Because we assume that an integration of both messages could increase the quality of informing. To achieve this, dynamically changing situations of people are dealt with. On the one hand, the assessed personal danger level of individual people was considered to choose a proper template for message construction. On the other hand, (i) the SAT information with regard to the most relevant situation of people and (ii) the devised personal action plans were used to generate various message components. Therefore, context-sensitive messages have been automatically generated. The functionalities specified for the DCIP-mechanism are consistent with the technical requirements presented in Chapter 2.

For the purpose of testing the proposed computational mechanisms, algorithms for realizing the presented functions are specified in the case of an indoor fire evacuation guiding application. In total, 45 required algorithms have been identified out. It can be seen that some of the algorithms are very simple, and some existing solutions can be directly used (e.g. ranking of people and situations). Some algorithms can be realized through adaptations from existing solutions (e.g. generating optimal path among predefined landmarks). The rest of the algorithms are considered as novel algorithms, which should be designed by ourselves (e.g. generating a penalty matrix to represent the impact of situations).

A reusability orientated classification of the algorithms is shown above in Table 3.7. In the next chapter, both the algorithms derived by source adaptation and the generated novel algorithms will be detailed from an implementation perspective.

**Table 3.7.** Classification of the algorithms

Categories	Required algorithms for the DCIP-M
Algorithms that are available in the literature	A2, A3, A5, A7, A8, A20, A21, A22, A23, A26, A29, A33, A39, A40, A41, A42, A43, A44, A45
Algorithms that need adaptations based on existing ones	A1, A4, A6, A11, A15, A16, A17, A18, A19, A31, A32, A35, A36, A37
Algorithms to be designed	A9, A10, A12, A13, A14, A24, A25, A27, A28, A30, A34, A38

### 3.8. References

- [1] Wan, J., Zhang, D., Zhao, S., Yang, L. T., & Lloret, J. (2014). Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. *IEEE Communications Magazine*, 52(8), pp. 106-113.

- [2] Feng, Y. H., Teng, T. H., & Tan, A. H. (2009). Modelling situation awareness for context-aware decision support. *Expert Systems with Applications*, 36(1), pp. 455-463.
- [3] Armand, A., Filliat, D., & Ibañez-Guzman, J. (2014). Ontology-based context awareness for driving assistance systems. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp.227-233.
- [4] Zhang, H.J., & Chen, X.J. (2014). A dynamic key context information based scenes recognition method. In: *Proceedings of the International Conference on Wireless Communication and Sensor Network*, pp. 307-311.
- [5] Dutoit, D., & Poibeau, T. (2002). Inferring knowledge from a large semantic network. In *Actes de la conférence Computational Linguistics. Association for Computational Linguistics*, pp. 232-238.
- [6] Blanco, E., & Vempala, A. (2015). Inferring temporally-anchored spatial knowledge from semantic roles. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 452-461.
- [7] De Marneffe, M. C., Grenager, T., MacCartney, B., Cer, D., Ramage, D., Kiddon, C., & Manning, C. D. (2007). Aligning semantic graphs for textual inference and machine reading. In: *Proceedings of the AAAI Spring Symposium*, pp. 468-476.
- [8] Ackoff, R.L. (1989). From data to wisdom. *Journal of Applied Systems Analysis*, 16(1), pp. 3-9.
- [9] Jennex, M.E. (2019). Using a revised knowledge pyramid to redefine knowledge management strategy. In: *Effective Knowledge Management Systems in Modern Society*. IGI Global. pp. 1-18.
- [10] Frické, M. (2009). The knowledge pyramid: A critique of the DIKW hierarchy. *Journal of Information Science*, 35(2), pp. 131-142.
- [11] Mautz, R. (2012). Indoor positioning technologies, Habilitation Thesis, Schweizerische Geodätische Kommission, pp. 1-127.
- [12] Padovitz, A., Zaslavsky, A., & Loke, S. W. (2006). A unifying model for representing and reasoning about context under uncertainty. In: *11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 1983-1989.
- [13] Cao, M., Sun, X., & Zhuge, H. (2018). The contribution of cause-effect link to representing the core of scientific paper-The role of Semantic Link Network. *PLoS One*, 13(6), e0199303.
- [14] Henriksen, K., & Indulska, J. (2004). Modelling and using imperfect

context information. In *PerCom Workshops*, pp. 33-37.

- [15] Shmoys, D. B. (1985). The traveling salesman problem: a guided tour of combinatorial optimization. E. L. Lawler, J. K. Lenstra, & A. R. Kan (Eds.). New York: Wiley, 3, pp. 1-463.
- [16] Rego, C., Gamboa, D., Glover, F., & Osterman, C. (2011). Traveling salesman problem heuristics: Leading methods, implementations and latest advances. *European Journal of Operational Research*, 211(3), pp. 427-441.





# Chapter 4

## **Implementation of a prototype computational mechanism, and testing its functionality and performance**

### **4.1. Objectives and methodological approach of research cycle 3**

#### **4.1.1. Objectives and activities**

The objective of this research cycle was to implement a prototype, which can be used as a basis to test the proposed concepts as well as their implementation as algorithms and data constructs. To this end, the work was dedicated to the functional and architectural implementation of the components (modules) of the DCIP-M and the various algorithms in executable forms. The functions and algorithms identified in the previous chapter were realized based on standard system engineering methodology, that is, the V-model. Integration of the components into modules and integration of the modules into a system is the next procedural step. Integration was done from: (i) structural, (ii) procedural, (iii) informational, and (iv) temporal (timing) aspects. In the case of software means and algorithms, a common element of the structural, procedural and informational integration is harmonizing of the input and output quantities (variables and values). After that, the possibility of continuous information processing is provided, including the properness of external inputs and the timing of the execution.

Then, the four constituents of the implemented prototype were tested individually in themselves and together in a specific application case. The objective of the tests was to validate the functionality of the implemented modules and the prototype as a whole. A simulation of a real-life scenario was implemented, which was used (i) to generate time-dependent contextual data for computations and (ii) to respond to the personalized messages for realizing dynamic context management. An

assumed indoor fire evacuation scenario was designed and the behaviors of people in the scenario were specified. Based on the simulated scenario, the feasibility of the implemented algorithms and data constructs was shown. In the simulative studies, scenarios generated based on the assumed real-life application was used.

In addition, this research cycle also includes a performance testing. The goal of the performance testing is to show if the overall performance of the implemented prototype is sufficient or not in a challenging target application case. Towards this end, proper profiling of computational performance with regard to the DCIP-M was carried out. Various performance descriptors are to be used. The performance of computation was quantitatively evaluated concerning computational time spent by the implemented functions and algorithms under different kinds of loads.

#### **4.1.2. Methodology applied in this research cycle**

Like the preceding RC2, this research cycle also included three phases: (i) an explorative phase, (ii) a constructive phase, and (iii) a confirmative phase. Therefore, it was completed following the concepts of the design inclusive research, as a methodological framing. The specific tasks completed in the explorative phase were: (i) reviewing the existing solutions that enable implementing some basic data structures and algorithms in the mechanism, e.g. representing layout of a building-block, path planning algorithms, (ii) collecting information about the fundamentals that support designing innovative algorithms, e.g. calculating impact indicators of situations and relevance indicators of situations to individual entities and (iii) exploring logical and computational techniques concerning the prototype level implementation to the reference application case, e.g. inferring entities jams. In addition, the constructive phase targeted the software level implementation of all modules and algorithms of the DCIP-M. The designed algorithms in Chapter 3, used data construct and working sequence of the functions in the prototype were specified. In the confirmative phase: (i) the feasibility of the implemented prototype mechanism was validated through operationalizing it for a concrete application case, and (ii) the computational performance of the prototype was evaluated. The results of these confirmative research actions provided the basis for assessing the applicability of the concepts and algorithms in multiple application cases.

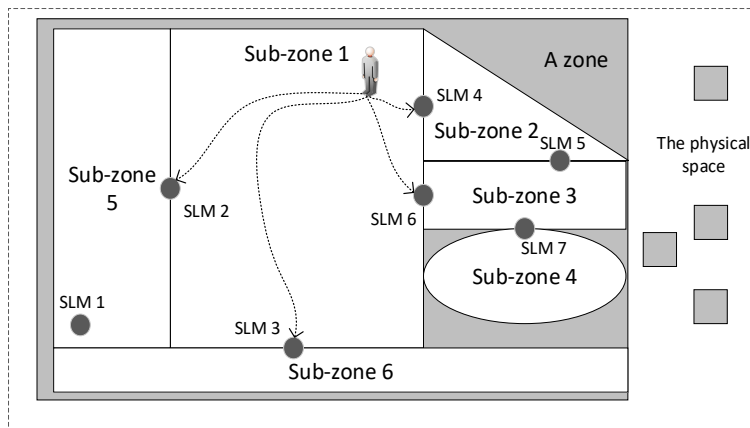
### **4.2. Fundamentals for the implementation**

#### **4.2.1. Digital representation of a concerned physical space**

Digital representation of a concerned physical space aims to capture relevant information about the spatial relationship between two entities (i.e. distances) in computation. It is a projection of the continuous physical space to a discrete

logical space. This digital (computational) representation serves as the basis for computing spatial relations among entities. To digitalize a physical layout, there are many principles and approaches available in the literature. A commonly used approach is graph theory [1]. In this way, information about the layout of a physical space can be represented by a graph-like model. The layout of a physical space is represented by a finite number of vertices (also named as nodes) in a topological space and a number of edges (also named as arcs) are used to link the vertices. The data assigned to the edges capture the spatial relations, e.g. distance, direction, and connectivity among the vertices [2]. Compared to map-like representations, such as the approach applied in [3], graph representation is more flexible and extendable [4]. Another approach is the matrix of orientation and adjacency and characteristic (MOAC) model reported in [5]. This model is normally applied to traffic management applications, as exemplified in [6]. In this work, we applied the graph theoretical diagramming to digitally represent the layout of a given physical space. In the following part, the technical details are introduced.

The entire physical space is divided into finite number zones. A zone is defined as accessible areas that a concerned entity is able to move from any location of the zone to another location on a completely inclusive path. The zone can be either 2-dimensional or 3-dimensional. The boundary of a zone is a closed path which is formed by non-intersecting line segments joint in a pair-wise way. For instance, a building can be a typical zone. In our study, we consider 2D zones only. A zone can be further divided into a finite number of non-overlapping sub-zones, e.g. a room in a building. The principle to divide a sub-zone from a zone is similar to the division of a zone from the physical space. The only difference is that the outline shape of a sub-zone should be a convex polygon. It means that the shortest path (linear path) between any two locations of a sub-zone remains inside of the



**Figure 4.1.** Designation of zones, sub-zones, and SLMs from the physical space

sub-zone. Thus, the associated entity is able to move from a position in a sub-zone to any other position through a continuous path without entering other sub-zones. An example is shown in Figure 4.1. A floor of a building can be considered as a zone of the physical space, while the floor is divided into several 2D sub-zones.

The boundary of a sub-zone,  $B_{sz}$ , is represented as an ordered set of locations:

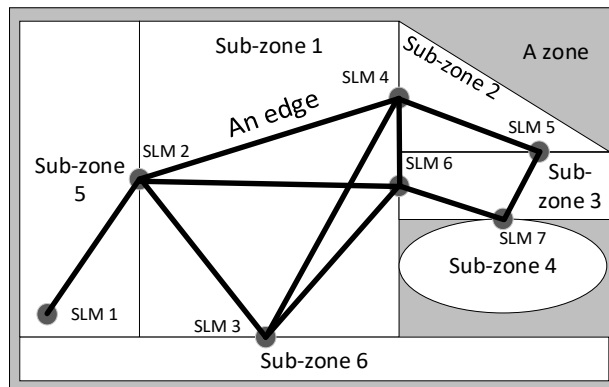
$$B_{sz} = \{P_1, P_2, \dots, P_n\} \quad (4-1)$$

where:  $P_i$  refers to the location of a vertex of the sub-zone. The locations in the set refer to the vertices of the subzones, which are arranged according to the clockwise order. The layout of a zone can be noted as the set including the boundaries of all sub-zones in it.

$$\Psi_{zone} = \{B_{sz_1}, B_{sz_2}, \dots, B_{sz_m}\} \quad (4-2)$$

Spatial landmarks (SLMs) are positions designated in the sub-zones. Two types of SLMs have been identified: Type-1 SLM is designated at the common border of two sub-zones and is used to represent the connectivity of the two sub-zones. For instance, an SLM designated at the position of a door of a room represents that the room (regarded as a sub-zone) and the corridor (regarded as another sub-zone) are connected since a person is able to enter the room through the door. Type-2 SLM is designated inside of a sub-zone and is used to represent an important position within the sub-zone, e.g. a position where a fire extinguisher is placed in that room. As shown in Figure 4.1, SLM 1 is of Type-2, while the other six SLMs are of Type-1.

For every sub-zone, at least one Type-1 SLM should be designated. It is used to specify the spatial location (e.g. a door) where an associated entity could enter into the sub-zone from another adjacent sub-zone. According to the principles of generating sub-zones, an entity is able to move from any position of a sub-zone to a designated SLM associated with the sub-zone through a continuous path without entering the other sub-zones, as shown in Figure 4.2. SLMs can be either static positions that permanently designated in the sub-zones or dynamically created/changed/removed from the sub-zones. For instance, in case of indoor fire, if an exit of the building is occupied by the fire or locked, the



**Figure 4.2.** Connecting the SLMs with edges

SLM represented for the exit should be removed from the sub-zones. It means that people cannot escape from the building through the exit.

The edges connecting the adjacent SLMs represent the spatial relationships of the sub-zones. Two sub-zones are disconnected if there is no (or cannot be) edge defined between them. For instance, as shown in Figure 4.2, although the sub-zone 4 and sub-zone 1 is next to each other, there is no SLM designated at the common border of them. Therefore, a person in sub-zone 1 has to enter sub-zone 3 before he/she may enter into sub-zone 4. In our research, we assumed that a zone belonging to SLMs is fully connected. It means that any SLM designated in the zone is linked with another SLM through a finite number of edges. In addition, there is no isolated SLM or sub-zone in the considered zone.

In graph theory and computer science, an adjacency matrix is a square matrix used to represent the relationship of finite vertices [2]. The adjacency matrix is symmetric due to the reason that the edge between two connected SLMs is non-directional. We used a weighted adjacency matrix/ distance matrix ( $DM$ ) to represent the relationships among the SLMs and the edges in between. The weights of edges were considered as the spatial distance between the connected SLMs. Therefore, based on the above-mentioned concepts, the layout information of a physical space can be represented digitally by the following triplet:

$$LI_{ps} = \{\Psi_{zone}, P_{SLM}, DM\} \quad (4-3)$$

where:  $\Psi_{zone}$  is a set of boundaries of sub-zones,  $P_{SLM}$  is the positions of SLMs designated in the sub-zones, and  $DM$  is the distance matrix.

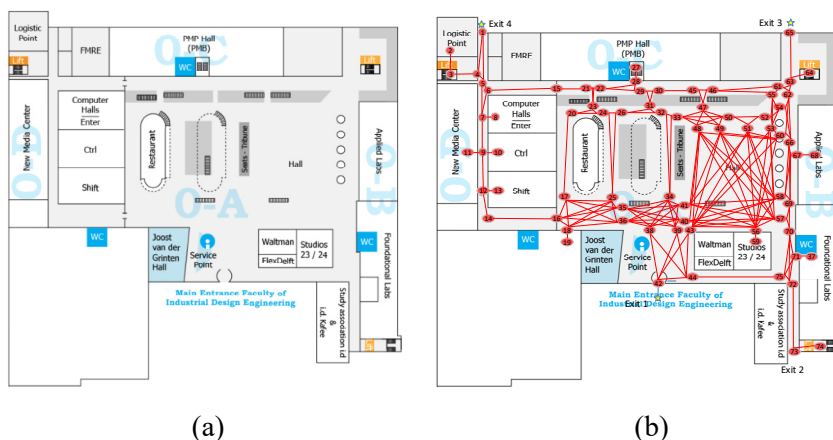
#### 4.2.2. Implementation of a computational simulation of fire evacuation scenarios

Implementation of the prototype of the computational mechanism involves two sets of activities: (i) implementation of the necessary computational algorithms with a view to the possible range of applications, and (ii) integration of the algorithms according to the information flows of a set of application cases. The implementation of the computational algorithms for simulation of indoor fire evacuation scenarios considered two major functionalities: (i) generating descriptive, time-dependent context data for processing, and (ii) representing the generated computational results visually. In general, the simulation considers time-varying scenarios within a given physical space, which is digitally represented according to the principles proposed in the previous sub-section. The motion with regard to three types of entities, including people, exits and fire, in the space were simulated. In addition, the behavior of entities is defined by several predefined rules and their reaction to their individual context. Based on this, the generated time-dependent data with regard to the entities were used as

the input of the implemented prototype, while the output of the prototype (i.e., personal action plans) was executed by the simulated people (e.g., speed up).

The simulated real-life indoor fire evacuation scenario was implemented in MatLab®. The ground floor of the Building IDE of TU Delft was digitally modeled by a 2D space (Figure 4.3.a). Its size was 130m\*100m (width\*height). The concerned space was considered as a single zone that was divided into 46 sub-zones, 74 SLMs and 114 spatial relations as shown in Figure 4.3.b. In addition, as shown in Figure 4.4.a, the initialized situation involved 80 people (represented by circles and diamonds), four exits (represented by solid cubes), and the location of a starting fire (represented by the hollow yellow cube). Represented by diamonds, 40 people are able to receive personalized messages provided by the system (i.e., they are informable people), while the people represented by circles cannot be informed by the system individually (i.e., they are uninformable people). Among the informable people, thirty ones will completely do whatever is recommended in the given messages, whereas ten people will not obey the given instructions and make decisions based on their own judgments.

It was assumed that the fire alarm is triggered when a fire is detected in the space. 70 people were considered to move towards the nearest exit with constant, but random speeds of motion (distributed normally, characterized by the mean value 0.75m/s, and the standard deviation 0.1). It had to be also assumed that some of them will not respond to the fire alarm due to physiological reasons (e.g. may be deaf or otherwise handicapped) or due to being heavily submerged in activities (e.g. listening music). Therefore, it was set in the simulation that 10 people neglect the fire alarm and stay around until the fire front reached them. In addition,



**Figure 4.3.** Representation of the concerned space layout: (a) the ground floor of Faculty IDE of TU Delft, (b) the specified spatial landmarks (SLMs) and edges among the SLMs

every person was characterized by a body volume that was considered as an obstruction of the movement of another person who intended to pass through. Another consideration in simulation was that if an object appears in front of a person within 3 meters, then the person may decide (i) to round the obstruction if the obstruction is motionless, (ii) to decrease the speed of motion if the obstruction is moving towards the same direction as the person does and (iii) to move backward with 50% speed if the obstruction is moving towards the person (to avoid any possible collision). Furthermore, it was assumed that if any person approaches the fire front closer than a threshold distance (10 meters), then they will change their paths and will move towards another randomly selected exit.

To simulate the proliferation of fire, the entire physical space meshed into  $120 \times 90$  cells. The fire was generated based on the cellular automata mechanism. Every second, the fire proliferates from a cell catching fire to its neighboring cells (8 cells) with a predefined probability, which was 12.5%. In addition, the maximum number of people that could pass through an exit per second was set to 1. This was a basis for the generation of people jams in front of exits. When the simulation started the fire appears. It was assumed that the fire was detected by the system at  $t=10s$  when the fire alarm works. The results of simulated scenarios without DCIP and personalized informing at  $t=50s$ ,  $t=100s$ , and  $t=200s$  are shown in Figure 4.4.b, Figure 4.4.c and Figure 4.4.d, respectively. Due to the reason that personalized informing was not considered in the simulation, informable people and uninformable people were not treated differently.

It can be observed from the simulation results that (i) people get involved in dynamically formed people jams during the escape process, and (ii) people may move towards fire due to the inappropriate informing method supplied (i.e. the fire alarm does not provide personalized guidance). As demonstrated by the fore-running simulation, dangerous situations have a great impact on the evacuation of people. This simulated scenario was considered as a test bed for the generation of time-dependent data for DCIP.

In addition, an indoor fire evacuation guiding system was simulated assuming that the generated locations of entities can be aggregated by deployed sensors. Indoor environment sensing often contains notable errors. Therefore, to represent a normal distribution of the errors, white noise with a normal distribution  $\varepsilon \sim N(\mu, \sigma^2)$ , where  $\sigma = 0.3$  and  $\mu = 0$ , were added to the location ( $x$  and  $y$  coordinates) of every person at each sampling time. The result is used in the computation as the actual location of the people. At  $t = 21 s$ , the computation of the DCIP-M is controlled to run. The generated position data was used as the input of the DCIP-M. As an outcome of the DCIP-M, personalized messages should be generated and provided to the people in order to manage the above-mentioned situations.

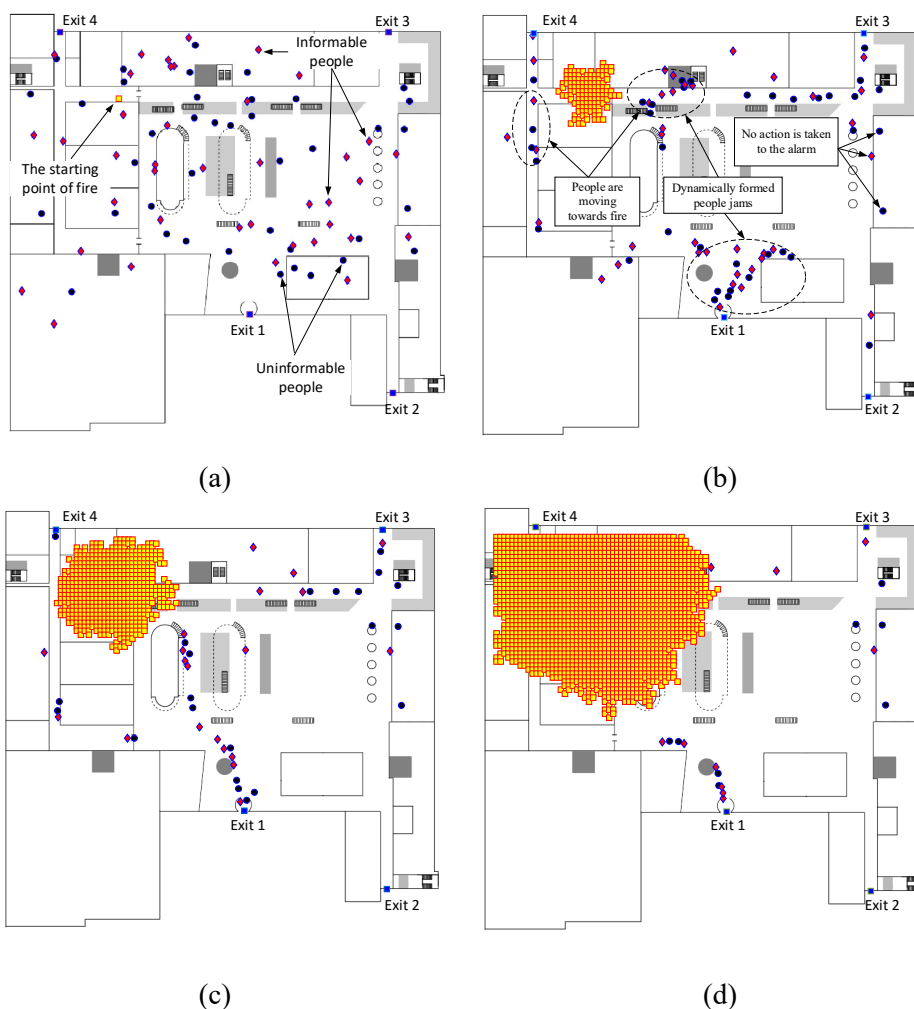


### 4.3. Implementation of the modules

The DCIP-M includes 45 interoperating algorithms. Some of these algorithms are standard (published), whereas others are novel and proprietary. Considering the space limitation, we will discuss only the so-called kernel algorithms, which are novel and significant from the realization of the modules and the DCIP-M as a whole. This is the reason why the identification numbers assigned to the particular algorithms show a discontinuity.

#### 4.3.1. The information representation module

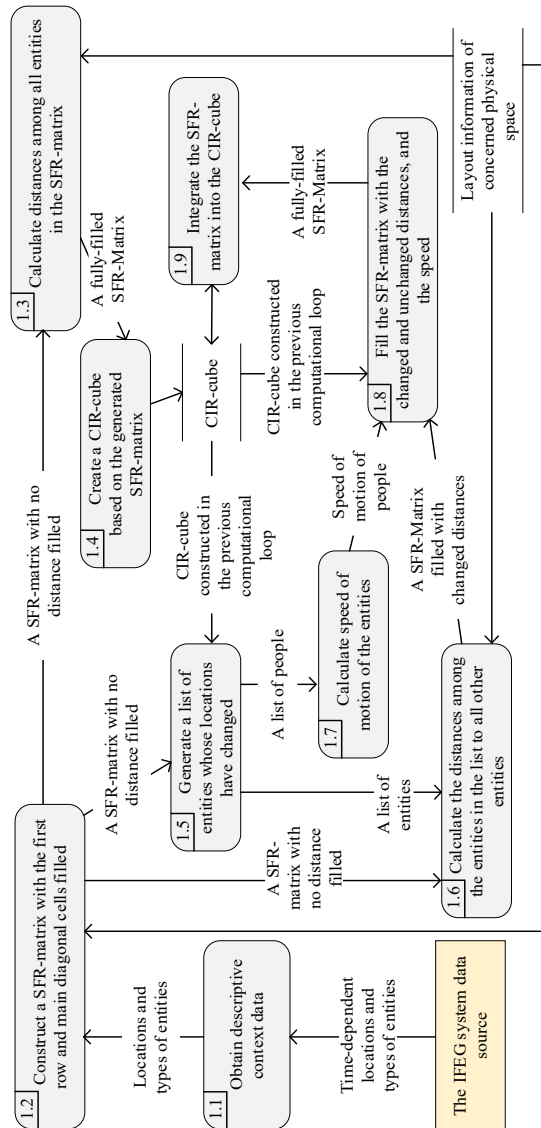
The information representation module includes 9 functions. Various control



**Figure 4.4.** The results of the computational simulations of an indoor fire evacuation scenario: (a) at  $t = 0$  s, (b) at  $t = 50$  s, (c) at  $t = 100$  s, and (d) at  $t = 200$  s

operations were implemented among the interoperating algorithms. An overview of the IRM can be indicated by the data flow diagram in Figure 4.5. It presents how the implemented functions are integrated based on their specified inputs and outputs. The layout information of a physical space is the static context information which is predefined since we assume that the layout of a building will not change during the computations.

As the first step of the computational processing, the time-dependent descriptive data of the entities considered in the reference application case were acquired. Three types of entities were considered: people,



**Figure 4.5.** Data flow diagram of the information representation module

fire (fronts and corpus) and exits. In the prototype, the computation started assuming that the location data of entities at a given point in time are fully aggregated. Thus, every set of data describes the types and locations of the entities at a given point in time, e.g., at  $t = t_s$ . The type of people is referred to as an attribute concerning if the person is able to receive messages from the system, e.g. an informing terminal, or not. The type of exit represents if the exit is

available for people to pass through or not.

To obtain descriptive context data, the following principles were applied to estimate the locations of people: From  $t = t_s$  to  $t = t_s + 3\Delta t$ , the locations of people were considered as the values delivered from the IFEG system (with noise). After  $t = t_s + 3\Delta t$ , the locations were calculated by using a Kalman-filter, which was specified as follows:

$$\hat{u}_k^- = \mathbf{F}\hat{u}_{k-1} \quad (4-4)$$

$$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q} \quad (4-5)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (4-6)$$

$$\hat{u}_k = \hat{u}_k^- + \mathbf{K}_k (z_k - \mathbf{H}\hat{u}_k^-) \quad (4-7)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- \quad (4-8)$$

where:  $\hat{u}_{k-1}$  is the estimated positions of people at time  $k-1$ ,  $\hat{u}_k^-$  is the predicted positions of time  $k$ ,  $\hat{u}_k$  is the estimated positions at time  $k$ ,  $\mathbf{F}$  is the state transition matrix,  $\mathbf{P}_{k-1}$  is the error covariance matrix at time  $k-1$ ,  $\mathbf{P}_k^-$  is the predicted error covariance of time  $k$ ,  $\mathbf{P}_k$  is the error covariance matrix at time  $k$ ,  $\mathbf{Q}$  is the covariance of the process noise,  $\mathbf{R}$  is the covariance of the observation noise,  $\mathbf{H}$  is the observation matrix.  $z_k$  is the observations at time  $k$ ,  $\mathbf{K}_k$  is the Kalman-filter gain.

In the implementation,  $u_k$  was specified as a quadruple:  $\{x_k, y_k, \Delta x_k, \Delta y_k\}$ , where:  $x_k$  and  $y_k$  are the  $x$  and  $y$  coordinates of a person, respectively, and  $\Delta x_k$  and  $\Delta y_k$  are the increments of these values concerning two successive computations.  $\mathbf{F}$  was specified as:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-9)$$

$\mathbf{H} = \mathbf{I}(4)$ ;  $\mathbf{R} = d_r^2 * \mathbf{I}(4)$ ;  $\mathbf{Q} = q_0 * \mathbf{I}(4)$ . The value of  $d_r$  was set as 0.3 (according to the mean value of the simulated white noise). Above,  $q_0$  is the coefficient of the noise covariance matrix, which should be optimally estimated before the Kalman-filter is properly applied.

Then, the sub-zone of entities was determined based on the locations of entities and the boundary information of all considered sub-zones,  $\Psi_{zone}$ . This is done by checking if an entity is located in any of the polygons shaped by the boundary of the sub-zones or not. An SFR-matrix of the size of  $(N+1)*(N+1)$ , where  $N$  is the

total number of entities, is created. The number of exits and people were calculated, while fire is considered as one entity occupying multiple locations. Then, identifications of entities are filled into the first row and the first column of the SFR-matrix. The content profile created for every person contains (i) identification, (ii) current location, (iii) type and, (iv) current sub-zone. The content profile of exits includes several aspects concerning (i) identification, (ii) referred SLM, (iii) type and (iv) status. The content profile for fire describes (i) identification, (ii) locations catching fire and (iii) sub-zones that catch fire and their neighboring sub-zones. The newly created content profile is linked to the SFR-matrix, according to the identification of the entity.

To calculate the distances between every two entities listed in the SFR-matrix, Algorithm 4 was implemented, which makes use of the Floyd-Warshall algorithm, to find the shortest paths between any two SLMs in the generated graph model [7]. The distances between two people, or a person and an exit were considered as the length of the nearest path followed by a person to meet another one. The distance

---

**Algorithm 4.** Calculate the distance between two entities

---

**Inputs:**  $loc_{e_1}$  : indoor location of an entity  $e_1$   
 $loc_{e_2}$  : indoor location of another entity  $e_2$   
 $id_{sz_{e_1}}$  : the ID of the sub-zone of  $e_1$   
 $id_{sz_{e_2}}$  : the ID of the sub-zone of  $e_2$   
 $Ll_{ps}$  : layout information of the concerned space  
**Outputs:**  $dis$  : the calculated distance

---

```

1: if ( $id_{sz_{e_1}} == id_{sz_{e_2}}$ ) do
2:    $dis \leftarrow$  calculate the linear distance between  $loc_{e_1}$  and  $loc_{e_2}$ 
3: else do
4:    $slm_{id_{sz_{e_1}}} \leftarrow$  determine the SLMs that belongs to  $id_{sz_{e_1}}$ 
5:    $slm_{id_{sz_{e_2}}} \leftarrow$  determine the SLMs that belongs to  $id_{sz_{e_2}}$ 
6:   for  $i \leftarrow 1, size(slm_{id_{sz_{e_1}}})$  do
7:     for  $j \leftarrow 1, size(slm_{id_{sz_{e_2}}})$  do
8:        $dis_{e1\_slm} \leftarrow$  calculate the linear distance between  $loc_{e_1}$  and  $slm_{id_{sz_{e_1}}}(i)$ 
9:        $dis_{e2\_slm} \leftarrow$  calculate the linear distance between  $loc_{e_2}$  and  $slm_{id_{sz_{e_2}}}(j)$ 
10:       $dis_{ss} \leftarrow$  the nearest distance between  $slm_{id_{sz_{e_1}}}(i)$  and  $slm_{id_{sz_{e_2}}}(j)$ 
11:       $a\_dis(i, j) \leftarrow dis_{e1\_slm} + dis_{e2\_slm} + dis_{ss}$ 
12:     end for
13:   end for
14:    $dis \leftarrow \min(a\_dis)$ 
15: end if
16: return  $dis$ 

```

---

between the fire and a person was considered as the value of the linear distance between the nearest fire front and the location of the person.

The determination of the changed locations of entities contain three steps: (i) loading the SFR-matrix generated in the previous computational loop, (ii) comparing the locations of entities in the previously generated SFR-matrix and the newly created SFR-matrix, and (iii) generating a list containing the identification of entities whose location change is bigger than a threshold (specified as 0.3 meter in the prototype). The unchanged distances in the SFR-matrix were taken over from the previously generated SFR-matrix. The difference between two successive estimated locations of entities was considered as the speed of motion of people. After a CIR-cube is constructed, the changes with regard to the attributes and locations of entities and distances can be known.

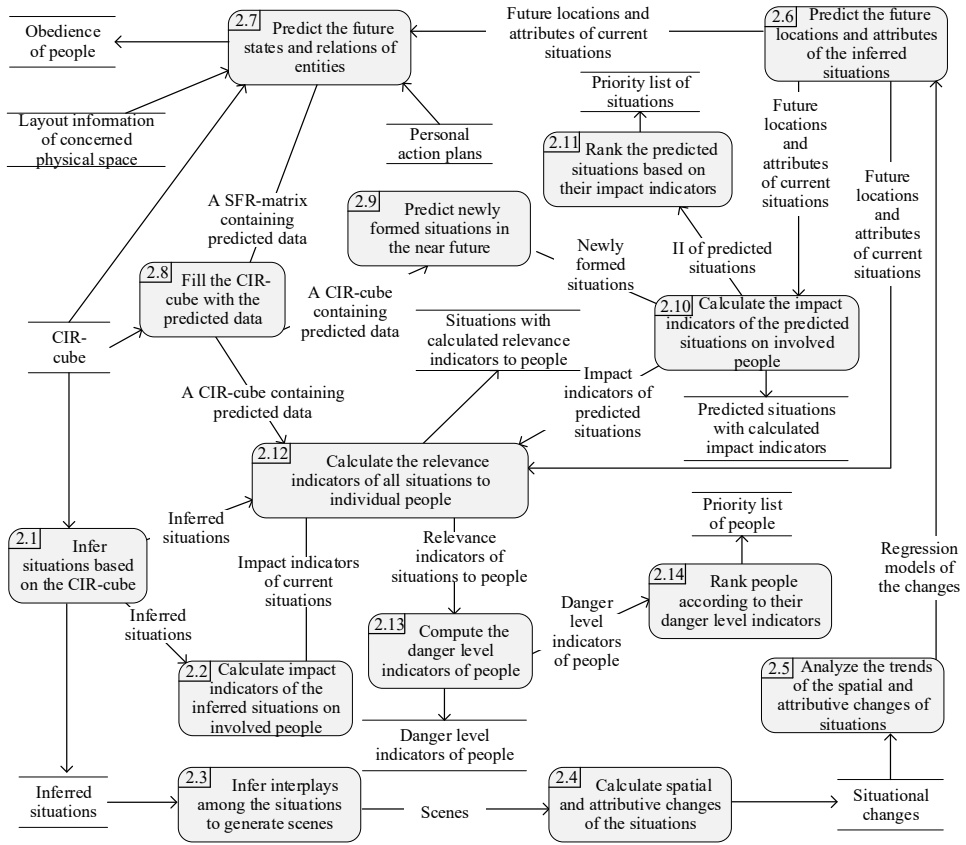
### 4.3.2. The context knowledge inferring module

Using the constructed CIR-cube, semantic knowledge about dynamic context can be inferred out. As shown in Figure 4.6, the data flow diagram of the KIM is derived from the general workflow. Inputs of the KIM are (i) CIR-cube, (ii) layout information of concerned physical space, and (iii) personal action plans, which are used to calculate the obedience of people.

For the IFEG system, three algorithms were developed for inferring three types of situations based on a constructed CIR-cube. The details of Algorithm 9 (infer floating people jams) is shown below. In the algorithm, the situation inferring is achieved by using the defined operators indicating the content of the CIR-cube and some rules. The fourth input parameter of the  $CIRC(i, j, t, T)$ , which indicates the time when the CIR-cube is constructed, was not used in reasoning. This is just a technical simplification, rather than a theoretical limitation. Since Algorithm 10 (infer anchored people jams) is very similar to Algorithm 9, it will not be separately discussed. In Algorithm 9 and 10, two basic units of people jams could be combined together to form a bigger people jam whenever these two basic units contain the same person or are closer than 3 meters.

In addition, the representation of a people jam captures: (i) the type of the situation, (ii) the entities involved in the situation, (iii) the time when the situation happens, and (iv) the location of the situation, which is calculated as the average locations of involved entities. The fire is considered as a situation with only one entity. The location of the fire is computed as the geometric center of the locations having a fire.

The impact indicator (II) of the inferred situations on involved entities was calculated. The II of fire was set to -1, while the II of a people jam was calculated by the following equation:



**Figure 4.6.** Data flow diagram of the context knowledge inferring module

$$II_{pj} = -0.02 * n^{R_v} \quad (4-10)$$

where:  $n$  is the number of people in a people jam,  $R_v$  is the ratio of speed reduction of people involved in the people jam, which is calculated by using  $R_v = 1 - \bar{v}_p / \bar{v}_m$ .  $\bar{v}_p$  is the average speed of motion of people in the jam at the time of calculation and  $\bar{v}_m$  is the maximum speed of motion that has been achieved by the involved people. Equation (4-10) was a specification of Equation (3-1). The exponential computation was designed according to our experience. Namely, the more people in a people jam are, the higher the importance of the people jam will be. In addition, the coefficient (i.e., -0.02) was given since we assume that a people jam with 50 people and the speed of motion of the people jam is 0 will lead to an extremely dangerous situation ( $II_{pj} = -1$ ).

When situations happening at two successive computational cycles are inferred, interplays of the inferred situations can be carried out. According to the principle for inferring different types of interplays, which are presented in Chapter 3, the

**Algorithm 9.** Infer floating jams of people

---

**Inputs:** *CIRC* : the generated CIR – cube  
**Outputs:** *inf\_situ* : the inferred floating people jams

---

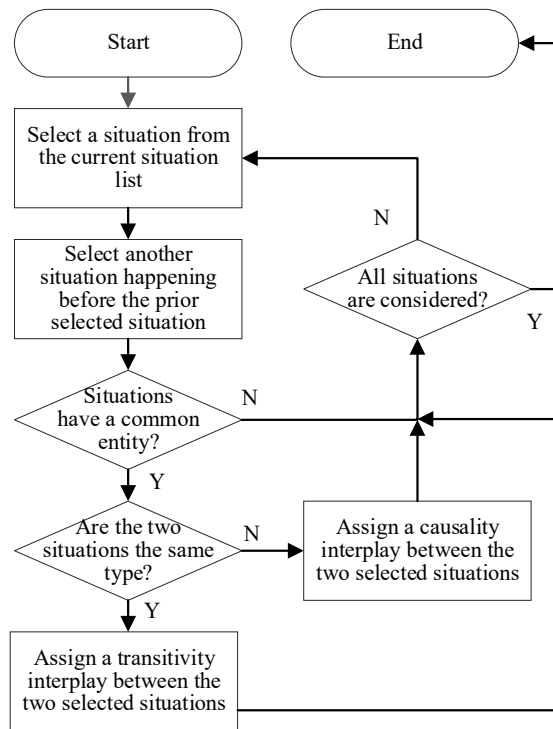
```

1:  s_cube ← size(CIRC)
2:  c_ss ← 0
3:    for i ← 2, s_cube(1) do // s_cube(1) – 1 is the total number of entities
4:      for j ← 2, i – 1 do
5:        if (CIRC{i,i,end}.type=='people')&&(CIRC{j,j,end}.type=='people')
          && (CIRC{i,j,end} – CIRC{i,j,end-1}<1) &&
          (CIRC{i,j,end} – CIRC{i,j,end-2}<1.5) &&
          (CIRC{i,j,end}<3) && (CIRC{i,i,end}.speed>0.1) &&
          (CIRC{j,j,end}.speed>0.1) do
6:          c_ss ← c_ss+1
7:          create a situation, inf_situ (c_ss)
8:        end if
9:      end for
10:    end for
11:    loop_temp ← 1
12:    int_temp ← 0
13:    while (loop_temp==1) do
14:      ext_temp ← 0
15:      s_s ← size(inf_situ)
16:      for i ← 1, s_s(1) do
17:        if int_temp == 1 do
18:          ext_temp ← 1
19:          int_temp ← 0
20:          break
21:        end if
22:        for j ← i+1, s_s(1) do
23:          if inf_situ(i) and inf_situ(j) contain a same entity do
24:            iterate inf_situ(i) and inf_situ(j) into inf_situ (s_s(1)+1)
25:            remove inf_situ(i) and inf_situ(j) from inf_situ
26:            int_temp ← 1
27:          break
28:          end if
29:        end for
30:      end for
31:      if ext_temp ==0 do
32:        loop_temp ← 0
33:      end if
34:    end while
35:    return inf_situ

```

---

workflow of inferring scenes is shown in Figure 4.7. It can be seen that situations inferred in the previous computational cycle were loaded for the computation. Situational changes were calculated based on the inferred transitivity interplays. To demonstrate on the situational changes, three aspects were implemented, including: (i) location changes of people jams, (ii) speed changes of people jams, and (iii) location changes of fire situations. Particularly, since the location and speed of people jams were considered as the average locations and speeds of people involved in the jam, the changes were computed based on the difference of the locations and speeds. Since fire proliferates in the space, the location change of fire was taken into account as the difference of the coverage of two fire situations.



**Figure 4.7.** Workflow diagram for inferring interplays among situations to generate scenes

The trend of changes with regard to people jams is represented by second-order regression models. The values of the run-time generated models are computed using the ordinary least squares method that estimates the parameters based on a minimum of the sum of squared residuals. The location changes of fire are represented by an incremental model, showing how the fire proliferates in all directions. Then, the generated regression models for location and speed changes and the incremental model for coverage change of fire are applied for predicting the future location and speed of motion of people jams.

In the above-mentioned prediction procedure, only one computational step in the near-future was taken into consideration. The prediction of the near-future locations of the entities is done by Algorithm 24, which is specified below. The algorithm predicts the future location of the entities by considering (i) their involvement in people jams, which influences their speed of motion, and (ii) the



**Algorithm 24.** Predict near-future locations of people

---

**Inputs:**  $Ll_{ps}$  : layout information of the concerned space  
 $CIRC$  : the generated CIR-cube at  $t$   
 $p\_sp_{jam}$  : predicted speed of motion of jams  
 $inf\_pj$  : inferred people jams

**Outputs:**  $p\_loc_e$  : the predicted location of an entity  $e$

---

```

1:  for  $i \leftarrow 2, \text{size}(CIRC, 1)$  do
2:    if ( $CIRC\{i, i, t\}.\text{type} == \text{'people'}$ ) do
3:       $temp\_i \leftarrow 0$ 
4:      for  $j \leftarrow 1, \text{size}(inf\_pj)$  do
5:        if ( $inf\_pj(j).\text{entity}$  contains  $CIRC\{i, i, t\}.\text{id}$ ) do
6:           $temp\_i \leftarrow 1$ 
7:           $t\_n \leftarrow j$ 
8:          break
9:        end if
10:     end for
11:     if ( $temp\_i == 1$ ) do %the entity is involved in a people jam
12:        $s\_p \leftarrow p\_sp_{jam}(t\_n)$ 
13:     else do // the entity is not involved in a people jam
14:        $s\_p \leftarrow CIRC\{i, i, t\}.\text{speed}$ 
15:     end if
16:      $t\_slm \leftarrow$  compute the SLM that the entity is moving towards
17:      $d\_t\_slm \leftarrow$  calculate the left distance to the target SLM
18:     if ( $d\_t\_slm > s\_p$ ) do
19:        $p\_loc_e(i) \leftarrow CIRC\{i, i, k\}.\text{loc} + s\_p$ 
20:     else do
21:        $t\_exit \leftarrow$  estimate the exit that the entity is targeting
22:        $n\_t\_slm \leftarrow$  compute the SLM that the entity will reach after  $t\_slm$ 
23:        $p\_loc_e(i) \leftarrow$  calculate the position that will be reached by the entity
24:     end if
25:   end for
26:  return  $p\_loc_e$ 

```

---

layout of the concerned space. If a person is involved in a people jam, then the speed of him/her is as the same as the predicted speed of motion of the people jam (one-step ahead), which is calculated based on the represented regression model.

In addition, to predict if a person will obey the given instructions. The following principles were considered in the implemented algorithm: (i) if there is no available personal action plan (before the first message is given), then all people who have informing terminals, e.g. smartphones, are believed as obeying people, (ii) for a person who is provided with personalized messages, the deviation between the action plan contained in the messages, which is a set of time-related positions, and the current location of the person is computed. If the deviation is bigger than a threshold, then the person is believed as a disobeying one. For the

IFEG system, the threshold was specified as 5 meters.

Having the predicted locations and attributes of entities, a new SFR-matrix was created, which can be included in the CIR-cube. The updated CIR-cube provides a basis for inferring situations happening in the near-future. Actually, the inferred situations include both newly formed situations and the future of existing situations. The latter were left out by checking if any entity in them has been considered in the existing situations. Finally, the predicted situations were ranked according to the calculated impact indicators in an ascending order, which means that the situation with the smallest negative impact indicator will be placed at the top of the list.

The calculated impact indicators were considered as a reference for calculating the relevance indicators (RIs) of all situations to people. Both historical situations and predicted situations were concerned according to Equation (3-10) proposed in Chapter 3. In the implementation, if a RI is bigger than a predefined threshold, then the referred situation is believed as relevant to the concerned person. Actually, all relevant situations of a person were sorted into a list based on the values of the calculated RIs. The indicators show the importance of the situations to the concerned person. In this way, the context of the concerned person can be handled by manipulating the important situations related to him/her. The danger level indicators of people were calculated by summing the RIs of all situations related to individual people. The calculated danger level indicators were used as a reference in ranking people into a priority list. In the implementation, if a person has a higher danger level indicator, the person will be placed to a higher position in the list.

### **4.3.3. The personal action-plan deriving module**

In general, the data flow diagram of the implemented action-plan deriving module (ADM) is shown in Figure 4.8. The input of the module includes: (i) CIR-cube, which contains attributes of entities, (ii) obedience of people, (iii) predicted situations with calculated impact indicators, (iv) layout information of the concerned physical space, (v) priority list of people and (vi) priority list of situations. The output of the module is the derived personal action plans. According to the basic concepts proposed in Chapter 3, every action plan contains information about (i) the path to be followed and (ii) the short-term action to be performed by an informable person.

To generate an escape path for individual people, a penalty matrix is constructed and its size is as the same as the constructed adjacency matrix (74\*74). Details with regard to the generation of a penalty matrix are shown in Algorithm 30. The generated penalty matrix enables the calculation of optimal paths among SLMs, which considered both the distance between any two SLMs, as well as the IIs of

**Algorithm 30.** Generate a penalty matrix to represent the impact of situations

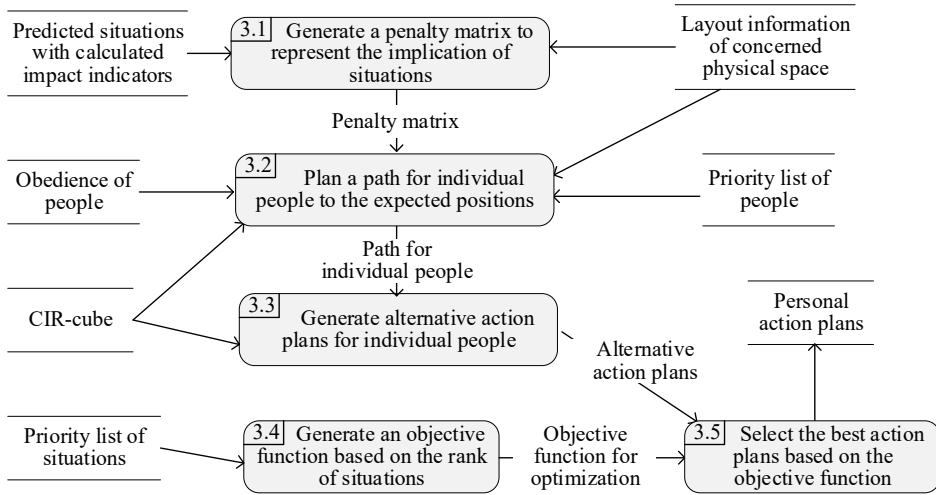
**Inputs:**  $p\_situ\_ii$  : predicted situations with calculated impact indicators  
 $LI_{ps}$  : layout information of the concerned space

**Outputs:**  $pe\_m$  : penalty matrix

```

1:  $pe\_m \leftarrow \text{zeros}(\text{size}(LI_{ps}.DM, 1), \text{size}(LI_{ps}.DM, 1))$ 
2: for  $i \leftarrow 1, \text{size}(p\_situ\_ii)$  do
3:    $situ\_sz \leftarrow \text{identify the sub-zone where the situation is happening}$ 
4: end for
5: for  $j \leftarrow 1, \text{size}(LI_{ps}.\Psi_{zone})$  do
6:   if  $(LI_{ps}.\Psi_{zone}(j))$  contains a situation do
7:      $slm\_tp \leftarrow \text{identify SLMs that belong to the sub-zone}$ 
8:      $ed\_tp \leftarrow \text{identify edges connecting any two SLMs in } slm\_tp$ 
9:      $pe\_m(slm\_1, slm\_2) \leftarrow pe\_m(slm\_1, slm\_2) + ii\_situ$ 
10:   end if
11: end for
12: return  $pe\_m$ 

```



**Figure 4.8.** Data flow diagram of the personal action-plan deriving module

situations that are happening on the path connecting the two SLMs, as shown in Algorithm 31. The optimization coefficient,  $a_{oc}$  in Algorithm 31, was specified as -500 and  $\bar{v}$  was specified as 1 m/s for the reference application. After the optimal paths among SLMs were obtained, the optimal paths of individual people to exits were computed and the details are shown in Algorithm 32. To achieve group optimal, the conditions for determining potential informants were specified as (i) people who have informing terminals and (ii) people who obey the given instructions.

**Algorithm 31.** Calculate the lengths of optimal paths among spatial landmarks

---

**Inputs:**      $pe\_m$      : penalty matrix  
                   $LI_{ps}$      : layout information of the concerned space

**Outputs:**    $slm\_r$      : optimal paths and path lengths among SLMs

---

```

1:   $c\_wm \leftarrow LI_{ps}.DM/\bar{v} + a_{oc} * pe\_m$ 
2:   $num\_slm = size(LI_{ps}.P_{SLM})$ 
3:   $path\_m, path\_l \leftarrow zeros(num\_slm, num\_slm)$ 
4:  for  $i \leftarrow 1, num\_slm$  do
5:    for  $j \leftarrow 1, num\_slm$  do
6:      if  $(c\_wm(i, j) == 0)$  do
7:         $c\_wm(i, j) \leftarrow 10000$ 
8:         $path\_m(i, j) \leftarrow j$ 
9:      end if
10:   end for
11: end for
12: for  $k \leftarrow 1, num\_slm$  do
11:   for  $i \leftarrow 1, num\_slm$  do
12:     for  $j \leftarrow 1, num\_slm$  do
13:       if  $(c\_wm(i, k) \geq 10000) \parallel (c\_wm(k, j) \geq 10000)$  do
14:          $sum\_temp \leftarrow 10000$ 
15:       else do
16:          $sum\_temp \leftarrow c\_wm(i, k) + c\_wm(k, j)$ 
17:       end if
18:       if  $(i \neq j)$  do
19:         if  $(c\_wm(i, j) > sum\_temp)$  do
20:            $c\_wm(i, j) \leftarrow sum\_temp$ 
21:            $path\_m(i, j) \leftarrow path\_m(i, k)$ 
22:         end if
23:       end if
24:     end for
25:   end for
26: end for
27: for  $i \leftarrow 1, num\_slm$  do
28:   for  $j \leftarrow 1, num\_slm$  do
29:      $s\_t \leftarrow i$ 
30:      $e\_t \leftarrow j$ 
31:     while  $(path\_m(s\_t, e\_t) \neq j)$  do
32:        $path\_l(i, j) \leftarrow LI_{ps}.WAM(s\_t, path\_m(s\_t, e\_t)) + path\_l(i, j)$ 
33:        $s\_t \leftarrow path\_m(s\_t, e\_t)$ 
34:     end while
35:   end for
36: end for
37:  $slm\_r.path\_l \leftarrow path\_l$ 
38:  $slm\_r.path\_m \leftarrow path\_m$ 
39:  $slm\_r.c\_wm \leftarrow c\_wm$ 
40: return  $slm\_r$ 

```

---

**Algorithm 32.** Calculate the optimal paths of people to exits

---

**Inputs:**    *CIRC*        : CIR-cube  
               *slm\_r*        : optimal paths and path lengths among SLMs  
               *LI<sub>ps</sub>*        : layout information of the concerned space

**Outputs:**   *i\_path\_e*    : optimal paths of individual people to exits

---

```

1:  for i ← 1, num_st do % num_st is the number of people
2:    slm_st ← determine the SLMs of the sub-zone of the person
3:    slm_et ← determine the SLMs that are assigned as exits
4:    pos_t ← 1
5:    for j ← 1, size(slm_st) do
6:      s_t ← slm_st(j)
7:      for k ← 1, size(slm_et) do
8:        if (s_t == slm_et(k)) do
9:          a_d(j,k) ← dis_e_slm
10:       else do
11:         dis_e_slm ← calculate distance between num_st(i) and s_t
12:         wa_d(j,k) ← slm_r.c_wm(s_t, slm_et(k)) + dis_e_slm
13:       end if
14:     end for
15:   end for
16:   row_n, col_n ← position of min(wa_d) in wa_d
17:   start_slm ← slm_st(row_n)
18:   end_slm ← slm_et(col_n)
19:   while 1 do
20:     i_path_e(i, pos_t) ← start_slm
21:     if (start_slm == end_slm) do
22:       break
23:     else do
24:       pos_t ← pos_t + 1
25:       start_slm ← slm_r.path_m(start_slm, end_slm)
26:     end if
27:   end while
28: end for
29: return i_path_e

```

---

In addition, to determine help-recipients, the following conditions were considered: (i) people who do not have an informing terminal and (ii) people who are motionless. Help-givers were determined from the potential informants and their paths were refined by Algorithm 34 to include the locations of help-recipients.

Actually, the determination of help-givers is an optimization problem since the generated solution should achieve (i) evacuation time of the last person should be as short as possible, (ii) average evacuation time of all people should be as short

as possible, and (iii) average length of all evacuation routes should be as short as possible. As presented in Chapter 3, a heuristic algorithm should be applied for dynamic context reasoning to approximately solve the NP-complete question. For this reason, Algorithm 34 was adopted from a Genetic Algorithm [8]. Technical details with regard to this algorithm are given as following.

The escape paths of all people were constructed as a chromosome. Every gene in a chromosome represents an escape path of a person (a sequence of SLMs). Thus the chromosomes are of the same length. This design facilitates crossover and mutation operations. A typical chromosome is shown in Figure 4.9, which includes (i) identification of the concerned people ordered according to the priority list, and (ii) evacuation paths generated for the people. A path is composed of a series of SLMs, which means that the people are assumed to follow the SLMs one by one. The default path of a person is the individually calculated optimal path to exits (Outcome of Algorithm 32). Particularly, ‘-1’ in the path of P\_ID: 35 in Figure 4.9 is an indicator, which means that the related person is a help-recipient and is waiting for the help from a help-giver (e.g. P\_ID: 31).

In the implemented Algorithm 34, both the number of chromosomes in a population and the number of maximum generation will be determined according to the results of functional testing. To generate a chromosome, help-givers were selected from the list of potential informants and will be able to provide help to people who have a lower danger level indicator than him/her. This is to ensure that help-recipients are informed by people who are safer than them.

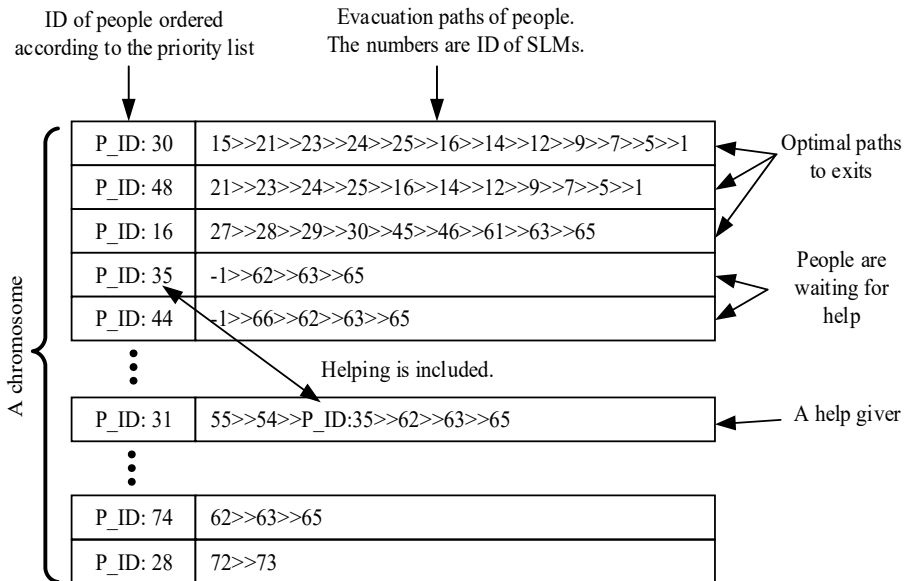


Figure 4.9. A chromosome generated for devising personal escape solutions

**Algorithm 34.** Revise the paths of help-givers to enable help-recipients be informed

---

**Inputs:**      $ifm\_hr$      : potential informants and help recipients  
                $p\_st$         : priority list of people  
                $slm\_r$        : optimal paths and path lengths among SLMs  
                $i\_path\_e$     : optimal paths of individual people to exits  
                $CIRC$         : the generated CIR-cube

**Outputs:**     $i\_path\_f$     : calculated paths for all people

---

1: population  $\leftarrow$  generate alternative escape solutions of all people  
 2: **for**  $j \leftarrow 1, maximal\_generation$  **do**  
 3:     cost  $\leftarrow$  feval (fitness\_func, population)  
 4:     cost, index  $\leftarrow$  sort(cost)  
 5:     population  $\leftarrow$  population( : , : , index(1: elite\_number))  
 6:     elite( : , : , j)  $\leftarrow$  population( : , : , 1)  
 7:     population  $\leftarrow$  exchange the help-recipients of two help-givers %Crossover  
 8:     population  $\leftarrow$  change the help giver of a help-recipient to another informant  
 9: **end for**  
 10:  $i\_path\_f \leftarrow elite( : , : , end)$   
 11: **return**  $i\_path\_f$

---

In addition, the fitness function for evaluating chromosomes was specified as:

$$\min (F) = a_1 \text{avg} \left( \sum te_{e_i} \right) + a_2 \max(te_{e_i}) + a_3 \text{avg} \left( \sum lp_{e_i} \right) \quad (4-11)$$

where:  $te_{e_i}$  is the evacuation time of a person,  $lp_{e_i}$  is the length of the evacuation path of a person.  $a_1$ ,  $a_2$  and  $a_3$  are weighting factors. These parameters are application-dependent and should be specified by application designers to achieve group optimal. In the prototype, they were determined as 5, 1 and 2, respectively.

The number reserved for a generation was 50% of the total population. To enable a fast convergence, two elite chromosomes were selected as parents of the crossover operation, and a help-giver in each of the chromosomes were considered to exchange their help-recipients. For the purpose of keeping the diversity of chromosomes, the rest 25% chromosomes were generated by the mutation operation, in which a help-recipient was randomly selected and another informable person was considered as the associated help-giver. When all generations were completed, the best solution was considered as the evacuation paths for all people.

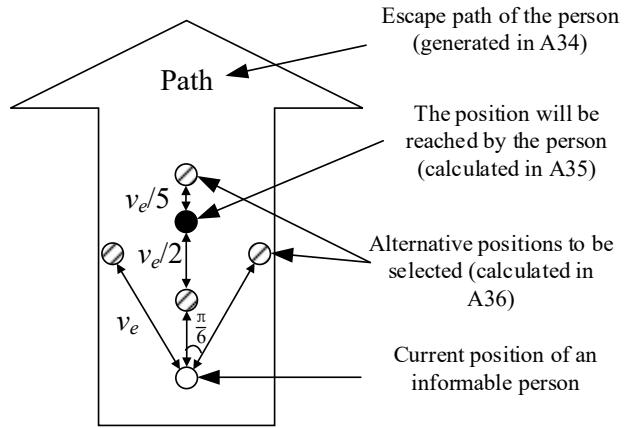
To generate short-term actions, the positions that will be reached in the near future by informable people were computed based on (i) the paths expected to be followed by them, (ii) their speed of motion, and (iii) the layout information of

the physical space. To demonstrate our mechanism for dynamic context information management, four alternative positions were calculated based on the speed of motion of the concerned person. The specific parameters for generation of the alternative positions are shown in Figure 4.10.

Here,  $v_e$  is the speed of motion of the

concerned person. In total, there are five alternative positions considered to be reached by the person. Accordingly, five alternative short-term actions can be generated corresponding to the positions, including (i) stay the same speed and direction of motion, (ii) increase the speed of motion, (iii) decrease the speed of motion, (iv) move to the left with the same speed of motion, and (v) move to the right with the same speed of motion. It was assumed that if the concerned person performs the corresponding action, then one of the alternative positions will be reached.

To handle dynamic situations towards expectations, the weights of situations should be calculated, depending on their IIs. In the case of the implemented fire evacuation simulation, situations with negative impacts were involved. The weight of a situation in optimization was calculated by the following equation:



**Figure 4.10.** Generation of alternative positions to be reached by an informable person

**Algorithm 38.** Generate an objective function based on the weights of situations

**Inputs:**  $w\_situ$  : weights of situations for optimization

**Outputs:**  $ob\_f$  : objective function for optimization

```

1: for  $i \leftarrow 1, num\_situ$  do
2:    $weight \leftarrow num2str(w\_situ(i, 1))$ 
3:    $situ\_temp \leftarrow num2str(w\_situ(i, 2))$ 
4:    $this\_situ \leftarrow strcat('situ(', situ\_temp, ')')$ 
5:    $this\_aspect \leftarrow strcat('+', weight, '*', this\_situ)$ 
6:    $ob\_f\_temp \leftarrow strcat(ob\_f\_temp, this\_aspect)$ 
7: end for
8:  $ob\_f\_temp2 \leftarrow strcat('ob\_f=@(situ)', ob\_f\_temp, ';')$ 
9:  $this\_rule \leftarrow char(ob\_f\_temp2)$ 
10:  $eval(this\_rule)$  % the string is executed as a code
11: return  $ob\_f$ 
    
```



$$weight(situ_i) = II_i / \sum II \quad (4-12)$$

where  $II_i$  is the impact indicator of the concerned situation, and  $\sum II$  is the sum of IIs of all situations. Then, an objective function was generated based on the calculated weights using Algorithm 38. The best action plans for people were selected by applying the generated objective function using Algorithm 39. In the reference application case, the dynamic context is managed by keeping people away from dangerous situations. This is the applied principle for selecting the best action plan for a person.

**Algorithm 39.** Select the best action plan based on the generated objective function

---

**Inputs:**     $ob\_f$         : objective function for optimization

$a\_ap$         : alternative action plans

**Outputs:**    $p\_ap$         : personal action plans

---

```

1:  for  $i \leftarrow 1, is\_num$  do %  $is\_num$ : the number of informable people
2:      for  $k \leftarrow 1, size(a\_ap, 1)$  do
3:          for  $j \leftarrow 1, situ\_num$  do
4:               $dis\_a\_s(j, k) \leftarrow$  calculate the distance to the situation
5:          end for
6:           $situ \leftarrow dis\_a\_s$ 
7:           $result(k) \leftarrow ob\_f(situ)$ 
8:      end for
9:       $[row, pos] \leftarrow \min(result)$ 
10:      $p\_ap(i) \leftarrow a\_ap(i, pos)$ 
11: end for
12: return  $p\_ap$ 

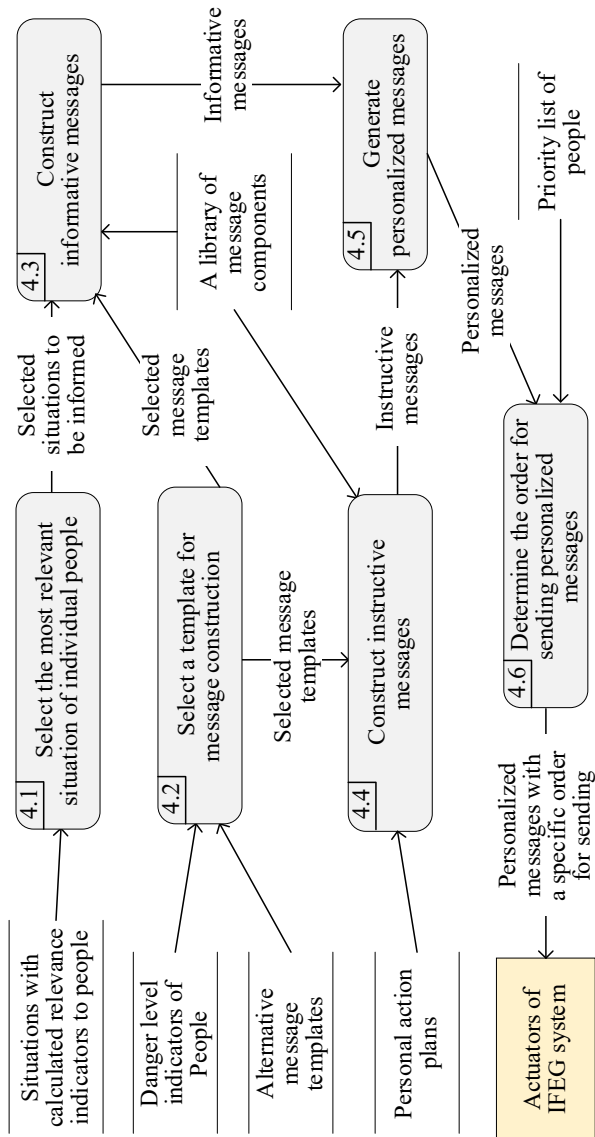
```

---

#### 4.3.4. The personalized message construction module

The implemented message construction module (MCM) includes six functions. The data flow diagram of the MCM is shown in Figure 4.11. The inputs of the prototype implementation are: (i) situations with calculated RIs to people, (ii) danger level indicators of people, (iii) message templates and message components which are predefined in a warehouse, (iv) priority list of people, and (v) personal action plans for informable people. The output of the MCM is the generated personalized messages with a specific order for sending. This output is used as a basis for providing message generation services for and to perform informing operations by the I-CPS application embedding the computational platform for dynamic context computation.

To select the most relevant situation of individual people, the situation that is characterized by the largest was concerned. As a next computational task, the function of message template selection regarded the computed sum of relevance indicators (SRI) of people as a reference. Another input of the function is the alternative message templates articulated and stored in a warehouse. To enable the generation of personalized messages for the fire evacuation case, several messages templates were designed, which are shown in Table 4.1. A message template was selected concerning two aspects. On the one hand, one of two conditions was judged according to the value of the calculated *SRI*, namely (i) normal circumstance ( $SRI \leq 0.1$ ), and (ii) emergency circumstance ( $SRI > 0.1$ ). On the other hand, the selection of a message template also depends on the type of the most relevant situation (having the largest *RI*).



**Figure 4.11.** Data flow diagram of the message construction module

On the one hand, one of two conditions was judged according to the value of the calculated *SRI*, namely (i) normal circumstance ( $SRI \leq 0.1$ ), and (ii) emergency circumstance ( $SRI > 0.1$ ). On the other hand, the selection of a message template also depends on the type of the most relevant situation (having the largest *RI*).

For each condition, a message template was used for constructing both informative and instructive messages. The first sentence of an informative message describes the status of the most relevant situation, while the second sentence of it describes the changes of the situation. In addition, when the personal danger circumstance is normal, the informative message includes information about the location of the situation. When the personal danger circumstance is emergent, the informative message includes information about the relationship between the situation and the target person. In addition, the first sentence of the instructive message shows the short-term action that should be performed by the person immediately. The second sentence of it represents the escape solution for evacuation.

**Table 4.1.** Predefined message templates for message construction

Value of <i>SRI</i>	Referred situation	Type of message	Predefined message templates	Examples of constructed messages
$\leq 0.1$	Fire	Informative message	(ds_name, link_v, spa_info), (ds_name, ds_verb, ds_att_adv)	Fire is in the computer room. Fire proliferates slowly.
		Instructive message	(p_a), ('please go to:', spa_info, 'to inform', h_r, 'about fire.', 'The path is:', p_path)	Please speed up the motion. Please go to the lab to inform P_ID:35 about fire. The path is 55>>54>>S_ID:35 >> 62>> 63>>65.
	People jam	Informative message	('A', ds_name, ('(, ds_att_1,')', link_v, spa_info), (ds_att_2, att_verb),	A people jam (10 people) is in front of exit 1. People number is decreasing.
		Instructive message	(p_a), ('The escape path is:', p_path)	Please keep the speed of motion. The escape path is: 72>>73.
$> 0.1$	Fire	Informative message	(ds_name, link_v, spa_re, 'you!'), (ds_name, ds_verb, ds_att_adv, ori_re, 'you'),	Fire is very close to you. Fire proliferates fast away from you.
		Instructive message	(p_a), ('The escape path is:', p_path)	Please speed up the motion. The escape path is: 55>>54>>62>>63>>65.
	People jam	Informative message	('A', ds_name, ('(,ds_att_1,')', link_v, spa_re, 'you!'), ('The', ds_name, ds_verb, ds_att_adv, ori_re, 'you!'),	A people jam (20 people) is in front of you. The people jam moves fast towards you!
		Instructive message	(p_a), ('The escape path is:', p_path)	Please move ahead on the left. The escape path is: 62>>63>>65.

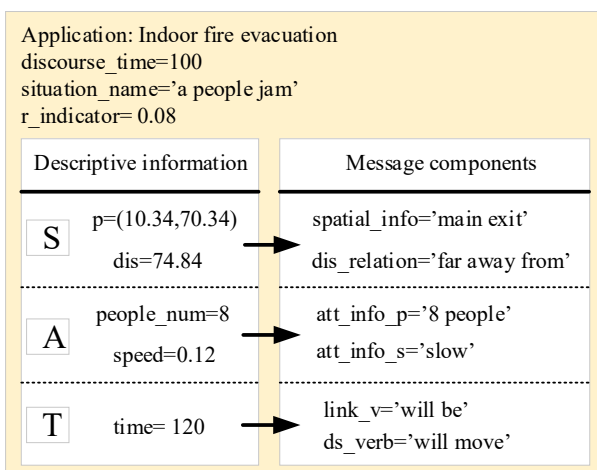
There are many vacant parts in the message templates. The SAT information describing the most relevant situation were converted into message components (i.e. words and phrases), which were placed in the selected message template. Table 4.2 shows a sample set of conditions and referred message components, which are stored in a library and used as an input of message construction. Based on this, message components were generated by comparing the relevant values of the SAT information of the concerned situation with the predefined threshold values. An example of converting the descriptive information of a situation to message components based on the implemented library is presented in Figure 4.12.

The generated personal action plans were converted to message components for the construction of instructive messages. According to the five alternative actions generated in Section 4.3.3, five sentences were predefined to indicate these five actions, including: (i) Please keep the speed of motion. (ii) Please speed up the

**Table 4.2.** Converting SAT information of a situation to message components

Type of information	Category	Conditions	Referred message components
Spatial information of a situation	Static: distance	$dis_{ij} \geq d_{vl}$	Very far away from
		$d_l \leq dis_{ij} < d_{vl}$	Far away from
		$d_s \leq dis_{ij} < d_l$	Close to
		$0 \leq dis_{ij} < d_s$	Very close to
	Dynamic: change of distance	$\Delta dis_{ij} \geq d_{ff}$	Fast away from
		$0 \leq \Delta dis_{ij} < d_{sf}$	Slowly away from
Attributive information of a situation	Static: speed of motion	$speed > s_f$	Fast
		$s_s < speed < s_f$	Mediate
		$s_s > speed$	Slow
	Dynamic: change of speed of motion	$\Delta speed > cs_i$	Speed increases
		$cs_d < \Delta speed < cs_i$	Speed is stable
		$cs_d > \Delta speed$	Speed decreases
Temporal information of a situation	Static: time of happening	$t_s(DS) - t_c \geq t_{ff}$	Far future
		$t_{nf} \leq t_s(DS) - t_c < t_{ff}$	Near future
		$t_n \leq t_s(DS) - t_c < t_{nf}$	Nearby
		$0 \leq t_s(DS) - t_c < t_n$	Now
		$t_c \leq t_s(DS) - t_c < 0$	Recent
		$t_{np} \leq t_s(DS) - t_c < t_c$	Near past
		$t_{fp} \leq t_s(DS) - t_c < t_{np}$	Far past
	Dynamic: variation of happening time	$\Delta t_s(DS) > 0$	Later
		$\Delta t_s(DS) < 0$	Earlier

motion. (iii) Please slow down the motion. (iv) Please move ahead on the left. and (v) Please move ahead on the right. In addition, as shown in the examples of generated sentences in Table 4.1, a personal escape solution was represented by a series of SLM, and no special message components were generated. Because of these, we assume that the servicing application (i.e. IFEG system) could convert the SLMs to sentences, e.g. represent the paths on a map.



**Figure 4.12.** An example of converting SAT information of a situation to various message components

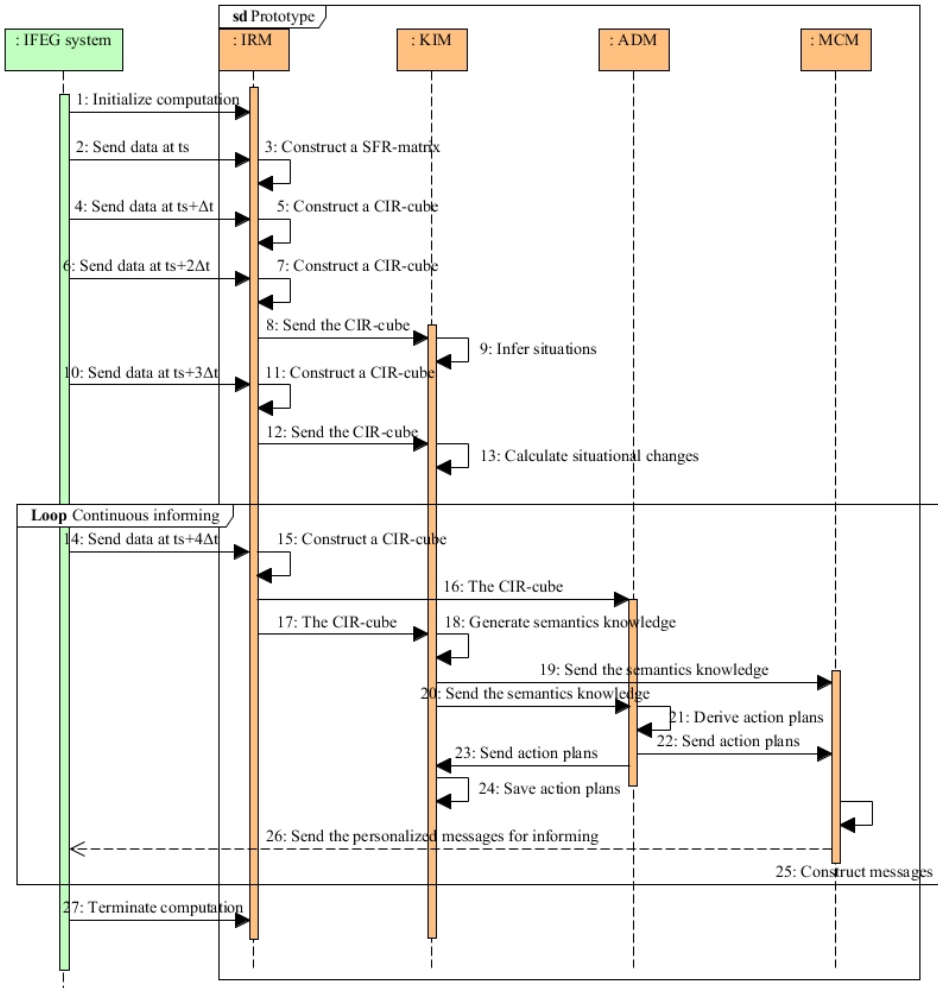
## 4.4. Implementation of the prototype

To enable continuous computation, the individually implemented modules were integrated into a prototype. Details with regard to the system-level implementation are introduced in this subchapter. Two aspects are focused, including (i) organization of the computational workflow among the modules, and (ii) specification of the data-level interrelation of the modules.

### 4.4.1. Procedural integration of the modules

Figure 4.13 shows the sequence relations among the simulated IFEG system and the implemented prototype. Two types of inputs should be provided by the IFEG system, including (i) control command which activates the working of the prototype, and (ii) contextual data that describes the entities attributes in a time-dependent manner (the increment of sending is  $\Delta t$ ). According to the proposed context knowledge pyramid model, computational integration of data, information, and knowledge is complemented by logical abstractions to generate semantic knowledge on different abstraction levels. Accordingly, the progression of building awareness is achieved by accumulatively generating, states, events, situations, and scenes to infer semantic knowledge. This is the main reason why the working sequence of the prototype involves multiple computational cycles.

Using the principles of integration and abstraction, the first computational cycle



**Figure 4.13.** Sequence diagram of computing in the implemented modules based on the data representation schemes

process states related to context entity and relations (CER), based on which an SFR-matrix is to be generated. Then, the prototype waits until the contextual data depicting the scenario at  $t_s + \Delta t$  is acquired. Events regarding CER can be calculated and a CIR-cube is initially generated. To enable the inferring of situations, two successively computed events are required. It means that knowledge with regard to situations can be known from the third computational cycle. Furthermore, two successively computed situations could form the basis to enable inferring of scenes, based on which a situation change can be calculated in the fourth computational cycle.

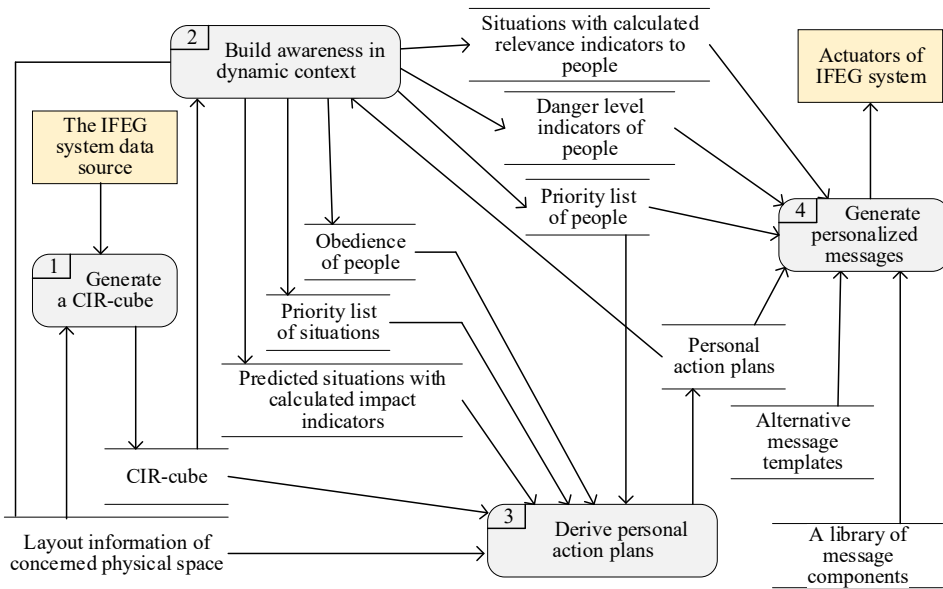
In the fifth computational cycle, the trend of the situational changes can be computed by the KIM. This is considered as the basis for generating personal

action plans and associated messages. Accordingly, the first personalized messages can be generated and sent to inform people by IFEG system. The knowledge obtained in the previous five computational cycles becomes the basis for the subsequent computational cycles. Personalized messages can be made in every computational cycle until a stop computational command is received.

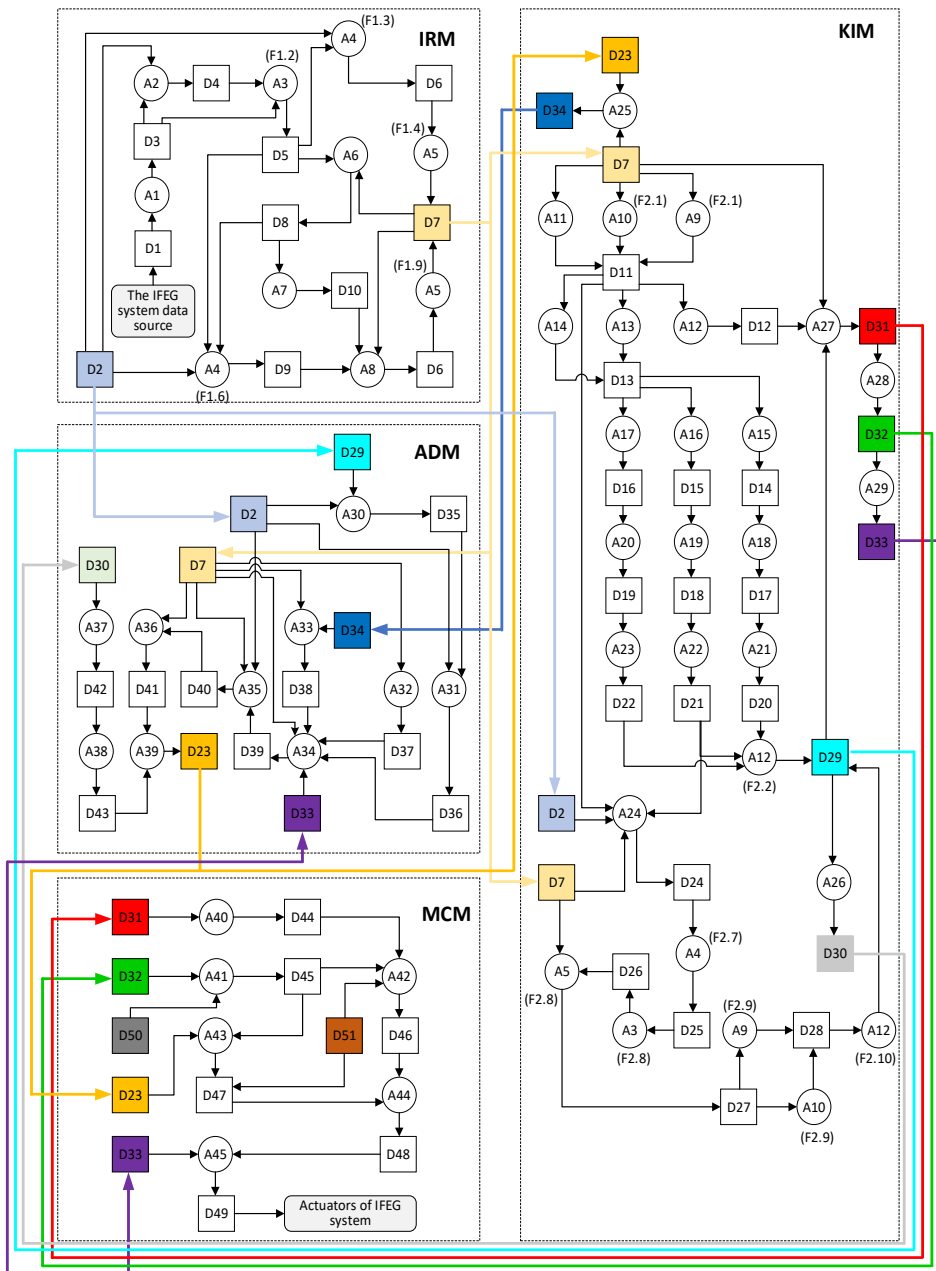
From the sequence diagram, it can be observed that the semantic knowledge inferred by the KIM is the key for generation of personalized action-plans and messages for the purpose of dynamic context management. To generate sufficient semantic knowledge, a continuous aggregation of the descriptive contextual data is needed. In our case, the inference of people jams requires consideration of two successive changes (events) regarding people distances and the future context is predicted by using two successive changes of the situations. It means that the solutions (e.g. messages) used for achieving dynamic context information management will be generated with a delay of  $4\Delta t + T_c$  (i.e. response time) after the ‘start computation’ command is firstly received, where  $T_c$  is the total computation time needed by the fifth computational cycle.

#### 4.4.2. Specification of the input and output couplings of the algorithms included in the prototype

Drawn from a bird’s-eye view, Figure 4.14 shows the system-level data flow diagram of the implemented prototype. In this figure, computational processes 1,



**Figure 4.14.** System-level data flow diagram (context diagram) of the implemented prototype



**Figure 4.15.** Inputs and outputs of the implemented algorithms

2, 3 and 4 refer to the IRM, the KIM, the ADM and the MCM, respectively. The input and output data among the implemented modules are shown.

In addition to the time-dependent data, (i.e. locations and types of entities)



**Table 4.3.** The input and output data of the implemented algorithms

Module	Data	Details	Data	Details
IRM	D1:	Time-dependent locations and types of entities	D6:	A fully-filled SFR-Matrix
	D2:	Layout information of concerned physical space	D7:	A CIR-cube
	D3:	Estimated locations and types of entities	D8:	A list of entities
	D4:	Sub-zone of entities	D9:	An SFR-Matrix filled with changed distances
	D5:	An SFR-matrix with no distance filled	D10:	Speed of motion of entities
KIM	D11:	Inferred situations	D23:	Personal action plans
	D12:	Current situations with calculated impact indicators	D24:	Future locations of entities
	D13:	Scenes	D25:	Distances among entities in the near future
	D14:	Location changes of people jams	D26:	An SFR-matrix containing predicted data
	D15:	Speed changes of people jams	D27:	A CIR-cube containing historical and predicted information
	D16:	Location changes of fire situations	D28:	Newly formed situations in the near future
	D17:	Regression models of location changes of people jams	D29:	Predicted situations with calculated impact indicators
	D18:	Regression models of speed changes of people jams	D30:	Priority list of situations
	D19:	Coverage increment of fire situations	D31:	Situations with calculated relevance indicators to people
	D20:	Future locations of people jams	D32:	Danger level indicators of people
	D21:	Future speed of motion of people jams	D33:	Priority list of people
	D22:	Future coverage of fire	D34:	Obedience of people
ADM	D35:	Penalty matrix	D40:	Positions that will be reached by informable people in the future
	D36:	Optimal paths and length of the paths among SLMs	D41:	Alternative action plans
	D37:	Optimal paths of individual people to exits	D42:	Weights of situations for optimization
	D38:	Potential informants and help recipients	D43:	Objective function for optimization
	D39:	Calculated paths for all people		
MCM	D44:	Selected situations to be informed	D48:	Personalized messages
	D45:	Selected message templates	D49:	Personalized messages with a specific order for sending
	D46:	Informative messages	D50:	Alternative message templates
	D47:	Instructive messages	D51:	A library of message components

provided by the IFEG system data source, three data constructs should be predefined and stored in the warehouse of the prototype. They are (i) layout information of the concerned physical space, (ii) alternative message templates, and (iii) a library of message components.

These three data constructs are dependent on the servicing application (i.e. the fire evacuation application in this case). They should be specified by application designers before the prototype can be used. The output of the prototype is the generated personalized messages with a specific order for sending, which will be delivered to the actuators of the IFEG system (e.g. personal smartphones).

A detailing of the input and output couplings of the modules on the algorithm level is shown in Figure 4.15. The specification of the input and output data used by the algorithms are listed in Table 4.3. In Figure 4.15, the algorithms are represented by circles, while the input and output data are indicated by rectangles. Particularly, colored rectangles indicate the exterior data used among the modules, while the white rectangles are data constructs temporarily used within the modules. In addition, several algorithms are reused in multiple functions, e.g. Algorithm 4 was used in both F1.3 and F1.6.

As indicated by the input and output couplings of the prototype, most of the algorithms depend on the outcomes of the previously executed algorithms. It means that the preciseness of the computational results generated by the previously executed algorithms has a significant impact on the subsequent algorithms. For instance, the computation of the algorithms conducted by the KIM, the ADM and the MCM strongly depends on the representation accuracy of the CIR-cube, which is the output of the IRM. Therefore, the computational accuracy of the algorithms and their impact should be further investigated together with a functional checking of the implemented prototype, which will be presented in the next sub-chapter.

## **4.5. Functionality testing of the prototype in the context of an IFEG System**

The objective of the functionality testing was to check (i) if the algorithms and modules will do as the same as what they are supposed to do and (ii) to what extent the specified functional requirements are fulfilled. The complete prototype system is naturally consisting of four modules. Functionality testing of them will be individually discussed in this subchapter. To this end, the implemented prototype has been placed in the context of the simulated reference application case (i.e. fire evacuation scenario). The generated descriptive contextual data in the reference application was fed as the input of the prototype and the output was examined. It's worth mentioning that the modules contained in the prototype system are interrelated. The testing of the modules cannot be completely separated. Accordingly, all input data of an individual module are assumed to be available and accurate before the testing of it was conducted. In addition, after the modules were individually tested, the prototype was tested as a whole. Varied real-life scenarios were considered and simulated, based on which dynamic

context information was managed.

The functionality testing conducted in this sub-chapter addresses two aspects. Firstly, the correctness of the computations by the implemented modules was evaluated. The target was to prove that the functions required by the DCIP-M can be achieved. In addition, computational errors and their impact that are induced from different experimental (application-oriented) setups, e.g. thresholds, were explored. In the following sections, details with regard to the functionality testing of the modules are given.

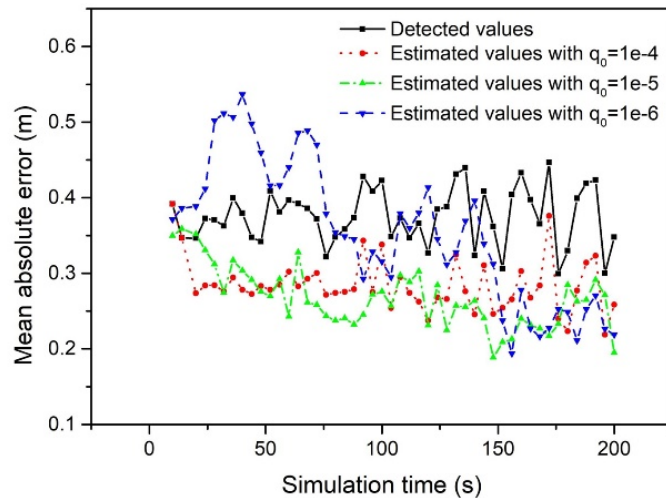
#### 4.5.1. Testing the information representation module

The basic requirement for the IRM is to properly represent the time-varying scenario in a digital way. In the simulated fire evacuation scenario, indoor position data of people were generated, which were considered as the real positions of people. Randomly generated white Gaussian noise was added to the indoor positions to represent biases of the sensed real data. Accordingly, the objective of F1.1 is to estimate the real indoor positions of people based on the detected data. The correctness of the estimation is crucial to the rest functions in the module, such as speed calculation and distance calculation. Since the position estimation was realized based on a Kalman-filter, the influence of different parameters of the Kalman-Filter is first investigated.

As claimed in the literature, the optimality of Kalman-filter depends on the quality of prior assumptions about the process noise matrix  $\mathbf{Q}$ , and the measurements noise covariance matrix  $\mathbf{R}$  [9]. As discussed in Section 4.3.1, the measurement noise  $\mathbf{R}$  was specified according to the predefined variance of the white noise, while

$\mathbf{Q}$  depends on the different estimated values of  $q_0$ . Accordingly, Different  $q_0$  were tested in order to optimize the functional performance of the implemented Kalman-filter.

In the test, the position data of people in the simulated fire



**Figure 4.16.** MAEs of the indoor position estimations with different process noise coefficients

evacuation scenario from  $t=0s$  to  $t=200s$  were used by the position estimation algorithm. The sampling rate of observation was set to 1. As shown in Figure 4.16, the mean absolute errors (MAE) between the estimated positions and the real positions of the people in the simulation concerning different values of  $q_0$  were calculated. Each value represented in the figure is the average value with regard to the estimation errors of all people at a given moment. The MAE between the detected positions and the real positions over the period of time are shown as well. It can be seen from the results that the best situation was found at  $q_0=10^{-5}$ . Either a smaller value  $q_0=10^{-6}$  or a bigger value  $q_0=10^{-4}$  would result in a larger estimation error. Particularly, when  $q_0=10^{-6}$  the estimation error is bigger than the detection error from  $t=21s$  to  $t=78s$ . The reason for the phenomenon can be explained as follows.

In the considered model for estimation, i.e., Equation (4-4), the movement of a person was simplified as a linear process. This is a normal principle applied for estimation the movement of objects by using Kalman-filter, e.g. vehicles, aircraft or spacecraft [11]. However, the actual motion of the person in an indoor environment may always experience non-linear processes. For instance, the person may move around a corner or may accidentally be influenced by people jams or other moving people. When the non-linear processes happen, a large estimation error will be produced. This can be proved by the estimation errors in case of  $q_0=10^{-6}$ . After the fire alarm started at  $t=10s$ , people are moving out from their individual rooms to the connected corridor or the main hall of the building. This is the reason why large estimation errors can be observed from  $t=21s$  to  $t=78s$ .

After  $t=150s$ , since most of the people are moving in the corridor and the main hall, the non-linear process is less likely to happen. During this period of time, a relatively small error could be observed. In addition, with a given covariance of the observation noise  $\mathbf{R}$ , when a relatively small  $q_0$  is applied, the state estimate  $\hat{u}_{k-1}$  relies more on the predicted positions of people at time  $t=k-1$  (calculated using Equation (4-4)). As a contrast, when a relatively large  $q_0$  is applied, the state estimate  $u_k$  more likely relies on the observed values,  $z_k$ . ( $u_k$  is the position estimation and  $z_k$  is the position observation). The estimations using  $q_0=10^{-6}$  are more sensitive to the non-linear processes.

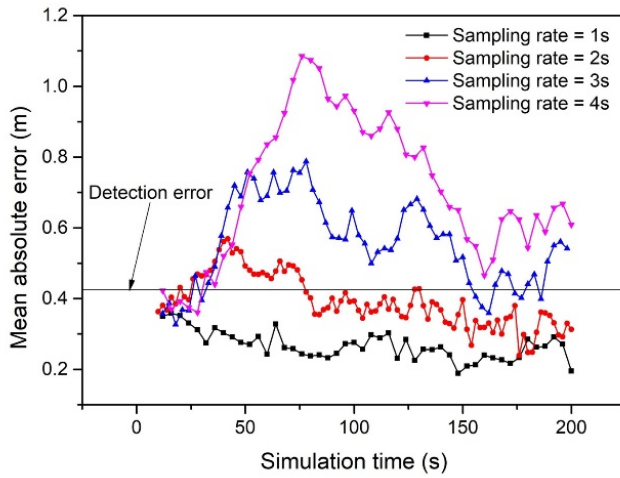
To efficiently represent the events happening in the physical space, the sampling rate plays an important role. Because an improper sampling rate may result in a delay and/or unreliable representation of events. Then, based on the optimized Kalman-filter, a test was taken to obtain the errors concerning the estimation of indoor positions of people with different sampling rates. The results are shown in Figure 4.17. The expectation of the detection error is shown in the figure by a straight line. It can be seen from the result that the best performance was found at sampling rate=1s. In addition, when the sampling rate is more than 2 seconds, the

estimation errors were very large, which made the estimated results worse than the detected results for most of the situations.

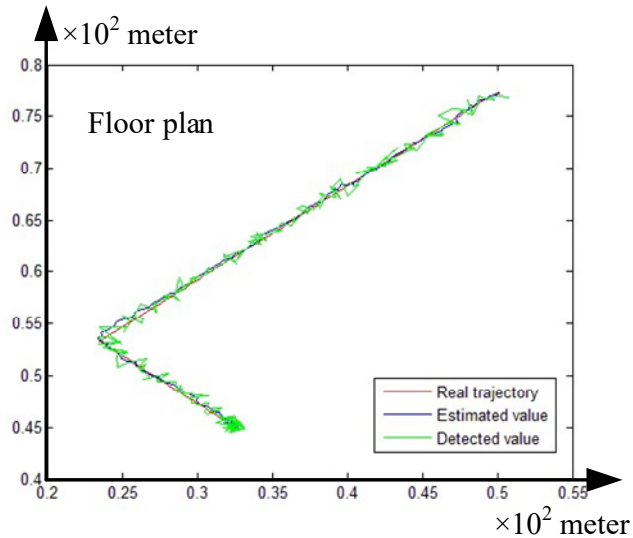
The nonlinear processes mentioned in the previous test are also the main reason that causes the estimation error. Because, if a large sampling rate is applied, the indoor

positions changes of a person in non-linear process cases (e.g. moving around a corner) cannot be timely observed. The increments in the  $u_k$  (i.e.  $\Delta x_k$  and  $\Delta y_k$ ) are relatively large, which may result in a large error in the calculation of  $\hat{u}_k$ . Therefore, it can be concluded that the applied approach to indoor position estimation is very sensitive to the sampling rate. For this reason, the sampling rate was determined as 1s in simulation time.

Based on the obtained parameters, a continuous estimation of the indoor positions of a given person over a period of time is shown in Figure 4.18. It can be observed that a good estimation of position was achieved when the concerned person suddenly changed the direction of movement. In this situation, the



**Figure 4.17.** Estimation errors of indoor positions with different sampling rates

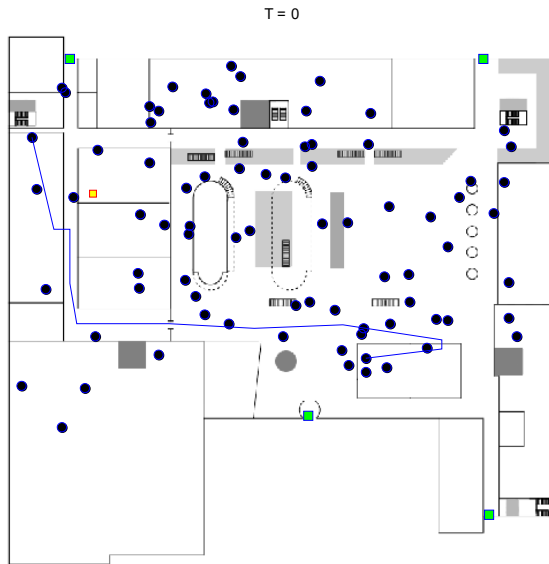


**Figure 4.18.** Estimation of the positions of a moving person

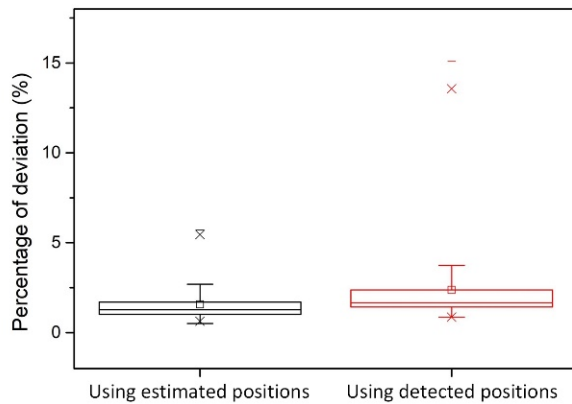
maximum estimation error was 0.39 meters. In addition, when the person is moving linearly, the average estimation error was less than 0.08 meters. Therefore, the results have shown that a good estimation has been achieved.

Another important function in the IRM is to calculate the distances among entities in the indoor environment.

It influences the correctness of the representation of spatial relations of entities. The distance calculation is based on a path-finding algorithm, which calculates the nearest path between two entities. Figure 4.19 shows the result of the path-finding algorithm applied between two selected entities. To test the correctness of the distance calculation, the distances among all people in the simulated fire evacuation scenario from  $t=0$  s to  $t=200$  s were calculated. The statistical deviations of the distances (calculated based on the estimated positions of people) are shown in Figure 4.20. As a comparison, the errors in distance calculation by using the detected positions directly are also presented. It can be seen that by using the optimized Karman-filter, the deviation of distance calculation decreased from 2.4% to 1.6%, while the maximum error decreased from 15.1% to 5.7%. In addition, if the detected position values are directly used for



**Figure 4.19.** Calculation of the distance between two concerned people



**Figure 4.20.** The percentage of deviation of distance calculation

distance calculation, large errors may accidentally happen.

The results imply that the estimation of indoor position is meaningful for distance calculation in the indoor environment. Although the expectation of the detection error is only 0.42 meter, it may lead to a significant error in distance calculation. Because a position error may lead a person to be judged into a different sub-zone. For instance, if a person is very close to the wall of a room, he/she may be considered to be staying in the neighboring room, which produces large errors in the distance calculation to other people. The distance calculation is very sensitive to the errors of indoor positioning. Based on the conducted functionality testing, it can be concluded that the implemented algorithms were proven to be efficient for calculating the indoor positions of people and the distances among them. Based on the calculated positions and distances, the CIR-cube is able to represent a time-vary scenario with high validity. The output of the implemented IRM can be used as a basis for the computations of the other three modules.

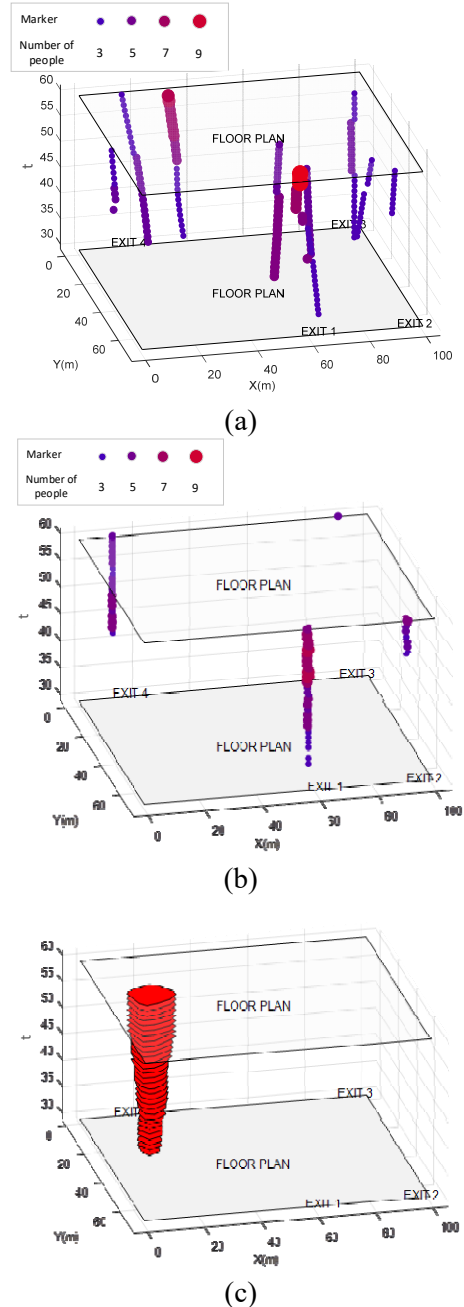
#### **4.5.2. Testing the context knowledge inferring module**

Based on the calculated CIR-cube, the situations inferred from the simulation within a given time interval from  $t = 30$  s to  $t = 60$  s are shown in Figure 4.21. The floor plan in the simulation was simplified to a rectangle. Figure 4.21.a. shows the inferred floating people jams, while anchored people jams at given positions are presented in Figure 4.21.b. Every people jam inferred at a given moment is represented by a point and the position of the point shows the center of the people jam. It can be seen from these two figures that the sizes of anchored jams were increasing due to the blockage caused by the people who tried to escape through the concerned exits.

In addition, since the fire was proliferating in the space, its boundary was changing in the concerned time period, as shown in Figure 4.21.c. It can be observed that the coverage of fire was expanding in simulation. The simulation results indicated that the concerned three types of situations can be efficiently represented from a time-varying process based on the constructed CIR-cube.

Since the situations shown are time-dependently inferred, transitivity and causality interplays among the inferred situations were obtained and a part of the inferred interplaying situations is shown in Figure 4.22. In this figure, every point refers to an inferred people jam. The red dashed lines represent for the causality interplays, while the black lines refer to the transitivity interplays. It can be observed from the result that, the individually inferred situations are bridged by the interplays, which form several 'chains' of situations. Each 'chain' is referred to as a scene according to concepts proposed in Chapter 3. Particularly, the scene defined by the causality interplay only contains two situations, while those formed by transitivity interplays may involve multiple interplaying situations.

Based on the generated scenes, semantic knowledge hidden from the time-varying scenario was interpreted. Firstly, the changes of the situations were identified. For instance, the movement of the floating people jams can be represented by the location changes of the situations over time. In addition, by checking the difference between two successive people jams, the entity joining and leaving from the people jams were identified. This can be a basis for management of the people jams (context) by informing involved people to perform certain actions (e.g. to keep away from a dangerous situation). Thirdly, the causality interplay implies that the floating people jams contribute to the formation of the anchored people jam. Accordingly, the anchored people jam can be considered as a consequence of a floating people jam that is moving towards the exit. This relation can be transit to a rule, which is used for predicting the consequence of any concerned floating people jam in similar cases. Therefore, the computational results imply that interplays enable a semantic interpretation of the procedural sequence of the varying scenario over a period of time. The inferred scenes can be represented and processed based on graph theory

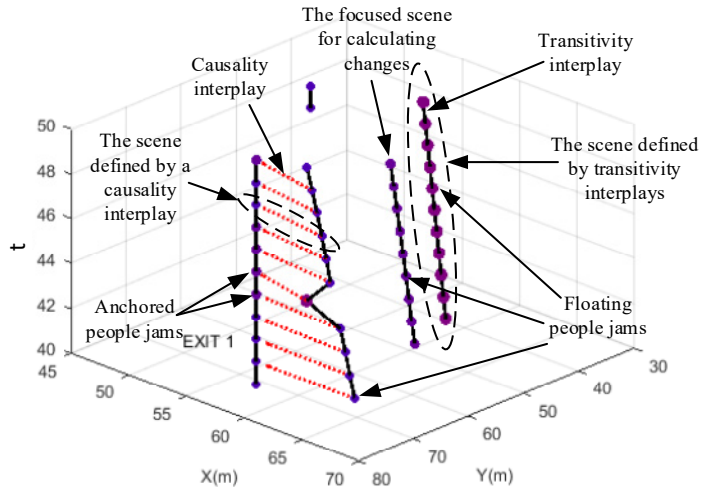


**Figure 4.21.** Simulation results of the inferred situations over a given period of time: (a) floating people jams, (b) anchored people jams, (c) proliferation of fire



technologies, e.g. semantic web, to enable learning, predicting or reasoning of the dynamic context.

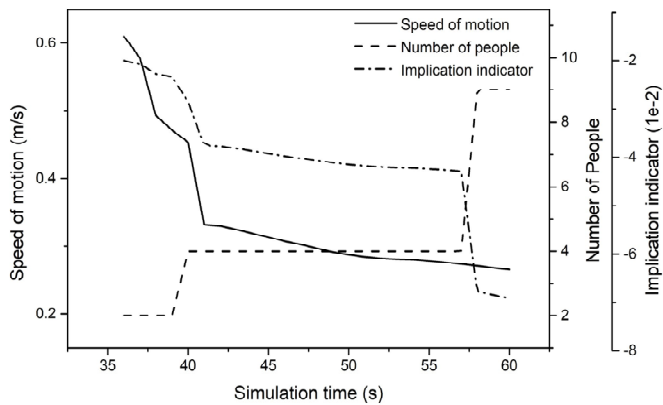
Then, the changes of jams formed by people within the given period of time were calculated by using the scenes defined by the transitivity interplays. Figure 4.23 shows two aspects of the changes of the



**Figure 4.22.** Inferred transitivity and causality interplays among people jams

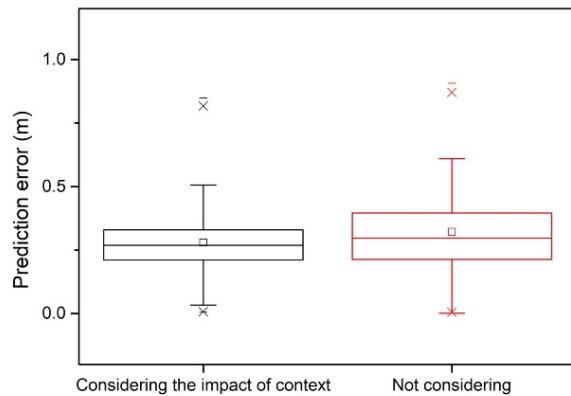
people jam marked in Figure 4.22, namely (i) the change of the number of entities involved in the people jam and (ii) the change of speed of motion. Based on the Equation (4-10), the impact indicators (IIs) of the people jams were calculated, which is also shown in Figure 4.23. It can be seen that, as the time eclipses, the number of people in the people jam increases and the speed of motion of the people decreases. Accordingly, the negative impact of the people jam (as a situation) increases (indicated by a decrease in the negative value). Therefore, the implemented algorithms were able to effectively calculate the II of dynamic situations over a period of time. In this way, the impact of dynamic context can be known by computing II of all situations.

One of the important purposes of using the interpreted impact of dynamic context is to facilitate context prediction. To prove this,



**Figure 4.23.** The changes and the calculated impact indicators of a people jam over a period of time

testing was made, in which the location changes of people in the time period from  $t = 25$  s to  $t = 200$  s were analyzed. At each time step of computation, the positions of people at the next time step were predicted according to the implemented algorithm. Then, the predicted values were compared to the actual positions of people (according to the simulation) one-time step later. The deviations



**Figure 4.24.** Prediction errors of people positions

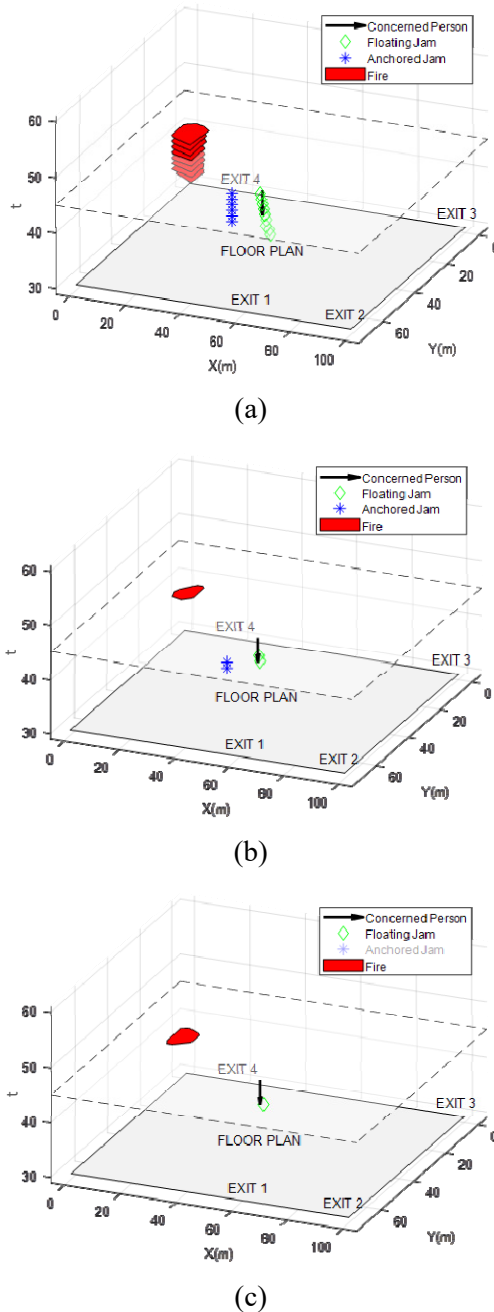
between the predicted values and the actual values were considered as prediction errors. The results of the testing are shown in Figure 4.24. To address the effectiveness of the proposed approach for context prediction, the errors of people positions predicted by using the implemented Kalman-filter were also calculated and shown in the figure. It's worth mentioning that our approach for location prediction considered the impact of dynamically formed people jams (dynamic context) and the impact of the layout of the concerned physical space (static context) on the motion of people. On the other hand, the implemented Kalman-filter individually predicts the future positions of people based on their historical position changes. Thus, the Kalman-filter-based approach does not take the impact of people context into consideration.

It can be seen from the results that by considering the impact of context, the 95 percentiles of the prediction error decreased from 0.67 m to 0.52 m, while the 75 percentiles of that decreased from 0.40 m to 0.32 m. In addition, there is no significant change with regard to the 25 percentiles of prediction error. The result indicates that the implemented algorithm effectively alleviated relatively large errors in position prediction, while it had limited improvement regarding relatively small errors. The reason can be explained as follows. As indicated by the testing results presented in Section 4.5.1, the average estimation error of indoor position was found at 0.27 m. Since the estimated positions were used for predicting the future positions of people, the estimation error naturally contributed to the prediction errors, which can hardly be eliminated by the applied prediction algorithms.

The prediction error between 95 and 75 percentiles mainly originates from the impact of context. For instance, if a person involved in a situation (e.g. enters into

a people jam), then the speed of motion of him/her might be suddenly changed. This piece of knowledge cannot be obtained from the historical position changes of him/her. Thus, the traditional approaches for position prediction, e.g., Kalman-filter cannot foresee the involvement and may believe that the person will move as what has been done in the past. On the other hand, the proposed approach calculated the speed of motion of the people who are involved in people jams and changed the predicted positions if people are moving around a corner. In this way, the impact of context has been taken into calculation, which improves the accuracy of prediction.

Furthermore, it can be seen that the maximum errors in both conditions are isolated and quite abnormal. These errors were generated due to the unanticipated changes of motion of people in the generated simulation, such as a sudden turn in movement due to the notice on the fire by people. Due to the lack of knowledge concerning this, both prediction approaches cannot deal with the unanticipated situations that cause significant errors. Therefore, the testing results have proved that DCIP facilitates context prediction. Different to traditional approaches, which are normally based on an analysis of a sequence of historical positions or attributes of individual entities, the proposed approach makes use of the



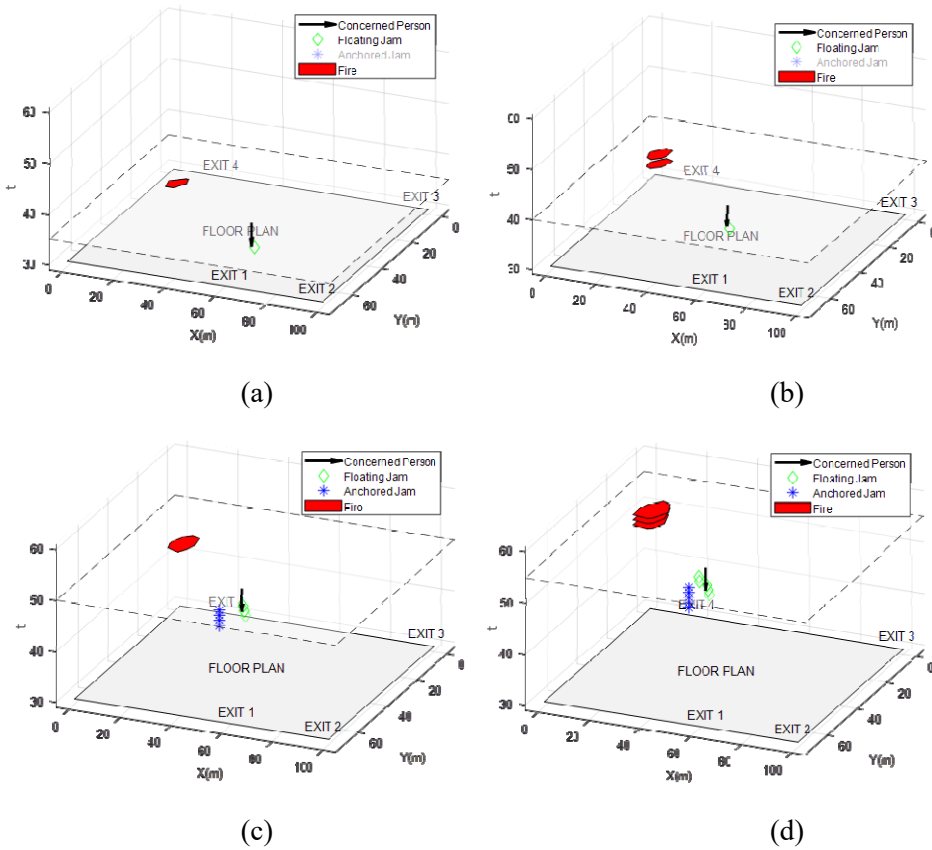
**Figure 4.25.** Determining personal context with different thresholds of RI: (a) 0.005, (b) 0.01, (c) 0.015

knowledge hidden from entity relations, which alleviated the prediction errors caused by non-linear processes (e.g. involving in a people jam).

Having the predicted positions, the newly formed situations in the near future were inferred out, and their impact indicators (IIs) were calculated according to the same principle as shown above. Then, relevance indicators (RI) between situations and people were calculated. Based on the calculated RIs, dynamic personal contexts were formed, where the threshold of RI plays an important role. A situation can be relevant to a person and become part of the personal context if its RI is higher than the threshold. Different thresholds may form different personal contexts. The specified personal context of an exemplified person (P\_ID = 039) at a given point in time (e.g.  $t = 45$  s) with different thresholds are shown in Figure 4.25. It can be seen that more situations were considered as relevant to the person when the threshold was relatively lower, whereas fewer situations were considered as relevant to the person when the threshold was relatively higher.

The simulation results imply that a low threshold makes the context relevance calculation mechanism sensitive to the situations happening around an entity. This might be required by emergency applications. However, the weak point is that the more situations are included in the personal context, the more computations might be needed for processing them. On the other hand, a higher threshold enables faster computations since the number of relevant situations is smaller. In practical applications, a proper threshold can be either specified by application designers or calculated in real-time according to the application needs. In the following part of the testing, the threshold was specified as 0.01.

The inferred personal contexts of the person at time points, including  $t=35$ s,  $t = 40$  s,  $t = 50$  s and  $t = 55$  s are shown in Figure 4.26. The personal context at  $t = 45$  s is shown in Figure 4.25.b. It can be observed that the personal context at  $t=35$ s includes two situations: (i) the floating people jam at  $t = 35$  s and (ii) the fire at  $t = 35$  s. This is because of the facts that (i) the II of the floating people jam was relatively low (actually contains 3 people), (ii) the fire was relatively far away from the person. However, the distance to the anchored people jam in front of Exit 1 is decreased as the concerned person moved towards Exit 1. The anchored jam became relevant to the concerned person at  $t = 45$  s. This can be observed by a comparison between the numbers of stars in Figure 4.26.b and Figure 4.25.b. In addition, since the people number of the floating people jam increased from 3 to 4 at  $t = 45$  s, the II of the floating people jam increased. This made the floating people jam happening at  $t = 46$  s relevant to the concerned person, as shown by the two diamonds in Figure 4.25.b. When the concerned person moved closer to Exit 1, the anchored people jam becomes more important to him. This can be underpinned by a phenomenon that the numbers of stars in Figure 4.26.c and Figure 4.26.d are more than that in Figure 4.25.b.



**Figure 4.26.** Identified personal context of a person at different points in time with a threshold = 0.01: (a)  $t = 35$  s, (b)  $t = 40$  s, (c)  $t = 50$  s, (d)  $t = 55$  s

Integration of personal contexts over a period of time was used to represent the dynamic context. A situation might be included in personal contexts specified at multiple points in time. For instance, the floating people jam happening at  $t = 46$  s was included in both of the personal context specified at  $t = 45$  s and the personal context specified at  $t = 46$  s. This implies that if a situation is important to a person, the history and future of it may also be important. In addition, the differences among multiple successive personal contexts indicate how the context is changing. The dynamics of personal context can be represented by (i) the change of the number of relevant situations and (ii) the change of attributes of the relevant situations.

Therefore, the simulation results validated that the proposed mechanism supports identifying and evaluating personal dynamic contexts of entities over a period of time. By using the II and the RI, the relevance between situations and entities can

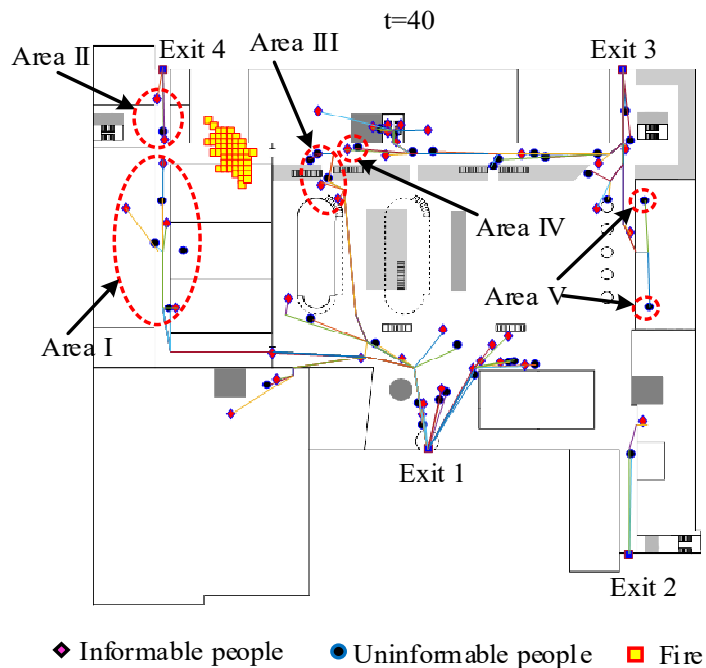
be handled. It provides an opportunity for customizing services to users by I-CPSs. The specified personal context enables the generation of a personal action plan for the purpose of dynamic context management.

### 4.5.3. Testing the action-plan deriving module

The implemented action-plan deriving module (ADM) is to generate personalized action plans for people in case of fire evacuation to achieve certain purposes of dynamic context information management. The specific objectives were to (i) evacuate all people safely from the space in the shortest time, (ii) reducing the number of people in people jams, (iii) keeping people away from dangerous situations. The first objective of the testing is to see if the implemented algorithms and function realize to do what they are supposed to do. Another consideration is to see if dynamic context information management can be properly realized by I-CPSs when the attribute diversity of people are considered, and If the proposed mechanism is able to adapt to the context changes (e.g. a disobeying happens).

As the first step, a penalty matrix was generated based on the calculated impact indicators of situations. Using the penalty matrix and the weighted adjacency matrix, optimal personal escape paths were computed. Figure 4.27 shows the result of the computation at a given moment, in which the generated escape paths were indicated

by the lines from people to their individual optimal escape exits. It can be seen that the people in 'Area I' and 'Area III' were considered to escape through Exit 1, instead of Exit 4, which is the nearest exit. This is because a large penalty value was given to the paths leading to the sub-zones that catch fire and



**Figure 4.27.** Personalized optimal escape routes

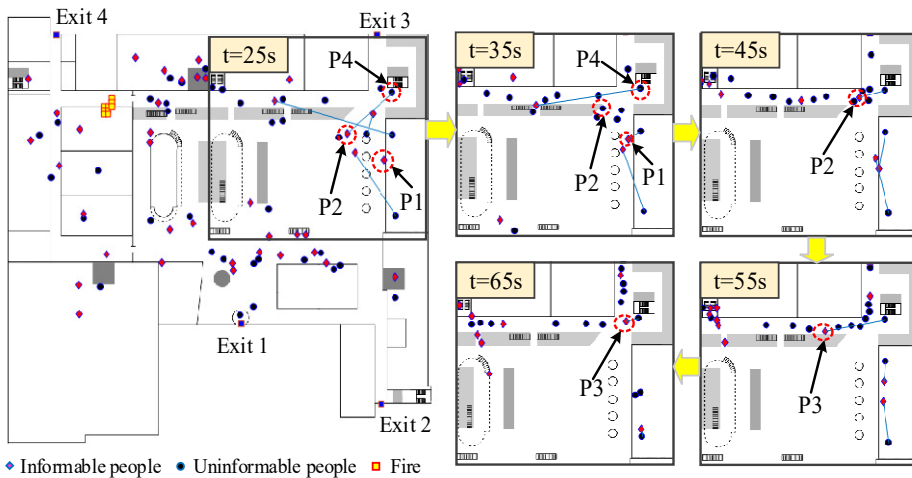
their neighboring sub-zones. In addition, people belong to ‘Area II’ were decided to escape through Exit 4 since they are very close to Exit 4 and the fire is left behind. Furthermore, two people in ‘Area IV’ were considered to escape through Exit 3 due to the people jam formed in front of Exit 1.

It’s worth mentioning that in the proposed path-finding algorithm (Algorithm 31), the optimization coefficient,  $a_{oc}$  was used as a weighting factor to enable a trade-off consideration between the length of a given evacuation path and the sum of impact indicators of situations happening on the path.  $a_{oc}$  was specified as -500 in the prototype for the test. When calculating the personalized optimal escape routes, a smaller  $a_{oc}$  means that the impact of situations is more weighted than the distance of paths in computation. Therefore,  $a_{oc}$  should be specified by application designers according to the practical needs.

In the simulation, both informable people and uninformable people were concerned. It means that the generated personalized escape routes can only be known by people who are informable by the system. As a result, uninformable people (represented by black cycles) in ‘Area I’ may not change their paths according to the computational results and will continue the walking towards Exit 4. In addition, the simulation also assumed 10 standing people after the fire alarm started. The people who are uninformable and standing (e.g. the people in ‘Area V’) may encounter dangers and may be the last ones of the evacuation process. The above-mentioned issues were considered to be solved by requesting help-givers to inform these uninformable people in person, in order to achieve a group-level optimization. As a demonstration, the latter scenario (requesting help-givers to find and inform uninformable standing people) was simulated. The solutions concerning group-level optimization were developed by using Algorithm 34. The process of computation is detailed as follows.

Figure 4.28 shows a typical scenario showing the computational results. Focusing on the right upper corner of the concerned space, the changes of the simulated real-life scenario from  $t = 25$  s to  $t = 65$  s were focused. It was assumed that the prototype started to compute at  $t = 21$  s, which means that the first set of personal escape solutions was generated at  $t = 25$  s. The computation results with regard to the determined relations between help-givers and help-recipients are shown, which are represented by the blue lines connecting diamonds and circles. The help-givers were informed to find the help-recipients and to move towards exits with them together. It’s worth mentioning that P1 was an informable person who did not move after the fire alarm started. For this reason, the system did not consider P1 as a help-giver at  $t = 25$  s. Instead, P1 was given the personalized optimal escape route leading to Exit 3.

After the escape solutions were sent to the concerned informable people, the mechanism monitored the actual behaviors of them and judged if they are really



**Figure 4.28.** A typical scenario showing the results of generating group-level optimal escape solutions in dynamic context

following the given solutions. At  $t = 35$  s, the prototype system found that there was a large deviation between the expected route of P2 and the developed action plan for P2. It means that P2 did not move towards the target help-recipient (P4). Thus, P2 was considered as an informable but disobeying person. Another informable one (P3) was selected to help P4. At the same time, P1 was instructed to move towards Exit 3 at  $t = 25$  s and P1 exactly did what was expected between  $t = 25$  s and  $t = 35$  s. Accordingly, P1 was considered as an informable person and a help-recipient was given. At  $t = 45$  s, P2 was considered as informable again and requested to help P4. This is because that from  $t = 35$  s to  $t = 45$  s, P2 did as the same as the given escape solution (move towards Exit 3) by coincidence. However, this coincident situation was recognized at  $t = 55$  s. The completion of the help request by P3 can be observed at  $t = 65$  s.

The convergence property of the implemented GA-based approach is shown in Figure 4.29, in which different population sizes ( $N_p$ ) were considered and used for generating escape solutions: 100, 200 and 300. A fast convergence speed can be observed. In addition, the computation times of the tests were 1.87, 3.06 and 4.53 seconds, respectively. Notably, the path-planning algorithm makes use of the computational result of Algorithm 31, which computes the optimal escape paths among SLMs in the concerned space with a consideration of the impact of dynamic context. Therefore, the experimental results indicate that the implemented GA-base algorithm is effective for finding the escape solutions for multiple people to achieve a group-level optimization in dynamic context. Since (i) the convergence property in all conditions were similar, (ii) the computation time was very sensitive to the population size, the population size and the maximum

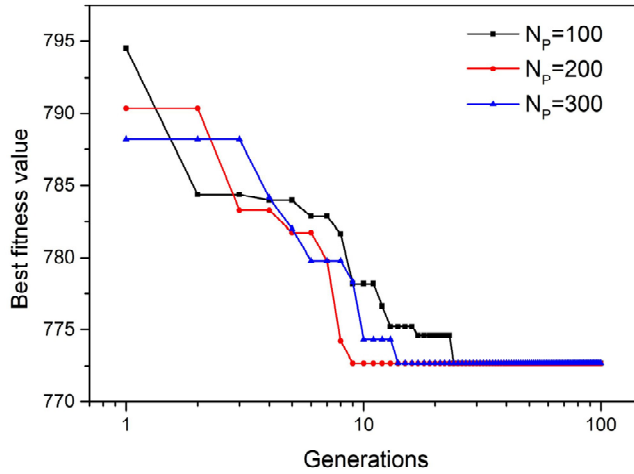


generation of GA were set to 100 and 100 in the prototype, respectively.

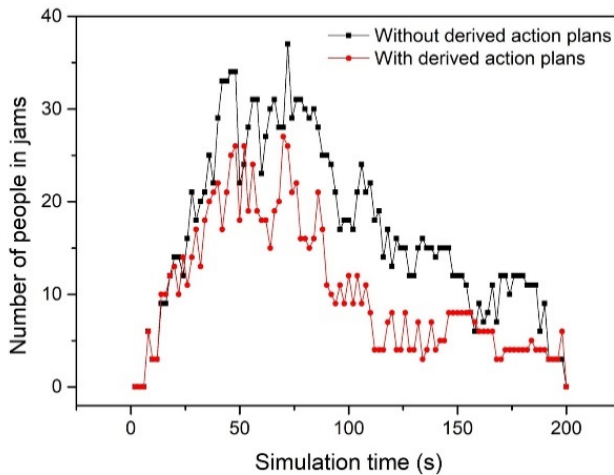
The specific objective for deriving short-term action plan was to keep people away from the negative situations as much as possible. In this way, the number of people involved in people jams was expected to be reduced. To validate the

capability of the reasoning mechanism at handling dynamic context information, a comparative study was conducted, in which two cases were considered. In the first case, every 5 seconds in simulation time, informable people were given both generated escape solutions and derived action-plans, while in the second case only escape solutions were given with the same informing frequency. The basic settings of the simulation were as the same as those proposed above. 40 informable people were included and all of them were set to follow the given instructions. In the test, the number of people in jams with and without informing the derived action plans was compared. The result of the comparison is presented in Figure 4.30.

It can be seen that after the fire alarm started, people jams were formed and the people number in the jams increased sharply. From  $t = 20$  s



**Figure 4.29.** Convergence property of the implemented GA-based path-finding approach

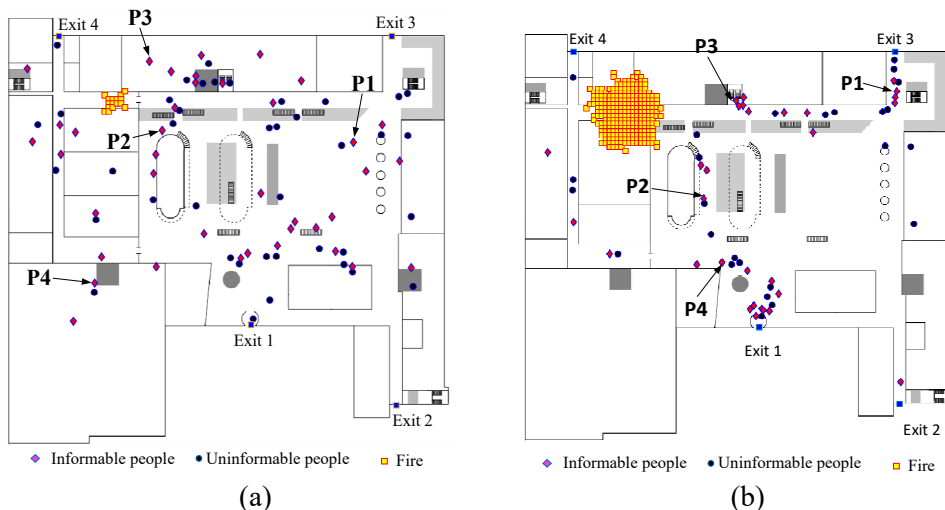


**Figure 4.30.** Comparison of the number of people in jams with and without providing information about the derived action plans

to  $t = 40$  s, there was no significant difference between the compared two cases. This is because within this period of time, people start to aggregate and many small people jams are formed. In this context, execution of the derived action-plans by people (e.g. reduce the speed of motion) may cause the formation of new (small) people jams. After  $t = 40$  s, the people number in jams in the case of informing is smaller than that in the case of not informing. After  $t = 90$  s, the number in case of ‘informing action plans’ is almost a half of that for the case of ‘not informing the action plans’. The experimental results have shown that the proposed reasoning mechanism can help to resolve people jams faster using the action plans. It is also better in terms of preventing the forming of people jams, as shown by the differences in the range of  $t = 40-90$  s. Based on the results of the test, the effectiveness of the proposed approach to dynamic context information management is proved. The expectation of the function has been achieved.

#### 4.5.4. Testing the message construction module

To test the functionality of the algorithms belonging to the MCM, four people (marked in Figure 4.31) were selected from the varying scenario to see what messages they can receive. To compare with the proposed MCM, another message construction approach was also implemented in which static context information (SCI) was used. It considered the location of the fire only and neglects the proliferation of fire and changes of people jams. Both approaches were used to generate messages by considering the completely same scenario. The personalized messages generated for the four concerned people based on the proposed MCM are shown in Table 4.4, whereas Table 4.5 presents the messages generated by using the SCI. The numbers contained in the messages are the



**Figure 4.31.** The scenario simulated for the testing of the MCM: (a)  $t=30$ s, and (b)  $t=90$ s

**Table 4.4.** Personalized messages generated based on the proposed MCM

Messages generated for the concerned people	
At t = 30s	At t = 90s
P1 Fire is in the corridor of the C block. Fire proliferates slowly. Please keep the speed of motion. Please inform P_ID_38 about the fire. The path is 54>>62>>P_ID_38>>63>>65>>Exit 3.	Fire is in the corridor of the C block. Fire proliferates slowly. Please keep the speed of motion. The path is 65>>Exit 3.
P2 Fire will be close to you! Fire proliferates slowly away from you! Please speed up the motion! The path is 24>>25>>35>>38>>42>> Exit 1.	Fire is in the corridor of the C block. Fire proliferates slowly. Please keep the speed of motion. The path is 25>>35>>38>>42>>Exit 1.
P3 Fire will be close to you! Fire proliferates slowly towards you! Please move ahead on the left. The path is 27>>28>>29>>30>>45>> 46>>61>>63>>65>>Exit 3.	Fire is in the corridor of the C block. Fire proliferates slowly. Please speed up the motion! The path is 29>>30>> 45>> 46>>61>>63>> 65>>Exit 3.
P4 Fire is in the corridor of the C block. Fire proliferates slowly. Please keep the speed of motion. The path is 19>>18>>36>>38>> 42>>Exit 1.	A people jam (12 people) is at the exit ahead. People number is increasing. Please reduce the speed of motion. The path is 38>>42>>Exit 1.

predefined spatial landmarks in the simulation. In a real-life scenario, they can be turned into real landmarks with meaning.

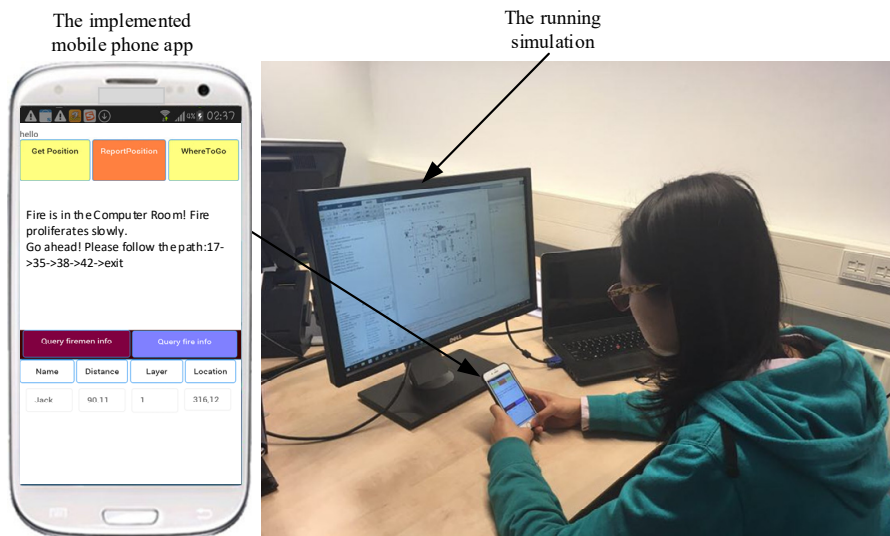
It can be seen from the results that the messages generated based on DCI contain not only the information about the most relevant situation of concerned people but also the information related to the change of the situations. In addition, when people were in emergency situations, e.g., P2 and P3 at  $t=30s$ , the messages generated by the MCM provided adequate information for people to know about their personal circumstances, e.g. close to the fire. When a person is in a relatively safe situation, e.g., P4 at  $t=90s$ , the MCM selected the people jam to inform. This is because the person was far away from the fire and the RI of the fire to P4 was lower than that of the people jam formed in front of Exit 1. In addition, information about the changes of the situation was also included, e.g. the people number is increasing, which is a piece of additional information and may help the person to make better judgments. On the other hand, the messages generated based on the static context, contain limited information with regard to the change of the scenario. Although the most critical information was contained in the messages, e.g. where the fire is and what the suggested personal escape solution is, the messages do not provide sufficient information about the personal context when the fire is not important to the person, e.g. P4 at  $t=90s$ .

**Table 4.5.** Personalized messages generated based on static context information

Messages generated for the concerned people	
At t = 30s	At t = 90s
P1 Fire is in the corridor of the C block. Please inform P_ID_38 about fire. The path is 54>>62>>P_ID_38>>63>>65>>Exit 3.	Fire is in the corridor of the C block. The path is 65>>Exit 3.
P2 Fire is in the corridor of the C block. The path is 24>>25>>35>>38>> 42>>Exit 1.	Fire is in the corridor of the C block. Please go ahead. The path is 25>>35>>38>>42>> Exit 1.
P3 Fire is in the corridor of the C block. Please leave the room. The path is 27>>28>>29>> 30>>45>>46>>61>> 63>>65>>Exit 3.	Fire is in the corridor of the C block. Please turn left. The path is 29>>30>>45>>46>> 61>> 63>> 65>>Exit 3.
P4 Fire is in the corridor of the C block. Please leave the room. The path is 19>>18>>36>> 38>>42>>Exit 1.	Fire is in the corridor of the C block. Please go ahead. The path is 38>>42>>Exit 1.

The results of testing have shown that personalized messages have been generated based on the applied template-based MCM. Each of the constructed personalized messages contains information with regard to the (i) the most relevant situation, (ii) changes of the situation, (iii) the action plan to be performed, and (iv) the generated escape solution.

Information contained in the messages is more sufficient than that contained in



**Figure 4.32.** The experimental setup for human evaluation

the messages which are generated by using SCI. In addition, the MCM makes use of the calculated personal danger level of people and considers it as a basis for selecting different templates for message construction. This enables representing the same content, but with different ways of rhetoric. Therefore, the generated messages adaptively represent the personal context of the (assumable) people, which is assumed to support decision-making by people in emergency situations.

Due to the reason that the messages generated by the prototype are designed to be read and interpreted by actual users, the feedback from the human evaluation is valuable. The ‘informing power’ of the generated messages by actual users was explored. The actual users are assumed to be placed at different positions in the simulated physical space. To evaluate the ‘informing power’ of the generated messages, questions were designed and given to the actual users to answer. The results of their answers were used as the indicators showing the goodness of the generated messages.

As a pilot study, altogether 18 human subjects (11 males and 7 females) were asked to evaluate the messages generated based on both approaches. The subjects were master students and Ph. D. students. The experimental setup is shown in Figure 4.32. Before the evaluation was taken, the subjects were asked to act as one of the people in the simulated fire evacuation scenario. The personalized messages generated by both approaches were shown to them on an implemented Android App during the play of the animation. After these, a focused questionnaire was designed for them to support collecting their opinions. The questions contained in the questionnaire are shown in Table 4.6.

The questionnaire contained five (informative) statements. For each statement, the subjects were asked to select one of the reflections based on their own judgment, including (i) grade 1: completely disagree, (ii) grade 2: partially disagree, (iii) grade 3: no sense, (iv) grade 4: partially agree, and (v) grade 5: totally agree. Actually, these statements were developed representing five aspects with regard to the quality of the generated message (QoM), which can be seen in Table 4.6. The

**Table 4.6.** The questionnaire designed for the human evaluation

Aspects of QoM	Statements	Grades
Usefulness	The messages are necessary for people to escape from hazardous situations.	(1-5)
Sufficiency	The messages contain sufficient information about the context of the people.	(1-5)
Informativeness	The information contained in the messages is clear and representative.	(1-5)
Added-value	The messages reduce the anxiety of people in hazardous situations.	(1-5)
Convincingness	People will obey the instructions.	(1-5)

results of the human evaluation are shown in Table 4.7. The evaluation results of each aspect include the mean value and the sample standard deviation of the 18 grades given by the subjects.

It was observed based on Table 4.7 that the mean values provided for the fourth statement, added-value, were the lowest in the case of both approaches. It indicated that providing personalized messages was of limited effectiveness in terms of eliminating the anxiety of people in a hazardous situation. In addition, the mean values concerning the fifth statement, convincingness, are the highest in both cases. It means that the involved subjects indeed tended to rely on the messages given to them. Most of them wanted to obey the instructions in the hazard-intense situation. Furthermore, in the case of the messages generated by the proposed MCM, the mean values for most of the considered aspects were higher than those generated based on the SCI, except for the fourth one. It means that the subjects preferred the messages generated based on the DCI except when the anxiety of people is considered. The reason could be explained as follows. There was more content contained in the messages generated by the MCM than those generated based on the SCI. People in hazardous situations may be more anxious when they are reading messages with a lot of information in comparison with circumstances when they provided with concise messages.

Particularly, in the answers concerning the fifth aspect, there were two ‘Partially agree’ options chosen by the subjects for the messages generated by the MCM, while one ‘Partially agree’, two ‘No sense’ and one ‘Partially disagree’ option was chosen for the messages generated based on the SCI. It means that some subjects tended to disregard the messages provided to them and may disobey the instructions. However, when the DCI is contained in the messages, the subjects showed a stronger will to obey the instructions, comparing to the situations that they read messages contain SCI information only. Messages generated based on SCI may cause suspicion of the people with regard to the correctness of the messages. On the other hand, some of the results show large deviations for the messages generated by SCI, which means that the opinions of the subjects were diverse, while their opinions on the messages generated based on the DCI were

**Table 4.7.** Results of the human evaluation

Aspects of QoM	Messages generated based on DCI		Messages generated based on SCI	
	Mean	Deviation	Mean	Deviation
Usefulness	4.33	0.84	4.22	0.65
Sufficiency	4.67	0.49	4.06	1.00
Informativeness	4.56	0.51	2.94	1.00
Added-value	3.61	0.85	3.78	1.00
Convincingness	4.89	0.32	4.56	0.92

more consistent. These results indicate that except for usefulness the difference between DCI and SCI based messages is more significant than represented by the difference of the mean value.

In the conducted human evaluation, 18 human subjects were involved. The sample size is only suitable for a pilot study. The obtained findings with regard to human opinions in the difference between the proposed MCM and the SCI-based messaging are not general enough. In addition, all the considered human subjects were professionals (master and Ph.D. students), who are able to clearly understand the objects of the test (e.g. static context and dynamic context, and fire evacuation), and give their answers after thoughtful consideration. Thus, we believe the aggregated answers were reliable and the findings were credible. However, the main drawback is that the involved human subjects were not diverse enough for drawing generalizable conclusions. All of the above-mentioned limitations will be considered as a basis for a more insightful investigation.

#### **4.5.5. Testing the prototype as a whole**

To have an overall view of the implemented prototype as a whole, a comparative study has been done to compare the simulated outcomes with different input data. This was considered to see if the computation of the prototype is independent of the input data. Another objective is to explore if different input data will influence the performance of dynamic context management. Two tests were taken, including (i) testing the prototype with different initial positions of the entities, and (ii) testing the prototype with different proportions of informable people.

##### **4.5.5.1. Testing the prototype with different initial positions of entities**

The main objective of this test was to prove the proposed computational mechanism is able to handle dynamic context with regard to diverse scenarios. Towards this end, five different escape scenarios were generated in the concerned physical space. The number and attributes of entities in the scenarios were the same as those presented in Section 4.2.2. However, the initial positions of people were different (randomly generated). In the action-plan deriving module, both long-term solutions and short-term actions were generated towards different objectives of dynamic context management. In order to compare the effectiveness of each of them, three different informing strategies were applied for each escape scenario:

- Strategy I: Informing the people about their (optimal) personalized escape solutions,
- Strategy II: Informing the people about the generated paths with the objective to achieve group-level optimization (e.g. by requesting someone to help), and

- Strategy III: Informing the people about the generated group-level optimal solutions and the action plans.

According to the requirements of the IFEG system, the quantitative evaluation indicators were: (i) the average escape time of all people (AET), (ii) the escape time of the last person in the evacuation process (TET), which is the worst case of evacuation, (iii) the average length of escape paths of all people (ALP), and (iv) the average value of people number in jams (APN), which is calculated by the following equation:

$$APN = \frac{\sum_{i=1}^{TST} NPJ_i}{TST} \quad (4-13)$$

where TST is the total simulation time,  $NPJ_i$  is the number of people involved in jams at a given moment. To calculate the APN, we focused on the period of time from  $t=1s$  to  $t=200s$  in all simulations. The above-mentioned three strategies were applied to handle the generated five scenarios. As a comparison, the case of ‘Not informing’ was also simulated. In total, 20 simulations were done. The four evaluation indicators were measured in the respective simulations. Calculated for each applied strategy, the mean values and the standard deviations (SD) are

**Table 4.8.** The results of testing the implemented prototype with different initial positions of the entities

Evaluation indicators	Not informing		Strategy I		Strategy II		Strategy III	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
TET (s)	500	0	500	0	234.8	20.4	234.7	20.3
AET (s)	146.6	4.3	119.2	5.2	100.4	5.6	100.5	5.8
ALP (m)	46.2	4.1	48.5	4.0	49.8	4.8	49.8	4.8
APN (people/s)	16.4	2.4	16	2.3	16.8	2.1	10.3	1.5

shown in Table 4.8.

The escape time of the last person in the evacuation process (TET) was set to 500 in ‘Not informing’ and in ‘Strategy I’. The above value is a fixed extreme value. It is applied because the simulation assumed 10 standing people, and in both cases, the people who were uninformed and standing were unable to escape from the space (i.e. devoured by the fire). By informing people with the generated group-level optimal solutions (Strategy II and Strategy III), the TET and AET values were sharply decreased and all people were evacuated safely. In addition, ‘Not informing’ has the lowest ALP value. Because uninformed people may be suggested to move towards the second or the third nearest exits according to ‘Strategy I’ due to the impact of situations (e.g. people belonging to the ‘Area I’ and ‘Area III’). When people were instructed to inform help-recipients by applying Strategy II and Strategy III, the ALP values were even bigger.



Particularly, the deviation values of ALP in the cases of ‘Strategy II’ and ‘Strategy III’ were bigger than those in the ‘Not informing’ and ‘Strategy I’ cases. The reason is that help-givers in cases of ‘Strategy II’ and ‘Strategy III’ may have a chance to inform help-recipients on their personalized (optimal) escape path, or to inform some help-recipients with certain detours. This depends on the positions of the help-givers and the help-recipients. However, in cases of ‘Not informing’ and ‘Strategy I’, people escaped depending on their own situations, and the mentioned detours unlikely happen. Furthermore, by informing people about the personalized action plans, APN kept to a low value with a small deviation. It means that the implemented algorithms worked well in all scenarios.

Simulation results have shown that the proposed approach is effective to manage dynamic context of people in time-varying scenarios. The correctness of the computation conducted by the prototype system has been proved in the simulated scenario. In addition, the implemented prototype was lack of sensitivity to the initial positions of people. For all generated scenarios, the functional expectations were achieved.

#### **4.5.5.2. Testing the prototype with different percentages of informable people**

In a real-life situation, an I-CPS may not be able to communicate with all people for various reasons. Some of them may not have the informing devices at hand, or they may be not aware of the existence of messages at the moment of informing. Only a part of people can be informed at a given moment. It implies that the percentage of informable people may influence the execution of the generated solutions and action-plans in this case, and consequently, influence the performance of dynamic context information management by I-CPSs. The objective of the second test is to clarify the influence of the ratio of informable and uninformable people on the proposed computational mechanisms in handling dynamic context.

In the test, the initial positions of people were considered as the same as those shown in Figure 4.4. Different ratios of informable people were randomly selected from the scenario, ranging from 0 (0%) to 80 (100%) with an increment of 10 people. In total, 9 simulations were taken. It was also assumed that all informable people will exactly obey the given instructions and will move with the speed of 0.75m/s. Except for the case of 80 informable people, 5 uninformable people were set as standing and waiting for help from informants. To assess the goodness of dynamic context management, the following performance indicators were used:

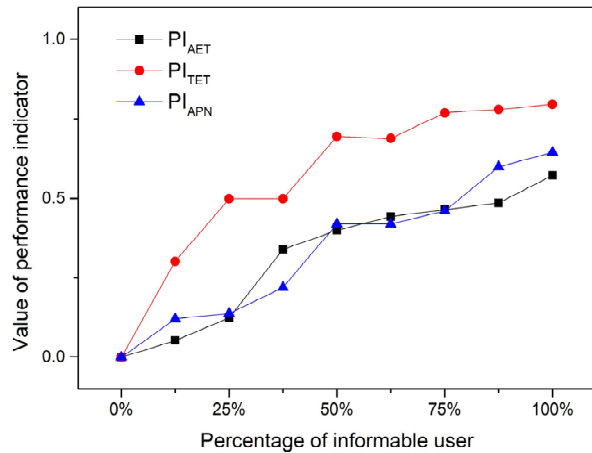
$$PI_{TET} = \frac{TET_{ws} - TET_a}{TET_{ws} - TET_{op}} \quad (4-14)$$

$$PI_{AET} = \frac{AET_{ws} - AET_a}{AET_{ws} - AET_{op}} \quad (4-15)$$

$$PI_{APN} = \frac{APN_{ws} - APN_a}{APN_{ws} - APN_{op}} \quad (4-16)$$

where  $TET_{ws}$ ,  $AET_{ws}$  and  $APN_{ws}$  are the values calculated from the worst case concerning the given evacuation scenario, respectively.  $TET_{op}$ ,  $AET_{op}$  and  $APN_{op}$  are the values calculated by considering the optimal escape case of the same.  $TET_a$ ,  $AET_a$  and  $APN_a$  are the values calculated in each of the running simulations. The worst escape case was defined as a simulation case without informing any people. The optimal escape case assumed that all people move towards and escape through their individual nearest exit, and the influence of fire and people jams were neglected. These two cases were used as reference cases and the proposed indicators imply how much the optimal case (or the worst case) can be achieved with a certain percentage of informable people. According to the definition and the simulated scenario (Figure 4.4),  $TET_{op}$ ,  $AET_{op}$  and  $APN_{op}$  were calculated as: 197.3s, 95.5s and 0people/s, respectively.  $TET_{ws}$ ,  $AET_{ws}$  and  $APN_{ws}$  were calculated as: 500s, 166.3s and 17.5people/s, respectively. Then, the results of the simulations are shown in Figure 4.33.

In general, all three calculated performance indicators were smaller than 1 in any of the conditions. This means that the parameters considered in the optimal case, e.g. no people jam is formed, can hardly be reached by changing the percentage of informable people only. In addition, all indicators monotonically grew as the proportion increased. It means that the more the informable people considered, the better the management of the dynamic context by the prototype will be. The best performance in terms of all three indicators was found in the condition that all concerned people are informable and controllable. In that sense, the I-CPS is equivalent to a controlling CPS.



**Figure 4.33.** The values of the various performance indicators under different percentages of informable people

In detail,  $PI_{TET}$  dramatically increased from 0 to 0.69, when 50% of the people were designated as informable. After that, a slight improvement could be seen when the proportion changed from 50% to 100%. It implies that a small percentage of informable people effectively facilitate the reduction of total evacuation time. This was achieved by utilizing informants to achieve group-level optimization. It also implies that the method also works in real life assuming that only a small percent of people is informable and follows the instructions given by the system. In addition,  $PI_{AET}$  and  $PI_{APN}$  were linearly increased as the percentage of informable people increased. A small percentage of the informable people (below 25%) had limited ability for people jams management.

The experimental result implies that the percentage of informable people is a very crucial factor in the case of I-CPSs operating in dangerous circumstances. This is anyway what would be assumed based on commonsense thinking. Also important is that it strongly influences the functional performance of dynamic context management. In addition, by applying 50% informable people, considerable performance can be achieved concerning the specified requirements of the indoor fire evacuation application.

## **4.6. Performance testing of the prototype**

### **4.6.1. Objectives of the simulation-based performance testing**

Normally, performance testing is used to test the performance of a given software or infrastructure under different circumstances. It can be used to examine responsiveness, stability, reliability, computation speed, and resource usage [12][13]. As indicated in the literature, typical approaches applied for performance testing are load testing, stress testing, spike testing, endurance testing, scalability testing, volume testing and configuration testing [14][15]. In the above approaches, typical measurements are response time, data transfer rate, network bandwidth, quality of results, throughput, workload efficiency, and reliability.

In terms of I-CPSs, since dynamic context may change rapidly, the generated solutions and messages should be strongly related to the referred context. Computational speed is a crucial factor for assessing the real-time computation capability of the DCIP-M. In addition, in many application domains of I-CPSs, for instance in traffic management, the number of entities defining the dynamic context may suddenly change. As our DCIP-M performs computation only with the changes of entities and their relationships, it is meaningful to investigate how our prototype copes with the sudden changes of the amount of input data streaming during run-time. As a primary prototype, our focus is to achieve quasi-real-time processing of DCI in the context of I-CPSs. Based on these, the

performance testing concerned in this sub-chapter addressed on the empirical properties of the implemented prototype, including (i) the computation speed, which implies the ability of real-time computation and (ii) computation stability, which refers to the ability for handling varying contexts. The implemented prototype was tested based on the simulated indoor fire evacuation scenario. All tests were run on a normal computer, using Intel 2.50 GHz Core i5 processor and 8 GB RAM.

#### **4.6.2. Setting the targets for the performance testing**

The first target of the performance testing was to explore how the prototype will perform under different load conditions. This is because the number of entities, their attributes and relations representing the dynamic context may be different in different application cases. To generate context semantic knowledge, all the entity attributes and relations should be checked. Thus, there is a strong relationship between the entity number and the computational speed in processing. Therefore, the first test was designed to check how the entity number influences the computational speed. In the test, the number of entities was considered as the load variation. The computational time that is spent by the implemented functions and algorithms were used as the performance indicators. Two performance indicators were specified, including (i) the time for computing a function and (ii) the time for executing an algorithm.

To evaluate the stability of the prototype in computation, the second target of the performance testing aimed to obtain insight into the consequences of sudden changes (increases and decreases) of computational loads. Since the entity number was considered as the computational load, sudden increases and decreases of entities numbers in the run-time of computation were applied as variations in the second test. Step response of the prototype to the variations was checked. To assess the stability of the prototype for handling dynamic loads, (i) computation time and (ii) overshoot rate were considered as the performance indicators.

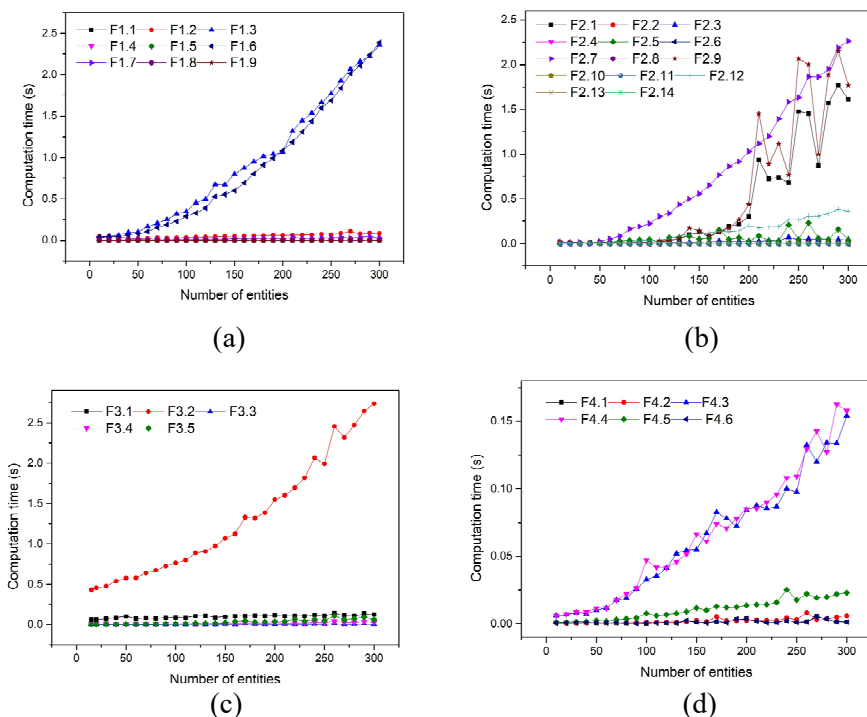
#### **4.6.3. Testing the empirical performance of the prototype under varied loads of computation**

In the concerned scenario, the number of people was varying from 5 to 295 with an increment of 10. Since there were four exits and a fire considered in the scenario, the total number of entities varied from 10 to 300 with an increment of 10. Accordingly, 30 different conditions were included. For each condition, five simulation runs were completed. The initial positions of people were randomly generated in every simulation run. In addition, all people were considered as informable people and set to obey the instructions. The computation times elapsed by the implemented functions were aggregated. The average value of the 5 simulations was believed as the final result for a concerned condition.

The testing results are shown in Figure 4.34. In general, there are several functions that are very sensitive to the number of entities, they are listed as follows.

- F1.3: Calculate the distances among all entities in the SFR-matrix.
- F1.6: Calculate the distances among entities in a list and all other entities.
- F2.1: Infer situations based on the CIR-cube.
- F2.7: Predict the future states and relations of entities.
- F2.9: Predict newly formed situations in the near future.
- F3.2: Plan a path for individual entities to the expected positions.

Since most of the computation time of the modules was spent by these functions, they are considered as the performance bottleneck of the implemented prototype. In the MCM, although the computation time of F4.3 and F4.4 are dramatically bigger than that of the rest functions, the total computation time of the MCM was less than 0.4s for constructing 300 personalized messages. This is much less than



**Figure 4.34.** The computation time elapsed by the implemented functions: (a) functions in the IRM, (b) functions in the KIM, (c) functions in the ADM, and (d) functions in the MCM

that of the above-listed functions. Actually, a linear relationship between the entity number and the needed computation time for message construction can be observed. This is because that the personalized messages have to be constructed in a one-by-one manner. The results proved that the template-based message generation approach is very time-efficient and can be used to facilitate personalized informing.

The computation time of F1.3, F1.6, and F2.7 steadily increased as the entity number increased. As a contrast, obvious fluctuations can be observed from the testing results of F2.1 and F2.9. This implies that the computation time of F2.1 and F2.9 not only related to the processed entity number but also has a strong relationship with some internal variables, which are situation-dependently generated. To investigate the reasons for the observed phenomenon, an algorithm-level analysis with regard to these time-consuming functions were further conducted. It's worth mentioning that both F1.3 and F1.6 contained only one algorithm, A4, which is also used by F2.7. In addition, both F2.9 and F2.7 computed based on A9 and A10. Accordingly, F2.1, F2.7, and F3.2 were focused and the time elapsed by the calculation with the related algorithms were aggregated. The settings for the algorithm-level testing were the same as for the function-level testing. The results are presented in Figure 4.35.

It can be seen from the result that, there are only three time-consuming algorithms in the prototype. Each of the algorithms is analyzed individually as follows.

- A4: Calculate the distance between two entities.

According to the approach to a digital representation of the physical space, every entity is located within a subzone. Every sub-zone may have multiple SLMs linking to other sub-zones. It means that if the two concerned entities are located in different sub-zones, the distance in between should be the length of the shortest path. The algorithm has to compute all the possible paths and check which SLM(s) belong to the shortest path. Therefore, the worst-case time complexity is  $O(m^2n^2)$ , where  $n$  is the total number of entities,  $m$  is the actual number of SLMs in a sub-zone among all sub-zones. On the other hand, if two entities are in the same sub-zone, then the algorithm only calculates the linear distance in between. The best-case time complexity is  $O(n^2)$ . The maximum  $m$  in the running simulation was set as 11, the computation time of distance calculation quadratically increased as the number of entity increased.

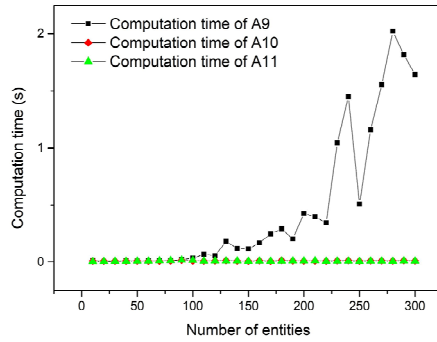
- A9: Infer floating people jams.

To infer floating people jams, the designed algorithm included two steps. Firstly, the distances between any two people were checked in order to generate a basic unit for inferring people jams. Then, the relations among the basic units were computed to see if several small floating people jams can be merged to form a

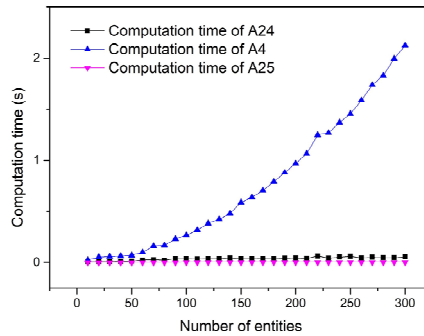
bigger people jam. Therefore, the worst-case time complexity is  $O(0.5n^2 + m^2)$ , where:  $n$  is the number of people, and  $m$  is the number of the basic unit inferred in the first step. There are two polynomials in the time complexity, which means that two factors influence the time complexity of the algorithm. This is the main reason for the large fluctuations. This algorithm should be further improved to enhance the stability and reduce the complexity for practical usages.

- A34: Revise the paths of help-givers to enable help-recipients to get informed.

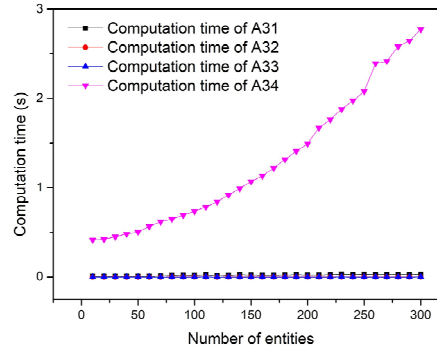
In Algorithm 34, the genetic algorithm was applied to generate the group-level optimal escape solutions. To calculate the fitness values of a chromosome, the length of the optimal route between potential help-givers and potential help-recipients should be computed. This is different from the distance calculation between two entities where the nearest path was considered. Therefore, the worst-case time complexity of the implemented GA is  $O(NKm^2n_gn_r)$ , where:  $N$  is the size of the population,  $K$  refers to the number of iterations,  $n_g$  is the number of potential help-givers,  $n_r$  is the number of potential help-recipients, and  $m$  is the maximum number of SLMs in a sub-zone among all sub-zones. In the test, population size was set to 100, while 100 iterations were considered in a



(a)



(b)



(c)

**Figure 4.35.** The computation time taken by the time-consuming algorithms: (a) algorithms in F2.1, (b) algorithms in F2.7, and (c) algorithms in F3.2

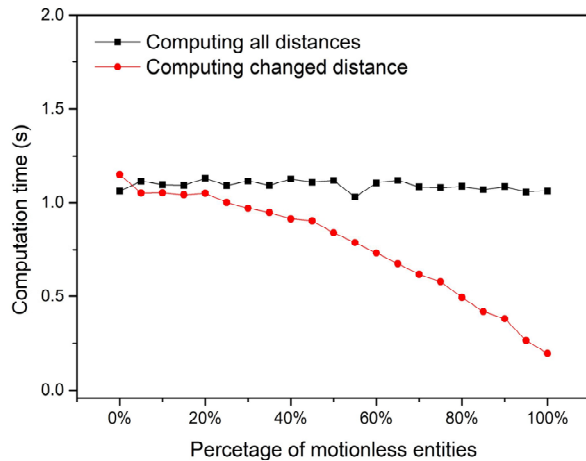
computation. Therefore, the relation between the computation time of A34 and the entity number is very similar to that of A4.

Based on the performance testing of the implemented functions and algorithms, it can be seen that the calculation of the distances among entities is very time-consuming.

Different to A34 and A9, which are application-dependent algorithms, the distance calculation is a general algorithm and can be

applied by multiple functions and for multiple applications. Therefore, we believe that this algorithm is the bottleneck for using the context semantics based on entity relations. Actually, in the proposed DCIP-M, from the second computational cycle only changed distances were recomputed. The principle was: if the positions of two entities do not change, then the distance in between does not change. We assumed that by focusing on the changed distance, the computation time for distance calculation could be reduced. To validate the effectiveness of this particular functional design, testing was conducted, in which 200 entities were considered in the simulation. Different percentages of movable and motionless entities were set. The computation time spent by computing all distances, and computing only changed distances among the 200 entities were compared. The result of the test is shown in Figure 4.36.

It can be seen, when a smaller percentage of entities do not change their positions, the proposed DCIP-M can significantly reduce the computation time for distance calculation. The unchanged distances can be obtained from the previous calculation (the SFR-matrix generated in the previous computational cycle). This strategy can be applied for applications where a majority of entities are motionless, e.g. car parking management system. Although there are many possible opportunities for further improvement, the proposed solution is a step towards quasi-real-time processing of dynamic context information.



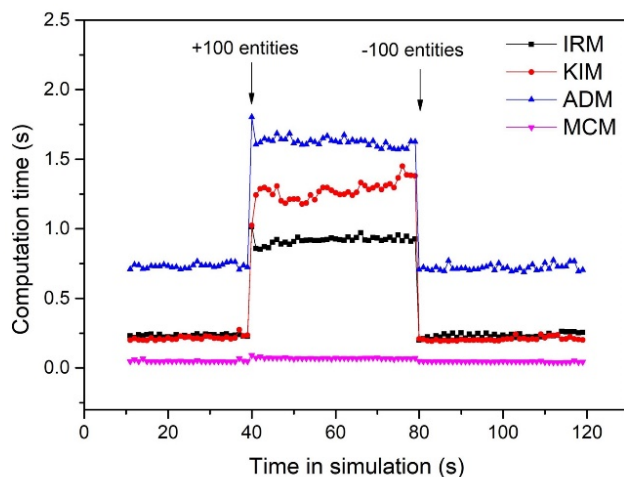
**Figure 4.36.** A comparison of the computation time spent on computing all distances and computing changed distances among 200 entities



#### 4.6.4. Testing the empirical performance of the prototype under dynamic loads of computation in run-time

To have a general overview of the sensitivity of the implemented modules to dynamic loads, a step experiment was done, in which a sudden increase and a sudden decrease of computation loads in run-time were considered. To this end, the input data was set as follows: (i) At  $t=0s$ , the initial number of entities in the simulation was set to 80. (ii) At  $t=40s$ , 100 additional entities entered into the scenario at randomly generated positions. (iii) At  $t=80s$ , 100 randomly selected entities suddenly disappeared from the scenario. (iv) At  $t=120s$ , the simulation ended. All the additional included entities were informable people and their speed of motion was set to  $0.5m/s$ . In addition, people cannot escape through the exits, which maintained the number of people to the designed values. Then, the computation times taken by the modules were aggregated over the concerned period of time.

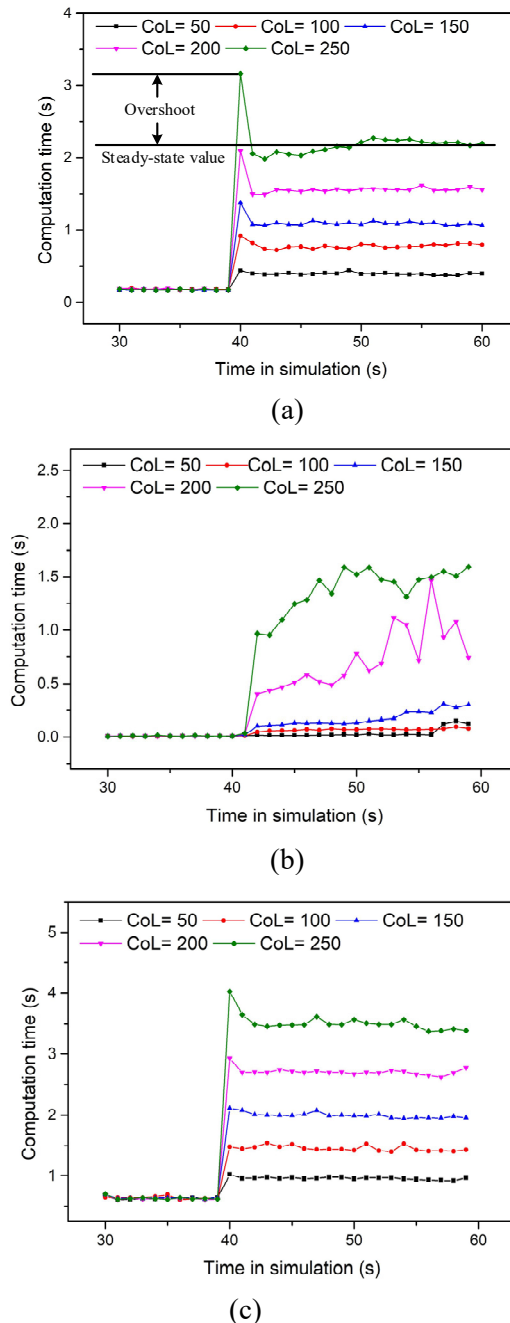
Figure 4.37 shows the results showing how the computation time of the modules changed when the dynamic loads were applied. It can be observed that, when the amount of input data suddenly increased, IRM, ADM, and MCM had a fast response. A peak value appeared in the computation time of these modules before the computation time entered into a steady period. However, the computation time of KIM gradually increased to its peak value with 4 simulation steps. After that, a number of fluctuations were produced. On the other hand, when 100 entities suddenly disappeared from the scenario, the computation time of all the four modules went to a normal level immediately, as the same as the starting period (from  $t=20s$  to  $t=40s$ ) of the simulation. No overshoot was observed. The results indicated that the IRM, ADM, and MCM were sensitive to sudden increases in computational load, while their responses to sudden decreases were quite stable. As discussed in the previous section, the most time-consumable function in KIM is the one used to infer people jams.



**Figure 4.37.** Computational performance of the prototype under dynamic loads

Since its computation time depends on the actual scenario, a fluctuation can be observed on its response to dynamic loads as well.

As underpinned by the findings from the previous section, there are several computationally time-consuming functions, including F1.3, F1.6, F2.1, F2.7, F2.9, and F3.2. Since these functions consumed a majority of the computation time of the prototype and are very sensitive to the number of entities. The overshoots in their responses to the increased loads may have a great impact on the performance of the entire prototype. The computation time of the other functions in the prototype was not comparable, thus the overshoots in their responses can be neglected. In addition, F1.3 was not considered since it is executed only once in a computation (in the first computational cycle). F1.6, F2.1, and F3.2 were focused on since all time-consuming algorithms were included. Accordingly, the step responds of F1.6, F2.1 and F3.2 were further investigated to see how they deal with different sizes of loads changes.



**Figure 4.38.** Performance of the selected functions under different sizes of load changes (CoL): (a) F1.6, (b) F2.1, and (c) F3.2

F1.6, F2.1, and F3.2 were tested individually by simulations. Different sizes of changes of entity number were applied at  $t=40s$  in each of the simulations. In the test, the changes of load (CoL) were 50, 100, 150, 200 and 250. The basic settings of the simulations were as the same as those mentioned above. The computation times of the functions were aggregated and shown in Figure 4.38.

Obvious transient overshoots can be observed from the responses of F2.1 and F3.2 to the applied CoL. The computation times of these functions concerning the steady-state period (from  $t = 50 s$  to  $t = 60 s$ ) were similar to the values measured in the corresponding conditions of static loads. For instance, the steady period of F1.6 in Figure 4.36.a. is about 2.2 seconds, which is similar to the computation time of A4 (the only algorithm included in F1.6) for processing the information with regard to 330 entities as a static load, as indicated by the trend of A4 in Figure 4.35.b. Therefore, the overshoots resulted from the sudden changes in load need more computation time for processing than static conditions. There are two possible reasons for the overshoots. On the one hand, the program needs more space in memory for caching the incremental input data. On the other hand, when the number of entity suddenly increases, new relations among entities (among new entities and between new entities and original entities) need to be computed. In addition, although no overshoot appeared in the response of F2.1, the function was not steadily executed when there were a large number of entities to be handled.

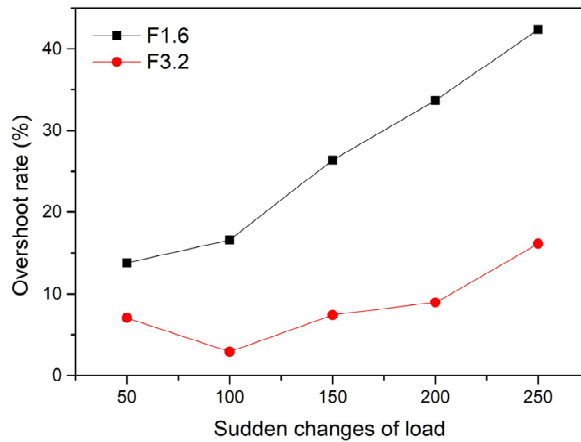
As indicated by the experimental results, different sizes of dynamic loads may produce different sizes of overshoots. The overshoot rates of F1.6 and F3.2 concerning different sizes of sudden changes of computational loads were calculated, which was defined as the amount of overshoot over the steady-state value in the response, as indicated by Figure 4.38.a. The calculated results are shown in Figure 4.39. It can be seen that when the size of load change increases, the overshoot rate increases in both functions. It means that the larger the dynamic loads are, the bigger their impact on the computational performance will be.

## **4.7. Reflections and conclusions**

### **4.7.1. Reflection on the implementation work**

Based on the implementation work, all the expected functionalities proposed in Chapter 3 were realized. After all required algorithms were implemented, a major revision of the prototype was taken for the purpose of functionality and performance testing. Because auxiliary functions that were tailored for conducting the tests (e.g. aggregating the computation time of algorithms) should be embedded into the basic functions of the software. But, they were neglected in the period of concept design. Before doing the implementation, we should specify the test cases based on the requirements collected earlier.

The integrated development environment (IDE) for software implementation was selected as Matlab<sup>®</sup>, which is an effective tool for handling matrix-based data constructs, such as the proposed SFR-Matrix and CIR-cube. It also provides interfaces to enable communication



**Figure 4.39.** The overshoot rate of the selected functions under different sizes of sudden changes in computational loads

with other development environments, e.g. Python. However, it showed a lower computational efficiency than our expectation. During the implementation work, much attention was given to the realization of the functionalities of the prototype, while the execution performance of the code was overlooked. There are many ‘for loops’ applied in the codes. The computation efficiency of the implemented functions can still be improved by using more efficient codes, e.g. using vectorized codes instead of the loop-based approach [16].

In addition, the presented work sequence and data streaming also imply the opportunity for utilizing parallel computing, which may reduce the total computational time and thereby reduces the response time to the request sent from the IFEG system. For instance, computing the distance between any two entities is an independent computation process. Distance calculation can be completed by using multiple processors at the same time. In addition, cloud computing might also be a solution for effectively obtaining the distances among entities. It can be applied for application cases that are required for processing context information with regard to a large number of entities.

#### 4.7.2. Findings from the functionality testing

In the completed work, both the functionalities of the entire prototype and the contained individual modules were tested. To represent the DCI regarding a given process, the error produced in the position sensing process is a critical factor. It influences the judgment of the sub-zone of entities and thus produces large errors in the calculation of the spatial relations (i.e. distances) among the entities. To

alleviate the sensing errors, the Kalman-filter has been used to estimate the indoor positions of people from the simulated sensing data. Based on the conducted functionality testing, the functional performance of the Kalman-filter was optimized. In addition, since the actual motion of entities in an indoor environment may always experience non-linear processes, the sampling rate of sensing plays an important role to the accuracy of indoor position estimation, which further influences the distance calculation. To enhance the validity of dynamic context information representation, a relatively high sensing frequency should be applied. However, the chosen time step in between two subsequent re-computations should be greater than the time needed for the completion of a cycle of information sensing, decision-making, and personalized informing.

Based on the conducted functionality testing of the KIM, the CIR-cube was proved to be an efficient data construct that integrates all possible states and events for DCIP in an application-independent way. It facilitates inferring situations using application-independent rules, which are designed as a combination of the contents of the CIR-cube and certain predefined conditions. In addition, taking the impact of static context (e.g. the arrangement of floor plan) and dynamic context (e.g. dynamically formed people jams) into consideration, the accuracy of context prediction can be enhanced. The inferring mechanism also includes a quantitative evaluation algorithm that elaborates the implication of situations on involved entities and the relevance of situations to the rest entities. In this way, the personal context of any entity at a given moment of time can be figured out. This is the basis for generating personalized services according to the personal context of individuals, e.g. personalized informing.

Based on the inferred semantic knowledge, personalized action plans were generated, including both long-term plans and short-term plans. The effectiveness of the developed action plans was validated through the conducted functionality testing of the ADM. In the concerned reference application case, the long-term plans were developed for the purpose of evacuating all people safely, while the short-term plans were developed to prevent people away from dangerous situations in their evacuation process. To this end, the impact of situations was first represented by a run-time constructed penalty matrix. Accordingly, both personal-level optimal escape solutions and group-level optimal escape solutions were developed. Process optimization was achieved concerning (i) the shortest average escape time of all people, (ii) the shortest escape time of the last person and (iii) the shortest average escape paths. In addition, the impact of dynamically formed situations was also used to generate an objective function, based on which the optimum short-term action plans for people to perform can be determined. In the simulated case, the number of people involved in people jams was reduced. It implies that the dynamic context information can be managed by informing people to perform certain action plans.

In the functionality testing of the MCM, personalized messages were generated, which adaptively represent the personal context of the (assumable) informable people. To test the quality of the generated messages (QoM), opinions from human evaluators were collected with regard to the usefulness, sufficiency, informativeness, added-value and convincingness. Despite the sampling size of the concerned human evaluators was limited, valuable findings were obtained. Firstly, most of the involved people believed that the proposed MCM provides more useful, sufficient, informative and convincing information about the personal context and expected actions than the messages constructed based on SCI only. In addition, when DCI is contained in the personalized messages, the involved people showed a higher level of agreement on the results of the QoM.

To test the functionalities of the overall prototype, different input data were used and several indicators were defined to enable evaluation of the goodness of the prototype with regard to dynamic context information management. It was found that the implemented prototype system worked well with different input conditions, e.g. different initial positions of entities. However, the capability for handling dynamic context largely depends on the attributes of people (e.g. informable or not). In the simulated case, 50% of informable people achieved considerable performance concerning dynamic context management.

#### **4.7.3. Findings from the performance testing**

This chapter also tests the computation performance of the implemented prototype. Several time-consuming functions and algorithms were sorted out, and their computation speeds were investigated. Three algorithms were considered as the performance bottleneck of the prototype, namely: (i) A4: Calculate the distance between two entities, (ii) A9: Infer floating people jams, and (iii) A34: Revise the paths of help-givers to enable help-recipients to get informed. Among these, A9 and A34 are algorithms dedicated to the fire evacuation application. However, A4 is a generally required algorithm for calculating the indoor distance of entities, which is the basis for handling entity spatial relations and constructing the SFR-matrix and the CIR-cube. In the proposed DCIP-M, after the second computational cycle, the IRM only recalculates the changed distances among entities, which has been proved to be an effective solution for reducing the computation time of distance calculation.

To evaluate the ability of the prototype for handling contextual changes, its response to sudden changes of computational load during run-time was investigated. It was found that the implemented modules were very sensitive to sudden increases in computational load and significant overshoots were generated in their step responds. On the other hand, the prototype was insensitive to the sudden decreases of computational load. Then, several time-consuming functions were focused and their performance to different sizes of load changes was

investigated. It was found that the overshoot rate in the respond of two functions was proportional to the size of the applied sudden load changes, including F1.6: Calculate the distances among entities in a list and all other entities, and (ii) F3.2: Devise long-term solutions for people to perform. The result indicates that the computational mechanism for representing and managing dynamic context information should be resilient and robust. The result also implies that by applying these functions (e.g. distance calculation) in real-life application cases which are required to process input data with sudden changes in run-time, a larger computation delay may be experienced than the corresponding condition with a static amount of input data.

The result of the performance testing was considered as a basis for further evaluation of the applicability of the DCIP-M from the empirical point of view. The formulated set of conditions e.g. varied computational loads and different sizes of load changes may be different in different applications. By using the characterized performance indicators, the applicability of the proposed mechanism in varying application cases can be foreseen. This is how we moved from a single case performance validation towards a multi-case performance validation. In the next Chapter, systemic applicability testing will be presented.

#### **4.7.4. Reflections on some recognized limitations**

Based on the research actions and the testing of the implemented prototype, some limitations have been recognized. These can be the starting point of future improvement of the mechanism.

- In the implemented prototype, three types of situations (i.e. anchored people jams, floating people jams and fire) were inferred and considered as the basis for generating personalized action-plans. Two types of interplays (i.e. transitivity and causality) were inferred and only transitivity interplay was used in computation for the purpose of predicting the future of situations. Actually, the potentials of the proposed concepts were not sufficiently represented. Other types of interplays, such as interactivity and similarity among situations may also enhance the accuracy of context prediction and should be focused on in the future work.
- In real-life I-CPS applications, there are three types of delays that may influence the performance of dynamic context handling, namely: (i) input delay, (ii) computational delay, and (iii) informational delay. The input delay is caused by context information sensing and aggregating, while the computational delay is caused by the computation time spend by functions. In addition, an informational delay is related to the message sending and reading by people. The influence of delay in informing was neglected in the conducted implementation and testing work. When using the prototype in real-life

application cases, its sensitivity to the different types of delays should be further investigated.

- According to the results of the performance testing, considerable computation time will be spent by the implemented algorithms when the DCI with regard to a large number of entities is to be processed. More implementation effort is needed to improve the computational efficiency of the algorithms, such as using binary space partitioning to increase the computational efficiency of distance calculation.
- Even if a physical prototype of the indoor fire evacuation could have been built, it could not be tested physically due to the hazardous phenomenon, which excludes real-life experimentations. The fundamental form of testing the prototype implementation of the computational mechanism was a context-sensitive digital simulation. This is identified as one of the limitations of the implementation work due to the lack of practical experience with regard to the reference application case. The application-dependent parameters (e.g. thresholds) should be articulated by application designers according to certain practical needs.

#### **4.7.5. Conclusions considering all findings**

The investigation of the findings from the functionality testing and the performance testing created a basis to conclude the implementation of the prototype of the computational mechanism. The main conclusions are given as follows:

- The integral handling of SAT information allowed the SFR-matrices and the CIR- cube to follow the dynamics of a critical real-life application case. The chosen means of representation made it possible to include, blend and manipulate context data from multiple sources, and to infer additional information about complex situations by monitoring states, events, situations and scenes.
- The conducted work allows progression of context building from a syntactical level to a semantic level. The growing semantic elements can be considered as problem-solving enablers to enhance the decision-making and reasoning operations by I-CPSs. The experimental results indicated that the proposed mechanism provides an effective approach for handling DCI and making decisions based on it.
- Due to the detection errors, indoor positions of the entities may deviate from their actual positions. This influences the accuracy of the representation of DCI. The configured Kalman-filter proved to be an effective tool for estimating the indoor positions of entities. It also facilitates the calculation of



distances among entities.

- The semantic knowledge inferred from situational changes can be used to improve the prediction accuracy of entity context that involves accidentally created non-linear processes. Taking the implication of situations into consideration is necessary for context information management for an indoor environment.
- The DCIP-M considers the actually produced implication of the situations on involved entities and the relevance of the situations to the rest entities. The proposed ‘impact indicators’ and ‘relevance indicators’ enables the investigation of the relations between situations and entities. Therefore, the DCIP-M allows capturing context-sensitive computations and lends itself to a run-time adaptive computational mechanism.
- The reasoning mechanism supports automatic inferring of the personal context of the people involved and interpretation of the meaning of a specific context in real-time. These emergent phenomena could be taken into consideration in the decision-making and controlling actions, such as the generation of informative instructive messages for people in hazardous-intense applications.
- By using an I-CPSs built around the proposed DCIP-M, the hazard in critical events and situations can be reduced. An opportunity for this is providing context-dependent messages for people who are involved. As demonstrated in the conducted tests, dynamically changing situations of people should be dealt with in order to increase the quality of informing.
- As indicated by the results of the human evaluation on the generated messages, people may disobey the instructions given to them in the considered application case, or in particular in hazardous situations. The disobedience of people should be considered as a part of the dynamic context of the people and handled by the I-CPSs. In addition, the quality of information contained in personalized messages could help people to make better judgments, at the same time, the obedience of the people to the given instructions can be stimulated.
- Calculating the distances among entities in an indoor environment is time-consuming. To enhance the computation efficiency, the proposed DCIP-M only focuses on the changed information in every new computational cycle, which is a step towards quasi-real time processing of DCI.
- In real-life application cases, the computational mechanism developed for representing and managing dynamic context information should be resilient and robust in order to maintain an acceptable level in facing overshoots in computation time, which are caused by dynamic changes with regard to entity states and relations.

## 4.8. References

- [1] Godsil, C. Royle, G.F. (2013). Algebraic graph theory, Springer Science & Business Media.
- [2] Harary, F. (1962). The determinant of the adjacency matrix of a graph, *Siam Review*, 4, pp. 202-210.
- [3] Berendt, B., Barkowsky, T., Freksa C., & Kelter, S. (1998). Spatial representation with aspect maps, in: *Spatial Cognition*, Springer, pp. 313-336.
- [4] Franz, G., Mallot, H., Wiener, J., & Neurowissenschaft, K. (2005). Graph-based models of space in architecture and cognitive science-a comparative analysis. In *Proceedings of the 17th International Conference on Systems Research, Informatics and Cybernetics*, pp. 1-8.
- [5] Samaan, N., & Karmouch, A. (2005). A mobility prediction architecture based on contextual knowledge and spatial conceptual maps, *IEEE Transactions on Mobile Computing*, 4(6), pp. 537-551.
- [6] Wang, C. Y., Hwang, R. H., & Ting, C. K. (2011). UbiPaPaGo: Context-aware path planning. *Expert Systems with Applications*, 38(4), pp. 4150-4161.
- [7] Cormen, T.H. Leiserson, C.E. Rivest, R.L. Stein, C. (2009). Introduction to algorithms, MIT press.
- [8] Mitchell, M. (1998). An introduction to genetic algorithms, MIT press.
- [9] Mohamed, A. H., & Schwarz, K. P. (1999). Adaptive Kalman filtering for INS/GPS. *Journal of Geodesy*, 73(4), pp. 193-203.
- [10] Singh, Y., & Mehra, R. (2015). Relative study of measurement noise covariance R and process noise covariance Q of the Kalman filter in estimation. *IOSR Journal of Electrical and Electronics Engineering*, 10(6), pp. 112-116.
- [11] Zarchan, P., & Musoff, H. (Eds.). (2013). Fundamentals of Kalman filtering: a practical approach. American Institute of Aeronautics and Astronautics, Inc..
- [12] Vokolos, F.I. Weyuker, E.J. (1998). Performance testing of software systems. In: *Proceedings of the 1st International Workshop on Software and Performance*, ACM, pp. 80-87.
- [13] Avritzer, A., Kondek, J., Liu, D., & Weyuker, E. J. (2002). Software performance testing based on workload characterization. In: *Proceedings of the 3rd international workshop on Software and performance*, ACM, pp.

17-24.

- [14] Weyuker, E. J., & Vokolos, F. I. (2000). Experience with performance testing of software systems: issues, an approach, and case study. *IEEE Transactions on Software Engineering*, 26(12), pp. 1147-1156.
- [15] Molyneaux, I. (2009). The art of application performance testing: Help for programmers and quality assurance. 'O'Reilly Media, Inc.
- [16] MathWorks (2019). Techniques to Improve Performance, [https://www.mathworks.com/help/matlab/matlab\\_prog/techniques-for-improving-performance.html](https://www.mathworks.com/help/matlab/matlab_prog/techniques-for-improving-performance.html), Visited on 2019/06/06.

# Chapter 5

## Applicability validation of the computational mechanisms

### 5.1. The objectives and approach of validation

#### 5.1.1. The research objectives

In the previous chapters, the conceptualization and implementation of the proposed multi-module DCIP-M for dynamic content representation and reasoning are presented. In the development process, two bodies of knowledge were synthesized: (i) theoretical knowledge concerning the fundamentals of and approaches to dynamic context information management, and (ii) practical knowledge related to a concrete reference application case of fire evacuation management in a critical situation. These provided a robust intellectual basis for the development of the reasoning mechanism, which can be used as a strongly context-dependent reasoning and messaging platform for a family of I-CPSs.

The presented promotion research focused on the algorithmic basis of an adaptable software platform but did not deeply elaborate on the necessary system interfaces, which are also needed for operation as a platform for I-CPSs. As mentioned above, the practical knowledge for supporting the development has been derived from a real-life case. The fire evacuation guiding application was used as a source of both application data and an empirical computational scenario. For this reason, it has been considered as a ‘reference application’. The four computational mechanisms included in DCIP-M, as well as their specific sets of algorithms, have been tailored to this particular application. On the other hand, these reasoning mechanisms have been designed to be applied without or with limited adaptation to (i.e. with the objective of reusability in) applications with similar critical phenomena.

Due to the differences in the principal phenomena, there can be countless practical I-CPSs applications. Recognition of the probability of potential

dissimilarities in other possible application raised the need for applicability validation. While the fire evacuation guiding application was handled as the reference application in the entire process of development of the tailored mechanisms and algorithms, for the purpose of validation a large number of so-called ‘target applications’ could be considered. Among these choices, we selected three applications which belongs to completely different fields of interest and knowledge, but require the four major functionalities provided by the multi-component DCIP-M developed by us: (i) efficient spatiotemporal representation of dynamic context information mechanism, (ii) inferring semantic context information based on the spatiotemporal representation, (iii) generation of action plans according to the situational dynamics of real-life processes, and (iv) generation and sending of context-based content and receiver sensitive messages to various people. As explained below, the adapted validation square approach was used in applicability validation. Our intention was to generate an applicability profile of multi-component computational mechanism (as well as of the methodology implied by this). An additional objective was to generate hints concerning the possible enhancement and necessary adaptations of the entire platform.

### **5.1.2. The application leading the development of the mechanisms**

As a reference application case an indoor fire evacuation guiding application has been selected and this guided the actions of knowledge aggregation, the conceptualization of possible reasoning mechanisms, and the implementation of a testable prototype and simulation of the expectable results under varying circumstances. The indoor fire evacuation system aims at providing personalized escape strategy for people in different situations. The main research tasks were: (i) understanding the phenomenon of indoor fire evacuation, (ii) constructing representation schemes for spatiotemporal context and content data, (iii) deriving semantic information and knowledge based on dynamic context information management, (iv) developing situation-dependent and personalized escape routes for all involved individuals in a quasi-real-time manner, and (v) sending informative and instructive messages to all individuals having communication possibility.

The building of the Faculty IDE has been selected as a place of the hypothetical fire, and its ground floor as the basis of spatial context information computation and evacuation strategy development. As discussed in the preceding chapters, the DCIP-M was developed to generate operational strategy and to synthesize personalized action plans for all involved individuals in the simulation, no matter if they were directly (through their smartphone) or indirectly (through the involvement of other individuals) notified about these plans. The DCIP-M

determines the best route for the concerned person to escape, updates the individual escape action options, and sends information about this to each person at a given sampling time. This indoor fire evacuation guiding application was used not only in the development of the dynamic context computation, action plan generation, and message construction and distribution mechanisms but also in their testing. A high-fidelity simulation of (i) the propagation of the fire and (ii) the behaviors of human, artefactual and natural entities was applied in order to correctly reproduce the presumed real-life case. The obtained experimental results proved the efficiency of the computational mechanisms and the interoperating algorithms. They also confirmed our hypothesis that the proposed DCIP-M is able to provide both descriptive and predictive knowledge about emergency situations as well as about the implications of the interplaying situations on the concerned entities in quasi-real time.

### **5.1.3. The methodological approach to validation**

Validation of a software engineering methodology was seen as a challenging issue by Lee and Rine, since it involves testing of data, algorithms, usability, and performance [1]. Many issues concerning software system validation are in the focus of recent studies. For instance, Eze et al. discussed various challenges associated with validation of self-managing and autonomic systems, and investigated the relations of validation approaches (such as system unit testing, real-world system testing, pervasive supervision, system model checking, system self-testing) and execution modes (generic, design-time, run-time, integrated, autonomous) [2]. Ahmad et al. analyzed validation techniques for safety-critical software such as (i) functional failure analysis, (ii) HAZard and OPerability Studies (HAZOP), (iii) failure modes and effects analysis), and (iv) fault trees analysis, and compared them according to their (i) efficiency, (ii) reliability, (iii) dependability, (iv) testability and (v) usability [3]. Brings et al. dealt with the issue of supporting early validation of cyber-physical system (CPS) specifications based on model-based prototype development. Their objective was the identification of defects in the system specification [4].

Gonzalez and Barr exposed the differences to be taken into account in the verification and validation of intelligent systems [5]. Feth et al. focused on the validation of open and heterogeneous systems, such as CPSs in the automotive domain, and proposed a simulation-based framework, which integrates AUTOSAR applications [6]. The virtual validation concerned the functional behavior and the performance of the software. Guarro et al. proposed a comprehensive, multi-level framework for validation and verification of model-based control and adaptive control systems [7]. The framework combines logic dynamic model constructs and the associated analysis processes to demonstrate compliance with the related aviation-system certification standards. Bradley

claimed that validation of system models within a multi-disciplinary design framework has three primary components (i) independent validation of the contributing analyses, (ii) validation of the overall performance and behavior on system-level, and (iii) validation for utility with respect to design decision making [8].

According to our survey work, there are both experimental and analytical (logical) approaches to validation of computational methods. The approach should be logically rigorous, internally consistent, and mathematically correct. However, the literature offers only limited insights with regards to the validation of computational methods. This is because the validation is largely influenced by the purpose of their applications. Accordingly, external appropriateness should be focused on. The typical aspects of validation in the concerned external appropriateness should be checked, for instance, in terms of usability, applicability, performance and overall utility.

It has been recognized that an adapted version of the validation square approach (VSA), initially proposed by Pedersen et al. [9, 10] can be a very powerful

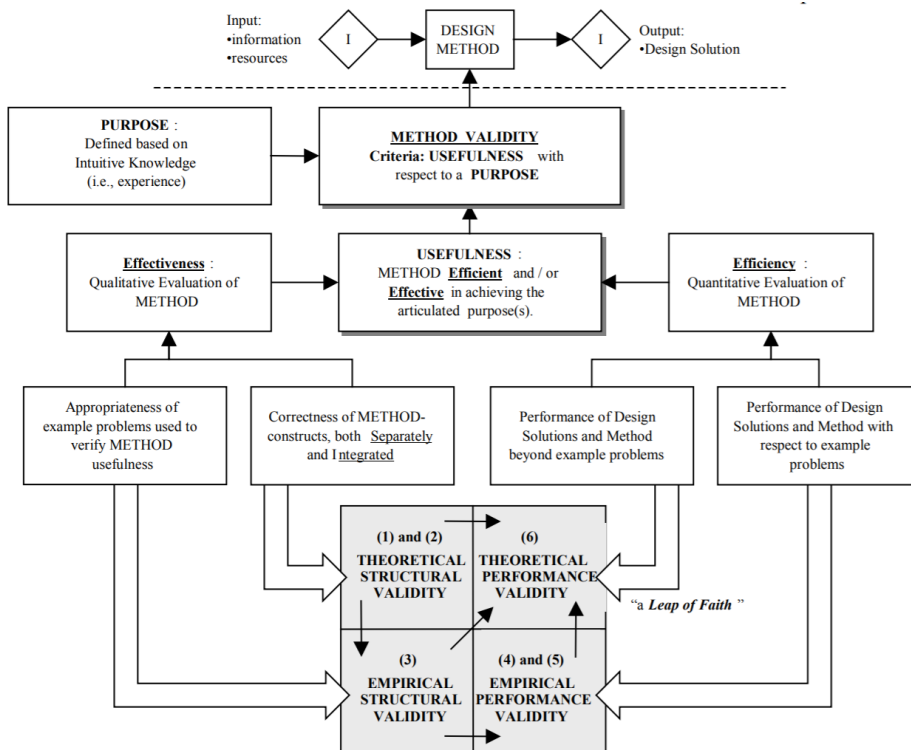


Figure 5.1. The reasoning model of the validation square approach [9] [10]

approach to external validation from the above perspectives, as shown in Figure 5.1. The fundamental assumption of the authors was that external validation of new knowledge is a process of building confidence in its usefulness with respect to an application purpose. This has been combined with their research interest that concerned the approach of verifying and validating a newly developed or existing design method in various applications. Their working hypothesis was that a design method should be useful with respect to a purpose. Instead of formal logic-based, rigorous and quantifiable validation approaches, they proposed a formal approach for validation of usefulness as a measure of external relevance, assuming that the internal consistency of the design method is guaranteed.

Therefore, the original VSA is organized around two criteria, namely (i) usefulness (meaning that a design method should enable designing correctly), and (ii) appropriateness (meaning that a design method should be able to produce utility). The former criterion was captured in VSA as ‘effectiveness’, while the latter criteria as ‘efficiency’. Reaching a balance with regards to fulfilling the criteria and having a reasonably manageable process was formulated as a requirement for VSA. The effectiveness of a design method can be assessed by a qualitative evaluation, while efficiency can be evaluated by a quantitative assessment. Effectiveness shows the appropriateness of a design method to example problems. Efficiency indicator of the performance characteristics of the solutions generated by the design method with regards to example problems and beyond the example problems. The quadrants of the validation square define four aspects of validation, which are: (i) theoretical structural validity, (ii) empirical structural validity, (iii) empirical performance validity, and (iv) theoretical performance validity.

#### **5.1.4. An adapted validation square approach**

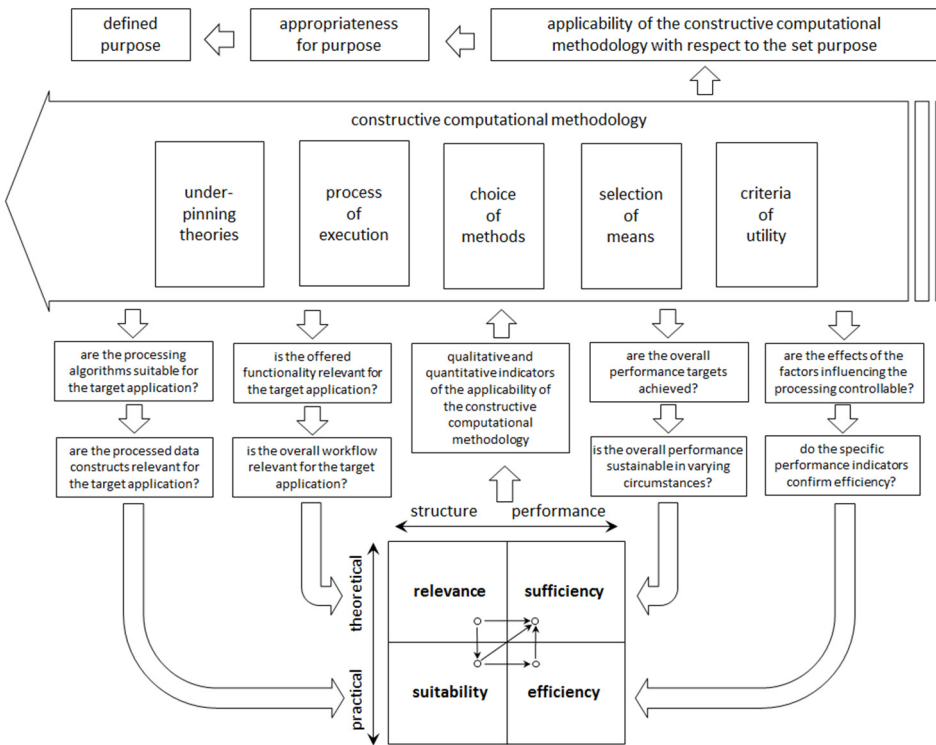
Based on the original VSA, an adapted validation square approach (AVSA) was considered for applicability validation of constructive computational methodologies. The overall strategy of applicability validation, the aspects and the criteria to be applied for the evaluation of the applicability of the proposed computational reasoning mechanism is shown in Figure 5.2. The AVSA considers (i) the theoretical procedural structure, (ii) the theoretical performance targets, (iii) the empirical procedural/information flow, and (iv) the empirical performance indicator as measures of applicability of the multi-component reasoning mechanism in real life application cases [11]. It provides an applicability profile of the mechanism in a particular application, and, by doing so, hints on necessary or possible adaptations.

As a concrete representation of the process structure proposed by a CCM, procedural flow charts can be used that can capture all computational transformations and their interrelationships (interoperation), if the procedural flow



chart of the process structure required by the target application is developed, then the two charts can be compared, and the congruencies and the differences can be identified and evaluated. For the purpose of checking the appropriateness of mechanisms and algorithms, and data, information and knowledge structures, and constructs, respectively, information processing flows and data structures can be used. This way, comparison and evaluation of what is proposed by the computational methodology and what is required by the target application can inform about the measure of empirical structural validity. In other words, it informs about whether the computational mechanisms and algorithms are able to compute what is needed in the specific application case and whether the data/knowledge needed by them can be provided and the output is an adequate input to other mechanisms/algorithms in that application case.

The procedural flow charts and the information processing flow and data structure models, respectively, can be used to decide on the range of applications for which the computational methodology appropriate without or with limitations. There is a possibility to generate indicators of the theoretical and empirical structural



**Figure 5.2.** The reasoning model adapted to the use of the validation square approach in applicability validation of constructive computational methodologies

appropriateness (soundness) by considering the deviations of the example applications (phenotypes) from the target application (genotype) for which the computational methodology has been developed. These indicators may be both qualitative (interpretative) and quantitative (statistical). This is again a difference with regards to the original VSA, in which the theoretical and empirical structural ‘validities’ were supposed to be evaluated only qualitatively.

The flow of knowledge elicitation for application validation, including the four indicators (relevance, suitability, efficiency, and sufficiency) and the questions formulating the fundamental criteria are shown in Figure 5.3. In the case of the applicability validation of CCMs, the empirical performance depends on multiple factors such as the efficiency of (i) the underpinning theory, (ii) the procedural flow, (iii) the collection of the methods, and (iv) the usable instruments. This implied the need for reconsideration of the performance validation strategy of the VS approach. It was proposed to be a quantitative process centered on efficiency that was considered from three aspects: (i) evaluation of usefulness of the design method in the case of some chosen example problem(s) with respect to the initial

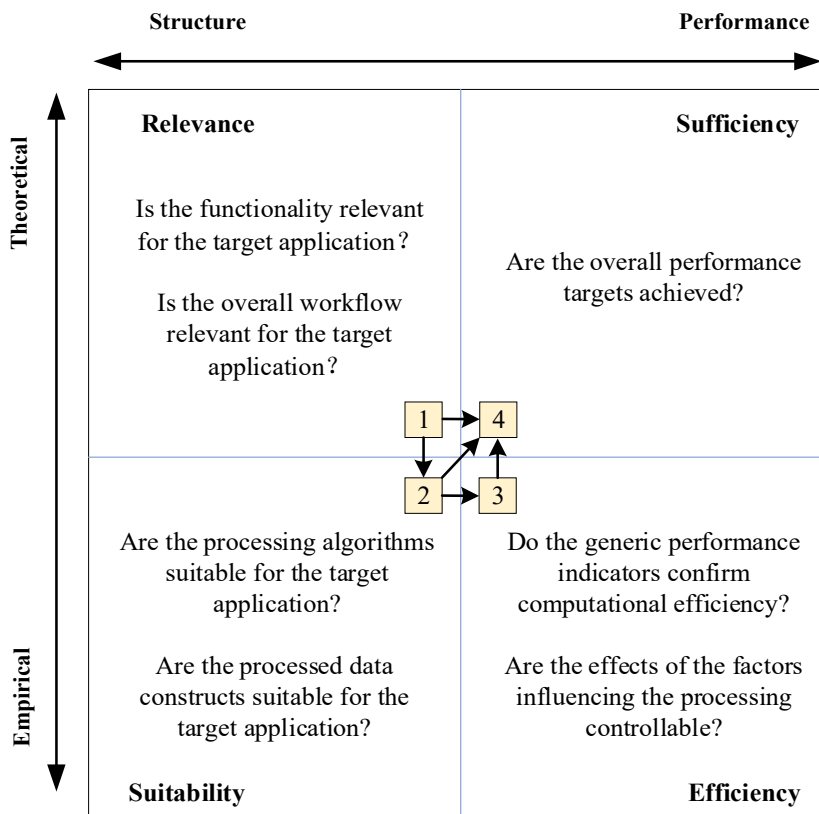


Figure 5.3. The process and indicators of applicability validation

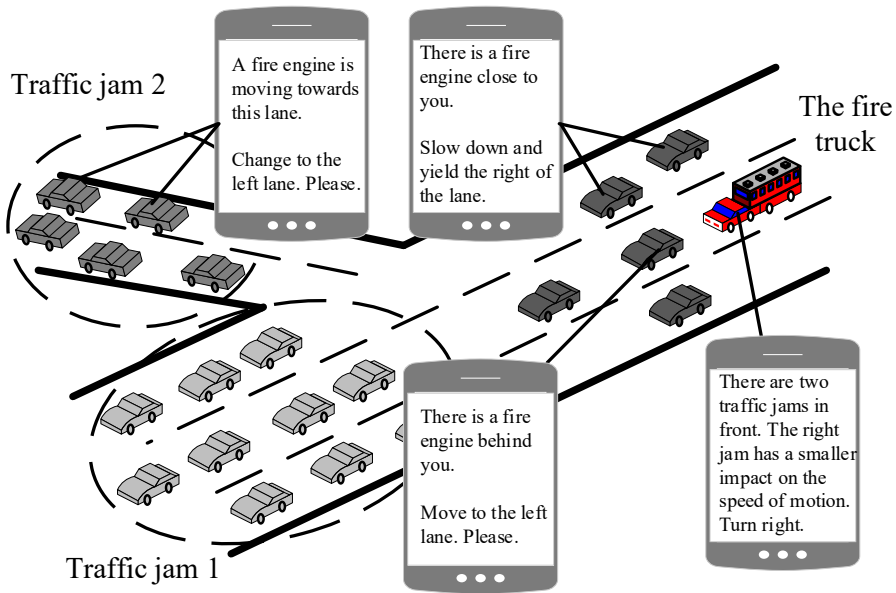
purpose, (ii) demonstration that the achievements are linked to the use of the design method in the chosen example problem(s) with respect to the initial purpose, and (iii) evaluation of usefulness beyond the chosen example problem(s) assuming limited changes in the initial purpose.

## **5.2. The considered other applications**

The proposed reasoning mechanism has been developed with an indoor fire evacuation support application in mind. Nevertheless, the generality of the mechanisms and most of the included algorithms make it possible to consider its application in other cases. However, this raises the need for investigation and validation of its applicability in other application cases. The matter of fact is that there are many real-life applications which can benefit from a direct or adapted application of the computational mechanism and/or the results produced by this. Three possible I-CPS applications have been considered in the chapter to validate the applicability of the computational mechanism. The applications include: (i) road traffic management, (ii) home caretaking assistant, and (iii) real-time football-play guiding. These are novel applications for smart I-CPS, which need dynamic context information processing. They form a small but representative set of applications where real-time context representation, situated reasoning, and personalized message communication play a crucial role. Detailed explanations with regard to the objectives, challenges, and requirements of these applications are given as follows.

### **5.2.1. Road traffic management system**

Due to the increase of vehicle number on city roads, unexpected traffic situations (e.g. traffic jams) may always happen, which may decrease the speed of motion of vehicles, increase the possibility of traffic accidents happening and increase the fuel consumption of individual vehicles [12]. The traffic situations can be managed through (i) providing real-time situation-dependent informative messages to individual drivers about the happenings around them in order to support their making decisions and (ii) providing personalized guidance to the relevant drivers based on vehicle-carried informing terminals to dissolve some existing situations or to prevent the happening of some unexpected situations in the near future. These call for the application of a road traffic management system (RTMS). As shown in Figure 5.4, a specific scenario considered for application of the RTMS is given as follows. A fire truck is moving on road for a fire distinguishing mission. There are two traffic jams accidentally formed on the high-way, which may influence the movement of the fire truck. In this case, the objective of the RTMS is to determine the optimal routes and actions for the related drivers (including the fire truck and the rest related cars) to enable the fire truck arrives at the target location as fast as possible.



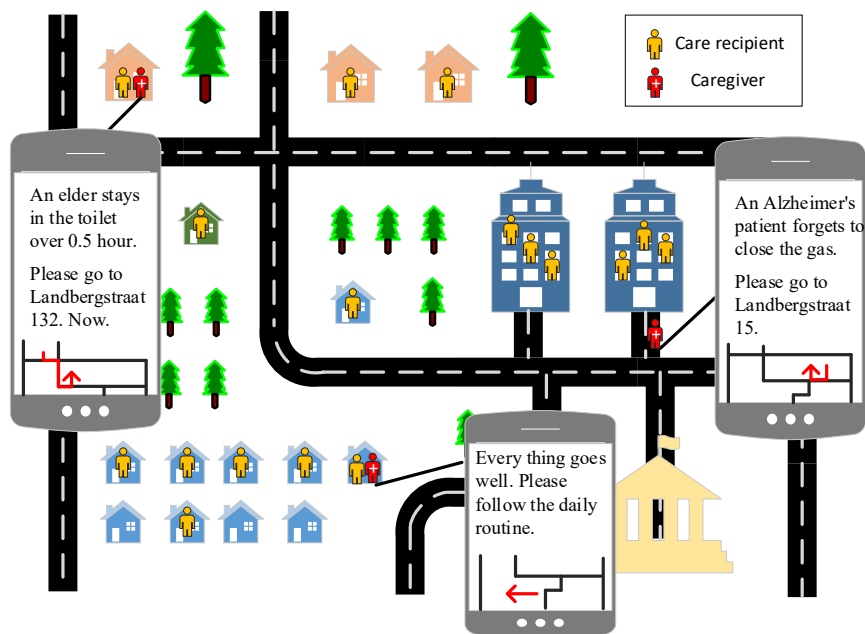
**Figure 5.4.** A road traffic management system

Processing dynamic context information is necessary for this type of application since the possible rapid changes of the state and situation of the individual vehicles (e.g. their motion attributes, the distance between them, etc.) have an impact on the decision-making process of the drivers (e.g. selecting a possible alternative road, stopping for a rest, etc.). The dynamically changing locations, attributes, and relations together form different situations that may also influence the decision-making process of the traffic management system. In order to be able to manage the different situations, the traffic management system should: (i) continuously monitor the emerging situations on the road, (ii) extract and manage the spatial, attributive and temporal data of the vehicles as well as the data about their spatiotemporal relations, (iii) infer high-level context knowledge and meta-knowledge concerning (a) the essence of situations, (b) the implications of the situations on the concerned entities, and (c) the relevance of the situations to these entities, (iv) prioritize the entities and situations and apply the prioritization in the decision-making process, (v) derive action plans for the particular drivers that are involved in a particular situation taking into account the overall system objectives, and (vi) construct personalized messages to the drivers according to the action plan they should follow, including all pieces of information they should be aware of.

### 5.2.2. Home caretaking assistant system

The recent increase of the aging population and the consequent increase of the incidence of chronic diseases are challenging the existing capacity of medical establishments (e.g. hospitals, sanatoria, care centers, etc.) [13], and are implying the need for alternative health care systems. One possible solution is providing caretaker services for care recipients (e.g. elderlies, patients, handicapped, etc.) in a home environment. In this application context, the caregivers (e.g. doctors, nursing staff) should be 'on-line' and operate in a virtual establishment. In order to be aware of the medical and/or well-being states of the related care recipients, they have to be aware of the situation in general as well as all emergent happenings. To make them informed, context-dependent real-time messaging is needed that selectively provides them with information regarding (i) what the overall personal situation of the related care recipients are, (ii) if anyone of the care recipient involved in any kind of possibly dangerous situations (e.g. staying in the washroom over a long period of time), and (iii) what actions are to be done by the caregivers assuming multiple care recipients in home environment (e.g. scheduling, route planning, instrumentation).

A specific scenario for the application of a home caretaking assistant system (HCAS) is shown in Figure 5.5. According to this simplified scenario, three caregivers provide daily care for a group of people. As their routine tasks, each of



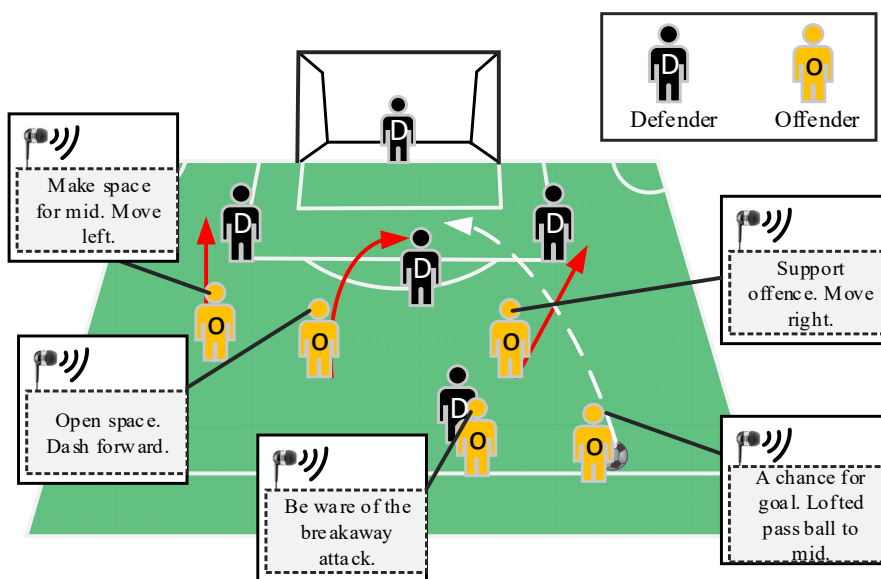
**Figure 5.5.** A home caretaking assistant system

the caregivers has to look after several people in a day and their moves should be done with high efficiency (with a minimal totaled motion path). However, in the case of emergent situations and circumstances, they have to act according to the incurred level of danger. They have to give up their daily routine and act as implied by these events. They may need to look after a patient who is not in the daily schedule or to change the priority concerning the care recipients since they are supposed to react immediately if a dangerous situation appears. However, if more than one dangerous situation happens almost at the same time, then they have to make decisions about a plan of actions. At developing their action plan, they have to take into consideration changing contexts. In order to be able to provide support for the caregivers' work, the HCAS should be involved in monitoring, manage the schedule, reason with the event and the servicing capacity of the caregivers, and provide action plans and information in a real-time manner.

To achieve this, the dynamic context of caregivers and care recipients (e.g. the changes of attributes of the care recipients and distances from them to the caregivers) should be processed. The dangerous situations in which the care recipients may be involved can be inferred based on various pieces of information aggregated in their home. The challenge is (i) to prioritize the care recipients based on the inferred situations and the distances to the caregivers, and (ii) to manage the spatiotemporal context of the caregivers (since every action takes time to get completed). Therefore, the HCAS is supposed (i) to aggregate information about the personal context of care recipients (e.g. activities) as well as about the caregiver (e.g. location), (ii) to infer knowledge about any dangerous situation happening in or around the home of the care recipients, (iii) to evaluate the relevance of the situations to caregivers, (iv) to derive action plans for the caregivers to follow (including the order of caretaking and the needed actions), and (v) construct personalized messages (including informative messages and instructive messages) for taking the care of all concerned care recipients.

### **5.2.3. Real-time football-play coaching system**

To enhance the performance quality and to increase the efficiency of football playing, training is indispensable. The practice involves not only physical exercises but also information about the best practices in certain situations. Using the context processing, situated reasoning, and action planning functionalities of smart I-CPSs, players (both professionals and amateurs) may be provided with (i) information in real-time about the specific situations happening during the game and (ii) instructions concerning where to be and what actions to do, and (iii) what to be prepared for. Due to miniaturization, the information provisioning can happen through wearables and/or portable equipment (e.g. mini wireless headset) during the training sessions. A digital trainer (like a human) may be in direct and real-time contact with all players simultaneously. The trainer may also decide on



**Figure 5.6.** A real-time football-play coaching system

who and when should be informed or instructed. This kind of I-CPS is seen as a new generation of sports assistive systems.

Figure 5.6 shows a specific hypothetical scenario in which the real-time football play coaching system (RFCS) can provide informational services. The offenders shown may be provided with personalized messages about the best momentarily strategy and collective actions to score a goal. Since the offenders may play different roles (e.g. scorer, supporters, and passer), the information or instruction can be selectively formulated according to the dynamically changing context. The RFCS may (i) develop strategies for completing the offense according to the dynamic changes of the context, (ii) generate the relevant types of actions based on the spatial position, attributes and role of the offenders, and (iii) instruct the players to replay the actions in the given situation.

In this scenario, the dynamic context of the football player includes (i) the changes of the attributes of the other players (e.g. location, speed of running, and physical agility), and (ii) the changes of the spatial relations (e.g. distances and orientations) among a concerned player, the other players, and the ball. The dynamic context of the individual players is decisive in terms of: (i) to which location to move on the playground, (ii) what speed to take to reach the target location, and (iii) what actions to take after receiving the ball. To advise the players on these, the RFCS should be able (i) to handle time-dependent descriptive information concerning all players and the ball on the playground, (ii)

to infer and predict situations (e.g. an open corridor, a blockage, a large space behind the defenders), (iii) to forecast the implications of the situations on players, (iv) to derive action plans that can be used at suggesting the players what to do, and (v) construct personalized messages for the players (e.g. quick counterattack, offense, defense, etc.).

#### 5.2.4. Comparison of the concerned target applications

The above description of the three target I-CPS application cases casts light on two facts. On the one hand, there are significant differences in terms of the case characteristics of the chosen target applications, which in turn define a reasonably broad spectrum of applications. Having a narrower or broader range of the case characteristics, many other applications with rather different purposes can be involved. On the other hand, enabling technologies applied by the chosen target applications may involve different constraints on computation, e.g. hardware. The realization of expected performance should depend on the empirical influencing factors. Therefore, in the rest of the sub-section, we focus on the comparison of the characteristics and influencing factors of the chosen target applications, in order to create a factual basis for the applicability analysis and validation.

We selected three major characteristics for the purpose of comparative applicability analysis, namely (i) the number of entities handled at a given point in time or in a specific time interval (computational time-increment), (ii) the response time required by the ‘happenings’ in the application case, and (iii) the sudden change of entity number involved between two successive computations. The specific quantitative values associated with these characteristics are indicated in Table 5.1. As references related to these emerging applications can hardly be found, the specified values of these emerging applications were more or less generated based on our assumption, rather than a concrete survey work. Although there might be some exceptional situations, we assumed that these values could properly fit most cases in the application contexts.

**Table 5.1.** A comparison of the application-implied requirements of the target I-CPS applications

No.	Characteristics	IFECS	RTMS	HCAS	RFCS
1	The number of entities handled at a time	Moderate (40-300)	Large (100-1000)	Moderate (20-100)	Small (23)
2	Required response time	Moderate (less than 5s)	Short (less than 1s)	Long (less than 10s)	Short (less than 0.5s)
3	The sudden change of entity number	Moderate (0-10)	Large (10-200)	Small (0)	Small (0)



As shown in Table 5.2, 8 influencing factors with regard to information sensing and messaging were concerned to represent the technical constraints in the concerned application cases. In terms of the IFEGS, Ultra-Wideband (UWB) is assumed to be applied for sensing the indoor positions of individual people based on the principle of fingerprinting. The reported accuracy of this approach is within 0.5 meters with a good preparation [14]. The maximum latency of information sensing depends on the frequency of signal transmission, which is estimated to be 0.25s (4 times per second) [15]. Since hand-held smartphones can be selected as informing terminals, personalized informing can be realized through a multi-hop WLAN/Wi-Fi networking, which may have a latency of informing varied from 5 ms to 20 ms [16].

In addition, RTMS may use GPS to obtain individual positions of vehicles on the road. The accuracy of positioning is within 4.9 meters under open sky [17]. The reported latency of positioning by using GPS is about 0.5s [18]. Due to the large-scale for traffic management, the 4G-mobile web may be selected as a way for sending personalized messages to drivers. The latency of message delivering ranges from 60 ms to 100 ms [19]. In the case of HCAS, GPS+ indoor camera may be jointly used to aggregate the location information of help givers and the indoor activities of help recipients. Different to the RTMS, HCAS only provide personalized messages to help givers and messages with detailed information

**Table 5.2.** A comparison of the technical factors that influence the empirical performance of the target I-CPS applications

Influencing factors	IFEGS	RTMS	HCAS	RFCS
Technology for information sensing	UWB	GPS	GPS + Indoor camera	Camera
Typical positioning accuracy	Moderate ( $< 0.5$ m)	Low ( $< 4.9$ m)	Low ( $< 4.9$ m)	High ( $< 0.1$ m)
Latency of information sensing	Moderate ( $< 0.25$ s)	Long ( $< 0.5$ s)	Long ( $< 0.5$ s)	Short ( $< 0.1$ s)
The power supply of informing terminals	Moderate (Hand-held)	High (Onboard)	Moderate (Hand-held)	Low (In-ear)
Technology for message sending	WLAN/WiFi (10Mbps)	4G Mobile web (100 Mbps)	4G Mobile web (100 Mbps)	Bluetooth 5.0 (2Mbps)
The latency of message receiving	Moderate (5-20 ms)	Long (60-100 ms)	Long (60-100 ms)	Short ( $< 3$ ms)
The length of personalized messages	Moderate (10-30 words)	Moderate (10-30 words)	Large (50-80 words)	Short (0-10 words)
Latency of message reading by people	Long (0- $\infty$ s)	Short (0 s)	Long (0- $\infty$ s)	Short (0 s)

(long messages) can be used to describe the situations about help recipients and the generated action plans to perform. Last, RFCS may use cameras to detect the real-time position and orientation of football players [20], which offers a 0.1-meter accuracy and 100 ms latency of positioning [21] [22]. In-ear devices should be applied as informing terminal for this application case. For this reason, in this case, Bluetooth 5.0 was assumed to be suitable for sending messages [23] [24]. The latency of Bluetooth 5.0 based communication is smaller than 3 ms [25].

It can be seen from the comparison that the empirical characteristics and the concerned aspects of influencing factors are varied with regard to the selected target application cases. Actually, each of the target application cases can be considered as a representation of a set of application cases with similar situations. Particularly, the IFEGS, which was considered as the reference application case for the conceptual development, implementation, and testing of the prototype, has moderate values in most of the concerned characteristics. Since the implemented prototype is proved to be efficient for DCIP for the reference application case, its performance to other application cases is to be evaluated concerning different characteristics and actual contexts. This is the basis to investigate the applicability of the DCIP-M.

### **5.3. Analysis of the relevance of the theoretical structure of the DCIP-M**

The assessment results of the relevance of the functionalities for the target applications are shown in Table 5.3. It can be seen that the relevance of the functionalities are qualitatively evaluated with three characters, including (i) relevant, which means that the functionality is fully required by the target application (ii) partially relevant, which refers to a situation that the functionality can partially fulfill the requirement of the target application, and (iii) not relevant, which means that the functionality is not required by the target application at all. In addition, the assessment results of the relevance of the workflow for the target applications are shown in Table 5.4. The assessment of the results for the particular target applications is explained below.

#### **5.3.1. Relevance of the functionalities to the RTMS**

In the case of RTMS, all the functions included in the DCIP-M are considered to be relevant. A detailed explanation of reason is given as follows. The descriptive context data with regard to the locations and attributes of vehicles on road should be aggregated by the RTMS in real-time. In addition, the SFR-matrix concept is suitable to be applied in the case of traffic management since the distances (spatial relations) among vehicles and the speed of motion of the vehicles are the objects to be computed. Furthermore, the distance changes of vehicles imply the

occurrence of dangerous situations (e.g. traffic jams, traffic accidents). The CIR-cube concept is suitable for representing the temporal relations of the distances. Due to the reason that the big changes in distances among vehicles on road might cause the formation of dangerous situations more easily, they are more important than the slight changes of distances. The RTMS should put computational attention on the big changes of distance in order to increase the speed of calculation and decrease the time spent for computations, as the same as the fire evacuation guiding case.

The dangerous situations and the implications should be known by the drivers and are important for making decisions on the individual path and actions for the drivers to follow. The related functions should be applied. In order to prevent the formation of some dangerous situations and to proactively manage the dynamic traffic situations, the functionalities that are used for context prediction are relevant to the RTMS. Next, since multiple dangerous situations (e.g. traffic jams) may happen at the same time and the management of vehicles and situations should consider a particular order of priority.

The impact indicator based prioritizing mechanism is very relevant to the RTMS. Moreover, since the relevance of dangerous

**Table 5.3.** Assessment of the relevance of the functionalities for the target applications

No.	The relevance of the functionalities			
	IFECS	RTMS	HCAS	RFCS
F1.1	●	●	●	●
F1.2	●	●	●	○
F1.3	●	●	×	○
F1.4	●	●	●	○
F1.5	●	●	× (AF1.10)	×
F1.6	●	●	●	×
F1.7	●	●	× (AF1.11)	○
F1.8	●	●	●	○
F1.9	●	●	●	○
F2.1	●	●	●	●
F2.2	●	●	●	●
F2.3	●	●	× (AF2.15)	●
F2.4	●	●	×	●
F2.5	●	●	×	●
F2.6	●	●	●	●
F2.7	●	●	●	●
F2.8	●	●	●	○
F2.9	●	●	●	●
F2.10	●	●	●	●
F2.11	●	●	●	●
F2.12	●	●	●	●
F2.13	●	●	●	×
F2.14	●	●	●	×
F3.1	●	●	×	×
F3.2	●	●	●	×
F3.3	●	●	× (AF3.6)	●
F3.4	●	●	●	●
F3.5	●	●	×	●
AF				AF3.7
F4.1	●	●	×	●
F4.2	●	●	●	×
F4.3	●	●	●	●
F4.4	●	●	●	●
F4.5	●	●	●	●
F4.6	●	●	×	×

●: Relevant, ○: Partially relevant, ×: Not relevant. **AF**: An additional function is needed

situations to drivers should be evaluated run-time, the relevance calculation mechanism is very relevant for the purpose of a comprehensive context management.

Actually, the implication of the dangerous situations (traffic jams) is that they may reduce the speed of motion of vehicles which intend to pass through.

**Table 5.4.** Comparing the dependency of the functionalities among the concerned applications

Func-tions	Functional dependency of the reference application	The difference of functional dependencies for the target applications		
	IFEGS	RTMS	HCAS	RFCS
F1.1	Input+F1.4+F1.9+F2.2+F2.4		-F2.4	
F1.2	F1.1			
F1.3	F1.2		×	
F1.4	F1.3		-F1.3+ F1.8	
F1.5	F1.2		×	×
F1.6	F1.5		-F1.5+AF1.10	×
F1.7	F1.5		×	-F1.5+F1.2
F1.8	F1.6+F1.7		-F1.7+AF1.11	-F1.6
F1.9	F1.8			
AF1.10			+F1.2	
AF1.11			+AF1.10	
F2.1	F1.9			
F2.2	F2.1			
F2.3	F2.2		×	
F2.4	F2.3		×	
F2.5	F2.4		×	
F2.6	F2.5		-F2.5+AF2.15	
F2.7	F1.9+F2.6+F3.5		-F3.5+F3.2	
F2.8	F1.9+F2.7			
F2.9	F2.8			
F2.10	F2.6+F2.9			
F2.11	F2.10			
F2.12	F2.10			
F2.13	F2.12			×
F2.14	F2.13			×
AF2.15			F2.2	
F3.1	F2.10		×	×
F3.2	F1.9+F2.7+F2.14+F3.1		- F3.1+F3.4	×
F3.3	F1.9+F3.2		×	-F3.2+F2.7+F2.10
F3.4	F2.11			
F3.5	F3.3+F3.4		×	- F3.4+AF3.7
AF 3.6			+F3.2	
AF 3.7				+F1.9+F3.4
F4.1	F2.12		×	
F4.2	F2.13		-F2.13	×
F4.3	F4.1+F4.2		-F4.1+F2.12	-F4.2
F4.4	F3.5+F4.2		- F3.5+AF3.6	-F4.2
F4.5	F4.3+F4.4			
F4.6	F4.5+F2.14		×	×

Dangerous situations on the road have a great influence on planning the paths of the vehicles to their expected destination. Therefore, the functions related to personalized solution generation are relevant to the RTMS. In the given scenario (Figure 5.4), the cars that are relevant to fire truck should be instructed with not only the information about the route leading to their individual destinations but also some instant actions (e.g. yield the right of the lane to the fire truck). To achieve this, the action-plan generation and selection functions are relevant to the RTMS.

To generate personalized messages for drivers, the relevant situations to drivers can be a basis, which means that the situation selecting function in the MCM is relevant for RTMS. Similar to the IFEGS, drivers might need to be informed with different rhetoric styles, such as be warned in emergency situations. This is the reason why the danger level-based selection of message template is relevant to this application case.

Furthermore, the RTMS may handle thousands of vehicles on the road at a time. The sending of personalized messages should follow a particular order, which is decided based on the run-time evaluated danger levels of the drivers. For this reason, the priority list based approach for messaging sending is relevant to the RTMS.

### **5.3.2. Relevance of the functionalities to the HCAS**

In the case of the HCAS, descriptive context data concerning the caregivers and the care recipients should be provided for computation. The distances between the caregivers and the care recipients are also important for scheduling the order for taking care of recipients and planning the optimal path to reach all care recipients in the shortest time. For this reason, the concept of SFR-matrices and CIR-cubes can be generated in run-time for the representation of the entity states and relations. Particularly, F1.3 (Calculate distances among all entities in the SFR-matrix) is not relevant since the distances among patients are meaningless. F1.5 (Generate a list of entities whose locations have changed) is not relevant since the HCAS does not need to focus on the entities whose locations have changed. In this case, an additional function (AF1.10) is needed to replace F1.5, namely “Predefine an entity list containing the identifications of the caregivers”. Furthermore, F1.7 (Calculate the speed of motion of the entities) is not relevant since neither the speed of motion of caregivers nor that of care recipients is important for scheduling the plans of care-taking. Instead, another additional function AF1.11 is required, which calculates the time-dependent changes of entity attributes (e.g. service capacity of caregivers, activities of care recipients). These aspects of changes may cause situations (e.g. care recipients may be in dangerous situations, a caregiver is busy) that should be considered by the system. Based on the changes, a CIR-cube is needed for handling the dynamic context.

Representing the spatial and attributive changes of the care recipients, the CIR-cube enables situation inferring in the case of HCAS. For instance, if a care recipient stays in the toilet over 0.5 hours, it implies that the care recipient may get trouble there. The impact of the inferred and predicted situations should be evaluated, and the proposed context assessment mechanism is relevant. Furthermore, historical changes of situations are not important in the case of HCAS. The reason is that the prediction of the future situation is not based on the trend of historical changes. Instead, the prediction may depend on the scene formed by interplaying situations, e.g. if a caregiver will arrive at the house of a care recipient soon, then the dangerous situations of the care recipient may be alleviated. Functions that developed for predicting the future of existing situations (i.e. F2.3, F2.4, and F2.5) are not relevant.

An additional function, AF2.15, is required to replace F2.3, which predefines a set of possible scenes used for predicting the consequences of dangerous situations. Due to the reason that the future states of caregivers and care recipients are crucial for scheduling the next steps of care-taking, this application also needs to infer if any new situation will happen in the near future. This is the reason why the scene concept is meaningful to this application. Last, the predicted situations should be treated according to a certain order generated based on their impacts and the care recipients should be prioritized according to their danger levels. Functions designed for ranking situations and people are relevant to the HCAS.

The reasoning functionalities required by the HCAS are used for generating an ordered list containing the names of care recipients for care-taking. In this application, the happening of any situation about care recipients does not change the time consumed by a caregiver to reach a target location on the map. Thus, the concept of the penalty matrix is not relevant at all. The proposed personal solution generation mechanism is still relevant. Furthermore, F3.3 (Generate alternative action-plans for individual people) should be replaced with a new function (AF3.6), namely “Specify the actions taken by the caregivers” due to the fact that when a target care recipient is assigned to a caregiver to be looked after, the actions that should be taken by the caregiver are determined. The actions depend on the situation that the target care recipient is involving. In this case, the action-plan selection is not relevant.

In the concerned application, the caregivers may need to know information about all important situations that the care recipients are involving. It means that the inferred situations do not need to be selected by the messaging module. The HCAS should also calculate the danger level of care recipients, and select a template for message construction. Last, we believe calculating the order for sending messages (F4.6) is not relevant to HCAS since the number of caregivers considered at a time is relatively small (3 in the concerned scenario of Figure 5.5), which may not cause congestion or delay in messaging sending.

### 5.3.3. Relevance of the functionalities to the RFCS

To process the DCI in the football-play coaching scenario, obtaining descriptive context data is a necessary step. Due to the reason that the orientations of the concerned football players and the relations of the orientations are important for inferring situations (e.g. there is a large space behind a defender, or a defender is moving towards an offender from the back), new SFR-matrixes and a new CIR-cube should be calculated, in which the orientations of the players are compared. The proposed CIR-cube only calculates the distances among entities. The related functions only partially fulfill the computational requirements. This means that they are partially relevant to the RFCS. Due to the fact that all the entities on the football pitch (including 22 players and a football) should be considered in computation, the mechanism enables focusing on changed information is not needed. F1.7 (Calculate the speed of motion of the entities) is partially relevant since other aspects of the time-dependent changes of the attributes of the players might also be considered as the basis of the inferring, reasoning and messaging operations (e.g. the level of skills, turning speed, etc.).

In the KIM, particular situations and the interplays among them should be inferred, which are important for generating the strategy of the offense of football playing. For example, the movement of a support player may attract the attention of defenders and create space for other offenders. Accordingly, the associated functions are very relevant to the RFCS. To predict the future of the inferred situations and to know their consequences, functions facilitating context prediction are very relevant. For instance, if the last offender is observed to move up on the field (opposing the goal), an offside trap is about to happen. Based on the impact and the future of inferred situations, the future states of entities and relations could be predicted. For instance, based on a situation that ‘the turning speed of a defender is slow’ the RFCS may predict a future state that “the defender will be tired soon”. Moreover, F2.8 (Fill the CIR-cube with the predicted data) is considered as partially relevant since the predicted relations with regard to the orientations of the players require an extended function to process. Furthermore, to proactively manage future situations, RFCS also need functions for inferring newly formed situations in the near future.

In the ADM, the football pitch is not a structured space and thus the path-planning is not a required function. As a contrast, the action-plan generation and selection mechanisms are very relevant to making decisions about the optimal actions to be taken by the football players. In the case of football playing, different tactics might be executed by the players. An additional function (AF3.7) is needed for deriving the action plans, namely “Generate an offending strategy based on the dynamic context.” The strategy should be considered when the best action plans for the players are selected.

To enable the player to be aware of the context as soon as possible, the most relevant situation should be selected. However, it does not need to select a template since message construction should apply short and concise message templates, which could be predefined, to enable the players to understand the messages easily. Functions used for generating personalized messages (i.e. F4.3, F4.4, and F4.5) are relevant, whereas ordering the messages for communication is not relevant since the number of messages to be sent is small.

#### **5.3.4. Relevance of the workflow to the RTMS**

According to our investigation, the dependency of the functionalities for RTMS is the same as that for IFEGS. It means that the workflow can be reusable for the RTMS directly. Specifically, since the CIR-cube can be constructed by focusing on the changing distances among vehicles, the dependencies of functionalities in the IRM are very relevant. Concerning the KIM, situations should be firstly inferred out, based on which the implication and interplays can be obtained. Predicting the future of traffic context should be achieved based on the inferred situations and their changes. In addition, both the implications of inferred and predicted situations should be calculated and considered as a basis for ranking traffic situations and ranking vehicles on the high way.

The major considerations with regard to the relevance of the dependencies in ADM are (i) the path-planning function should depend on the impact of dynamically formed situations, which means that the penalty matrix should be generated before the paths for drivers are planned, (ii) the generation of alternative action plans should consider the determined paths for drivers to follow and (iii) the selection of alternative action plans for drivers should depend on the impact of traffic situations. With regard to the MCM, construction of informative messages depends on the relevance between traffic situations and individual drivers, while construction of instructive messages depends on the derived action plans. Both are essential for generating personalized messages.

#### **5.3.5. Relevance of the workflow to the HCAS**

In the IRM, as discussed in the previous section, some function (e.g. F1.3, F1.5, and F1.7) are not relevant for the HCAS. It means that their computational results will not be available for the subsequent functions. For this reason, calculation of the distances among entities in a list and all other entities and filling the SFR-matrix with the changed and unchanged distances, and the speed should depend on the results of the additionally included functions, including (i) predefine an entity list containing the identifications of the caregivers, and (ii) calculate the time-dependent changes of entity attributes. In the KIM, an additional function (AF2.15) is used for predicting future situations, which depends on the calculated



impact indicators of situations. The predicted information will be future used for predicting the future states of entities.

To generate optimal paths for the caregivers, an objective function should be constructed by considering the impact of situations, which considers the priority list of situations that the care recipients are or will be involved in. In addition, the caregivers may also disobey the instructions, which should be identified and considered in the generation of paths. In turn, the generated paths are used as the basis for predicting the obedience of caregivers, which means that the prediction of future context depends on the generated long-term solutions for people in this application. Furthermore, to generate personalized messages, the actions to be taken by the caregivers form the basis for generating instructive messages. Thus, the generated personal action plan should be excluded from the dependency list of F4.4 (construct instructive messages), while AF3.6 (specify the actions taken by the caregivers) is to be included.

### **5.3.6. Relevance of the workflow to the RFCS**

In case of football coaching, since two functions (i.e. F1.5 (Generate a list of entities whose locations have changed) and F1.6 (Calculate the distances among entities in a list and all other entities)) are irrelevant to this application, the calculation of the speed of motion of the entities could depend on the constructed SFR-matrix with the first row and diagonal cells filled. In addition, the only difference between the workflows of KIM for IFEG and for RFCS is that football players do not have danger levels to be assessed. The related functional dependencies are not applicable.

In the ADM, to generate alternative action plans for the players, the obedience of players, the SAT context data and the II of predicted situations should be known. It means that the action-plan generation depends on an overall consideration of the CIR-cube containing future information, predicted future states and relations of entities, and the calculated impact indicators of the predicted situations. Then, to generate an offending strategy based on the dynamic context (completed by the additionally included function), the constructed CIR-cube should be used. Based on the selected strategy, the best action plans for the players can be selected out from the alternatives. Furthermore, the workflow of the MCM for the RFCS is quite simple. Generation of personalized messages only considers the most relevant situation of individuals and the generated personal action plans.

## **5.4. Analysis of the suitability of the empirical structure of the computational mechanisms**

As introduced in Chapter 3, there are 8, 21, 10 and 6 algorithms in the IRM, KIM, ADM, and MCM, respectively. An overall view of the suitability of the

algorithms to the targets applications is shown in Table 5.5. Due to the diverse requirements of the target applications and the different attributes of the treated entities and situations, it can be seen that the qualitative evaluation considers three levels with regard to the suitability, namely (i) suitable (marked by ●), which means that the implemented algorithms can be directly used in the target application, (ii) suitable with a parametric modification (marked by ○), which means that the algorithms can be applied if some parameters are adjusted according to the requirements of the target applications (e.g. thresholds), and (iii) not suitable (marked by ×), which means that the concerned algorithms are not suitable at all.

Based on the suitability analysis of the algorithms, the applicability of the data constructs can be determined. The result of the assessment is presented in Table 5.6. Two qualitative levels were involved, including (i) suitable (marked by ●),

**Table 5.5.** Assessment of the suitability of the implemented algorithms

Func.	Algo.	IFECS	RTMS	HCAS	RFCS	Func.	Algo.	IFECS	RTMS	HCAS	RFCS
F1.1	A1	●	○	×	○	F2.8	A3	●	●	○	○
F1.2	A2	●	●	●	×		A5	●	●	●	●
	A3	●	●	○	○		A7	●	●	×	●
F1.3	A4	●	●	×	×		A8	●	●	●	●
F1.4	A5	●	●	●	●	F2.9	A9	●	○	×	×
F1.5	A6	●	○	×	×		A10	●	○	×	×
F1.6	A4	●	●	●	×	F2.10	A12	●	○	×	×
F1.7	A7	●	●	×	●	F2.11	A26	●	●	●	●
F1.8	A8	●	●	●	●	F2.12	A27	●	○	○	○
F1.9	A5	●	●	●	●	F2.13	A28	●	○	○	×
F2.1	A9	●	○	×	×	F2.14	A29	●	●	●	×
	A10	●	○	×	×	F3.1	A30	●	●	×	×
	A11	●	×	×	×	F3.2	A31	●	●	×	×
F2.2	A12	●	○	×	×		A32	●	●	●	×
F2.3	A13	●	●	×	●		A33	●	×	×	×
	A14	●	●	×	●		A34	●	×	×	×
F2.4	A15	●	×	×	×	F3.3	A35	●	●	×	●
	A16	●	×	×	×		A36	●	○	×	○
	A17	●	×	×	×	F3.4	A37	●	●	●	●
F2.5	A18	●	×	×	×		A38	●	●	●	●
	A19	●	×	×	×	F3.5	A39	●	●	×	×
	A20	●	×	×	×	F4.1	A40	●	●	×	●
F2.6	A21	●	×	×	×	F4.2	A41	●	○	×	×
	A22	●	×	×	×	F4.3	A42	●	○	○	○
	A23	●	×	×	×	F4.4	A43	●	○	○	○
F2.7	A24	●	●	×	×	F4.5	A44	●	●	●	●
	A4	●	●	●	×	F4.6	A45	●	●	×	×
	A25	●	○	○	○						

●: suitable, ○: suitable with a parametric modification, ×: not suitable

which means that the data construct is able to sufficiently represent the needed content for information processing in the target application context, and (ii) not suitable (marked by ×), which means that the data construct is either unneeded or insufficiently represents the required information. A detailed explanation of the assessment for each of the target applications is given in the following sections.

**Table 5.6.** Assessment of the suitability of the data constructs

Data	Related algorithms for IFECS	Suitability			Data	Related algorithms for IFECS	Suitability		
		RTMS	HCAS	RFCS			RTMS	HCAS	RFCS
D1	A1	●	●	●	D25	A3	●	●	●
<b>D2</b>	A2, A4, A30, A31, A24, A35	●	●	×	D26	A5	●	●	×
D3	A2, A3	●	●	●	D27	A9, A10	●	●	×
D4	A3	●	●	×	D28	A12	●	●	●
D5	A4, A6	●	●	×	<b>D29</b>	A26, A27, A30	●	●	●
D6	A5	●	●	×	<b>D30</b>	A37	●	●	●
<b>D7</b>	A5, A6, A8, A9, A10, A11, A24, A27, A32, A33, A34, A35, A36	●	●	×	<b>D31</b>	A28, A40	●	●	●
D8	A7, A4	●	●	●	<b>D32</b>	A29, A41	●	×	×
D9	A8	●	×	×	<b>D33</b>	A45, A34	●	●	●
D10	A8	●	×	●	<b>D34</b>	A33	●	●	●
D11	A12, A13, A14, A24	●	●	●	D35	A31	●	×	×
D12	A27	●	●	●	D36	A34	●	×	×
D13	A15, A16, A17	●	●	●	D37	A34	●	●	×
D14	A18	●	×	×	D38	A34	×	×	×
D15	A19	●	×	×	D39	A35	●	●	×
D16	A20	×	×	×	D40	A36	●	×	●
D17	A21	×	×	×	D41	A39	●	×	●
D18	A22	×	×	×	D42	A38	●	●	●
D19	A23	×	×	×	D43	A39	●	●	●
D20	A12	●	×	×	D44	A42	●	●	●
D21	A12, A24	●	×	×	D45	A42, A43	●	●	×
D22	A12	×	×	×	D46	A44	●	●	●
<b>D23</b>	A25, A43	●	●	●	D47	A44	●	●	●
D24	A4	●	●	●	D48	A45	●	●	●
					<b>D49</b>	Output	●	●	●
					<b>D50</b>	A41	●	●	×
					<b>D51</b>	A42, A47	●	●	×

●: suitable, ×: not suitable. Marked by **bold**: cross-module data construct

### 5.4.1. Suitability of the algorithms to the RTMS

In the case of traffic management, locations of vehicles should be estimated in a time-dependent manner. The applied Kalman-filter should be optimized according to the practical considerations, e.g. use a proper estimate of the noise covariance matrix. In addition, since RTMS manages vehicles within a structured physical space (i.e. the interconnected roads), the algorithm developed for judging the ID of the sub-zone of people is suitable if the traffic network is represented by the proposed digital representation mechanism.

In this context, the developed algorithm for calculating the distance among entities is suitable. In order to apply the rest of the algorithms belonging to the IRM, only the thresholds used for judging if the location of an entity changes or not should be modified. This is because the concerned traffic management might neglect vehicles that are moving with a small change of locations between two successive computational cycles.

To infer traffic situations, floating traffic jams and anchored traffic jams are similar to people jams in principle. However, the thresholds of distance among entities used for judging the formation of traffic jams and people jams are different. In addition, algorithms dedicated to processing information about the fire (i.e. A11, A17, A20, and A23) are not suitable in this case. Actually, for traffic management, more situations should be inferred and various algorithms are needed, e.g. the motion of a fire truck, collision of two vehicles, traffic accidents and road maintenance. Furthermore, the proposed algorithm for calculating impact indicator of people jams is suitable to be applied since the impact of traffic jams can be calculated based on the number of involved vehicles and the speed reduction of the same. However, the coefficient might be different from the fire evacuation application.

The algorithms proposed for predicting the future of people jams are not suitable to be applied for predicting traffic jams. Because our solution only calculates the location of a people jam as the geometric center of the shape formed by people involved in the people jam. Calculation of the speed of motion of people jams follows the same principle as well. This algorithm works when the number of entities is relatively small. However, a traffic jam might include hundreds of vehicles, whose location cannot be simply represented by a point and the algorithm for calculating the location changes cannot be directly applied. For the same reason, the attributes of existing situations (e.g. location, speed of motion) cannot be predicted using regression model-based method.

Then, the algorithm for predicting the future locations (i.e. A21) cannot be applied. Due to the fact that threshold used for judging the obedience of informable people might be different between the reference application and the traffic management, some parameters in the algorithm for predicting the

obedience of people need to be modified before the algorithm can be properly applied. Furthermore, the algorithm used for calculating the relevance between situations and people considers some application-dependent parameters (e.g. maximum time and maximum distance) in computation, which should be adjusted for the traffic management application. The threshold used for calculating the danger level indicators of entities should be modified as well.

To enable path-planning for vehicles, the penalty values used for constructing penalty matrices should be adjusted. In addition, traffic management does not need ‘informants’ to provide help in emergency situations, algorithms dedicated for realizing group-level optimization in case of fire evacuation (i.e. A33 and A34) are not suitable to be applied. Due to the difference between the capabilities of a vehicle and a person with regard to the execution of actions (e.g. different speed of motion), the alternative action plans calculated should be adjusted according to the specific requirements for traffic management (e.g. change the lane). To generate personalized messages, in this case, message components and templates should be revised, which means that (i) message component generation, and (ii) message template selection needs a parameter level modification.

#### **5.4.2. Suitability of the algorithms to the HCAS**

In the case of home care-taking, the estimation of the locations is not a suitable algorithm since this application is not sensitive to the accuracy of locations of caregivers. More information about the attributes of the care recipients (e.g. age, gender, preference, etc.) and about the caregivers (e.g. equipment status, service capacity, etc.) should be included in their content profiles. The proposed distance calculation algorithm is suitable to be applied for calculating the distance among caregivers and care-recipients. Accordingly, the algorithm developed for calculating the speed of motion of entities is not a crucial factor for decision-making in this application.

Jams formed by the aggregation of a number of entities and an entity representing a hazardous phenomenon (i.e. the fire) do not need to be considered in this application. This implies that all the algorithms related to the jams and the fire, such as algorithms in F2.1, F2.2, F2.6, F2.7, F2.9, and F2.10, are not relevant. Although the application context is different, the algorithms related to processing the contents of CIR-cubes can still be used (e.g. calculating distances and constructing SFR-matrices and CIR-cubes). In addition, the algorithm for predicting the obedience of people is suitable for evaluating the obedience of caregivers unless an adequate diversity threshold is given. Furthermore, the HCAS may have a different principle for calculating (i) the relevance between situations and entities and (ii) the danger level of entities, which means that the application-dependent thresholds used for calculating the relevance indicator and danger levels should be adjusted.

In the ADM, the algorithm for determining the optimal paths of individuals can be applied since the care-recipients can be considered as different exits for the caregivers to choose. Algorithms articulated for generating the objective functions (i.e. A37 and A38) are also suitable. Furthermore, to generate informative messages for describing the situations that are involved by the care-recipients, application domain specific message templates and message components should be predefined. Therefore, the related algorithm (i.e. A41, A42, and A43) are partially suitable for HCAS, while message integration algorithm can be properly applied.

### **5.4.3. Suitability of the algorithms to the RFCS**

For football coaching case, the algorithmic solutions for determining sub-zone of entities and calculating distances among entities are not suitable since the linear distances among players should be considered. More attributive aspects regarding the players should be considered, which means that a modified algorithm for constructing a content profile can be suitably applied.

The algorithms used for inferring particular situations for football play should be articulated. Due to the reason that the algorithms included in F2.1, F2.2, F2.4, F2.6, F2.9, and F2.10 are articulated for the fire evacuation cases, they cannot be suitably applied for football playing. However, since the interplays inferring algorithms (i.e. A13 and A14) are designed in an application-independent way, they can still be applied to explore the interplays of situations in the football case, e.g. the consequence of an offensive opportunity. The algorithms for predicting the locations of people cannot be applied for predicting those of football players since the players may always change their speed of motion intentionally. Last, the obedience prediction is still suitable for judging the temporal obedience of players in case of RFCS, and algorithms for calculating the relevance indicators between situations and players can be used if several thresholds in the algorithms are modified.

To devise personal action-plans for dynamic context information management, the RFCS could compute based on the positions that will be reached by informable people on their path. Using these positions, the algorithm for generating alternative positions for people is also suitable but should be revised since more alternative action plans to be performed by concerned football players might be generated. The action-plan selection algorithm is not suitable in this case. This is because the principle for selecting action plans should not only consider the objective function, but it also depends on the generated strategy in the additional function (i.e. AF3.7). In the MCM, after concise and representative message components are predefined, personalized messages could be constructed.

#### **5.4.4. Suitability of the data constructs to the RTMS**

All cross-module data constructs employed by the DCIP-mechanism are suitable to be applied for the RTMS. Firstly, the layout information of a high way could be represented by the proposed approach and different sub-zones could be generated to enable distance calculation among vehicles. In addition, the SFR-Matrix and CIR-cube integrate SAT data with regard to the vehicles and their spatial relations, which is considered to be an effective way of representing the DCI of vehicles. Particularly, for a given cell belonging to the upper and lower triangular parts of an SFR-matrix, the content can be used to represent the length of travel route from one vehicle to another. For the purpose of monitoring the execution of action plans by any informed drivers, the obedience of drivers, stored by an array with a set of Boolean values, is also an important aspect for generating adaptive strategies for context management. Furthermore, the inferred semantic knowledge, including (i) the priority list of situations, (ii) the priority list of people, (iii) impact indicators of predicted situations, (iv) danger level indicators of people, and (v) relevance indicators between situations and people, are all useful for making decisions for the purpose of traffic management. Although different situations may be involved, and varied messages templates and complements may be applied, the related data construct (i.e. D50 and D51) are suitable to the RTMS since they enable message construction describing both context information and instruction. In terms of inner-module data constructs, data constructs dedicated for describing fire situation (i.e. D16, D19, and D22) are not suitable. Moreover, because the regression model approach is not suitable for context prediction in case of RTMS, the employed data constructs (i.e. D17 and D18) are not needed.

#### **5.4.5. Suitability of the data constructs to the HCAS**

Concerning the cross-module data constructs, danger level indicators of people are not suitable to be applied in case of HCAS. Because the danger level of caregivers is not important for message editing. The rest data constructs can be properly used and the reasons are as the same as explained above. However, many inner-module data constructs cannot represent sufficient information with regard to the management of care-taking context. For instance, the speed of motion of people is unnecessary information for this application. In addition, due to the reason that neither people jams nor fire situations should be considered, data constructs used for representing attributes of the situations (i.e. D14, D15, D16, D17, D18, D19, D20, D21, and D22) are irrelevant to this application case. In the ADM, there are many data constructs that are used for algorithms in case of a fire evacuation, such as, the penalty matrix, optimal paths, and length of the paths among SLMs, and potential informants and help recipients. All the above-mentioned information is not required for HCAS. Furthermore, the scheduling of

care-taking does not need to generate short-term action plans for caregivers. For this reason, data constructs employed for storing action-plans are not needed.

#### **5.4.6. Suitability of the data constructs to the RFCS**

The football pitch is an open space. For this reason, the proposed data constructs are not suitable to represent the layout of physical space in this application context, such as sub-zone, SLMs, etc. In addition, both the distances and the orientation relations among football players are crucial for inferring situations and making decisions. We argue that the CIR-cube (D7) cannot sufficiently represent the DCI in this application case. Accordingly, all the related data constructs (e.g. D5, D6, D9, D26, and D27) are not suitable as well. Furthermore, path-planning is not a required function in case of football playing. Data constructs for path-planning for indoor cases (i.e. D35, D36, D37, D38, and D39) are not suitable for RFCS. Lastly, due to the fact that concise and brief messages are required, message composition is not a required function. Both instructive and informative messages can be obtained directly from a library of predefined messages according to certain situations.

### **5.5. Analysis of the empirical performance efficiency of the computational mechanisms**

This sub-Chapter introduces the details about the empirical performance evaluation of the DCIP-M in different application cases. In Chapter 4, a performance testing of the implemented prototype was conducted with regard to both static and dynamic computational loads. The implemented algorithms behaved differently when different conditions were applied. Accordingly, the results obtained from the performance testing were projected to the context of the three target application cases concerning the characteristics and technical factors presented in Section 5.2.4. The process applied for efficiency analysis is detailed as follows.

Regression models showing the relations between varied static computational loads and the computation time for executing the implemented algorithms were established based on the obtained results from the performance testing. Then, the required computation time of algorithms in each of the target application cases were estimated based on the generated regression models and the practical characteristics of the target application cases, such as (i) the maximum possible number of entities handed at a time (the worst case) and (ii) the maximum sudden changes of entity numbers (the worst case). After this, the estimated computation time of algorithms was compared to the allowed computation time (ACT) in each of the application contexts, which is calculated by using:

$$ACT = ART - (LTS + LTM) \quad (5-1)$$



where *ART* refers to the allowed response time in a given application case, *LTS* and *LTM* represent for the practical latencies of information sensing and messaging, respectively. If the estimated computation time of an algorithm is higher than the *ACT*, then the algorithm is considered as ‘not effective’. Otherwise, the algorithm is assessed as ‘effective’. The estimated computation time (ECT) and the results of the assessment (RA) of the implemented algorithms in the target application cases are presented in Table 5.7.

In addition, in the functionality testing of the implemented prototype, several factors that influence the DCI management in the case of IFEGS were identified. This sub-Chapter also evaluates the effect of these factors in the target application cases. Two qualitative levels were used in the assessment, including (i) high,

**Table 5.7.** Assessment of the efficiency of the algorithms in the target applications

Algo.	RTMS		HCAS		RFCS		Algo.	RTMS		HCAS		RFCS	
	CT(s)	RA	CT(s)	RA	CT(s)	RA		CT(s)	RA	CT(s)	RA	CT(s)	RA
A1	5e-3	●	Δ		4e-4	●	A24	4e-1	×	Δ		Δ	
A2	9e-1	×	4e-2	●		Δ	A25	5e-2	●	5e-4	●	3e-4	●
A3	7e-3	●	1e-4	●	1e-4	●	A26	3e-3	●	2e-4	●	4e-5	●
A4	3e1	×	3e-2	●		Δ	A27	8e-1	×	4e-2	●	1e-2	●
A5	2e-2	●	2e-3	●	2e-3	●	A28	2e-2	●	2e-3	●		Δ
A6	2e-2	●		Δ		Δ	A29	6e-3	●	9e-4	●		Δ
A7	1e-2	●		Δ	2e-3	●	A30	7e-2	●		Δ		Δ
A8	8e-4	●	5e-4	●	2e-4	●	A31	2e-3	●		Δ		Δ
A9	2e1	×		Δ		Δ	A32	5e-2	●	3e-3	●		Δ
A10	9e-2	●		Δ		Δ	A33		Δ		Δ		Δ
A11		Δ		Δ		Δ	A34		Δ		Δ		Δ
A12	5e-4	●		Δ		Δ	A35	2e-2	●		Δ	8e-5	●
A13	2e-1	●		Δ	1e-3	●	A36	4e-2	●		Δ	1e-4	●
A14	1e-1	●		Δ	1e-3	●	A37	1e-1	●	1e-2	●	8e-3	●
A15		Δ		Δ		Δ	A38	4e-2	●	3e-3	●	1e-4	●
A16		Δ		Δ		Δ	A39	8e-2	●		Δ		Δ
A17		Δ		Δ		Δ	A40	1e-2	●		Δ	6e-4	●
A18		Δ		Δ		Δ	A41	4e-2	●		Δ		Δ
A19		Δ		Δ		Δ	A42	1e0	×	5e-3	●	7e-3	●
A20		Δ		Δ		Δ	A43	1e0	×	5e-3	●	7e-3	●
A21		Δ		Δ		Δ	A44	1e-1	●	8e-4	●	1e-3	●
A22		Δ		Δ		Δ	A45	9e-3	●		Δ		Δ
A23		Δ		Δ		Δ							

●: the algorithm is effective, ×: the algorithm is not effective, Δ: the algorithm is not needed

**Table 5.8.** Assessment of the effect of several influencing factors on the DCI management of the concerned application cases

No.	Influencing factors	IFECS	RTMS	HCAS	RFCS
1	Accuracy of positioning	High	Low	Low	High
2	Latency of information sensing	High	High	Low	High
3	Length of personalized messages	High	High	Low	High
4	Obedience of informable people	Low	Low	High	High

which means that the factor is very crucial for DCIP in the concerned application case, and (ii) low, which means that the factor has limited impact. The results of the assessment are shown in Table 5.8.

### 5.5.1. Efficiency of the algorithms in the RTMS application

In the case of traffic management, the ACT is estimated to be smaller than 0.4s. For this reason, three algorithms cannot be completed within this period of time, including determination of the sub-zone of entities, calculating the distance among entities and inferring traffic jams. Particularly, the computation time spent on calculating the distances among 1000 vehicles was estimated to be 23 seconds. However, the number of vehicles on the road may suddenly change.

According to attributes assumed in Table 5.1, the size of change might be a maximum of 200 vehicles. In addition, as recommended by the performance test with regard to the dynamic loads, a 30% overshoot should be taken into estimation of the computation time of this algorithm in case of RTMS. It means that, if the proposed approach is applied, at least 30 seconds should be reserved for the distance calculation. Therefore, distance calculation based on the used DCI representation requires a more effective algorithm in the case of traffic management.

In addition, predicting the near-future locations of entities (A24) cannot be properly applied for context prediction of traffic situations, while relevance calculation (A27) is not suitable for calculating the relevance between situations and vehicles. The time cost of these two algorithms depends on the number of inferred situations. In practice, they can only be applied if the number of situations to be handled is small. Furthermore, the implemented algorithms for message component generation (i.e. A42 and A43) cannot construct personalized messages for 1000 drivers within the required computation time. This implies that the proposed messaging strategy, which prioritizes users at sending messages, is useful. Actually, when the number of entities is relatively large, the RTMS can dynamically select a part of the drivers to inform during runtime, such as

informing the most relevant driver to a concerned situation for the purpose of management. This may reduce the computation time of these two algorithms.

### **5.5.2. Efficiency of the algorithms in the HCAS application**

According to the assumed characteristics, the allowed response time for HCAS is quite long (assumed to be 10 seconds). It means that the HCAS has enough time to complete all the required computations. Therefore, all the related algorithms could be effectively applied in this application context. Specifically, since only the distances between caregivers and care recipients are meaningful for making decisions for caregivers, the number of distances to be calculated in this case is relatively small. The distance calculation algorithm can be executed in real-time. In addition, only caregivers are required to be informed at a time, which means that informing caregivers can be done in a very short time.

### **5.5.3. Efficiency of the algorithms in the RFCS application**

For the purpose of football play coaching, the allowed computation time is 0.4s. In a general sense, all the relevant algorithms are effective for managing the dynamic context with regard to only 23 entities. As explained earlier, all time-consuming algorithms (i.e. A4, A9, and A34) are irrelevant to this application case. Actually, in the calculation of the spatial relations among entities (e.g. distances and orientation relations), the influence of the layout of the physical space is not involved. For instance, only linear distances among entities are to be calculated. It implies that the investigation of the spatial relations among entities in this application context is more time-efficient than the reference application. In addition, due to the fact that only offenders are to be informed, 11 personalized messages are required to be constructed at a time, which could be realized in real-time.

### **5.5.4. Effects of the influencing factors on the performance in the RTMS application**

According to our findings from the functionality testing, the accuracy of indoor positioning had a crucial impact on the distance calculation for the purpose of DCI representation. However, the relatively low accuracy of positioning in the case of RTMS has a limited impact on the DCI representation. Because the RTMS is typically applied over a physical space covering several to tens of kilometers. And, the error in positioning does not influence the judgment on the sub-zone (section of road) of vehicles. Therefore, the accuracy of positioning has little influence on the distance calculation among vehicles on a highway. On the other hand, the latency in information sensing may have a considerable impact on the DCI representation and short-term action-plan deriving. Because, (i) vehicles may move with a fast speed of motion and (ii) drivers may accidentally change

the movement speed of vehicles, which may result in a large deviation between the represented context information and the actual context. For instance, in the demonstrated case shown in Figure 5.4, the car that blocks the fire truck may change its position (e.g. move to the side of the fire truck) within 0.5s. Due to the latency of information sensing, the RTMS may consider that the car still blocks the fire truck and generate wrong instructions to inform the driver. Furthermore, the length of personalized messages is also deemed to be an important factor in managing traffic contexts. The reason is that a long message may attract the attention of the drivers while driving cars. Furthermore, as it was shown by the evaluation of the generated messages by human subjects, a long message may increase the anxiety of the people in emergency situations.

### **5.5.5. Effects of the influencing factors on the performance in the HCAS application**

In the case of the HCAS, neither the latency nor the accuracy of information sensing plays a critical role in the performance of DCI management. This is because that the HCAS does not need caregivers to perform any instant (short-term) actions. The real-time capability is not strongly required in this application context. Since caregivers have enough time to read the generated messages, the length of personalized messages does not influence the empirical performance of the computational mechanism too much. In the case of HCAS, only several (e.g. three in the demonstrative example shown in Figure 5.5) caregivers are to be informed at a time. If any of them does not follow the given instructions, the system can hardly generate an alternative solution. This is the reason why we believe the obedience of informable people has a high influence on the DCI management.

### **5.5.6. Effects of the influencing factors on the performance in the RFCS application**

During the football playing case, a slight change with regard to the attributes of players and spatial relations among them may result in a different interpretation of the dynamic context. Since the context changes rapidly, all concerned factors have a crucial impact on the performance of the RFCS. Firstly, the accuracy of positioning and latency of information sensing are important since they may influence the inference of situations, such as an offside, open shot. The RFCS should respond to any event concerning the contextual changes. In addition, football players should react to the personalized messages immediately after the (voice) messages are received. If the messages are too long, it may take a long period of time for reaction by players. Thus, personalized messages should be concise and short in this application case. Furthermore, the offenders should cooperate together to achieve the generated group-level optimized solutions. Any

disobeying action taken by a player may have an impact on the execution of action plans by other players.

## 5.6. Analysis of the theoretical performance sufficiency of the computational mechanisms

### 5.6.1. Revisiting the theoretical performance targets

There is a wide range of concrete and abstract theoretical performance targets (TPTs) can be claimed for a complex computational mechanism. While the performance indicators can be evaluated quantitatively, TPTs need a qualitative evaluation due to their abstract and often tentative nature. Adhering to the same strategy that was applied in the context of the empirical performance analysis, we will focus on the generic performance targets that can be applied equally well to each target application. For this reason, the following three TPTs have been chosen for this part of the applicability analysis:

- **Adaptation need**, which is judged by considering the total effort that is needed to adapt the elements of the modules (e.g. algorithms, data constructs) to the target application. (Its value is expected to be small).
- **Preparation effort** (should be low), which is judged by considering the total preparation of a module (e.g. specification of thresholds in algorithms, and data constructs) when it is applied in the target application.
- **Dependability** (should be low), which is judged by considering the total dependency of the computational results generated by an adopted and prepared platform (all modules together) to the varying circumstances in the target application.

Below, we project these TPTs to the three target applications and apply the following reasoning logic at deciding on the applicability:

- a. If either the testing of the functionality, or the testing of the overall workflow, or both, results in a negative outcome, then the proposed mechanism cannot be applied in the target application cases, since the expectations for its theoretical structural relevance is not fulfilled.
- b. If either the testing of the processing algorithms, or the testing of the processed data constructs, or both, concludes with a negative result, then the proposed mechanism cannot be applied in the target application chases, since its empirical structural suitability is not validated.
- c. If either the specific performance indicators suggest poor efficiency or the factors influencing the computational efficiency have large effects on the

outcomes (and makes the computation unpredictable), or both concurrently appear, then the proposed mechanism cannot be applied in the target application cases.

- d. If the theoretical structural relevance, the empirical structural suitability, and the empirical performance be all validated, then the sufficiency for an application depends on the achievement of the TPTs.

On the basis of the results obtained in Sections 5.3. - 5.5, items (a), (b) and (c) are known concerning each of the target application cases. Actually, three indicators can be used to represent the assessment results with regard to the previous three aspects, namely (i) theoretical relevance indicator (*TRI*), (ii) empirical suitability indicator (*ESI*), and (iii) empirical efficiency indicator (*EI*). For each indicator, a percentage value can be calculated, showing the proportion of applicable elements (e.g. function, algorithms, data constructs) of a module for each of the target applications. For instance, if 2 algorithms and the related data constructs in a module (a mechanism) are suitable to be applied for a target application, and there are 10 algorithms in the module (a mechanism) in total, then we claim that the *ESI* of the mechanism is 20% in case of the target application.

As indicated by the proposed method, the individually assessed results should be jointly synergized to enable a sufficiency evaluation of the computational mechanism. For this reason, an overall applicability indicator (*OAI*) is defined, which is calculated by using the three individual indicators. The calculation is realized based on the following equation:

$$OAI = \sqrt[3]{TRI * ESI * EI} \tag{5-2}$$

As shown, the geometric mean value among the three indicators is concerned because they are related to each other. Therefore, a qualitative evaluation of the proposed computational mechanisms with regard to the TPTs is made, in which three levels are concerned, including (i) high, (ii) medium, and (iii) low. The principles applied for evaluating TPTs are presented in Table 5.9. The first TPT (adaptation need) was assessed based on the calculated *OAI*. The principle implies that if a low *OAI* is obtained, then a computational mechanism will need a high adaptation work

**Table 5.9.** Principles applied to qualitatively assess the TPTs

Assessed levels	Principles
Adaptation need	
High	$OAI \leq 25\%$
Medium	$25\% < OAI < 75\%$
Low	$OAI \geq 75\%$
Preparation effort	
High	$ESI \leq 25\%$
Medium	$25\% < ESI < 75\%$
Low	$ESI \geq 75\%$
Dependability	
High	$OSI \geq 75\%$
Medium	$25\% < OSI < 75\%$
Low	$OSI \leq 25\%$

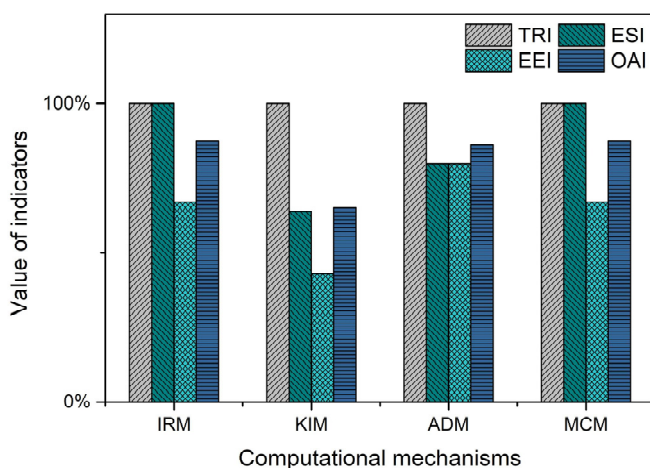
before it can be applied in the target application. The second TPT (preparation effort) was assessed only based on the calculated ESI. Because we believe that the preparation effort for using a computational mechanism in an application depends on how much change work with regard to the algorithms and data constructs is needed.

Moreover, as discussed in sub-chapter 5.5, there are multiple influencing factors that may affect the empirical performance of the I-CPSs dedicated to the target applications. Since the proposed platform (multiple modules) is supposed to be applied as a computational kernel for the I-CPSs, the dependability of the computation taken by the platform should rely on its sensitivity to the influencing factors. For this reason, we defined an overall sensitivity indicator (*OSI*) to represent the proportion of influencing factors that an adopted and prepared I-CPS is sensitive to. The *OSI* is concerned in the evaluation of the dependability of the entire mechanism as a whole. The applied principle indicates that the more influencing factors affect the practical computation, the higher the dependability of the platform will be. In this situation, a reliable computational result may not be obtained in varying circumstances.

It can be seen that the assessment principle indicates what level of compliance and what distribution profiles of compliance the computational mechanism have. Below, the sufficiency will be characterized by considering the fulfillment of the theoretical performance targets.

### 5.6.2. Achievement of the overall performance targets in the RTMS application

As shown in Figure 5.7, the values of *TRI*, *ESI*, *EEL*, and *OAI* for the RTMS are



**Figure 5.7.** Calculated applicability indicators of the proposed mechanisms for RTMS

**Table 5.10.** Assessment results of the TPTs in case of RTMS

TPTs	IRM	KIM	ADM	MCM	Platform
Adaptation need	Low	Medium	Low	Low	Low
Preparation effort	Low	Medium	Low	Low	Low
Dependability	-	-	-	-	Medium

calculated. It can be seen that the proposed mechanisms for representing dynamic context information (IRM) and constructing personalized messages (MCM) are applicable to the RTMS. The main drawback of using these two mechanisms is that some algorithms are not effective enough for real-time computation. Accordingly, the time-consuming algorithms (e.g. distance calculation) should be further improved before the mechanisms can be properly used. For the rest two mechanisms (KIM and ADM), although all the proposed functions are relevant, several new algorithms and data constructs should be developed.

According to the principles, the assessment results of the TPTs are presented in Table 5.10. The results have shown that the adaptation need and preparation effort for using the proposed mechanisms in the traffic management application was relatively low. Since traffic management may involve situations different from those considered in fire evacuation application, a medium adaptation and preparation work is required.

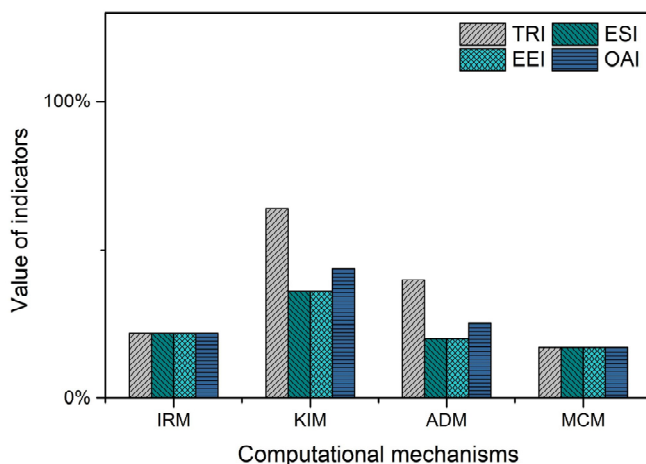
To evaluate the applicability of the platform as a whole, mean values of the *OAI* and *ESI* of the four modules were calculated. The ‘adaptation need’ and ‘preparation effort’ of the platform were qualitatively evaluated according to the same principle proposed in Table 5.9. It can be seen that relatively low adaptation and preparation work is needed. It means that the computational mechanisms dedicated to a fire evacuation guiding system have high compliance with a road traffic management system. For this reason, we believe the proposed platform can be used for management of dynamic traffic situations.

In addition, since the RTMS is only sensitive to the latency of information sensing and length of personalized messages, the computation taken by the platform is moderately dependent on the practical varying circumstances. This implies that in some circumstances, such as (i) large latency of information sensing happens, or (ii) long personalized messages are to be constructed, expected empirical performance of the platform may not be realized if it is used for the RTMS.

### 5.6.3. Achievement of the overall performance targets in the HCAS application

Concerning the HCAS, the assessment indicators are calculated and shown in Figure 5.8. It can be seen that only a few computational elements in the proposed





**Figure 5.8.** Calculated applicability indicators of the proposed mechanisms for HCAS

four modules, e.g. functions and algorithms, can be properly applied. In addition, the HCAS does not have a strict requirement on the response time of computations and the number of entities to be handled at a time is relatively low. For this reason, all the suitable algorithms can be effectively executed within a required computation time.

The qualitative assessment results with regard to the TPTs are shown in Table 5.11. It can be seen that the relatively low values of the indicators result in a moderate adaptation need and a high preparation effort for applying the entire platform for the HCAS. Particularly, most of the functions in the IRM and MCM should be revised, and new algorithms should be designed to satisfy the technical requirements. In addition, since the HCAS is only sensitive to the obedience of informable people, the dependability of the computation taken by the platform is relatively low in the case of HCAS. Actually, the informable people are caregivers who are obligated to finish the care-taking tasks. After an adequate adaptation and preparation work, the computation executed by the platform could provide reliable results for the HCAS.

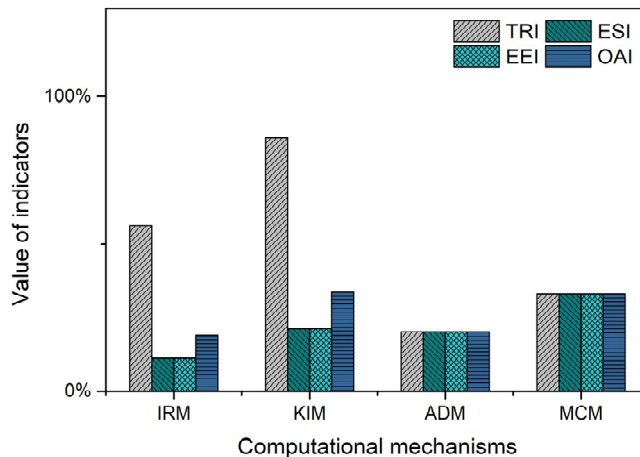
**Table 5.11.** Assessment results of the TPTs in case of HCAS

TPTs	IRM	KIM	ADM	MCM	Platform
Adaptation need	High	Medium	Medium	High	Medium
Preparation effort	High	Medium	High	High	High
Dependability	-	-	-	-	Low

#### 5.6.4. Achievement of the overall performance targets in the RFCS application

The calculated indicators in the case of RFCS are shown in Figure 5.9. High values of the *TRI* of the IRM and KIM are obtained, which means that most of the functions articulated for the IRM and KIM are relevant to the RFCS. However, due to the unsuitable data constructs, (e.g. this application may process the orientation relations among football players), most of the algorithms are not suitable to be applied, which resulted in low *ESIs* of the IRM and KIM, respectively.

By using the proposed principles, assessment results of the TPSs are presented in Table 5.12. It can be seen that moderate adaptation and high preparation effort should be paid for the purpose of using the proposed platform (all mechanisms) for the RFCS. In addition, all the considered influencing factors may significantly affect the practical computation of the RFCS. It means that the computation taken by the platform is dependent on the practical setups (e.g. hardware, communication networking) if it is applied for an RFCS. Therefore, to enable the application of the platform, specification of the system setups will need a dedicated consideration.



**Figure 5.9.** Calculated applicability indicators of the proposed mechanisms for RFCS

**Table 5.12.** Assessment results of the TPTs in case of RFCS

TPTs	IRM	KIM	ADM	MCM	Platform
Adaptation need	High	Medium	High	Medium	Medium
Preparation effort	High	High	High	Medium	High
Dependability	-	-	-	-	High

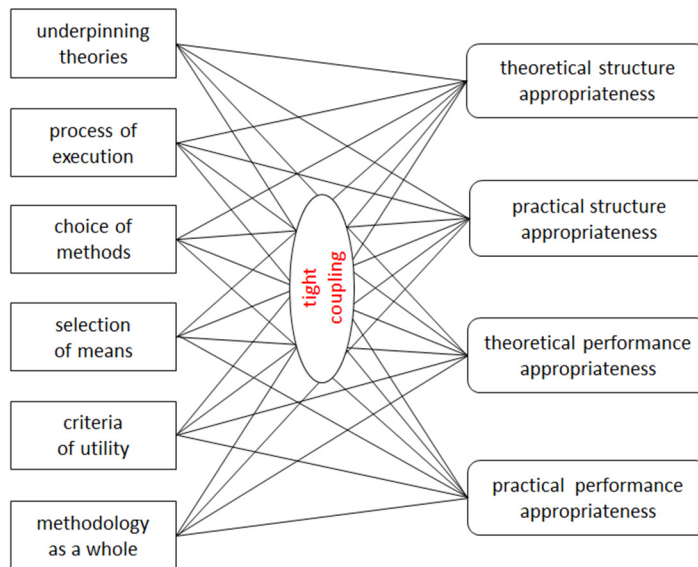
## **5.7. Some reflections on the findings and approach of validation**

### **5.7.1. Reflections on the approach of applicability validation**

In the tradition of scientific inquiry, internal validation and external validation goes hand-in-hand, since a body of knowledge, which suffers from internal biases due to lack of procedural rigor, and/or the one, which is not appropriate for or has serious limitations with regards to its intended purpose, can be considered as scientific of knowledge. Focusing on useful design methods, the original VSA targeted external validity, which manifests as the process of building confidence in the usefulness of a body of knowledge with respect to a purpose. In the context of design methods, utility validation is driven by the question of whether the method provides design solutions ‘correctly’ (effectiveness) and whether it provides ‘correct’ design solutions (efficiency). Correct in this context are design solutions with acceptable operational performance, which are designed and realized with less cost and/or in less time.

It can be inferred logically that AVSA rests on the assumption that internal validity is guaranteed. This obviously needs specific measures in the process of theory generation and design methodology development, as well as rigorous testing when the result of the process is available. It is also worth mentioning that AVSA also recognizes that ‘formal, rigorous and quantitative’ validation cannot be applied without problems in certain areas of engineering research, which much less rely on subjective statements than on physical experimentation and mathematical modeling.

In the three application cases, the applicability of the computational methodology of the DCIP-M was based on the pieces of evidence obtained in terms of functional/procedural relevance, the suitability of the algorithm and the data constructs, and the efficiency and sustainability of the performance. If the computational methodology of DCIP-M fails in any one of these aspects, then the theoretical performance targets (sufficiency) cannot be fulfilled and there is no way to conclude about the applicability of the DCIP-M positively in the considered application. If relevance and suitability are positively evaluated, then the empirical performance plays a decisive role in the judgment. If the relevance and suitability indicators present partial compliance only, and the efficiency designate a partial completion, in that case, the functional structure, the procedural flow, the algorithms and data structures, and the computational measures of the DCIP-M should be adapted to those required by a specific application.



**Figure 5.10.** Tight coupling among the methodological enablers and the aspects of validation

### 5.7.2. Conclusions

Over multiple applications in various studies, some deficiencies of the original validation square approach (VSA) have been recognized, which are: (i) lack of explicit focus on formal verification and internal validation, and (ii) not being tailored to the epistemological and methodological challenges of constructive computational methodologies (CCMs). In addition, it is of importance to consider that any epistemic-methodological construct (e.g., design method, creative tool, computational methodology) should in the first place be applicable to the task at hand, and just then can fulfill the usefulness requirements (i.e., to resolve the applicability versus utility issue). These stimulated our efforts towards the proposed adaptation of the original VSA.

As a framework of validating design methods, VSA is based on a combination of qualitative and quantitative perspectives (like a cross-case study), as well as of theoretical and empirical viewpoints (like experimental research). This makes it relevant for validation of CCMs, in the case of which both structural appropriateness and performance appropriateness need to be assessed from both theoretical and empirical points of view. In addition to using meaningful and reliable information, the advantage of AVSA is that it can be used to explore the range and/or the extent of applicability of various other constructive computational methodologies too. The reason is that, as shown in Figure 5.10,

AVSA establishes a tight coupling among the enablers embraced by a CCM and the aspects of theoretical and empirical validation.

We proposed to judge the applicability of a constructive computational methodology in terms of four aspects and various indicators related to them. The measure of theoretical structural appropriateness is relevance that can be evaluated by considering whether (i) the functionality offered by the CCM is relevant for the target application, and (ii) the overall computational workflow is relevant for the target application. The empirical structural appropriateness is measured by suitability that can be evaluated by considering if (i) the processing algorithms are suitable for the target application, and (ii) the processed data constructs are suitable for the target application. The empirical performance efficiency is measured in terms of practical efficiency, which can be captured by two indicators, namely, if: (i) the specific performance indicators confirm efficiency, and (ii) the effects of the factors influencing the processing are controllable. Finally, the theoretical performance is expressed in terms of overall sufficiency, which can be characterized by (i) the extent of achieving the overall performance targets, and (ii) the level of sustainability of the overall performance in varying circumstances. The completed applicability-testing case study confirmed that these measures are adequate and expressive.

Concerning our computational mechanisms, the proposed CIR-cube is an application-independent approach that is able to represent SAT data with regard to entity states and relations. It can be directly applied in applications that make decisions based on distances among entities and can be used with a minor adaptation to other applications that require the processing of other types of relations, e.g. orientation relations among people, e.g. in case of football training. In addition, the context knowledge pyramid is a general concept model that fits many smart system applications with the objective of dynamic context information management. The computational mechanisms proposed for calculating the impact of situations and the relevance between situations and entities can be adapted to other I-CPS applications. Accordingly, the approach to the formulation of personal context in time-varying processes is transferable. Furthermore, the adopted genetic algorithm based approach with the dedicated dynamic context assessment mechanism is proved to be a feasible and effective approach to generating personal escape solutions. Finally, the proposed message construction mechanism made use of a template-based approach, which can be adapted to other I-CPS applications with a pre-definition of the templates and message components.

### **5.7.3. Future research opportunities**

From the many opportunities for doing research in this particular field, we would like to expose three near-future research opportunities that are of importance and can result in practical benefits:

- Combination of the AVSA with rigorous verification of logical properness and validation of internal correctness
- Investigation if the concept of AVSA can be used for purposes of external validation other than usefulness and applicability validation
- Development of a dedicated computational tool that would support the validation work of and computational methodology and software tool developers.

## 5.8. References

- [1] Lee, S. W., & Rine, D. C. (2004). Case study methodology designed research in software engineering methodology validation. In *SEKE*, pp.117-122.
- [2] Eze, T., Anthony, R. J., Walshaw, C., & Soper, A. (2011). The challenge of validation for autonomic and self-managing systems. In *Proceedings of the Seventh International Conference on Autonomic and Autonomous Systems*, pp. 128-133.
- [3] Ahmad, W., Qamar, U., & Hassan, S. (2015). Analyzing different validation and verification techniques for safety critical software systems. In *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 367-370.
- [4] Brings, J., Bohn, P., Bandyszak, T., Föcker, F., & Daun, M. (2016). Model-based prototype development to support early validation of cyber-physical system specifications. In *REFSQ Workshops*. pp. 1-15.
- [5] Gonzalez, A. J., & Barr, V. (2000). Validation and verification of intelligent systems-what are they and how are they different? *Journal of Experimental & Theoretical Artificial Intelligence*, 12(4), pp. 407-420.
- [6] Feth, P., Bauer, T., & Kuhn, T. (2015). Virtual validation of cyber physical systems. *Software-engineering and Management*. Bonn: Gesellschaft für Informatik e.V., pp. 201-206.
- [7] Guarro, S., Ozguner, U., Aldemir, T., Knudson, M., Kurt, A., Yau, M., & Kwon, S. (2016). Formal validation and verification framework for model-based and adaptive control systems. In *NASA Formal Methods Symposium*, pp. 227-233.
- [8] Bradley, T. H. (2008). Modeling, Design and energy management of fuel cell systems for aircraft, Doctoral dissertation, Georgia Institute of Technology.
- [9] Pedersen, K., Emblemavag, J., Bailey, R., Allen, J. K., & Mistree, F. (2000).

- Validating design methods and research: the validation square. In *ASME Design Engineering Technical Conferences*, pp. 1-12.
- [10] Pedersen, K. (1999). Designing platform families: an evolutionary approach to developing engineering systems. Doctoral dissertation, Georgia Institute of Technology.
- [11] Els Du Bois. (2013). A designerly methodology for software development. Doctoral dissertation, Delft University of Technology.
- [12] De Souza, A. M., Brennan, C. A., Yokoyama, R. S., Donato, E. A., Madeira, E. R., & Villas, L. A. (2017). Traffic management systems: A classification, review, challenges, and future perspectives. *International Journal of Distributed Sensor Networks*, 13(4), pp. 1-14.
- [13] Song, Y. (2014). The development of a homecare platform to support informal caregivers. In *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V01AT02A013-V01AT02A013.
- [14] Mautz, R. (2012). Indoor positioning technologies. Habilitation thesis, ETH Zurich.
- [15] Stelios, M. A., Nick, A. D., Effie, M. T., Dimitris, K. M., & Thomopoulos, S. C. (2008). An indoor localization platform for ambient assisted living using UWB. In *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, pp. 178-182.
- [16] Sui, K., Zhou, M., Liu, D., Ma, M., Pei, D., Zhao, Y., & Moscibroda, T. (2016). Characterizing and improving wifi latency in large-scale operational networks. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 347-360.
- [17] GPS Accuracy. [cited on 2019.04.11]; Available from: <https://www.gps.gov/systems/gps/performance/accuracy/>.
- [18] Liu, Z. (2001). The evaluation of the accuracy and measurement of latency of internet-based DGPS and RTK-GPS Positioning. In *ION*.
- [19] Grigorik, I. (2013). High Performance Browser Networking: What every web developer should know about networking and web performance. O'Reilly Media, Inc.
- [20] Buchheit, M., Allen, A., Poon, T. K., Modonutti, M., Gregson, W., & Di Salvo, V. (2014). Integrating different tracking systems in football: multiple camera semi-automatic system, local position measurement and GPS technologies. *Journal of sports sciences*, 32(20), pp. 1844-1857.
- [21] Leser, R., Baca, A., & Ogris, G. (2011). Local positioning systems in (game)

- sports. *Sensors*, 11(10), pp. 9778-9797.
- [22] Gandhi, H., Collins, M., Chuang, M., & Narasimhan, P. (2010). Real-time tracking of game assets in American football for automated camera selection and motion capture. *Procedia Engineering*, 2(2), pp. 2667-2673.
- [23] Bluetooth, S. I. G. (2016). Specification of the Bluetooth System-Covered Core Package version: 5.0. Bluetooth Special Interest Group.
- [24] Yaakop, M. B., Malik, I. A. A., bin Suboh, Z., Ramli, A. F., & Abu, M. A. (2017). Bluetooth 5.0 throughput comparison for internet of thing usability a survey. In *2017 International Conference on Engineering Technology and Technopreneurship (ICE2T)*, pp.1-6.
- [25] Collotta, M., Pau, G., Talty, T., & Tonguz, O. K. (2018). Bluetooth 5: A concrete step forward toward the IoT. *IEEE Communications Magazine*, 56(7), pp. 125-131.





# Chapter 6

## Overall conclusions and recommendations

### 6.1 Reflections on the work and the results

The main objective of the Ph.D. research was to develop a computational engine for DCIP, which enable I-CPSs to behave smartly in varying circumstances. DCIP was supposed to facilitate context-dependent reasoning in operational/computational time-sensitive applications. Driven by the objective, four research cycles were planned for addressing specific aspects of DCIP, including: (i) aggregation of the knowledge and requirements concerning approaches to DCIP in I-CPSs, (ii) conceptualization of computational mechanisms of the context processing engine, (iii) implementation of a testable prototype of the computational mechanisms, and (iv) applicability validation in different application contexts. Based on the results of the completed research cycles, this chapter provides a summary and self-evaluation of the work and the results. Specifically, it deals with (i) the knowledge aggregated and the requirements explored for DCIP, (ii) the conceptual framework that integrates the modelling, inferring, reasoning and messaging mechanisms for DCIP, (iii) the implemented prototype, which comprises a set of operationally interconnected algorithms, and (iv) the outcomes of the applicability validation of the prototype in various I-CPS application cases.

#### 6.1.1. Research cycle 1

The scientific objective of the first research cycle was (i) to get insight into the phenomenon of DCIP, (ii) to learn the state of the art and the cutting edge research results, and (iii) to organize the obtained DCIP knowledge in a robust knowledge framework with regard to the planned constructive research activities. Thus, the orientation of the work in this research cycle was defined by the guiding questions presented in Chapter 1. The main findings of the research are as follows:

***RQ 01: What is the current state of the art in research and development of DCIP?***

Our knowledge aggregation focused on five aspects of DCIP, (i) dynamic context representation, inferring of hidden knowledge, reasoning with DCI, context-dependent message generation, and software enablers. These five aspects were analyzed with the goal to identify the utility and limitations of existing approaches and methods of DCIP.

- (i) Our literature study has explored that dynamic context representation approaches can be both model-based and non-model-based. While model-based context representations (e.g. context ontologies, context modeling languages) have their strengths in rapidly capturing context elements and their relationships based on predefined models, non-model based approaches (e.g. data-driven context representations) offer flexible representations that lend themselves for inferring dynamic context knowledge based on run-time obtained data. Our study also showed that I-CPS typically require flexibility and generalizability for dynamic context representation. This finding implies that future research should focus on novel data-driven context representation that facilitates inferring and reasoning about DCI.
- (ii) Current approaches to building context awareness are facilitated by either inferring of hidden knowledge or predicting future situations. Several methods (e.g. Bayesian networks, rule-based inference) have been developed for inferring hidden knowledge, which can address a particular aspect of SAT relationships in various domains of applications from mobile computing to cognitions of humans. Despite these early successes, there were no generic approaches proposed that are able to infer SAT knowledge elements in a single framework. Our study has shown that predicting future situations also remained to be a research challenge. While existing methods are able to predict future events, handling higher level semantics (e.g. predicting situations and their relationships) remained to be an unsolved issue. Our findings implied that building context awareness needs a simultaneous treatment of hidden knowledge over SAT information, as well as prediction methods that can forecast the situational changes in the future.
- (iii) Computational reasoning with dynamic context was explored to obtain a comprehensive overview of methods capable to treat incompleteness and uncertainty of context information. We have reviewed probabilistic and non-probabilistic approaches in our study. While the former methods are effective in treating uncertainty and incompleteness of context information with the condition that the probabilistic values of context information are known or can be determined runtime. This condition, however, is limiting

their application in I-CPS as finding the probability of events and situations in a dynamic context at runtime is a major bottleneck. Non-probabilistic methods such as case-based reasoning and rule-based reasoning have also been widely used for reasoning with DCI. These methods are effective in reasoning about known events and situations, but they have their limitations in terms of treating emerging events and situations. For this reason, there is a need to explore the potentials of automatic rule generation based on inferred knowledge and real-time obtained data.

- (iv) Message generation mechanisms are essential means of I-CPSs for managing and controlling information exchange between humans and machines. Message generation has dominantly been implemented based on application domain dependent templates, which are typically predefined at the time of the design of the system. While we found some recent efforts for natural language generation using deep learning algorithms, they are typically made for correcting the semantics of messages in order to produce grammatically correct sentences or to correct incomplete text. We have investigated the applicability of template-based message generation for DCIP in our research. The main finding is that DCI is rarely considered as a basis for messaging operations (e.g. message construction and delivering) in informing systems. The main reason is that there is no effective tool available for assessing characteristics (e.g. relevance, quality, impact) of the personal context of users.
- (v) Many software enablers have been proposed in order to facilitate the development of computational mechanisms for DCIP. Three aspects have been surveyed, including (i) computational frameworks, (ii) software platforms, and (iii) reusable knowledge resources. We have found that most of the existing solutions focus on describing context either on data level or on information (event) level. Consequently, they lack the capability of handling the knowledge associated with a dynamic context that addresses not only the capture of situational changes but also the interpretation of the interrelationships and implications of the changes.

Therefore, our overall finding from the knowledge aggregation is that current research related to DCIP is still in its infancy. Limited research effort has been paid to investigate approaches to DCIP in I-CPS applications. Efficient solution for real-time management of dynamic context information is rarely reported in the literature. Circumventing these limitations is in the center of our research inquiry and development efforts.

***RQ 02: What are the general and technical requirements for a computational mechanism for multi-functional DCIP from the perspective of smart I-CPSs?***

Though the literature did not address specifically the issue of general requirements for DCIP related to smart I-CPSs, some indications on the general expectations could be found. In Chapter 2, we have explored application-related requirements from two sources, including (i) general application-related requirements (GAR) originating in multiple I-CPS applications, and (ii) specific application-related requirements (SAR), which are related to an indoor fire evacuation case. One of the important expectations is that context should be interpreted on multiple different abstraction levels by I-CPSs, in order to facilitate interpretation of the meaning and implication of context changes (self-awareness) and to adapt to them (self-adaptation). In addition, some semantically enriched representation of context is needed in order to enable a sufficient computational understanding and awareness building of time-varying context of stakeholders. Furthermore, dynamic context information management by I-CPSs entails the need to generate time-dependent action plans, which are to be performed by stakeholders. To achieve this, I-CPS applications should not only capture and represent the changes of various contextual elements (e.g. situations) but also monitor the relations between the contextual elements and relevant stakeholders.

With regard to the technical requirements for implementing DCIP in I-CPSs, the literature was even more limited. Actually, this issue seems to be an under-attended one in the current literature. Therefore, we had to make postulates concerning technical needs. For instance, it was postulated that dynamic context information management should be realized with the consideration of all concerned entities, stakeholders, processes, situations, and event characteristics. Informative and instructive message generation should be adapted to the stakeholders and their assumed/observable reactions to given personalized messages, e.g. obedience to the given messages. DCIP should be realized in a quasi-real-time manner, to guarantee the relevance of the generated message contents to the real-life context experienced by the target stakeholder. Furthermore, in hazard-intense I-CPS applications, DCIP should predict the changes of existing situations in dynamically changing contexts and should be able to forecast the occurrence of near-future situations in a proactive way. These extracted and postulated general and technical requirements have been further concretized and operationalized at the conceptualization of the knowledge framework of the reasoning engine.

***RQ 03: What are the recognized limitations of the existing DCIP solutions with regard to enhancing system smartness?***

As we discussed in Chapter 1, only some primary research progress has been achieved with regard to self-awareness and self-adaptation, which are fundamental features of the second generation (2G-) smart CPSs. These two characteristics were considered as the starting points to investigate the limitation of existing smart operations. Our observation was that the recently proposed

DCIP approaches have not been specifically developed according to the generic and specific requirements posed by I-CPSs. As a consequence, publications dedicated to the investigation of the appropriateness of the proposed solution to I-CPSs were not found at the time of completing our literature study. In the past three years, only a very limited number of publications appeared, which were deemed to be marginal to the DCIP and I-CPS interrelationship.

Our research pointed at the fact that there are three major criteria that should be fulfilled in order to achieve an efficiency of DCIP in I-CPS context. First, the representation should allow high-level dynamics with regard to the input data. Second, the context information processing should be able to derive additional information that may be only implicitly conveyed by the context data or maybe an only implication of them. Thirdly, the whole mechanism of DCIP should be flexible and adaptable to various application cases, since it is not possible to tailor a DCIP-M to a rather broad spectrum of applications. These system-related requirements were taken into consideration in the completed development and implementation of the proposed computational mechanisms.

### **6.1.2. Research cycle 2**

The second research cycle aimed at developing an underpinning theory and a computationally effective methodology for DCIP-M. The work in this research cycle was guided by the open issues found in the literature study. The main findings were as follows:

***RQ 04: What representation can be used for describing context data in a cohesive manner?***

Though many different representation methods can be considered to be relevant, actually the search space immediately narrows if we consider the three criteria mentioned above. The proposed ‘knowledge pyramid’ analogy-based context representation facilitates a logical/semantic progression from (i) states, through (ii) events and (iii) situations, to (iv) scenes. It provides a basis for inferring situations and their implications based on the interplays of situations. It also facilitates deriving semantically-rich, procedural knowledge about time-varying processes (e.g., interpretation of the causalities). The two-dimensional spatial feature representation (SFR) matrix provides a structured and flexibly up-datable representation of the descriptive information related to spatial entities, attributes, and relations. It also lends itself to an effective computational mechanism, since it primarily focuses on capturing the deviations at the time of re-computing the contextual states, events, situations, and scenes. The context information representation cube (CIR-cube) allows cohesive handling of spatial, attributive and temporal context data. It also facilitates connecting computational time to physical time, which is important to identify events related to contextual entities

and their relationships.

***RQ 05: What functionalities is the pursued computational mechanism supposed to provide?***

The identified main tasks of the DCIP-M are: (i) representing the dynamic context information, (ii) inferring semantic knowledge from dynamic context for building awareness, (iii) deriving personalized action-plan based on context reasoning, and (iv) constructing and sending personalized messages. Realization of a robust and high-performance solution required the distribution of the main functions over multiple component mechanisms (also called as modules). Four models were considered, including IRM, KIM, ADM, and MCM. The functionality of the computational algorithms was derived by a functional decomposition. On the other hand, the application of the ‘divide and conquer’ principle also raised the need for guaranteeing interoperability of the component mechanisms. It had to be achieved not only on module level, but also on the level of algorithms and data constructs.

***RQ 06: What way can the needed different functionalities be integrated to form a computation mechanism (context processing engine)?***

In the proposed context knowledge pyramid model, computational integration of data, information, and knowledge is complemented by logical abstractions to generate semantic knowledge on different abstraction levels, which enables a progression of building awareness of context from a syntactical level to a semantic level. Using the principles of integration and abstraction is sufficient for deriving semantic context information. This is the main ideation regarding how to integrate the decomposed functionalities together to form a computational engine. As discussed in the scheme-level specification of the DCIP-M, the elementary functions contained in the four modules are not linearly arranged but are integrated according to particular algorithmic purposes.

The IRM calculates the changed distances among entities whenever a new CIR-cube is requested to be generated. By doing so, the IRM does not need to re-calculate the unchanged spatial relations in a new computational cycle. This is important for real-time DCIP since the computation time needed for constructing an SFR-matrix (CIR-cube) can be decreased, especially in cases which involve many static entities. In addition, it can be seen from the articulated workflow of the system that semantic knowledge with regard to state, events, situations, and scenes of the dynamic context is generated based on continuous monitoring of a varying process. Over a given period of time, system awareness is built through multiple computational cycles (by using the descriptive context data aggregated at multiple time points). Accordingly, the amount of semantic knowledge gradually increases as the number of computational cycle increases.

***RQ 07: What way can the fulfillment of the explored requirements by the proposed functionality and architecture be tested and approved?***

The functionalities specified for the DCIP-mechanism are consistent with the technical requirements explored in Chapter 2. Most of the explored requirements are fulfilled, except for some practical considerations, such as security and self-initiative adaptations. Below, we discuss the most important requirements that have been fulfilled by each of the component computational mechanisms.

The first requirement specified for DCIP-M is to enable high-level dynamic representations with regard to the input data. To this end, a sophisticated data structure called CIR-cube was proposed. The CIR-cube captures not only spatial and attributive relationships among entities, but also the relationships between the changes in the spatial and attributive domain and in the temporal domain. The CIR-cube is a dynamic construct by nature, which supports the dynamic representation of data with regard to entities and their relations according to various computational requirements (e.g. sampling time, allowed computational time). Based on the CIR-cube, situations can be inferred out by using articulated reusable rules, which integrate the contents of the CIR-cube with predefined conditions.

Second, we also explored that the context information processing should be able to derive additional information that is implicitly conveyed by the descriptive data. The proposed inferring mechanism includes a quantitative approach to calculating the relevance of situations to individual entities considering the actually produced impact of the situations on involved entities and the SAT relations of the situations to the rest entities. The proposed ‘impact indicator’ and ‘relevance indicator’ enable real-time assessment of the situations that are relevant to people. This enables generating additional knowledge about the relations between people and context, such as danger level, a priority list of people and situations. Furthermore, the inferring mechanism considers judging the obedience of people by comparing the instruction given to people and the actual actions performed by people. All the above-mentioned knowledge is implicitly hidden in dynamic context, which proves that the requirement is fulfilled.

Thirdly, to achieve the requirement with regard to dynamic context information management, the proposed action-plan devising mechanism generates both long-term solutions and short-term actions for people to perform, towards certain objectives for dynamic context management. The generation of alternative personal action-plans considers the attributes of people, while the selection of the best action-plan depends on the impact of the predicted situations in the near future. Foreseeable situations that will happen in the near future were prioritized, based on which personal action plans are generated for people to enable a



proactive adaptation to the context changes.

To facilitate personalized informing, both informative and instructive messages are generated in the message construction module. On the one hand, dynamically changing situations of people are dealt with. The assessed personal danger level was considered as a basis for choosing a proper template for message construction. On the other hand, the SAT information with regard to the most relevant situation of people and the devised personal action plans were used to generate personalized messages. Thus, context-sensitive messages have been automatically generated.

### **6.1.3. Research cycle 3**

The main objective of research cycle 3 was to implement a prototype which enables a demonstration of the proposed concepts. The findings related to the research questions can be summarized as follows:

#### ***RQ 08: What algorithms and data structures are needed for the realization of the design functionalities?***

Every module includes a number of interoperating algorithms, which are specified in Chapter 3 and Chapter 4. Their functionality and computational implementation are also discussed. Due to their input and output data-implied functional connections, the majority of the algorithms are dedicated to the tasks that they have to handle in the modules. Only a few of them is reused in more than one module. The data processing tasks the particular algorithms are supposed to provide were derived through a systematic functional decomposition. The feasibility and computability were checked by multiple in-process analyses. The outcome of these analyses indicated the need for further decomposition. This had a positive effect on reducing the complexity of providing massive data constructs for certain algorithms. At the beginning of the work, the utility of certain approaches could not be forecasted. For instance, the appropriateness of the penalty matrix approach was questioned, but, in the end, it proved to be a simple but effective tool for representing the (computed) implications of the inferred situations for context-based reasoning purposes. The genetic algorithm-based personal path computation also proved to be a robust, effective and integrative solution.

#### ***RQ 09: What way can the functional modules and the whole of DCIP-M be brought to a software level realization?***

In this work, a prototype of the DCIP-M was established in Matlab®, which provides many basic algorithmic solutions. The implementation work involved many existing software enablers. Some of them are directly used in order to achieve the goal of required functions or sub-functions. For instance, the

algorithm ‘*inpolygon*’ provided by Matlab® was applied to judge if a person is in a sub-zone or not. This is the basis for calculating the identification of a sub-zone that a person is located in. On the other hand, some algorithms were adapted according to the particular purposes of usage. For instance, the function used for generating personal escape solutions was adopted from a genetic algorithm (GA). We applied the basic principle and workflow of the GA provided by Matlab®, but articulated the operations of GA, e.g. chromosome generation, fitness evaluation, to achieve a particular usage. Furthermore, in the implemented prototype, there are many innovative algorithms created by us, such as the algorithmic solution proposed for calculating the impact of situations and the relevance between situations and people.

***RQ 10: What way can the functionality and performance of the implemented DCIP-M prototype be validated?***

Due to the nature of the reference application, validation of the components and the whole of the computational mechanisms had to be based mainly on simulations. In the functionality tests, the simulation results provided evidence that the representation mechanism enables integral handling of SAT information, which generates SFR-matrices and CIR-cubes to follow the dynamics of a critical real-life application case. In addition, the chosen means for awareness building made it possible to infer additional knowledge implicitly conveyed by contextual data by monitoring states, events, situations and scenes. In the simulated cases, the proposed reasoning mechanism can efficiently evacuate all people (both users of informing systems and non-users) within the shortest time while preventing the people away from involving in dangerous situations. Finally, a human evaluation was considered to validate the functionality of the generated messages. According to the results of the evaluation by human subjects, the messages generated by the proposed computational mechanisms were useful, sufficient, informative and convincing.

The performance tests were also conducted based on various simulations. The number of entities processed by the prototype was considered as computational load. The performance of the implemented prototype was tested under both static computational loads and dynamic computational loads. It was found that calculating the distances among entities in an indoor environment is time-consuming. Accordingly, efficient algorithmic solutions are needed to save the computation consumption for processing relations among entities. To this end, the proposed DCIP-M only focuses on the changed information in every new computational cycle, which is a step towards quasi-real time processing of DCI. In addition, the testing results with regard to dynamic loads highlighted a requirement that the computational mechanism developed for representing and managing dynamic context information should be resilient and robust to deal with

computational consequences caused by dynamic changes of entity numbers and relations, such as overshoot in computational time.

It can be seen that although most of the tests were done in a simulated scenario, many valuable conclusions have been found. These theoretical findings can be considered as a basis for further investigations of DCIP in real-life application cases.

#### **6.1.4. Research cycle 4**

The main objective of the last research cycle was to test the concepts and the implemented prototype in different application contexts to clarify their applicability. The working research questions were:

***RQ 11: What relevant application cases, other than the reference application case, can be considered for the application of the DCIP-M?***

Throughout the development process of the project, a real-life application case, namely an indoor fire evacuation guiding (IFEG) application, was used as a computational scenario and a source of application data/knowledge. To test the applicability of the computational mechanism, three target applications, computationally resembling the IFEG application, have been considered. The main features considered for selecting the applications were (i) the number of entities handled at a time, (ii) required response time of the application to inform people, and (ii) the sudden change of entity number. According to the validation results obtained in Chapter 4, these three aspects have a great impact on the functionality and performance of the computational mechanisms.

Accordingly, the simulated real-life application cases were: (i) road traffic management, (ii) home caretaking, and (iii) coaching football training. Each of them represents a novel application opportunity of a smart I-CPS. These application cases strongly need dynamic context information processing, and their characteristics with regard to the above-mentioned three aspects are different. We believe each of the selected application cases could represent a class of applications with similar characteristics. The research intention was to generate an applicability profile of the DCIP-M in each target application (and to hint at necessary adaptations). The major issue was to make the applicability evaluation factual and the results logically unquestionable.

***RQ 12: Based on what methodological approach can the applicability of the DCIP-M be tested?***

Applicability testing of constructive computational methodologies is an important issue for both the academia and for the industry since many of them have been developed with a view to the need of a particular application. This, however,

negatively influences software and algorithms reusability and the efficiency/economy of development. In our work, the objective of applicability testing is to explore the reliable range of applications of a method and using this knowledge carefully at making practical decisions. To this end, we adopted the reasoning model of validation square approach (AVSA) and made it better fitting for applicability validation of constructive computational methodologies.

The validation square is divided into four quadrants based on two aspects of consideration. The left two quadrants are used for structural validation and the right two quadrants consider performance validation. In addition, the upper two quadrants deal with theoretical validation, while the lower two quadrants correspond to the domain-specific validation. Based on the scheme of the validation approach, the concrete research activities for each of the quadrants can be driven from four quadrants, which define different aspects of validation. The quadrants are shown as follows:

- (i) Theoretical structural validity deals with the internal consistency of a computational mechanism and checks the logical soundness of the constructs both individually and integrated.
- (ii) Empirical structural validity deals with the appropriateness of a computational mechanism to chosen example problem(s) with the intention of having correct results.
- (iii) Empirical performance validity concerns the ability of a computational mechanism to produce useful results for the chosen example problem(s), and finally.
- (iv) Theoretical performance validity concludes about the capability of the computational mechanism to produce useful results beyond the chosen example problem(s).

Actually, these validation aspects should be taken according to a particular semantic and procedural arrangement. Details with regard to the procedural dependency have been introduced in Chapter 5. The benefit of the AVSA is that it takes the computational performance of a mechanism into consideration. Thus, it uses different performance indicators and facilitates both quantitative and qualitative evaluation.

***RQ 13: What transferable knowledge can be obtained based on the applicability testing of the computational mechanism?***

In the tradition of scientific inquiry, internal validation and external validation goes hand-in-hand, since a body of knowledge, which suffers from internal biases due to lack of procedural rigor, and/or the one, which is not appropriate for or has serious limitations with regard to its intended purpose, can be considered as

scientific knowledge. As a framework of validating design methods, the original validation square approach (VSA) is based on a combination of qualitative and quantitative perspectives (like a cross-case study), as well as of theoretical and empirical viewpoints (like experimental research). This makes it relevant for validation of constructive computational mechanisms (CCM), in the case of which both structural appropriateness and performance appropriateness need to be assessed from both theoretical and empirical points of view. In addition to using meaningful and reliable information, the advantage of our adapted validation square approach (AVSA) is that it can be used to explore the range and/or the extent of applicability of various other constructive computational methodologies too. The reason is that AVSA establishes a tight coupling among the enablers embraced by a CCM and the aspects of theoretical and empirical validation.

Much transferable knowledge has been obtained concerning the proposed methodology. A systematic decomposition of the computational mechanism is transferable to other similar developments. The overall mechanism has been decomposed to component mechanisms, which all have their specific functionality, but they also interoperate with each other in the computational process. The component mechanisms consist of a set of standard or proprietary algorithms and the related data constructs. This approach allows not only a component-based development but also compositional implementation, in which the algorithm and the data constructs should be in synergetic relationships and should be designed accordingly. From the performance point view, the applicability of a CCM should not consider its computational efficiency of the employed algorithms and functions in time and space, but also involve practical factors that influence the computation, e.g. hardware devices, network conditions, the capacity of power supply etc. Using quantitatively calculated indicators, the applicability of the CCM in different applications can be compared.

Concerning our computational mechanisms, the proposed CIR-cube is an application-independent approach that is able to represent SAT data with regard to entity states and relations. It can be directly applied in applications that make decisions based on distances among entities and can be used with a minor adaptation to other applications that require the processing of other types of relations, e.g. orientation relations among people, e.g. in case of football training. In addition, the context knowledge pyramid is a general concept model that fits many smart system applications with the objective of dynamic context information management. The computational mechanisms proposed for calculating the impact of situations and the relevance between situations and entities can be adapted to other I-CPS applications. Accordingly, the approach to the formulation of personal context in time-varying processes is transferable. Furthermore, the adopted genetic algorithm based approach with the dedicated dynamic context assessment mechanism is proved to be a feasible and effective

approach to generating personal escape solutions. Finally, the proposed message construction mechanism made use of a template-based approach, which can be adapted to other I-CPS applications with a pre-definition of the templates and message components.

## 6.2. Propositions

### 6.2.1. Scientifically-based propositions

*Proposition 1: In the case of I-CPSs, the conventional passive and static processing of context information need to be transferred into an active and inferable dynamic context information processing approach.*

Our knowledge aggregation explored that the majority of the existing solutions for context information processing describes the context in a passive and static way either on data or information level. I-CPSs, on the other hand, typically operate in dynamically changing environments that are influenced by unknown and unpredictable events. Computational representation of dynamic context information hence needs to be able to capture the dynamic and emerging properties of context information. Semantic knowledge (e.g. the impact of an event on the entities defining dynamic context) implicitly hiding in the dynamic context information should be interpreted and represented explicitly. To this end, I-CPSs are required to be able to reason with the body of knowledge based on an active representation of context, where semantic rules are to be extracted from augmentable inferences.

*Proposition 2: Computational frameworks for dynamic context information processing should at least rely on an efficient and integrative management of spatial, attributive and temporal context data.*

Efficient representation of context data is a cornerstone for achieving real-time dynamic context information processing. In the studied (reference) application indoor fire evacuation guiding application, spatial, attributive and temporal data were taken into consideration. This was underpinned by our (unproven) null hypothesis which claimed that situations, such as people jams, can be inferred based on the temporal changes in spatial relations and attributes of entities. One form of managing situations was informing stakeholders about the need or necessity to change their locations or attributes. To achieve this, dynamic context semantics should be generated by means of integration and abstraction operations, which required integrative handling of the time-dependent spatial, attributive and temporal data.

*Proposition 3: If dynamic context information is captured in a sequence of spatial relation matrices, then only contextual deviations are*

*needed to be computed.*

Representing dynamic context information as a sequence of relations of spatial, temporal and attributive information allows observing the deviation of contexts. This approach not only provides an explicit representation of the context dynamics, but it also simplifies computation. Computing only the changes in the relationships enables quasi-real time computing makes inferring and reasoning more efficient and at the same time, it contrasts static and dynamic context information. Our research results showed that using this approach, a quasi-real-time computation can be achieved on an average laptop computer for highly complex contexts involving more than 85 entities, 85\*85 relationships, and 10 situations.

***Proposition 4:*** *Though dynamic context information processing mechanisms need to be generalized with a view to the application cases, their integration should be established on algorithm level.*

A general computational framework is a composition of multiple functional components that are implemented by a finite set of algorithms. The applicability testing of our dynamic information processing mechanism showed that although the same functional components can be applied to different application cases, their order of composition is dependent on the application. This finding implies that a general computational framework should offer functional and algorithmic solutions in the form of composable components, but it should not fix the order of information processing. It should facilitate the design of computational mechanisms for dynamic context information processing by enabling application-dependent composability of the functional and algorithmic components.

***Proposition 5:*** *Context-based reasoning needs to simultaneously capture both physical and computational events.*

Physical events refer to the contextual changes happening in the physical world, while computational events are used to explicitly represent the physical events by the dynamic context information processing mechanism. Actually, physical events are contextual changes supposed to happen in the physical space, but computed in the virtual space as computational events, and interpreted in the physical space to capture the meaning of them. Context-based reasoning should connect the computational events with the physical events to enable automatic monitoring of the physical process. This proposition casts light on the problem of event synchronization in the physical and digital spaces. Unsynchronized processing of events may lead to conflicting information and incorrect conclusions about the context that will negatively influence the results of the temporal aspect of reasoning. Temporal relationships of events and situations are essential for reasoning about the context and predicting future events and situations. Capturing

both physical events and computational events facilitates generating correct semantic knowledge for context-based reasoning.

**Proposition 6:** *The ability of the context information representation (CIR-) cube to model both computational events and physical events is a step towards real-time computing.*

Real-time operation of I-CPSs is one of the essential requirements for reliable servicing in many application domains including traffic management and sports coaching. In our research, a new data construct (namely CIR-cube) have been proposed for cohesive handling of spatial, attributive and temporal context data. SFR matrices are gradually appended into the CIR-cube as the computation progresses, resulting in an arrangement of entity and relation states according to the time dimension. The time-sequencing arrangement of SFR matrixes enables mapping the contextual changes of the physical world to the changes of the values in the corresponding cells of the subsequent SFR-matrices. Processing of dynamic context by taking computational events and physical events into consideration enables realizing event-triggered computations. In the conducted performance testing, it was found that the calculation of relations among entities in a structured environment is rather time-consuming. The experimental results have shown that monitoring the changed context in sequential computations can reduce the computational time, which is an effective step towards real-time computing.

**Proposition 7:** *It is possible to infer certain semantic context knowledge based on a syntactic data representation using data aggregation, integration, and abstraction.*

In this research work, a stratified model called context knowledge pyramid has been created. The context knowledge pyramid represents four types of context elements with different abstraction levels, namely states, events, situations, and scenes. High-level semantic context elements (e.g. situations and scenes) are generated by integrating and abstracting low-level syntactic context elements (e.g. states and events). Accordingly, integration and abstraction were considered as a knowledge enhancement mechanism. It has been recognized that relying on the principle of the ‘knowledge pyramid’ (i.e. by combined data aggregation, integration and abstraction procedure) additional semantic information (which is a kind of ‘hiding in the context model’) could be partially extracted to support semantic reasoning.

**Proposition 8:** *Management of dynamic context knowledge needs a replica of human abstraction in computation.*

Humans have the cognitive capacity for making abstractions for forming judgments, learning from experience, and making inferences. Traditional computational context modeling solutions, that are able to capture static and



dynamic context information, only applied context integration (or composition). We argue that dynamic context information processing mechanisms should be extended with abstraction for generating and managing the semantic context. Abstraction enables smart systems to give new semantic meaning to a compound of low-level context elements. However, the limited abstraction capability of the existing systems is the bottleneck for achieving a higher level of smartness, that covers the ability to learning from experiences, applying generalized knowledge in different contexts of application, and forming a judgment.

**Proposition 9:** *By enabling prediction of future situations, scenes create a basis for generating personalized action plans.*

In this research, we have introduced a context knowledge pyramid, in which the scenes have the highest level of abstraction and the highest level of semantic interpretation of context. Scenes represent spatial, causal and temporal relationships of situations. The changes of the relationships of situations serve as a knowledge base for (i) generating various possible (feasible) scenes, and (ii) inferring about the implications of the situational changes, which provides a basis for predicting the near-future situations. Rely on the predicted situations, personal action plans can be generated to proactively handle the dynamic context e.g. by preventing the happening of negative situations or promoting the happening of positive situations in the near future.

**Proposition 10:** *Template-based message generation facilitates articulation of the content of context-sensitive personalized messages.*

In I-CPS applications, the dynamic context of stakeholders may change rapidly. If the content of messages is required to be sensitive to the personal context, then a time-effective approach should be applied for message construction. The template-based solution provides a simple, flexible, and specific enough way of generating personalized messages by integrating predefined message components. Comparing with other approaches, e.g. model-based construction, the template-based approach is able to construct messages is a time-efficient way. In addition, a message template contains several elements. The message component (e.g. words, phases) for each of the elements is selected from alternatives. This enables the articulation of the content of messages, which is the most relevant to the given stakeholders.

**Proposition 11:** *The trust of stakeholders in communication can be increased by situation-dependently constructed messages.*

In hazard-intense situations, the trust of stakeholders in communication is an important issue since if stakeholders suspect on the given instruction, then the execution of it may not be realized and the system objective may not be achieved easily. The survey results of the conducted work have shown that the content of

personalized messages influences the trustiness of stakeholders in communication. If situation-dependently constructed messages are given to the stakeholders, then the sample people showed a stronger will in following the instructions.

**Proposition 12:** *A forerunning performance validation and an applicability validation are inseparable in the case of the quadrant-based validation approach.*

In this research work, the results of the forerunning performance validation are used to show the properness of the articulated algorithms in the applied application case (the fire evacuation application). Several performance indicators were defined, such as the number of entities considered at a given time. Actually, these indicators may be different in different but similar application cases. For instance, in the case of traffic management, the number of entities considered at a given time is larger than that considered for the fire evaluation. By formulating a set of conditions, the performance of the algorithms can be foreseen in varying application cases. This is the basis of how the performance validation of a computational mechanism in a single case can move towards multiple cases of performance validation, which is the aim of the proposed quadrant-based validation approach for applicability testing.

### **6.2.2. Socially flavored propositions**

**Proposition A:** *Since current artificial neural networks are not aware intuitively of the semantic meaning of knowledge, their capability to reason on a high abstraction level is limited.*

**Proposition B:** *Observing, thinking and acting by stakeholders in simulated scenarios will always differ from that in real-life emergency situations.*

**Proposition C:** *Managing and interpreting scenes by humans are complicated in real-life cases, but even more complicated in a virtual simulation environment.*

**Proposition D:** *There are many successful implementations of self-capabilities in the nature, but transferring them to smart cyber-physical systems needs sufficient understanding of their quintessence.*

### **6.2.3. Self-reflective propositions**

**Proposition  $\alpha$ :** *A more sufficient abstraction capability is needed for studying abstraction.*

**Proposition  $\beta$ :** *Smart systems will never be smarter than their designers.*

## 6.3. Possible future research work

### 6.3.1. Short-term follow-up research

Although five years' time has been paid in this research topic and certain scientific findings were obtained. This research work can still be improved and short-term follow-up research can be expected. The possible topics may include:

In the CIR-cube, the distances among all entities were calculated, but most of them were not used when inferring a certain type of situations (like people jams). Actually, the physical space could be dynamically divided into a number of local areas. Focusing on the entity relations within each of the divided area may eliminate the calculations on useless context relations, which may reduce the time complexity for construing a CIR-cube.

In the conducted work, we only considered and tested the implication of situations having negative impacts on the involve stakeholders in computation. Reasoning in dynamic contexts that are formed by both positive situations and negative situations could be an interesting follow-up research topic. The type and implication of situations on involved entities could be estimated and learned in real-time based on synergic processing of the actual changes of situations and the consequences caused by the changes.

As a pilot study, the findings of this research work reply on a simulated scenario, where the behaviors of entities (e.g. fire and people) were predefined. One possible short-term research topic can be a testing of the computational mechanism in real-life application cases. Descriptive context data is aggregated in real-time based a deployed wireless sensor network and customized messages are to be delivered to stakeholders through personal mobile devices. The reactions of stakeholders can be further investigated when they are reading the 'real' messages.

As a complement of the digital simulation, real-life application testing will be conducted. The planned internal validation will focus on the assessment of the underpinning concepts, theories, and method in order to optimize the operation of the dynamic spatial context reasoning mechanism and the SCA platform. Through the external validation, the effectiveness of the SCA platform will be tested in another application. As a possible application of the dynamic spatial context reasoning mechanism, a smart vehicle navigation system will be considered.

### 6.3.2. Long-term research opportunities

Many research efforts related to context-aware computing has been made during the past 30 years. Nowadays, people are still interested in this research domain and increasing research attention is given. The research about dynamic context

representation and management is still in its infancy and a proliferating future is expected in this research area. Long-term research opportunities are identified as the following:

In the presented work, descriptive context data was assumed to be accurately and perfectly acquired. However, it can hardly be achieved in real-life applications due to the limitations of existing sensing technologies, e.g. the sensing the indoor locations of stakeholders. Dynamic context representation makes use of the acquired context data, and hence the generated knowledge may deviate from the real-life situations, which causes uncertainty in representation. Probability-based sequential computations may create research opportunities for handling uncertainty in a dynamic context, which may facilitate context prediction for a varying scenario.

Another long-term research goal is to transfer the implemented prototype to a computational platform, which is able to provide run-time DCIP services (e.g. information modeling, inferring, reasoning and messaging) for multiple resembling I-CPS applications. This system-level problem-solving platform (e.g. cloud-based solution) should integrate application-independent mechanisms and facilitates the development of smart systems.

Understanding and realizing the features of smart systems will remain a promising research direction in the coming decades. An important step is striving after a ‘truly’ self-adaptation, which requires smart systems to automatically generate rules to make decisions depends on their own. Our work has shown the potentials for generating rules based on the semantic knowledge hidden from dynamic context, as a kind of meta-knowledge. Any generated rule should be optimized, validated and approved by the system itself before it can be applied for self-adaptation.



# List of figures

## Chapter 1

- Figure 1.1. The generic enablers of developing CPSs
- Figure 1.2. Typical articulation of the functionality of CPSs
- Figure 1.3. Synthesis of system technologies towards Industry 4.0
- Figure 1.4. The V-model of realization of CPSs as complex systems
- Figure 1.5. Subsequent generations of smart CPSs
- Figure 1.6. Trends of the evolution of data-driven self-control
- Figure 1.7. The ‘labyrinth of challenges’ that smart CPSs developers are facing
- Figure 1.8. Adaptation opportunities of smart CPSs
- Figure 1.9. Basic constituents of an informing CPS
- Figure 1.10. The focus of the research
- Figure 1.11. Methodological framing of the promotion research

## Chapter 2

- Figure 2.1. Reasoning model for the literature study
- Figure 2.2. The process flow of requirement exploration, synthesis, and processing
- Figure 2.3. The general operational flow for dynamic context information management by I-CPSs
- Figure 2.4. Demonstrative image of the indoor fire evacuation guiding application case that served as a reference
- Figure 2.5. Consistency checking of the aggregated requirements

## Chapter 3

- Figure 3.1. Progression of understanding context on different abstraction levels
- Figure 3.2. The dynamic context knowledge pyramid
- Figure 3.3. The mechanism for inferring semantic knowledge from dynamic context
- Figure 3.4. The interplays of situations
- Figure 3.5. The composition of context description on different abstraction levels

- Figure 3.6. Interpretation of the dynamic context of an entity
- Figure 3.7. The impact of situations on a concerned entity
- Figure 3.8. The spatial feature representation matrix (SFR-Matrix)
- Figure 3.9. The context information reference cube (CIR-cube)
- Figure 3.10. Conceptual design of the functionalities of the DCIP-mechanism
- Figure 3.11. Inferring situations and scenes from the CIR-cubes generated in multiple computational sessions
- Figure 3.12. Predicting the future of inferred situations based on the trend of attributive changes
- Figure 3.13. Predicting the future states of entities concerning the impact of situations
- Figure 3.14. Calculation of the spatial and temporal relations between situations and entities
- Figure 3.15. Representing the impact of situations based on the spatial relation model of a given physical space
- Figure 3.16. An action plan for a person to perform
- Figure 3.17. Construction of personalized messages based on a message template
- Figure 3.18. The generic framework of a multi-module computational platform for I-CPSs
- Figure 3.19. Integration of the functionalities of the computational mechanisms
- Figure 3.20. Obtaining semantic knowledge using multiple computational cycles

## **Chapter 4**

- Figure 4.1. Designation of zones, sub-zones, and SLMs from the physical space
- Figure 4.2. Connecting the SLMs with edges
- Figure 4.3. Representation of the concerned space layout
- Figure 4.4. The results of the computational simulations of an indoor fire evacuation scenario
- Figure 4.5. Data flow diagram of the information representation module
- Figure 4.6. Data flow diagram of the context knowledge inferring module
- Figure 4.7. Workflow diagram for inferring interplays among situations to generate scenes
- Figure 4.8. Data flow diagram of the personal action-plan deriving module
- Figure 4.9. A chromosome generated for devising personal escape solutions
- Figure 4.10. Generation of alternative positions to be reached by an informable person
- Figure 4.11. Data flow diagram of the message construction module
- Figure 4.12. An example of converting SAT information of a situation into various message components
- Figure 4.13. Sequence diagram of computing in the implemented modules based on the data representation schemes

- Figure 4.14. System-level data flow diagram (context diagram) of the implemented prototype
- Figure 4.15. Inputs and outputs of the implemented algorithms
- Figure 4.16. MAEs of the indoor position estimations with different process noise coefficients
- Figure 4.17. Estimation errors of indoor positions with different sampling rates
- Figure 4.18. Estimation of the positions of a moving person
- Figure 4.19. Calculation of the distance between two concerned people
- Figure 4.20. The percentage of deviation of distance calculation
- Figure 4.21. Simulation results of the inferred situations over a given period of time:
- Figure 4.22. Inferred transitivity and causality interplays among people jams
- Figure 4.23. The changes and the calculated impact indicators of a people jam over a period of time
- Figure 4.24. Prediction errors of people positions
- Figure 4.25. Determining personal context with different thresholds of relevance indicators
- Figure 4.26. Identified personal context of a person at different points in time with a threshold = 0.01
- Figure 4.27. Personalized optimal escape routes
- Figure 4.28. A typical scenario showing the results of generating group-level optimal escape solutions in dynamic context
- Figure 4.29. Convergence property of the implemented GA-based path-finding approach
- Figure 4.30. Comparison of the number of people in jams with and without providing information about the derived action plans
- Figure 4.31. The scenario simulated for the testing of the MCM
- Figure 4.32. The experimental setup for human evaluation
- Figure 4.33. The values of the various performance indicators under different percentages of informable people
- Figure 4.34. The computation time elapsed by the implemented functions
- Figure 4.35. The computation time taken by the time-consuming algorithms
- Figure 4.36. A comparison of the computation time spent on computing all distances and computing changed distances among 200 entities
- Figure 4.37. Computational performance of the prototype under dynamic loads
- Figure 4.38. Performance of the selected functions under different sizes of load changes
- Figure 4.39. The overshoot rate of the selected functions under different sizes of sudden changes in computational loads

## **Chapter 5**

- Figure 5.1. The reasoning model of the validation square approach



- Figure 5.2 The reasoning model adapted to the use of the validation square approach in applicability validation of constructive computational methodologies
- Figure 5.3. The process and indicators of applicability validation
- Figure 5.4. A road traffic management system
- Figure 5.5. A home caretaking assistant system
- Figure 5.6. A real-time football-play coaching system
- Figure 5.7. Calculated applicability indicators of the proposed mechanisms for RTMS
- Figure 5.8. Calculated applicability indicators of the proposed mechanisms for HCAS
- Figure 5.9. Calculated applicability indicators of the proposed mechanisms for RFCS
- Figure 5.10. Tight coupling among the methodological enablers and the aspects of validation

# List of tables

## Chapter 2

- Table 2.1. An overview of existing context information processing platforms  
Table 2.2. Typical I-CPS applications and the computed context information

## Chapter 3

- Table 3.1. Examples of rules used for inferring situations  
Table 3.2. Functions to be realized by the DCIP-M  
Table 3.3. Specified algorithms for the IRM for the IFEG application  
Table 3.4. Specified algorithms for the KIM for the IFEG application  
Table 3.5. Specified algorithms for the ADM for the IFEG application  
Table 3.6. Required algorithms for the MCM for the IFEG application  
Table 3.7. Classification of the algorithms

## Chapter 4

- Table 4.1. Predefined message templates for message construction  
Table 4.2. Converting SAT information describing a situation to message components  
Table 4.3. The input and output data of the implemented algorithms  
Table 4.4. Personalized messages generated based on the proposed MCM  
Table 4.5. Personalized messages generated based on static context information  
Table 4.6. The questionnaire designed for the human evaluation  
Table 4.7. Results of the human evaluation  
Table 4.8. The results of testing the implemented prototype with different initial positions of the entities

## Chapter 5

- Table 5.1. A comparison of the application-implied requirements of the target I-CPS applications

Table 5.2.	A comparison of the technical factors that influence the empirical performance of the target I-CPS applications
Table 5.3.	Assessment of the relevance of the functionalities for the target applications
Table 5.4.	Comparing the dependency of the functionalities among the concerned applications
Table 5.5.	Assessment of the suitability of the implemented algorithms
Table 5.6.	Assessment of the suitability of the data constructs
Table 5.7.	Assessment of the efficiency of the algorithms in the target applications
Table 5.8.	Assessment of the effect of several influencing factors on the DCI management of the concerned application cases
Table 5.9.	Principles applied to qualitatively assess the TPTs
Table 5.10.	Assessment results of the TPTs in case of RTMS
Table 5.11.	Assessment results of the TPTs in case of HCAS
Table 5.12.	Assessment results of the TPTs in case of RFCS

# Abbreviations

<b>ACT</b>	allowed computation time
<b>ADM</b>	action-plan devising module
<b>AET</b>	average escape time of all people
<b>ALP</b>	average length of escape paths of all people
<b>APN</b>	average of people number in jams
<b>ATR</b>	additional technical requirements
<b>AVSA</b>	an adapted validation square approach
<b>CCMs</b>	constructive computational methodologies
<b>CE</b>	context engineering
<b>CER</b>	context entity and relations
<b>CIR-cube</b>	context information reference-cube
<b>CPS</b>	cyber-physical system
<b>CPSoS</b>	cyber-physical system of system
<b>CT</b>	computation time
<b>DCI</b>	dynamic context information
<b>DCIP-M</b>	dynamic context information processing mechanism
<b>DL</b>	danger level
<b>DM</b>	distance matrix
<b>EI</b>	empirical efficiency indicator
<b>ESI</b>	empirical suitability indicator
<b>GAR</b>	general application-related requirement
<b>GSR</b>	general system-related requirement
<b>HCAS</b>	home caretaking assistant system
<b>I-CPS</b>	informing cyber-physical system
<b>IFECS</b>	indoor fire evacuation guiding system
<b>II</b>	impact indicator
<b>IRM</b>	information representation module
<b>KIM</b>	knowledge inferring module

<b>LW</b>	local world
<b>MCM</b>	message construction module
<b>MREP</b>	modeling, representation, extraction, and processing
<b>nG-CPS</b>	n-th generation cyber-physical system
<b>NLG</b>	natural language generation
<b>NPJ</b>	number of people involved in jams
<b>OAI</b>	overall applicability indicator
<b>OSI</b>	overall sensitivity indicator
<b>QoM</b>	quality of messages
<b>RD</b>	research design
<b>RFCS</b>	real-time football play coaching system
<b>RI</b>	relevance indicator
<b>RiDC</b>	research in design context
<b>RTMS</b>	road traffic management system
<b>SAR</b>	specific application-related requirements
<b>SAT</b>	spatial, attributive and temporal
<b>SCI</b>	static context information
<b>S-CPS</b>	smart cyber-physical system
<b>SD</b>	standard deviation
<b>SFR-matrix</b>	spatial feature representation matrix
<b>SLM</b>	spatial landmark
<b>SRI</b>	sum of relevance indicators
<b>SSR</b>	specific system-related requirement
<b>SST</b>	syntax to semantics transition
<b>TET</b>	total escape time
<b>TPT</b>	theoretical performance target
<b>TRI</b>	theoretical relevance indicator
<b>V-model</b>	validation and verification strategy model
<b>VSA</b>	validation square approach
<b>WLAN</b>	wireless local area network
<b>WRQ</b>	working research question
<b>WSN</b>	wireless sensor network

# Summary

## **Utilizing dynamic context semantics in smart behavior of informing cyber-physical systems**

### **Background of the research**

Context is interpreted as a body of information dynamically created by a pattern of entities and relationships over a history of situations. Computational handling of dynamically changing contexts and the consideration of rapidly changing situations in awareness building, situated reasoning, and proactive adaptation of smart cyber-physical systems has been recognized as an important research phenomenon. The main reason is that there are many real-life processes whose smart control and self-\* behavior require quasi-real time processing of context information. Though processing time-varied context information has been addressed in the literature, domain-independent solutions for reasoning about time-varying complex and critical activity scenarios are scarce. Thus, explicit generation and utilization of dynamic context semantics in smart behavior of informing cyber-physical systems (I-CPSs) is a frontier endeavor.

The intention of the thesis was to provide an effective representation and processing methodology of predefined kinds of dynamic context data, to enrich context information with derived semantics, to address real-life applications that need reasoning and decision making for development of some sort of action plans, and eventually, to realize a dynamic context information processing mechanism (DCIP-M) as a kernel component of a reasoning platform for I-CPSs. Our overall objective was to make using dynamically changing context information in decision-making by application-specific smart CPSs possible. Towards this end, we needed purposeful representation and handling techniques for dynamically changing context information that allowed reducing the time of information input and computational processing, in harmony with critical short-time happenings.

### **Research problem and question**

The research challenge has been conceptualized as a design research problem with a

flavor of information science and engineering research. Starting out of the above formulated objective, the major tasks were formulated as follows: (i) obtaining insight in the essence of dynamic context information and exploring the state of the art in dynamic context information management, in particular in I-CPSs, (ii) development of a conceptual framework, which captures entities, relationships, attributes, and changes in space and time, (iii) transferring the conceptual framework to a multi-functional computational mechanism that can be used as a kernel of a reasoning platform for specific I-CPSs, (iv) using evacuation of a building in fire as a source of empirical knowledge and as a practical case study throughout the completed research, and (v) conceptualization, design, implementation and testing the performance and applicability of the proposed computational mechanism.

Our work was challenged by the idea of generalizing the proposed multi-functional computational mechanism (referred to as a DCIP-M) to be able to handle dynamically changing spatial and attributive context information in other target application cases, such as protection in disaster, crowd management, and medical rehabilitation. Therefore, the guiding research question has been formulated so as: In what way can semantic information obtained from dynamic context information and how can it be utilized in certain elements of smart behavior (such as situation awareness building, situated reasoning, decision making and action planning) of I-CPSs?

## **Research methodology and content**

From a methodological point of view, both a detailed research model and a research design have been elaborated to bring the scope of research and the procedure of research into harmony. The general nature of the research was what is typical in information system research and engineering. The methodological framing of the research project was done accordingly, and it helped transfer the theoretical framework into a testable implementation. As far as the overall research process is concerned, it has been divided into a stream of four research cycles.

### **Research cycle 1**

The overall objective of the first research cycle was knowledge aggregation concerning DCIP. This research cycle was methodologically framed as a research in design context process, which has been decomposed into an explorative part and a confirmative part. To aggregate the state-of-the-art knowledge related to DCIP, the work surveyed the current approaches (i) to representing dynamic context, (ii) to building situation awareness in a dynamic context, (iii) to reasoning with dynamic context, (iv) to context-dependent messaging. It also reviewed available software enablers for DCIP. As a complement of the theoretical knowledge platform creation, requirements were specified from four sources, including (i) general system-related requirements, (ii) specific system-related requirements, (iii) general application-related requirements, and (iv) specific application-related requirements. The first research cycle was concluded by (i) structuring the requirements, (ii) validation of the requirements, (iii) checking the consistency and coherence of the requirements.

## **Research cycle 2**

Methodologically framed as design inclusive research, the research cycle two was dedicated to ideation and conceptual modeling of the parts and the whole of the DCIP-M. The theoretical fundamentals of ideation and conceptualization rested on four pillars: (i) a semantically enriched model for managing context information, (ii) an information structure model for representing spatial, attributive and temporal context data, (iii) a set of principles for inferring semantic knowledge from dynamic context, and (iv) an approach for deriving action plans for proactive management of and reasoning with dynamic semantic context. To understand context, four levels of information engineering constructs have been introduced: states, events, situations, and scenes, which formed a basis for the design of the functions and workflow of the DCIP-M. Four computational modules were included in the DCIP-M: (i) information representation module, (ii) knowledge inferring module, (iii) action-plan deriving module, and (iv) message construction module. To validate the feasibility of the proposed components of the DCIP-M, a plan for a software prototype implementation was developed.

## **Research cycle 3**

The third research cycle was dedicated to the functional and architectural implementation of the modules of the DCIP-M and the various algorithms in executable forms. This research cycle was also framed methodologically as a design inclusive research cycle. The main activities of implementation were: (i) algorithm-level implementation of all modules of the prototype, (ii) functional and structural integration of the modules and (iii) specification the input and output variables of the prototype. To confirm the prototype, domain-dependent knowledge for indoor fire evacuation guiding was carried out, and the usability of the implemented modules was tested. A simulated real-life scenario was implemented as a testbed, in which the ground floor of Faculty IDE of TU Delft was considered as a reference case. In the simulation, personal action plans and messages for informable stakeholders were generated for managing emergency situations using the implemented prototype. The confirmative part of the research cycle focused on the functional and performance testing of all critical algorithms and the investigation of their proper interoperation and results.

## **Research cycle 4**

In the fourth research cycle, applicability validation of the DCIP-M was considered. Methodologically, this research cycle was completed as a practice driven research. The major issue was to demonstrate and evaluate the applicability of the developed DCIP-M in application cases different from the reference case. Four aspects of applicability, namely (i) theoretical structural relevance, (ii) empirical structural suitability, (iii) empirical performance efficiency, and (iv) theoretical performance sufficiency were considered at the application of the adapted validation square approach. In the applicability investigation, three differing real-life application cases



have been considered: (i) road traffic management, (ii) home care-taking assistant, and (iii) coaching football training.

## **Main findings and conclusions**

Our major findings can be summarized as follows: The traditional passive and static representation and processing of context information need to be transferred into an active and inferable DCIP approach in the case of I-CPSs. I-CPSs typically operate in dynamically changing environments that are influenced by unknown and unpredictable events. Our research confirmed that these can be achieved in the case of spatial, attributive and temporal context data in an efficient and integrative manner. The ability of spatial feature representation matrices with regards to capturing both computational events and physical events is a step towards implementation of real-time context computing.

In addition, a computational framework relevant for DCIP should rely on efficient and integrative management of spatial, attributive and temporal context data. This is because that efficient representation of context data is a cornerstone for achieving real-time DCIP. Applying data integration and abstraction makes possible to generate these synthetic context entities and derive semantic information. We showed that, based on this, a stratified computational model can be developed, which allows inferring semantic context knowledge. The model facilitates to infer changes of context data, information and knowledge level, and this way, inferences about happening or expectable states, events, and situations can be made, including their implications. By enabling prediction of future situations, scenes create a basis for generating personalized action plans.

It was also found that if dynamic context information is captured in a sequence of relation matrices, then only contextual deviations are needed to be computed. In our work, dynamic context information was represented as a sequence of relations of spatial, temporal and attributive information (i.e. by the CIR-cube), which allows observing the deviation of contexts. This approach not only provides an explicit representation of the context dynamics, but it also simplifies computation. Computing only the changes of the relationships enables quasi-real time computing, makes inferring and reasoning more efficient and at the same time it contrasts static and dynamic context information.

Last but not the least, we concluded that generalization of DCIP mechanisms can be made on algorithm level, but their integration depends on the application cases. The applicability testing of our DCIP-M showed that although same functional components can be applied to different application cases, their order of composition is dependent on the application. This finding implies that a general computational framework should offer functional and algorithmic solutions in the form of composable components, but it should not fix the order of information processing. It should facilitate design of computational mechanisms for DCIP by enabling application-dependent composability of the functional and algorithmic components.

# Samenvatting

## Gebruikmaking van dynamische contextsemantiek voor slim gedrag van informerende cyberfysische systemen

### Achtergrond van het onderzoek

Context kan worden geïnterpreteerd als een dynamisch geheel van informatie dat wordt bepaald door een patroon van entiteiten en relaties gedurende een historie van situaties. Computationeel omgaan met dynamisch veranderende contexten en inachtneming van snel veranderende situaties in het opbouwen van bewustheid, gesitueerd redeneren en proactieve aanpassing van cyberfysische systemen wordt gezien als een belangrijk onderzoeksonderwerp. De belangrijkste reden is dat voor het realiseren van slimme aansturing en zelf\*-gedrag in veel processen die zich in de werkelijkheid afspelen, quasi-realtime verwerking van contextinformatie benodigd is. Hoewel het verwerken van over de tijd veranderlijke contextinformatie in de literatuur aan bod komt, zijn domeinonafhankelijke oplossingen voor het redeneren over veranderlijke complexe activiteitsscenario's zeldzaam. Daarom is het streven naar slim gedrag van informerende cyberfysische systemen (I-CPS'en) waarin dynamische contextsemantiek expliciet gegenereerd en gebruikt wordt, grensverleggend.

De bedoeling van dit onderzoek was om een effectieve weergave- en verwerkingsmethodologie te bieden voor voorgedefinieerde soorten contextgegevens, om zo de contextinformatie te verrijken met afgeleide semantiek voor praktische toepassingen waarin een soort besluitvormings- of redeneerplannen moeten worden gemaakt en, uiteindelijk, om een dynamisch contextinformatieverwerkingsmechanisme (DCIP-M) te realiseren als kerncomponent voor een redeneerplatform voor I-CPS'en. Ons algemene doel was het gebruik mogelijk te maken van dynamisch veranderende contextinformatie bij besluitvorming door toepassings specifieke slimme cyberfysische systemen. Daarvoor nodig waren zinvolle weergave- en hanteringstechnieken voor dynamisch veranderende contextinformatie die, in harmonie met kritieke kortstondige gebeurtenissen, de tijd nodig voor informatie-

invoer en computationele verwerking konden bekorten.

## Onderzoeksprobleem en onderzoeksvraag

De onderzoeksuitdaging is geconceptualiseerd als een ontwerponderzoeksprobleem met een vleugje informatica en technisch onderzoek. Uitgaande van de hierboven geformuleerde doelstelling, werden de belangrijkste taken als volgt geformuleerd: (i) inzicht verkrijgen in de essentie van dynamische contextinformatie en de stand van zaken verkennen in dynamisch contextinformatiebeheer, in het bijzonder in I-CPS'en, (ii) ontwikkeling van een conceptueel kader, dat entiteiten, relaties, attributen en veranderingen in ruimte en tijd vastlegt, (iii) overdracht van het conceptuele raamwerk naar een multifunctioneel berekeningsmechanisme dat kan worden gebruikt als kernel van een redeneerplatform voor specifieke I-CPS'en, (iv) evacuatie van een in brand staand gebouw gebruiken als een bron van empirische kennis en als een praktische case study gedurende het hele onderzoek, en (v) conceptvorming, ontwerp, implementatie en testen van de prestaties en toepasbaarheid van het voorgestelde rekenmechanisme.

Ons werk werd uitgedaagd door het idee om het voorgestelde multifunctionele computermechanisme, genaamd DCIP-M, te generaliseren om dynamisch veranderende ruimtelijke en attributieve contextinformatie te kunnen verwerken in andere doeltoepassingsgevallen, zoals bescherming bij rampen, crowd management en medische revalidatie. Daarom is de richtinggevende onderzoeksvraag geformuleerd als: 'Op welke manier kan semantische informatie worden verkregen uit dynamische contextinformatie en hoe kan deze worden gebruikt in bepaalde elementen van slim gedrag (zoals situatiebewustzijnsontwikkeling, gesitueerd redeneren, besluitvorming en actieplanning) van I-CPS'en?

## Methodologie en inhoud van het onderzoek

Vanuit methodologisch oogpunt zijn zowel een gedetailleerd onderzoeksmodel als een onderzoeksontwerp uitgewerkt om de reikwijdte van onderzoek en de procedure van onderzoek op elkaar af te stemmen. Het algemene karakter van het onderzoek is zoals gangbaar in onderzoek naar, en engineering van informatiesystemen. Het onderzoeksproject was dienovereenkomstig methodologisch ingekaderd, zodat het theoretische kader kon worden omgezet in een testbare implementatie. Wat het totale onderzoeksproces betreft, was het opgesplitst in vier opeenvolgende onderzoekscycli.

### Onderzoekscyclus 1

Het algemene doel van de eerste onderzoekscyclus was kennisaggregatie met betrekking tot DCIP. Deze onderzoekscyclus werd methodologisch opgezet volgens het proces van onderzoek-in-ontwerpcontext, waarin een exploratief deel en een bevestigend deel te onderscheiden zijn. Om de state-of-the-art-kennis met betrekking tot DCIP in kaart te brengen, zijn bestaande benaderingen bestudeerd voor (i) het weergeven van dynamische context, (ii) het opbouwen van situatiebewustzijn in een

dynamische context, (iii) het redeneren met dynamische context, (iv) het genereren van contextafhankelijke berichten. Daarbij zijn ook de beschikbare software-enablers voor DCIP bekeken. Als aanvulling op het creëren van theoretische kennisplatforms, werden op vier deelgebieden eisen vastgelegd: (i) algemene systeemgerelateerde eisen, (ii) specifieke systeemgerelateerde eisen, (iii) algemene applicatie-gerelateerde eisen, en (iv) specifieke toepassingsgerelateerde eisen. De eerste onderzoekscyclus werd afgesloten door de eisen (i) te structureren, (ii) te valideren en (iii) te controleren op onderlinge consistentie en coherentie.

## **Onderzoekscyclus 2**

Deze onderzoekscyclus, methodologisch opgezet als ontwerpinclusief onderzoek, was gewijd aan ideeënvinding en conceptueel modelleren van de DCIP-M, zowel het geheel als de onderdelen ervan. De theoretische grondbeginselen van ideeënvinding en conceptualisatie berustten op vier pijlers: (i) een semantisch verrijkt model voor het beheer van contextinformatie, (ii) een informatiestructuurmodel voor het weergeven van ruimtelijke, attributieve en tijdelijke contextgegevens, (iii) een reeks principes voor het afleiden van semantische kennis uit dynamische context, en (iv) een aanpak voor het afleiden van actieplannen voor proactief beheer van, en redeneren met, dynamische semantische context. Om de context te begrijpen, zijn engineeringconstructies op vier niveaus geïntroduceerd: toestanden, gebeurtenissen, situaties en scènes, die een basis vormden voor het ontwerp van de functies en workflow van de DCIP-M. De DCIP-M kent vier rekenmodules: (i) informatieweergavemodule, (ii) kennisafleidende module, (iii) actieplanafleidingsmodule, en (iv) berichtconstructiemodule. Om de haalbaarheid van de voor de DCIP-M voorgestelde componenten te valideren is een plan voor een software-prototype-implementatie ontwikkeld.

## **Onderzoekscyclus 3**

De derde onderzoekscyclus was gewijd aan de functionele en architecturale implementatie van de modules van de DCIP-M, en van de verschillende algoritmen in executeerbare vorm. Ook deze onderzoekscyclus is methodologisch opgezet als een ontwerpinclusief. De belangrijkste implementatieactiviteiten waren: (i) implementatie van alle prototypemodules op algoritmeniveau, (ii) functionele en structurele integratie van de modules en (iii) specificatie van de input- en outputvariabelen van het prototype. Om vast te stellen of het prototype functioneert zoals beoogd werd domeinafhankelijke kennis van indoor- evacuatiebegeleiding ingezet, en werd de bruikbaarheid van de geïmplementeerde modules getest. Als testbed werd een gesimuleerd levensecht scenario opgezet gebaseerd op de begane grond van faculteit IO van de TU Delft. Voor informeerbare betrokkenen werden in de simulatie persoonlijke actieplannen en berichten voor noodsituatiebeheersing gegenereerd met behulp van het geïmplementeerde prototype. Het confirmerende deel van de onderzoekscyclus richtte zich op het testen van functionaliteit en prestaties van alle kritische algoritmen, alsmede het beproeven van de correctheid van hun interoperabiliteit en uitkomsten.

## Onderzoekscyclus 4

In de vierde onderzoekscyclus werd de validatie van de toepasbaarheid van de DCIP-M ter hand genomen. Methodologisch gezien is deze onderzoekscyclus uitgevoerd als praktijkgericht onderzoek. Het belangrijkste element was beproeving en evaluatie van de toepasbaarheid van de ontwikkelde DCIP-M voor andere toepassingen dan de referentiecasse. Vier aspecten van toepasbaarheid, namelijk (i) theoretische structurele relevantie, (ii) empirische structurele geschiktheid, (iii) empirische prestatie-efficiëntie, en (iv) theoretische prestatie-efficiëntie werden meegenomen onder toepassing van een aangepaste vorm van de validatie-kwadrantenbenadering. Het toepasbaarheidsonderzoek omvatte drie verschillende levensechte praktijkcases: (i) verkeersmanagement, (ii) thuiszorgassistentie en (iii) coaching bij voetbaltraining.

## Belangrijkste uitkomsten en conclusies

Onze belangrijkste bevindingen kunnen als volgt worden samengevat: in het geval van I-CPS's moet de traditionele passieve en statische weergave en verwerking van contextinformatie worden omgezet in een actieve en afleidbare DCIP-aanpak. I-CPS's werken doorgaans in dynamisch veranderende omgevingen die worden beïnvloed door onbekende en onvoorspelbare gebeurtenissen. Ons onderzoek bevestigde dat deze kunnen worden bereikt in het geval van ruimtelijke, attributieve en temporele contextgegevens op een efficiënte en integratieve manier. Het vermogen van ruimtelijke functie-representatiematrices met betrekking tot het vastleggen van zowel computationele gebeurtenissen als fysieke gebeurtenissen is een stap in de richting van implementatie van real-time context computing.

Bovendien moet een voor DCIP relevant berekeningskader gebaseerd zijn op efficiënt en integraal beheer van ruimtelijke, attributieve en temporele contextgegevens. Dit komt omdat die efficiënte weergave van contextgegevens een randvoorwaarde vormt voor het realiseren van realtime DCIP. Door gegevensintegratie en abstractie toe te passen, kunnen deze synthetische contextentiteiten worden gegenereerd en semantische informatie worden afgeleid. We hebben laten zien dat op basis hiervan een gestratificeerd computermodel kan worden ontwikkeld, waarmee semantische contextkennis kan worden afgeleid. Het model maakt het mogelijk om veranderingen in contextgegevens, informatie en kennisniveau af te leiden, en op deze manier kunnen conclusies worden getrokken over actuele of verwachte toestanden, gebeurtenissen en situaties, inclusief de implicaties ervan. Door voorspelling van toekomstige situaties mogelijk te maken, vormen scènes een basis voor het genereren van gepersonaliseerde actieplannen.

Er werd ook gevonden dat als dynamische contextinformatie wordt vastgelegd in een reeks relatiematrices, alleen contextuele afwijkingen hoeven te worden berekend. In ons werk werd dynamische contextinformatie weergegeven als een reeks relaties in ruimtelijke, temporele en attributieve informatie (d.w.z. door de z.g. contextinformatie-referentiekubus), waarmee veranderingen in contexten kunnen worden waargenomen. Deze benadering biedt niet alleen een expliciete weergave van de

contextdynamiek, maar vereenvoudigt ook de berekening. Door alleen de veranderingen in relaties mee te nemen wordt quasi-realttime berekening mogelijk, vindt afleiden en redeneren efficiënter plaats en worden tegelijkertijd de verschillen tussen statische en dynamische contextinformatie uitgelicht.

Last but not least zijn we tot de conclusie gekomen dat generalisatie van DCIP-mechanismen op algoritmeniveau kan plaatsvinden, maar dat hun integratie afhangt van de toepassing. Uit de toepasbaarheidstests van onze DCIP-M bleek dat hoewel dezelfde functionele componenten op verschillende toepassingsgevallen kunnen worden toegepast, hun volgorde van samenstelling afhankelijk is van de toepassing. Deze bevinding impliceert dat een algemeen rekenraamwerk functionele en algoritmische oplossingen moet bieden in de vorm van samenstelbare componenten, maar dat dit de volgorde van informatieverwerking niet mag bepalen. Het ontwerpen van rekenmechanismen voor DCIP zou worden vergemakkelijkt als toepassingsafhankelijke configureerbaarheid van de functionele en algoritmische componenten mogelijk zou zijn.



# Acknowledgments

Completion of a doctoral thesis is really a hard job. It would not be possible done by my own effort. I would like to thank all the people who directly and indirectly helped me to finalize the thesis. In particular, I wish to express my gratitude to Professor Imre Horváth, for giving me the opportunity to work on this exciting project. Throughout the past few years, Professor Horváth showed me, by his example, what a good scientist should be. I have received many support, advice, and mentorship from him to tide over the scientific difficulties. I will always remember our debate as a deep, profitable and pleasurable experience.

I am especially indebted to Dr. Zoltan Rusák, who gave me intensive, professional and daily guidance of the research. It has been my great privilege to benefit from his incredibly large range of expertise, point of views and ways of scientific thinking. Many thanks to Dr. Wilfred van der Vegte for nice discussions and the deal about Dutch life in general. I also appreciate his help in translating the summary of the thesis into Dutch. Many thanks to the members of my dissertation committee, who have offered professional and considerable guidance with regard to the improvement of the book.

My years at TU Delft would also not have been enjoyable without the reliable friends and talented colleagues that I had here. Never had I imagined that I would be surrounded by such companionship from them. Thanks to Sirasak and Tima for sharing your wonderful moments and pleasure to me, and encouraging me to overcome frustrations. Thanks to Jolanda, Sara, and Mariska for the support of my research and living in Delft.

Last but not least, I would like to thank my parents. Their love is with me in whatever situation I met. Most importantly, I wish to thank my beloved wife, Miao Wang for her unending care, support and trust. She has defended me in seemingly indefensible times, and has looked after my happiness with fondest attention. I am forever indebted to my families.





# Biography

Yongzhe Li was born in Daqing, China on August 31st, 1987. In 2010, he received his bachelor degree in Electronics and Information Engineering from the Harbin Institute of Technology (HIT), China. He finished master education from the School of Materials Science and Engineering of the HIT in 2012. In the same year, he got a certificate of International Welding Engineer (IWE) from the International Institute of Welding



(IIW). From 2012-2019, he worked on a Ph.D. research project at the State Key Laboratory of Advanced Welding and Joining of HIT, and received his first doctoral title in the area of wire and arc additive manufacturing. From 2014 onwards, he was doing his second Ph.D. research project at the Faculty of Industrial Design Engineering, Delft University of Technology (TUD), under the financial support from the Chinese Scholarship Council (CSC). His current research interest includes smart cyber-physical systems; context information management; process planning of additive manufacturing; robotics and intelligent welding.