# Conceptualization of a real-time information processing platform for context-aware informing cyber-physical systems

Li, Yongzhe; Horvath, Imre; Rusak, Zoltan; van der Vegte, Wilfred; Zhang, Guangjun

**Publication date**
2016

**Document Version**
Final published version

**Published in**
Proceedings of the TMCE 2016

**Proceedings of TMCE 2016,** May 9-13, 2016, Aix-en-Provence, France, edited by I. Horváth, J.-P. Pernot, Z. Rusák.


# CONCEPTUALIZATION OF A REAL-TIME INFORMATION PROCESSING PLATFORM FOR CONTEXT-AWARE INFORMING CYBER-PHYSICAL SYSTEMS

**Yongzhe Li**
Faculty of Industrial Design Engineering
Delft University of Technology, and
State Key Laboratory of Advanced Welding and Joining
Harbin Institute of Technology
y.li-8@tudelft.nl

**Imre Horváth**
**Zoltán Rusák**
**Wilfred van der Vegte**
Faculty of Industrial Design Engineering
Delft University of Technology
{ i.horvath, w.f.vandervegte, z.rusak }@tudelft.nl

**Guangjun Zhang**
State Key Laboratory of Advanced Welding and Joining
Harbin Institute of Technology
zhanggj@hit.edu.cn

## ABSTRACT

*Informing cyber-physical systems (I-CPSs) are designed to accomplish sensing, reasoning and informing activities in dynamic context. In order to simplify and accelerate the design and implementation process of multiple context-aware I-CPSs, we are developing an information sensing, computing and actuating (SCA) platform that can be used as a central module of these systems. This paper presents the concept of a SCA platform. The functionality of the platform includes development of context-dependent strategies to adapt the sensing, reasoning and informing behaviors of the platform to various dynamic contexts. There are four constituents of the platform: (1) a generic kernel, (2) built-in elements, (3) add-on components, and (4) system interfaces. The paper also discusses both the internal and external integration mechanism of the SCA platform, which can be customized according to the needs of specific I-CPS applications by extending the generic kernel with various functional built-in elements and add-on components. The feasibility and applicability of the platform have been tested through a case study: an indoor fire evacuation guiding system. The proposed platform provides a useful package of functionalities, alleviates the burden of developers, and speeds up the development of applications specific context-aware I-CPS.*

## KEYWORDS

Informing cyber-physical systems, SCA platform, context aware computing, applications development

## 1. INTRODUCTION

Cyber-physical systems (CPSs) are physical artifacts and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core [1]. According to our conceptualization, CPSs are smart actor systems, which are able to reason in various dynamic contexts and to develop context-dependent strategies to adapt themselves to the need of specific application scenarios. The three major functionalities of CPSs are sensing, reasoning, and actuation. One specific form of actuation is informing, that is, providing information to involved stakeholders in dynamically changing contexts. Thus, one specific category of CPSs is formed by informing cyber-physical systems (I-CPSs), which employ various informing modalities to communicate the results of their context-dependent reasoning to stakeholders (application users) in order to achieve certain objectives, with much more smartness than in the case of a mobile tourist informing system [2], or a personal navigation system [3]. In I-CPSs, the appliances vanish into the background and provide services for each user by sending personalized instructions through an informing terminal [4].

To provide better services for stakeholders and users, implementation of a high-level of context-awareness is a key requirement for many I-CPSs. 'Context-awareness' refers to a capability based on which an I-

CPS can aggregate, interpret, and use context information and automatically adapt its functionalities to the context [5]. In context-aware systems, context information is generally defined as any information related to people, places or objects that are relevant for operations of the systems [6]. In such applications, the context is dynamic as the users move through their environment and the environment is changing as well [7]. Therefore, the behavior of I-CPSs should be dynamic, i.e. these systems are supposed to have capabilities to adapt their sensing, reasoning and informing functionalities to dynamic contexts. The dynamism of context-dependent operations is a key to the success of pervasive and ubiquitous computing [8] [9].

In order to simplify the development of application specific, context-aware I-CPSs and services, designers have proposed to use a central platform and to integrate more and more components on platform level [4]. A platform can accelerate modelling, design, and implementation processes of a given I-CPS, and provide greater safety and security [10]. Therefore, there is a legacy of developing platforms for multiple context-aware I-CPSs that can provide certain generic functions of context-dependent information processing. The four constituents of the platform are: (i) a generic kernel, which offers generic functionalities for context-dependent information processing, (ii) build-in elements, which are application specific algorithms and data (embedded in the generic kernel), (iii) add-on components, which are external sensing and actuating components, and (iv) system interfaces, which connect the add-ons to the generic kernel. Based on these four constituents, the SCA platform can facilitate the development of multiple context-aware I-CPSs, which can change their behaviors in real-time in order to enhance system performance. The functionality of the platform includes development of context-dependent strategies to adapt the sensing, reasoning and informing behaviors of the system built around the platform to various dynamic contexts.

This paper proposes a possible concept for a SCA platform. It also discusses the internal and external integration mechanisms of the SCA platform, which can be customized according to the needs of specific I-CPS applications by extending the generic kernel with various functional build-in elements and add-on components. The paper is organized as follows: In Section 2, a brief overview of the related research and development work is presented. In Section 3, the proposed concept of the SCA platform is discussed, including the requirements, functions, architecture, software components, and integration mechanisms. Section 4 analyses the feasibility and applicability of the SCA platform by ‘virtually’ putting it into a fire evacuation guiding system. Our conclusions are given in Section 5.

## 2. BRIEF OVERVIEW OF RELATED RESEARCH AND DEVELOPMENT

Context-aware applications have been widely studied in recent years. Specific, though limited, attention was given to operation in dynamic contexts. Several frameworks and platforms have been developed in order to simplify and accelerate the development of context-aware applications, but without proposing application independent solutions. For instance, the Context Toolkit was designed as the first attempt to support rapid prototyping of context-aware applications [11]. This framework integrates different functional components to acquire, interpret, store and discover context information modelled by key/value pairs. Another example is the framework used in the Stick-E Notes system [12] [13]. The details of context information are considered as the content of a simple metaphor, Stick-E Notes, which are attached to entities to represent the related context information. A context manager class responds to the notification that satisfies a particular condition based on the triggering-checking mechanism. JCAF is a Java-based context-awareness infrastructure and programming API [14]. In this framework, the basic API defined as Entity, which has a Context specified by a set of ContextItems. The abovementioned three frameworks predefined context information as a set of data. They cannot interpret data on a high level of abstraction, since no inference engine has been considered.

A different approach is represented by the context management framework proposed for distributed context-aware mobile applications [15]. A blackboard-based context manager stores context information gained from multiple sources and serves it to clients in a hierarchical structure. Lun and Cheng proposed a framework based on four layers, namely context acquisition layer, context processing layer, context selection layer, and context application layer [16]. Another layered framework is SOCAM, which is a service-oriented middleware for handing ontology-based context models [17]. ParcTabs and WildCAT are two frameworks that focus on dynamic context information processing [18] [19]. Changes of a particular type of information are represented and treated as events. Hence, actions are made by the system based on the interpretation of different predefined events. Other similar approaches have also been proposed in recent years [20] [21]. Current information processing frameworks address the basic procedures of context information processing, such as context information acquisition, interpretation and actuation. However, most of the procedures are predefined and cannot adapt dynamically to context changes.

Development of generalized hardware, software and cyberware for reusable platforms is one of the major research challenges associated with the implementation of CPSs [22] [23]. Several platforms have been designed to support the development of context-aware applications based on the idea of a centralized architecture. 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 [24], employs a cloud server to maintain the information collected from carry-on wallets, which sense personal information and bio-signals of patients. If an unexpected pattern of information is detected by the wallets, appropriate actuations will be triggered by the cloud server. In other cloud-based platforms, such as the context-aware platform [25] and the RaaS platform [26], 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.

In addition, a platform can be built in a distributed form and manner. For instance, the collaborative and distributed search platform presented in [27], and the mobile networking platform proposed in [28], 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 [29], 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 apply a hybrid architecture, based on which context management and solution development are made in both a centralized and a distributed manners. For instance, robots in the assistive robot platform presented in [30], 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 EvIM for real-time complex event discovery [31], and the citizen sensing & actuation platform [32], 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.

## 3. CONCEPTUALIZATION OF THE SCA PLATFORM

In order to be able to support the development of multiple context-aware I-CPSs, the SCA platform should enable context-dependent information processing and adaptive or dynamic operation of these systems. Context-dependent operation can enhance smartness of system performance. Context-dependent operations in I-CPSs may include; (i) context-dependent sensing, (ii) context-dependent reasoning, and (iii) context-dependent informing. Therefore, sensing, reasoning and informing behaviors can be enhanced by execution of real-time control strategies developed considering the dynamic changes of the context.

The objective of context-dependent sensing is to control (and improve) the behavior of sensors in real-time, e.g. maximizing the lifetime of sensor nodes, enhancing the quality of information collected from sensors, and dealing with accidental failures. Context-dependent sensing makes use of adaptive sensing strategies and protocols to control and reconfigure sensors or sensor networks. The objective of context-dependent reasoning is to generate relevant contents and making decisions for users on what to inform considering both the objectives and the context of the system's operation. Context-dependent reasoning is supposed to be able to choose a proper reasoning algorithm depending on the available context information. The results of context-dependent reasoning should be delivered to users through context-dependent informing operations. Thus, the objective of context-dependent informing is to develop an informing strategy on how to communicate the users in a most effective way, in order to promote or ensure an effective informing process.

Consequently, each context-dependent operation should be in a synergy with multiple preparatory operations. From another perspective, it raises the need for organizing a chain of system operations. What it means is that the context-dependent sensing operations, which are realized based on a real-life scenario of information sensing, should provide intelligence for computation of control strategies for particular sensors, and actuation of the information providers of the I-CPS should also be based on these control strategies.

The basic hypothesis of our research work has been that the SCA platform may encapsulate certain context-dependent operations of I-CPS as generic functions of its kernel part. By defining and integrating application specific elements and components, the kernel can be used to develop multiple I-CPS applications. The target platform has been designed with the focus on applications, where mobile devices are used as both aggregating (information collecting) and informing (instruction messaging) terminals. The mobile devices can aggregate personal information of users and inform users by presenting instructive messages. A wireless sensor network (WSN) has been applied to aggregate additional situational information, including information of non-users and the environment. Therefore, the reasoning mechanism of the platform should simultaneously control both the mobile devices and the WSN to perform all context-dependent operations.

Having these in mind, in the rest of the paper we present the conceptual blueprint of the SCA platform. First, we introduce the requirements and functions of the context-dependent operations in Section 3.1 and Section 3.2, respectively. In Section 3.3, we present the architecture of the SCA platform. The most crucial part of it, the software constituents are discussed in Section 3.4. Then, in Section 3.5, we demonstrate the integration mechanisms of the SCA platform. We will concentrate on the integration of the application specific components with the generic kernel module, because it is essential at realizing application specific I-CPS based on the platform. Then, we describe the workflow of the SCA platform in Section 3.6.

## 3.1. Some basic requirements for context-dependent operations

In general, the SCA platform is required to develop strategies about how to adapt and enhance the sensing, reasoning, and informing operations according to the available dynamic context information in real-time. For this reason, there is a need for a reasoning engine that is able to apply multiple various reasoning mechanisms. To develop the strategies, the reasoning engine needs and should manage relevant information for each context-dependent operation. The strategies should regulate the necessary actuations for controlling context-dependent sensing and informing. The sensors or sensor networks, as well as the informing terminals should perform the control strategies without significant delays, in order to enable a timely completion of the context-dependent sensing and informing actions.

Most of importance is that the working of these context-dependent operations closely depends on each other. Each operation has to respond to the requests of other operations. For instance, the sensors and/or sensor networks should provide sufficient information for various reasoning and informing actions of the system. If some pieces of information are required for informing, a sensing strategy should be generated and operationalized to reconfigure the WSN in order to acquire these pieces of information, e.g. by switching on a sensor in a distributed sensor node. Typically, the result of a reasoning operation is a precondition of a context-dependent informing operation, since the platform has to know what to inform the users about before developing informing strategies on how to inform the users.

## 3.2. Functional description

The basic requirements presented in Sub-section 3.1 have been taken into consideration at designing the main functions of the SCA platform. A detailed description of these functions is given below.

The main function of context-dependent sensing part of the SCA platform is to adapt the sensing behavior of a WSN to the dynamic context. Specific functions of the context-dependent sensing are: (i) aggregate real-time information from the physical world, (ii) manage context information used for developing control strategies, (iii) interpret the need of information from other context-dependent operations, (iv) develop a proper strategy to control the WSN, (v) translate the control strategy into control signals, (vi) deliver the control signals to distributed sensor nodes, and (vii) execute actions according to the control signals.

The main function of context-dependent reasoning part of the platform is to make inference based on the available set of information and perform reasoning according to the dynamically changing context as needed for strategy development and control. Specific functions of the reasoning engine are: (i) analyze the situation based on the information aggregated, (ii) select a proper reasoning mechanism according to the context information, (iii) manage the context information, which is required by the reasoning mechanism, (iv) query information from the sensing components, and (v) generate contents for informing by using the reasoning mechanism.

The main function of the informing part of the platform is to execute the informing of the stakeholders and to adapt the informing actions according to the changes in the context. Specific functions of the context-dependent informing module are: (i) manage context information, which is required by the reasoning mechanisms, (ii) query information from the context-dependent sensing components, (iii) select an informing method for a particular stakeholder based on the strategy and the context information, (iv) construct instructive messages for each particular stakeholder, (v) deliver the messages to their mobile devices, and (vi) execute the informing actions by interacting with the mobile devices.

### 3.3. Architecture of the SCA platform

Several patterns of information processing have been identified in context-aware applications [33]. Based on the functional description of the context-dependent operations, we designed an SCA platform architecture, which is shown in Figure 1. The SCA platform consists of five groups of components: (i) mobile devices, (ii) application server, (iii) service computer, (iv) WSN gateway, and (v) distributed WSN nodes. These five groups of components are integrated through Internet connection and a wireless local area network (WLAN).

The mobile devices are able to aggregate personal information about the users and present the instructive messages to them. In the SCA platform, Android (smart) phones are considered as mobile devices. Thus, the SCA platform related software stored the Android phones is designed based on the Android operating system. The aggregation of personal information includes many things, for instance, collecting input information about users, e.g. profile/schedule of users, and sensed personal information, e.g. location of users. The user interface is designed in a way that it can facilitate both collecting user specific personal data and presenting instructive messages to the users. The mobile devices communicate with the service computer through an application server. A message broker is designed to collect information from mobile devices and forward them to the service computer and to distributed instructive messages to the mobile devices. In addition, the mobile devices communicate with the application server based on the TCP/IP protocol. The application server assigns an IP address to each mobile device, in order to be able to provide personalized instructions for each user.

The service computer is responsible for managing context information and developing strategies for context-dependent operations by various reasoning mechanisms. This also includes the strategies for controlling the network of sensors and the strategies concerning how to inform the users. The historical context information is stored in the local database. The specific results of processing (interpretations of) the context information are stored in the local knowledgebase. A detailed description of the specific software components of the SCA platform is presented in Section 3.4. The distributed WSN nodes are responsible for: (i) aggregating dynamic information from a real-life setting scenario, (ii) making it available for the service computer through a WSN gateway, and (iii) executing control actions delivered from the WSN gateway. On the one hand, the WSN gateway translates the sensor readings into situational information. On the other hand, it translates the control strategies into control signals. The WSN nodes communicate with the gateway through a wireless local area network (WLAN), while the gateway communicates with the service computer through an Ethernet connection. The Zigbee (IEEE 802.15.4) sensor nodes are designed to report sensor readings and execute control signals.

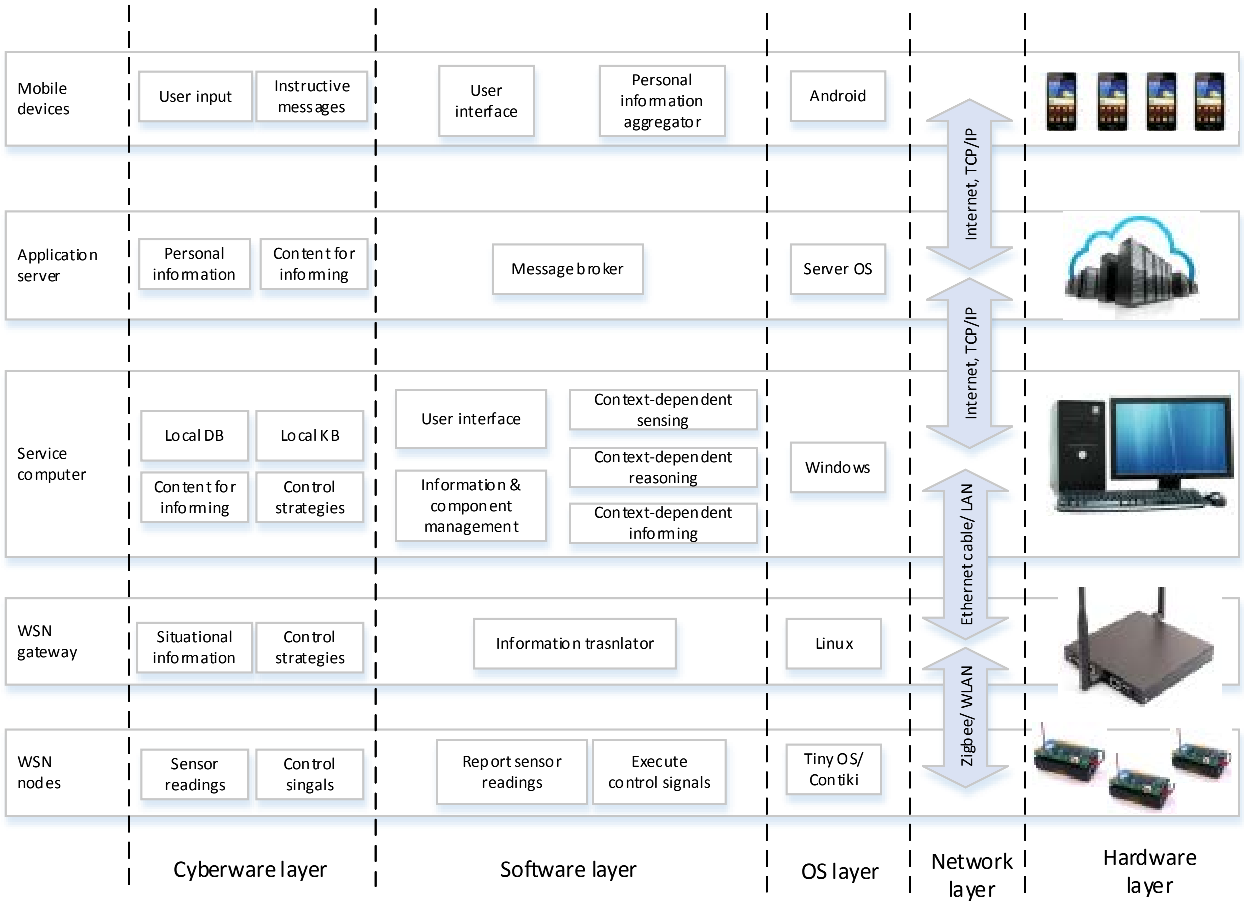


**Figure 1** Architecture of the SCA platform

## 3.4. Software components of the SCA platform

In this sub-section, we introduce the major software components. As shown in Figure 2, the software components are organized in three layers: (i) context-dependent computation layer, (ii) management layer, and (iii) user interface layer. The software components in these three layers are actually interconnected. These all are processed by the service computer. The working status of the context-dependent computation modules depends on the concrete application scenario and some predefined conditions. For example, in the case of an indoor fire evacuation guiding I-CPS, the system instructs the users to escape and provides information about the best escape route alternatives for the users, when fire is detected in a building.

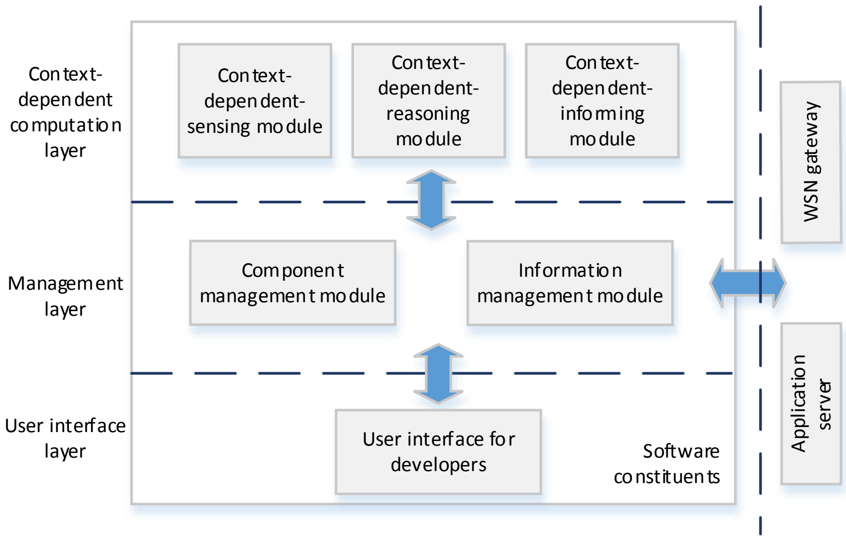


**Figure 2** Specific software components of the platform

The software components residing in context-dependent computation layer are used to develop strategies to enable the SCA platform (and the hosting I-CPS) to adapt the sensing, reasoning and informing behaviors to the dynamic context. This layer comprises three modules, which are: (i) the context-dependent-sensing module (CDSM), (ii) the context-dependent-reasoning module (CDRM), and the context dependent-informing module (CDIM). The CDSM module develops control strategies to control the sensors in order to adapt and enhance the sensing performance. The CDRM module makes decisions on the expected behavior of each user by using various reasoning mechanisms. The CDIM module develops informing strategies considering all users and the changing situation. The screen or speakers (buzzers) available in the mobile devices can be selected as actuators to perform the informing actions. The CDIM module is also responsible for developing personalized instructive messages depending on the context, e.g. personal preference, if known.

The task of the software components residing in the management layer is to manage the context-dependent computation modules and to provide context information for developing context-dependent strategies. Accordingly, information used by the platform is managed in the information management module (IMM), while the working states of the context-dependent computation modules, and the results developed by these modules, are managed by the component management module (CMM). Information aggregated from the mobile devices and WSN nodes are delivered to the IMM through the application server and the WSN gateway. The user interface layer is designed for the developers of I-CPS applications. Based on the standardized internal interfaces, the developers are able to develop algorithms according to the set objectives of the application.

## 3.5. Integration mechanisms

The SCA platform consists of four fundamental constituents: (i) a generic kernel, which offers generic functionalities for a class of context-aware I-CPS applications, (ii) build-in elements, which are application-specific algorithms or data (embedded in the generic kernel), (ii) add-ons, which are application specific external components, including mobile devices and sensor nodes, and (iv) system interfaces, connect the add-ons to the generic kernel. By integrating these four types of constituents, the SCA platform can be used to develop multiple context-aware I-CPSs, which will be able to change their sensing, reasoning and informing behaviors to adapt and enhance system performance in real-time.

Using the SCA platform, the I-CPSs application development should happen in two consecutive phases: (i) internal integration, and (ii) external integration, using different integration mechanisms. These two integration mechanisms used in system architecting also represent two different phases of the development of an I-CPS application. Internal integration focuses on the build-in elements, whereas external integration focuses on the add-on components. The objective of internal integration is to enable the generic kernel to provide context-dependent strategies for the target application. Therefore, application specific build-in elements, including algorithms or data, should be integrated in the kernel. Based on the respective software components run by the servicing computer, the kernel part of the SCA platform implements the generic functions needed for the context-dependent operations, such as the functions for context information management, component management, etc. On the other hand, the objective of the external integration is to establish connection with the distributed sensing and actuating components, such as the mobile devices and the WSN nodes, to perform the sensing and informing actions.

### Internal integration of the build-ins

In order to develop strategies for the context-dependent operations, the application developer has to make a decision on the necessary application-specific build-in elements and integrate them in the generic kernel of the SCA platform. There are four types of application specific build-ins: (i) specifications of the conditions to be considered at

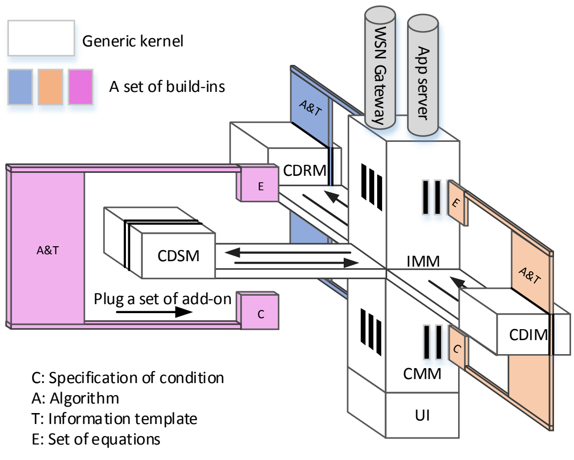


**Figure 3** Internal integration mechanism of the SCA platform

managing context-dependent computation modules, (ii) algorithms used for the development of strategies for context-dependent operations, (iii) information templates used for managing context information, and (iv) set of equations used for deriving necessary context information. The internal integration mechanism of these four types of build-ins is shown in Figure 3.

The specifications of the conditions include predefined information patterns in the CMM, which represent particular situations. Assuming that any one of the situations may happen, the SCA platform should respond to a particular condition and activate the concerned context-dependent computation modules to develop applicable strategies. For example, in the case of an indoor fire evacuation guiding I-CPS, the detection of fire in a building can be considered as a condition. Having this condition perceived by the system, the context-dependent reasoning module is activated and charged with the task of developing the best possible escape routes for the recognized users. Another type of build-in elements are the algorithms used by the reasoning engine in each context-dependent computation module. These algorithms are actually predefined in the CDSM, CDRM and CDIM modules. Each of the algorithms is able to develop strategies for a respective context-dependent operation based on the available situation-pertinent information.

The different context-dependent operations consider different pieces of information as context. Therefore, the context information has to be represented, managed, and stored in the SCA platform by using a context data model. For each context-dependent operation, a set of information patterns, which are incorporated in the context data model, are predefined. The reasoning engine makes use of the relevant pieces of information included in the patterns at the development of strategies for a context-dependent operation. An information pattern template is a type of build-in, which includes the key information elements of the context information. The CDSM, CDRM and CDIM modules use different information templates to query the values of the keys information elements from the IMM. Last but not least, the sets of equations are used for deriving context information necessary for developing strategies for context-dependent operations by the IMM. The reasoning mechanisms in each context-dependent computation module may require IMM to derive information based on the data sensed by the WSN as well on the information queried from the database. As an example for the former, an example can be calculation of the speed of a moving object based on its location changes.

It should be mentioned that these four types of build-in elements are not independent from each other, but have various interconnections. The condition specifications are also used to decide on when a context-dependent operation can be made. Then, the reasoning mechanism included in the context-dependent computation module develops possible strategies using these conditions in association with the predefined algorithms. The algorithm requires a particular set of information, whose keys are defined in an information template. In order to provide the values carried the keys for each context-dependent computation module, the IMM may be required to derive information based on the related equations. Eventually, these interconnections make it necessary that the relevant build-in elements needed to realize a context-dependent operation can be encapsulated as a set to integrate into the generic kernel.

### External integration of the add-ons

There are two types of add-ons considered for the SCA platform: (i) the WSN nodes, and (ii) the mobile devices. The bridge used to interface the add-ons and the computer is the WSN gateway and the application server. The external integration mechanism of the platform is shown in Figure 4.

The apps designed in each Android (smart) phones enable the phone to exchange messages with the application server through Internet communication, which is based on the TCP/IP protocol. The IP address of a mobile phone is used for the identification of user. This identification is necessary in order to make the SCA platform capable of

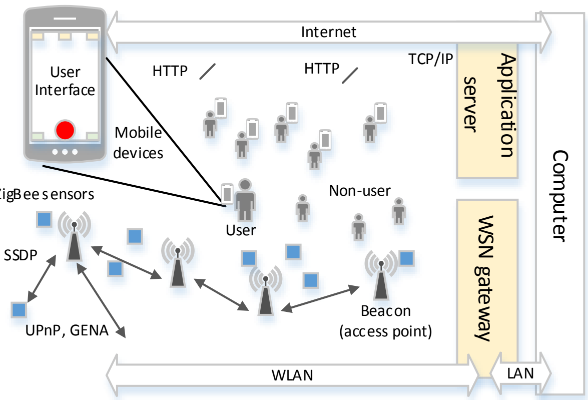


**Figure 4** External integration mechanism of the SCA platform

providing personalized services (messages and instructions). The application server and the mobile phones communicate through HTTP packages. In addition, the ZigBee sensors are able to communicate with the beacons (access point), and WSN gateway through WLAN. The beacons are responsible for managing the dynamic joining and leaving of sensor nodes. The SCA platform support multiple standard protocols, such as universal plug and play (UPnP), general event notification architecture (GENA), and simple server discovery protocol (SSDP).

## 3.6. Operation workflow of the platform

The SCA platform implements a rather complex workflow, which also extends to the cooperation of other modules of the application-specific I-CPS in which the platform is embedded. However, below we restrict the discussion to those platform operations, with which the platform supports the overall operation of I-CPSs. This operation workflow is made up by the logically and temporarily arranged operations of the modules of the platform. It is shown in Figure 5.

Let us assume for the discussion that application developers have completed the information templates, equations, algorithms and conditions using the user interface on the service computer. Also assume that the platform manages mobile devices and WSN nodes through a WSN gateway and application server. As a starting event, the sensor readings collected by the WSN (arrow 1) are processed by the WSN gateway. The cloud-based application server aggregates personal information concerning the users (arrow 2). The structured information is interpreted and sent to the IMM by the interface (arrow 3). By using the equations predefined by the application developer, the IMM is able to construct a content information model, which is referred to as dynamic spatial information reference model (DSIRM). This content information model is made available for the CMM (arrow 4). Based on the conditions predefined by the application developer, CMM is able to select an activating flag (arrow 5), which is then used to trigger and set the working status of the context-dependent computation modules. According to the flag, these modules select an information template, which is provided by the application developer (arrow 6).
Based on the information templates, IMM is able to construct a context information model, which is a subset of the DSIRM (arrow 7). The context-dependent computation modules select possible strategies based on the context information model. These strategies are composed from algorithms defined by the application developers in order to adapt and enhance the sensing, reasoning and informing behaviors in conceived situations by the respective module. Then, the results of context-dependent computation (reasoning) are sent to the CMM (arrow 8). The feedback messages, which are based on the results of the context-dependent computation, are used to select activating flags (arrow 9). For instance, the result of the context-dependent reasoning can be used to trigger the

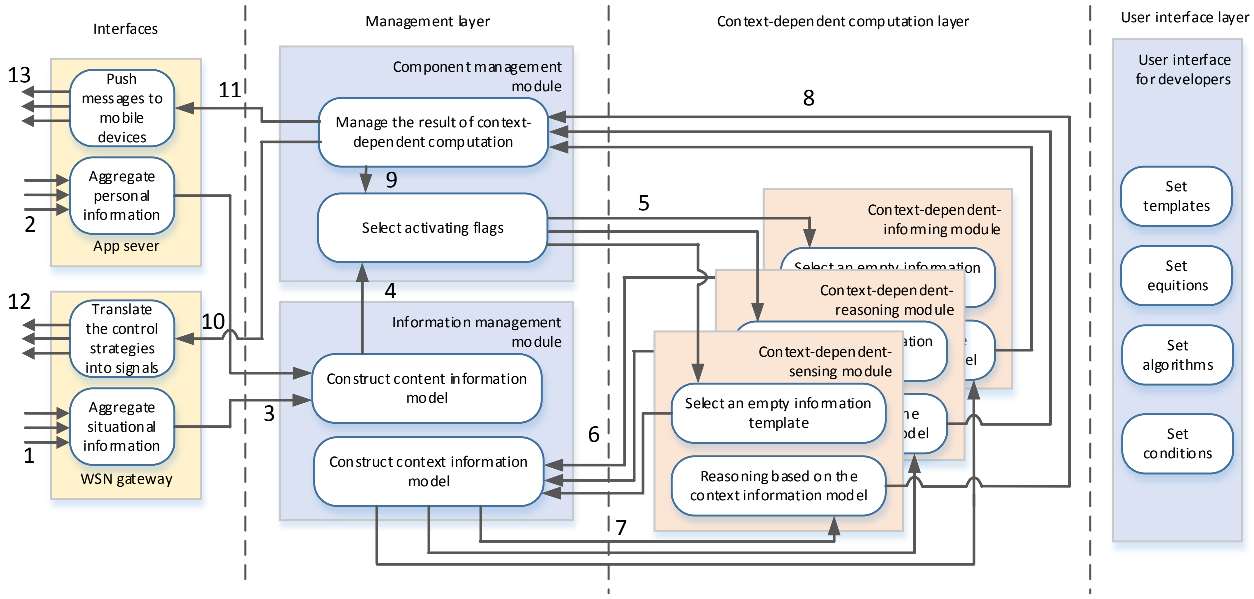


**Figure 5** Operation workflow of the SCA platform

operation of the informing module. Therefore, the control strategies for the WSN will be delivered to the WSN gateway (arrow 10) and translated into control signals (arrow 12). The instructive messages will be created by the informing module and delivered to the cloud-based application server (arrow 11) and the processable messages will be pushed to mobile devices to provide instructions to each included users (arrow 13).

## 4. CASE STUDY: INDOOR FIRE EVACUATION GUIDING SYSTEM

A possible application of the proposed SCA platform is an indoor fire evacuation guiding system (IFEGS). Actually, many functions of the platform have been conceptualized by considering this as a demonstrative application. Development of the IFEGS includes two steps: (i) internal integration of build-in elements, and (ii) external integration of add-on components. Based on the application specific build-in elements and add-on components, context-dependent operations can be realized. An overview of the IFEGS is shown in Figure 6.

The primary objective of the IFEGS is to provide personalized guidance and instructions for the mobile (smart) phone users to escape from a burning building. Obviously, in case of fire within a building, the optimal escape route for a person is not always the shortest path out of the building, since it may be: (i) affected by fire or smoke, (ii) fully occupied by other people, or (iii) inaccessible due to a jam formed

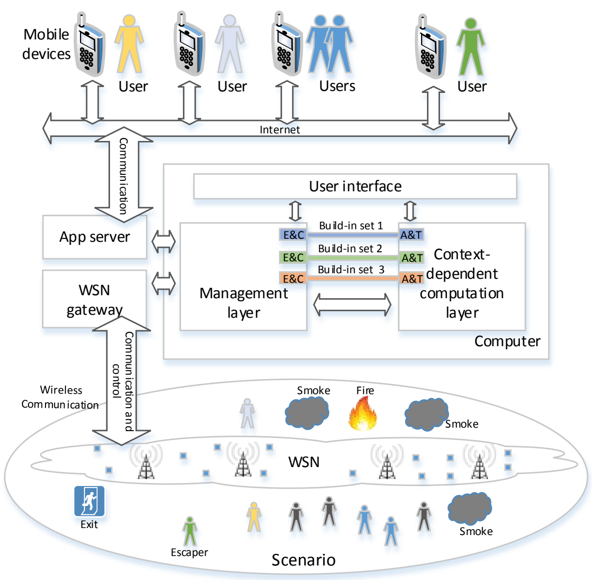


**Figure 6** An overview of the fire evacuation guiding system

by panicking people. In addition, (iv) not every person in the building uses a mobile phone, or (v) have one at hand, and (vi) even active mobile phones may not receive the instructive messages from the system. Furthermore, (vii) the people who have received instructive messages on their personal mobile devices might neglect them, or (viii) just partially obey or even disobey the instructions. Moreover, (ix) poor performance of the WSN in the building, or (x) unexpected system behavior may be experienced for all kinds of reasons, such as a sudden death of sensor nodes due to fire. These real-life happenings should be considered as conditions at the development of IFEGS.

The sensing part of the IFEGS aims at aggregating information from both normal and emergency situations in the building based of conceived situation-dependent strategies. In a normal situation, the sensor nodes work alternatively and periodically to save energy. However, in case of fire, all the sensors should be activated in order to aggregate as much information as possible. If one of the sensor nodes happens to run out of energy, or becomes destroyed by fire, the rest of the sensor nodes should take over its duty. That is, the WSN nodes should be able to change their sensing behavior depending on the dynamic context information. On the other hand, it needs a placement and a strategy of control that enables them to adapt to the new situation and maintain the requested system operation.

The context-dependent reasoning function of the system should create a customized and optimal escape route for each user. The location changes of other people and the propagation of dangerous fire fronts should be considered at the planning of the escape routes. In order to avoid a jam of people or a dangerous spot, the motion of other people and the danger should be forecasted. In addition, some people might be uninformable, since they might not be users of the system, or may neglect the instructions given to them. At the same time, it can be considered only with strong restrictions that an informable user may assist with informing other (digitally non-informable) persons concerning their escape routes.

The context-dependent informing operations enable the development of instructive messages for the users according to the specific situation of each user. The informing module is responsible for: (i) making decisions on the informing contents, and (ii) selection of the proper way and method of informing. In order to accomplish these tasks, the decisions should be made based on the latest context information of each user. Two informing methods have been chosen for IFEGS: (i) voice messaging, and (ii) textual messaging. An implication (meaning) orientated processing of dynamic spatial information is crucial at strategy development for the context-dependent operations. For example, to inform a user about how to escape from a burning building in shortest time, the IFEGS needs to forecast if a jam of people may be expected based on the on-going location changes of all the people in the building. By reasoning with the dynamic spatial context, IFEGS can adapt its operations to emergent situations and optimize its performance.

The first set of build-in elements is designed to realize the context-dependent sensing operation:

- The condition specifying the start of fire and the death of a sensor node.
- The algorithms, which are used to decide if all the sensors should be activated and to develop routing strategies for the delivery of control messages to distributed nodes.
- The information templates, in which the keys of information required by the algorithms are included, such as, 'velocity of fire: __'.
- The equations to derive context information, such as '$v_f = \Delta L_f / \Delta t$', where $v_f$ refers to the velocity of fire, $\Delta L_f$ refers to location changes of fire, and $\Delta t$ refers to a period of time.

The second set of build-in elements is designed to realize the context-dependent reasoning operation:

- The condition specifying the start of fire and a person considered as an informable user.
- The algorithms which are able to generate an escape route for the informable user.
- The information template used by the context-dependent reasoning module, which includes the locations of relevant people and the dangers within the building, and the distances between the informable user and other people and the dangers.
- The equations used to calculate the distance between an informable user and other people and the dangers.

The third set of build-in elements is designed to realize the context-dependent informing operation:

- The conditions specifying the start of fire and completeness of context-dependent reasoning operation.
- The algorithms used to decide on the method of informing,
- The information templates used by the algorithms, such as 'Language preference =__'.
- The equations that can be used to calculate the velocity of fire.

The first type of add-on is the mobile phones, in which an app is designed to aggregate the indoor locations of users and present the instructive messages to the users. The mobile phones are able to inform the users by either textual messages and voice messages depending on degree of emergency. Another type of add-on used in the IFEGS is the WSN nodes. In each node, a temperature sensor and a smoke sensor are integrated to detect the location of fire and smoke. Furthermore, each person in the building is equipped with a RFID tag, based on which the indoor location of the person can be obtained based on the deployed RFID signal senders and receivers in the building.

## 5. CONCLUSION

In this paper, we proposed a concept for an SCA platform that supports the development of multiple context-aware I-CPSs. The SCA platform consists of five groups of components: (i) mobile devices, (ii) application server, (iii) service computer, (iv) WSN gateway, and (v) distributed WSN nodes. These five groups of components are integrated through Internet connection and WLAN communications. The platform enables the realization of context-dependent sensing, reasoning, and informing operations. Based on the internal and external integration mechanisms, various functional build-in elements can be enclosed in, and add-on components can be appended to the generic kernel of the SCA platform. As a case study part, we demonstrated the usability of the platform in an indoor fire evacuation guiding system. We discussed the objectives and specific functionalities of each context-dependent module, as well as the possible application specific build-in elements and add-on components.

When certain predefined conditions are detected in the SCA platform, the component management module activates the context-dependent computation modules. The context-dependent computation modules query context information from the information management module by sending information templates. Therefore, context information management and context-dependent computations are allocated to different components as independent operations. This increases scalability, feasibility and applicability of the platform. The proposed platform offers an open and flexible solution for the realization of context-dependent operations. In our further work, we will focus on the implementation and testing of the SCA platform.

## ACKNOWLEDGMENTS

This work is sponsored by Chinese Scholarship Council (CSC).

## REFERENCES

[1] Rajkumar, R., Lee, I., Sha, L., and Stankovic, J., (2010), Cyber-physical systems: The next computing revolution, in: Proceedings of the 47th ACM/IEEE Design Automation Conference, Anaheim, pp. 731-736.

[2] 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, Vol. 15, pp. 26-47.

[3] Saeedi, S., Moussa, A., and El-Sheimy N., (2014), Context-aware personal navigation using embedded sensor fusion in smartphones, Sensors, Vol. 14, No. 4, pp. 5742-5767.

[4] Baldauf, M., Dustdar, S., and Rosenberg, F., (2007), A survey on context-aware systems, International Journal of Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, pp. 263-277.

[5] Wang, C.Y., Hwang, R.H., and Ting, C.K., (2011), UbiPaPaGo: Context-aware path planning, Expert Systems with Applications, Vol. 38, No. 4, pp. 4150-4161.

[6] Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., and Steggles, P., (1999), Towards a better understanding of context and context-awareness, in: Proceedings of Handheld and Ubiquitous Computing, Vol.1707, pp. 304-307.

[7] Gallacher, S., Papadopoulou, E., Abu-Shaaban, Y., Taylor, N.K., and Williams, M.H., (2014), Dynamic context-aware personalisation in a pervasive environment, Pervasive and Mobile Computing, Vol. 10, No. 2, pp. 120-137.

[8] Kaemarungsi, K., Krishnamurthy, P., (2012), Analysis of WLAN's received signal strength indication for indoor location fingerprinting, Pervasive and Mobile Computing, Vol. 8, No. 2, pp. 292-316.

[9] Weiser, M., (1993), Some computer science issues in ubiquitous computing, Communications of ACM, Vol. 36, No. 7, pp. 75-84.

[10] Hatcliff, J., King, A., Lee, I., Macdonald, A., Fernando, A., Robkin, M., Vasserman, E., Weininger, S., and Goldman, J.M., (2012), Rationale and architecture principles for medical application platforms, in: Proceedings of the Third International Conference on Cyber-Physical Systems, IEEE/ACM , pp. 3-12.

[11] Dey, A.K., Abowd, G.D., and Salber, D., (2001), A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, Human-Computer Interaction, Vol. 16, No. 2-4, pp. 97-166.

[12] 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.

[13] 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, ACM, Orlando, pp. 261-264.

[14] Bardram, J., (2005), The Java context awareness framework (JCAF) - A service infrastructure and programming framework for context-aware applications, In Proceedings of Pervasive Computing, Springer, Berlin, Vol. 3468, pp. 98-115.

[15] Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H., and Malm, E.J., (2003), Managing context information in mobile devices, Pervasive Computing IEEE, Vol. 2, No. 3, pp. 42-51.

[16] Lun, Y.L., Cheng, L.L., (2011), The research on the model of the context-aware for reliable sensing and explanation in cyber-physical system, Procedia Engineering, Vol. 15, pp. 1753-1757.

[17] Gu, T., Pung, H.K., and Zhang, D.Q., (2004), A middleware for building context-aware mobile services, in: Proceedings of the IEEE Vehicular Technology Conference, Milan, Vol. 5, pp. 2656-2660.

[18] Shilit, B.N., (1995), A context-aware system architecture for mobile distributed computing, Columbia University, New York, pp. 95-118.

[19] 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, ACM, Grenoble, pp. 1-7.

[20] 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, pp. 663-670.

[21] Kim, M., Stehr, M.O., Kim, J., and Ha, S., (2010), An application framework for loosely coupled networked cyber-physical systems, in: Proceedings of the 8th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Hong Kong, pp. 144-153.

[22] Park, K.J., Zheng, R., and Liu, X., (2012), Cyber-physical systems: Milestones and research challenges, Computer Communications, Vol. 36, No. 1, pp. 1-7.

[23] Sanislav, T., Miclea, L., (2012), Cyber-physical systems--Concept, challenges and research areas, Control Engineering and Applied Informatics, Vol. 14, No. 2, pp. 28-33.
[24] Bhatti, A.M., Masud, M., (2014), Context-aware intelligent wallet for healthcare, International Journal of Computer Applications, Vol. 96, pp. 31-36.
[25] Martín, D., Lamsfus, C., and Alzua-Sorzabal, A., (2016), A cloud-based platform to develop context-aware mobile applications by domain experts, Computer Standards & Interfaces, Vol. 44, pp. 177-184.
[26] Chihani, B., Bertin, E., and 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.
[27] Shen, C.Y., Yang, D.N., and Chen, M.S., (2012), Collaborative and distributed search system with mobile devices, IEEE Transactions on Mobile Computing, Vol. 11, No. 10, pp. 1478-1493.
[28] Wan, J.F., Suo, H., Yan, H.H., and Liu, J.Q., (2011), A general test platform for cyber-physical systems: Unmanned vehicle with wireless sensor network navigation, Procedia Engineering, Vol. 24, No. 8, pp. 123-127.
[29] Seshasayee, B., (2008), Middleware-based services for virtual cooperative mobile platforms, Georgia Institute of Technology, Atlanta, pp. 82-98.
[30] Oniga, S., (2013), ICT tools for smart homes and assisted living for elders, in: Proceedings of conference on Advances in Wireless Sensor Networks, Debrecen, pp. 41-46.
[31] Dao, M.S., Pongpaichet, S., Jalali, L., Kim, K., Jain, R., and Zettsu, K., (2014), A real-time complex event discovery platform for cyber-physical-social systems, in: Proceedings of International Conference on Multimedia Retrieval, Glasgow, pp. 201.
[32] Crowley, D.N., Curry, E., and Breslin, J.G., (2013), Closing the loop: from citizen sensing to citizen actuation, in: Proceedings of the 7th International Conference on Digital Ecosystems and Technologies, IEEE, Menlo Park, pp. 108-113.
[33] Adel, A., Laborie, S., and Roose, P., (2014), Semantic context-aware adaptation platform architecture, Procedia Computer Science, Vol. 32, pp. 959-964.