

A synthetic agent to simulate decisional behaviour of designers working with an active recommender framework system

Horváth, Imre; Tepjit, Sirasak

DOI

[10.3233/JID-220015](https://doi.org/10.3233/JID-220015)

Publication date

2024

Document Version

Final published version

Published in

Journal of Integrated Design and Process Science

Citation (APA)

Horváth, I., & Tepjit, S. (2024). A synthetic agent to simulate decisional behaviour of designers working with an active recommender framework system. *Journal of Integrated Design and Process Science*, 27(2), 83-109. <https://doi.org/10.3233/JID-220015>

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.

A synthetic agent to simulate decisional behaviour of designers working with an active recommender framework system

Imre Horváth^{a,*} and Sirasak Tepjit^b

^a*Faculty of Industrial Design Engineering, Delft University of Technology, Delft, the Netherlands*

^b*Faculty of Engineering, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand*

Abstract. Application-specific reasoning mechanisms (ASRMs) development is a rapidly growing domain of systems engineering. A demonstrative implementation of an active recommender system (ARS) was realized to support designing ASRMs and to circumvent procedural obstacles by providing context-sensitive recommendations. The specific problem for the research presented in this paper was the development of a synthetic validation agent (SVA) to simulate the decisional behaviour of designers and to generate data about the usefulness of the recommendations. The fact of the matter is that the need for the SVA was raised by the pandemic, which prevented involving groups of human designers in the recommendation testing process. The reported research had three practical goals: (i) development of the logical fundamentals for the SVA, (ii) computational implementation of the SVA, and (iii) application of the SVA in data generation for the evaluation of usefulness of recommendation. The SVA is based on a probabilistic decisional model that quantifies decisional options according to the assumed decisional tendencies. The three key concepts underlying the SVA are (i) decisional logic, (ii) decisional knowledge, and (iii) decisional probability. These together enable generation of reliable data about the decisional behaviours of human designers concerning the obtained recommendations. The completed tests proved the above assumption.

Keywords: Synthetic validation agent, decisional behaviour, decisional variables, decisional modes, shared design knowledge, usefulness of recommendation

1. Introduction

1.1. Background of research

The work reported in this paper is a part of a multi-year promotion research project. To support the development of application-specific reasoning mechanisms (ASRMs) for smart cyber-physical systems (S-CPSs), this project was aimed at ideation and design of an active recommender framework (ARF) (Tepjit, 2022). One example of ASRMs is the computational enablers (collaborative algorithms and software agents) of automated parking. Actually, this application case was used in the promotion research both as the target of research/design and as a real life problem context. In simple words, the ARF is a special recommender system that advises on how to overcome a hindrance in the design process of an ASRM rather than on what products are relevant for a particular customer/user to buy. The objective of the ARF development was to provide support services by using its runtime process monitoring and context-sensitive decision support mechanisms concurrently. The ARF provides

*Corresponding author: Imre Horváth, Faculty of Industrial Design Engineering, Delft University of Technology, Delft, the Netherlands. Tel.: +31 152783520; E-mail: i.horvath@tudelft.nl.

recommendations to compensate designers for their possible knowledge deficit associated with their decision making and/or to assist them to overcome procedural obstacles in design problem solving (Tepjit et al., 2019).

The development of the ARF was a multi-step process including steps such as (i) deriving requirements and exploring opportunities, (ii) service oriented ideation and overall conceptual modelling as a knowledge- and reasoning-intensive software system, (iii) functional and architectural specification and detail design, (iv) defining the scope of a demonstrative implementation, (v) adoption/development of the required computational mechanisms and algorithms, (vi) customization of the demonstrative implementation to the target application, (vii) testing the performance of the demonstrated ARF, and (viii) validation of the usefulness of the recommendations in the application context. The work reported in this paper is related primarily to this last activity.

1.2. Goals of the reported research

The primary objective of validation of recommendations was to demonstrate their ‘usefulness’ for the intended use. Usefulness was deemed a composite indicator that concerned features such as appropriateness, relevance, accuracy, comprehensibility, informativeness, and acceptability in the given context. In the literature, usefulness is usually interpreted as utilitarian goodness and not as instrumental (usability), pragmatic (problem solving), technical (performance), hedonic (pleasure), and/or transcendental (mystic) goodness. Our hypothesis was that a useful recommendation delivers information content that (i) is relevant for the continuation of the design process, (ii) the designer can recognize if it corrects or not, and (iii) the designer can operationalize the best next design action based on it. Thus, utilitarian goodness has been defined as the strategic objective and the specific indicator of validity. Accordingly, assuming that the recommendations generated by the ARF were internally valid (that is, correct and un-biased), we aimed at external validation of results, rather than exploring biases and other influential factors through internal validation. From the perspective of the validator, a combined validation scenario was operationalized, which involved the researchers as investigators, but substituted the subjects (recommendation assessing designers) by an SVA at performing the usefulness validation.

The usefulness (and timeliness) of the recommendations generated by various decision support systems are usually tested with the involvement of human stakeholders. It is widely documented that representative research groups and reference groups of subjects are formed considering the needed subject characteristics, principles of probability, and statistical significance. Usually, (i) the subjects are asked to complete given tasks according to certain scenarios, (ii) their responses are recorded, their behaviours are observed and documented, data about their performance are collected, and they are interrogated afterwards, and (iii) the research data are statistically processed and conclusions are drawn. Applying the same principles, we have developed a detailed plan to study the usefulness of the recommendations provided by the ARF software by involving human stakeholders, i.e., designers (involved in the logical specification) and software engineers (involved in computational implementation) of reasoning mechanisms. The plan included experimental sessions to study situations in which they face procedural obstacles or are held back by knowledge deficits.

Though the above scenario was believed to work well procedurally, it could not be implemented. The reason was the restrictions caused by the COVID pandemic, which did not allow us to get practicing designers involved in experimental design processes and to test the usefulness of the ARF-provided recommendations in practical problem-solving cases. Therefore, instead of a human-inclusive validation approach, we had to look for an alternative approach. We hypothesized that a dedicated computational simulation-based approach may be appropriate. Based on the study of the related literature, it was

concluded that a surrogate means supported by the principles of agent technology could be applied with minimal limitations. Consequently, conceptualization and realization of a synthetic validation agent (SVA) was formulated as the first goal of the recommendation validation orientated part of the promotion research. The SVA was conceived as a high-fidelity surrogate for various human designers that provides data on their decisional behaviour and possible choices when using the individual recommendations generated by the ARF. From a functional viewpoint, the programmable surrogate was supposed to (i) be familiar with possible design routes and design actions, (ii) be informed about the state of progression, (iii) reproduce decision-making actions, and (iv) mimic social behaviour of human designers.

1.3. Methodological approach of the research

The methodological approach was simultaneously influenced by the overall objective of the embedding promotion research project and the specific objective of the confirmatory part of the research. The objective of the promotion research was to generate correct recommendations for designers and to test their usefulness in the preferred application context. The specific objective of development and deploying the SVA was generation of decisional behaviour data for the evaluation of the usefulness of recommendations. That is, the SVA was intended to assist validation, but not to completely replace researchers in the process. The assumption was that the SVA mimics the decisional behaviour patterns of designers, while the generated data carry information about the potential of the designers to overcome an experienced procedural obstacle based on the obtained recommendation. The objective of validation was to check how well the data generated by the SVA correspond to what could be expected from human designers.

Accordingly, the research cycle for the SVA-based validation study included (i) an exploratory phase, (ii) a constructive phase, and (iii) a confirmatory phase. The exploratory phase involved (i) a systematic literature study, (ii) generation of an explanatory theory, and (iii) a functional requirement synthesis. The constructive phase comprised of (i) overall concept development, (ii) elaboration of the decisional logic, knowledge, and probability, (iii) development of the computational algorithms, and (iv) coding and verification of the SVA. The confirmatory phase involved (i) systematic data generation by the SVA, (ii) investigation of the generated data sets, and (iii) enhancement of the functional parameter values of the SVA. As these steps indicate, the ARF was only implicitly involved in this research cycle, namely through generating recommendations in various application situations and through the use of the knowledge included in the reference process protocol of the ARF. The SVA introduced a new but inseparable aspect of validation of recommendations, namely validation of the proper operation of the agent itself. This will be in the centre of discussion in the remaining part of the paper.

1.4. Contents and structuring of the paper

The next section provides an overview of (i) the aspects and techniques of validation, (ii) the current results in computerization of validation, (iii) the agent technologies for surrogating human decision making, and (iv) comparable efforts towards the development of validation agents. The third section discusses the essence and integration of decisional logic, decisional knowledge, and decisional probability. The fourth section provides information about the functionality of the implemented SVA and discusses the testing of essential sub-functions in application. The fifth section summarizes the major conclusions and open issues.

2. Overview of the related and preceding research

2.1. Understanding, modelling, and replicating human decision making and decisional behaviours

Research in human decisional behaviour involves multiple disciplines and a wide range of activities, from observation and experimentation through understanding and modelling to reproduction and application. Decision-making is the process of selecting a course of actions or a particular object from among multiple alternatives including (i) analysis of the alternatives, (ii) evaluation of the uncertainties and hazards, and (iii) choosing the best alternative based on some criteria (Wainwright, & Mulligan, 2004). The starting point of human decision making is selection of a belief or an action. Decision theory is based on the principle of maximum expected utility (Dastani et al., 2001). If a human or software agent is faced with a decision between prospects (beliefs or actions of uncertain outcome), then the option that has the highest expected utility should be chosen. The prospect theory advises that choosing the prospect with the highest expected utility and minimal risk is the most rational decision because it will maximize the agent's utility in the long term (Tversky & Kahneman, 2015).

However, human decision-making often transcends formal logic and rationality models. Widely used in modelling human behaviours in computational systems, classical decision theory and cognitive psychological theories recognize bounded rationality of humans. As Chung and Silver (1992) discussed, the early efforts to build (rule- or frame-based) expert systems typically applied linear models of expert decision-making behaviour, which were derived by logistic regression or statistical methods. The latest literature argues that human decision-making processes are non-linear and decisional models should capture their intrinsic non-linearity. The experiments done by Kim et al. (2008) showed that non-linearity in the decision-making process is positively related to the predictive validity of the decisions and that higher non-linearity in the decision-making process raises the need for higher non-linearity in computational decision models. Though much research shows that statistical linear models are better in many fields, non-linear models (such as various learnt neural network models) have higher potentials and show better performance in hard-to-explicitly-model applications.

The study of Parker and Fischhoff (2005) investigated to what extent (i) individuals show consistent performance differences across typical behavioural decision-making tasks, and (ii) how those differences correlate with plausible real-world correlates of good decision making. They found that, apart from individual differences, the decision-making performance was predictably related to measures of (i) basic cognitive abilities, (ii) cognitive styles, (iii) developmental conditions, and (iv) risk-taking behaviours suggesting poor real-world decision making. Cognitive neuroscience advises that both (i) a model-free system that uses values cached from the outcome history of alternative actions, and (ii) a model-based system that considers action outcomes and the transition structure of the environment of multiple brain systems contribute to the choice behaviour. Based on this, Shahar et al. (2019) examined the psychometric properties of model-based and model-free estimates in a two-stage decision task. Huang et al. (2020) discussed a specific human decision-making behaviour model, called human drift diffusion model, for controlling multi-robot systems. This concentrates on the evolution of human-decision information and uses a threshold of such information (determined based on the Bayesian risk criteria) to trigger timed human interaction.

Rosenfeld and Kraus (2018) showed the differences between the thinking of single and multiple decision makers, and analysed (i) expert driven paradigms (e.g. prognostic reasoning, utility maximization, level-k argumentation, competitive games, quantal response, cognitive hierarchy, non-monotonic reasoning), (ii) data-driven paradigms (knowledge mining, machine learning, deep learning, transfer learning), and (iii) additional hybrid approaches. Van Duinen et al. (2016) argued and evidenced that heuristics and social networks play an important role in decision-making under risk, as was claimed by theoretical and experimental studies by psychological and behavioural sciences. The concept of

behaviour networks was introduced by Maes (1989) as a means to combine reactive and deliberative decision making. It uses a mechanism of activation spreading to determine the best behaviour. Based on this, Dorer (2009) worked out the concept of extended behaviour networks (EBNs) in which the mechanism of activation spreading was changed so as to make it possible to use activation as a measure of expected utility of a behaviour.

Robinson (2003) used a simulation-based approach to elicit knowledge about human decision-making in combination with an expert system shell (XpertRule) to learn the decision-making strategies of humans. Yu and Choi (2005) discussed a behaviour decision model for robots that is based on (i) artificial emotions, (ii) various motivations, and (iii) dynamic personality models. The major goals of their work were to connect the behaviour decision model to emotional expression and to realize adaptability to various environments through interactions. Wu et al. analysed the composition of the human decision-making process in a human-robot interaction context and proposed a human decision-making behaviour model representing both the supervision behaviour and intervention behaviour of humans. Zhang et al. (2018) used the ACT-R cognitive architecture to represent decision-making behaviours and processes of humans aiming at real-life dynamic settings, which are characterized by constant changes and large individual differences in making a decision in duo-choice and multi-choice HCI tasks. Raghavan and Baras (2019) considered the binary hypothesis testing problem with finite observation space, and compared the probability of error achieved when the hypothesis testing problem is solved in the classical (Kolmogorov) probability framework and in the non-classical (von Neumann, quantum-like) probability framework. The above-mentioned authors suggest that classical probability models suffer from limitations in modelling human decision making.

2.2. *Fundamentals and recent results of computer-aided and -automated validation*

Gibson and Powell-Evans (1998) gave reasons and evidence of why validation is an integral methodological element of good scientific research practice targeting objective predictive credibility. The general goal is to generate and use objective evidence to confirm that a theory, an artefact, a process, or any other thing, fulfils certain requirements and does what it is supposed to do (Thacker, et al., 2004). Research validation has a very broad literature, which offers different vocabularies, interpretations, principles, and approaches for the different fields of concerns. Robinson (1997) discussed many difficulties of validation. For instance: (i) there is no such thing as general validity, (ii) there may be no real world to compare against, (iii) different interpretations of the real world exist, (iv) inaccuracy of real world data frequently occurs, and (v) there is not enough time to verify and validate everything. Validation is also a well-known concept in systems engineering where it provides information about appropriateness and dependability (Engel, 2010). The above circumscribed notional complexity is the reason why we briefly discuss the fundamentals, aspects, and techniques of validation, as well as its use in our context, below.

Validation differs from verification and can have internal or external orientation (Ryan & Wheatcraft, 2017). Internal validity is the extent to which the research design and the conduct of a study or realization are likely to have prevented systematic bias and, therefore, the results may be considered reliable. External validity expresses the extent to which the (internally valid) results of a study or realization can be held true in various application contexts and/or for other cases involving, for example, different people, places, or times. It is challenging to capture the real meaning and manifestation of external validity with one single characteristic or measure (Rothwell, 2005). Depending on the time of completion, validation can be: (i) prospective, (ii) concurrent, or (iii) retrospective validation. Based on the periodicity of the application, it can be: (i) single-time validation, (ii) revalidation after change, and (iii) periodic revalidation. According to the involved actors, (i) conventional (human planned, executed, and evaluated), (ii) combined (human planned, surrogate assisted, and human evaluated),

and (iii) automated (surrogate planned, executed, and evaluated) validation scenarios are possible. Here the word ‘surrogate’ stands for some sort of special software agent.

In the last two decades, both computer-assisted validation and automated self-validation have become a pivotal issue in software development, but also in systems and knowledge engineering. Self-validation of generated and fused data by quasi-redundant sensors and sensor networks is an example of attempts to automate validation (Frolik et al., 2001). The main frontline themes of research are such as (i) appropriateness and application related performance testing, (ii) validation of dynamic system and control models, and (iii) validation of self-adaptation and self-evolution (Yang et al., 2019). In the context of validation computational models, Kwasniewski (2009) discussed the relationships among (i) reality of interest, (ii) conceptual/mathematical models, (iii) computational models/mechanisms, and (iv) validation experiments, considering both the creative and the assessment activities. Monschein et al. (2021) differentiated (i) internal validity, (ii) external validity, (iii) construct validity, and (iv) conclusion validity as important dimensions of testing software adaptation and evolution. Automation of validation methods and processes is a top-of-list item and challenge in the development of self-adaptive and self-evolving systems. In line with the growing complexity and autonomy of systems, the involvement of human validators should be decreased or cannot be considered at all. This influences both design-time validation and run-time validation of fitness for purpose of software systems and hybrid (cyber-physical) systems.

To support design-time validation, various system model-based simulation and artificial learning-based forecasting models and techniques have been proposed (Meng et al., 2020). Due to their uncertain and changing relationship to the empirical evidence, both suffer from a credibility issue. Steyerberg et al. (2001) and Steyerberg et al. (2003) analysed the internal validation of predictive models and the biases and precision of their internal and external validation, respectively, in the case of small samples. Klügl (2008) argued that validity is the basic prerequisite for every simulation model and it vindicates the use of the agent-based simulation paradigm. She discussed different types of alternative validation, such as (i) empirical validation or statistical validation, (ii) conceptual model validation or operational model validation, and (iii) structural validation or process validation. She claimed that the possible level of validity strongly depends on the validation technology used and that the highest level of validity is possible only if both informal and formal validation techniques are applied. Guerini and Moneta (2017) proposed a method for empirical validation of specific theoretical and computational simulation models that generate artificial time series data comparable with real-world data. Ramspek et al. (2021) investigated the issues of external validation of prognostic models in a non-engineering context (i.e. in the case of prediction of clinical events and decision-making).

Moghadam (2019) claimed that automated testing activities (such as automated test case generation, automated validation, fusion of experimental data, etc.) reduce human effort and cost, and increases test coverage and executional performance. Typically, performance evaluation of software-under-test is conducted to (i) measure the performance metrics, (ii) detect the functional problems in varying execution conditions, and (iii) identify non-functional problems and violations of requirements. As a trend, agent-based technologies are intensively studied in the above contexts (Kumaresen et al., 2020). The testing agent learns an optimal policy to generate an effective workload efficiently and is able to reuse the learned policy in further similar testing scenarios. Smart tester agents equipped with learning capabilities and strategies generate implicit models that can be reused in several analogous situations. Instead of using an explicit model-based approach and user behaviour patterns, Moghadam et al. (2021) used smart reinforcement learning to implement a testing agent for efficient automation of workload testing and optimization in many decision-making problems.

Concerning the field of economics, Guerini and Moneta (2017) argued that agent-based models pose a serious methodological problem. Some researchers are afraid that the use of machine learning and deep learning converts the decision-making process and decision excavation into a categorization

problem. The fact of the matter is that development of an advanced decision model requires systematic and synthesized empirical studies, which include various areas and circumstantial factors affecting a predictive accuracy of a model or a system. Therefore, psychologists deem it essential to understand how humans make decisions and judgments under uncertainty that is often-times a characteristic of the operation of networked multi-agent systems. The inherent complexity of artificial intelligence and machine learning-based prediction models complicates risk calculation, and internal and external validation and explanation of 'black-box' models are still an open issue.

2.3. Agent technologies for surrogating human decision making and efforts towards development of validation agents

Modelling and reproduction of individual and collective human decisional processes, styles, and patterns are widely addressed in the field of artificial intelligence research. It requires capturing such influencing factors as intelligence, knowledge, wisdom, experiences, but also common sense, creativity, intention, emotions, apobetics, and social contexts. Towards this end, various clever reasoning mechanisms have been combined with human context ontologies (like SOUPA, mIO!, FOAF, PiVOn, and 3LConOnt). Nevertheless, high-fidelity computer-reproduction of complicated human decisional behaviours has remained a challenging matter. It is influenced by a large number of internal and external factors. Due to their operation in closed local worlds, artificial agents feature the so-called double bounded rationality (Burciu and Hapenciuc, 2010). Nevertheless, smart computational agents have been recognized as a promising technology to build next-generation decision-intensive systems (Zha et al., 2003). In spite of the limitations, they can replace humans in many decision-intensive tasks.

From the large number of agent decision-making models discussed in the literature, Balke and Gilbert (2014) analysed 14 different agent decision-making architectures and discussed their aims, assumptions, and suitability as modelling agents in computational simulations. They identified two dimensions of designing agent architectures: (i) the cognitive level of the agents, and (ii) the social level of the agents, and provided guidelines for the development of the associated agent decision-making models. The agent-based model approach enables properties emerging from the interaction among heterogeneous and bounded rationality agents. Claiming that reliability and usability of software during its operational state is crucial, Manzoor et al. (2011) implemented an autonomous agents-based framework using the JADE for testing networked software in run-time. This framework arranges (i) system administration, (ii) lab testing, and (iii) software testing agents in a layered structure. Choi and Choi (2002) reported on a test agent system that employs intelligent agent characteristics of autonomy, social ability, and intelligence rules, and that includes (i) user interface agent, (ii) test case selection agent, (iii) test execution agents, and (iv) regression test agent, to provide active assistance to the autonomous tester. The three tester agents use three different methods for (i) data driven testing, (ii) white-box testing, and (iii) black-box testing.

Intelligent agents are seen as a promising technology to deal with the development and testing of complex systems. Silveira et al. (2013) proposed an approach that uses rational agents who are able to test other rational agents. According to their conceptualization, the rational validation agent involves (i) a tester agent, (ii) a monitoring agent, and (iii) one or more task environment representation agents to indicate the performance of the tested agent. Kant and Thiriot (2006) used a multi-agent mechanism (MAM) to model one human decision maker. In order to soften the typically too rational behaviour of agents when faced with real-world human data and to take the bounded rationality of humans into account, the authors employed an entire multi-agent system, in which each agent is in charge of a particular sub-process of the whole decision-making process. Caire et al. (2004) proposed a framework for testing multi-agent systems, which considers two operation level model. On the first level, the agents are identified as atomic entities and the correctness of the activities carried out by them as a single

agent is checked through a number of different cases (agent-test). On the second level, the specific agent tasks are identified and a set of tests are completed related to all capabilities of a given agent (task-test).

In the agent-based validation methodologies, (i) both single agent and multiple (collaborating or competing) agents are taken into account, (ii) they are deployed to both non-agent-based systems and agent-based systems, (iii) they are used for both design-time prognostic and run-time self-variation, and (iv) they can execute both data-driven and model-based validation. In addition to the application context and the goals of validation, the required intelligence and automation levels and the opportunities for self-learning are the most influential factors in the methodological approach. Contrary to the expectable reliability, moderate development efforts, and specialization, single decision validation agents are scarcely used. Designing intelligent agents that interact proficiently with people necessitates the modelling of human behaviour and the prediction of their decisions. Jager and Janssen (2012) proposed a conceptual framework for developing agent rules to capture main behavioural drivers and processes. They investigated repetition, imitation, inquiring, and optimization as four strategies to capture consumer behavioural opportunities and managed to simulate cognitive decisional processes of moderate complexity, without considering social contexts and morality in agents. Mutanu and Kotonya (2018) proposed a self-learning orientated framework that includes (i) an application context mechanism, (ii) a sensor manager mechanism, (iii) an adaptation manager mechanism, and (iv) a machine learning engine, and provides adaptation validation as a (negotiated) service.

Testing models is a core activity for verification and validation of systems. Yin and McKay (2018) found that the literature is limited to concrete real-world methods and practical instruments for system model validation, while it is rich in theoretical framework and overall strategies. Already 20 years ago, Grundy and Ding (2002) warned that automatic validation of whether software components meet their requirements under a particular deployment scenario is a challenging task. Other researchers have also come to the conclusion that determining whether certain properties are valid for a given software is a computationally hard (and in general undecidable) problem. Kumar and Goyal (2012) attempted to automate testing of multi-agent systems and to execute the process in a continuous fashion using a dedicated autonomous tester agent, which continuously interacts with the agents under test by means of message passing. Thiriote and Kant (2006) proposed a multi-agent system, in which each agent (autonomous entity) is responsible for a particular sub-process of the whole decision-making process.

Nguyen et al. (2008) developed a tool for deriving (goal-oriented, ontology-based, random, and evolutionary mutation) test cases semi-automatically from goal-based analysis diagrams to generate meaningful test inputs based on agent interaction ontology. Friess et al. (2009) discussed a multi-agent-based simulator for testing anti-spam software, in which agents represent spammers and legitimate email users. Each agent follows hardcoded algorithms with randomization in areas such as message selection and speed of an agent's reply to a message. Rodríguez et al. (2014) classified the different ontology-related methods for human activity recognition and behaviour mining as data-driven and knowledge-based methods. Jain and Patel (2021) proposed a model to quantify the user context and provide semantic contextual reasoning based on a diagnostic belief algorithm, which computes the confidence of the decision as a function of available resources, premises, exceptions, and desired specificity.

2.4. Summary of the major finding

Validation of appropriateness, trustworthiness, performance, and other characteristics of models and implementations is strongly required by hardware, software, cyberware, and hybrid system engineering and utilization. Therefore, understanding, modelling, and replicating human decision-making and

decisional behaviours have been approached by many disciplines, but yet not in a multi-disciplinary or trans-disciplinary manner. Because of this, as well as of the innate complexity of the phenomenon, only specific aspect theories are available, instead of an integrative scientific theory. The proposed computational models and methods also reflect this theoretical plurality. Due to the rapidly growing complexity, heterogeneity, dynamics, and adaptiveness of software systems, validation methodologies have reached the most advanced level in this domain of interest (Asadollahi et al., 2009). Validation in the design stage goes together with predicting limitations, while validation in run-time needs sophisticated self-validation approaches. While the previous disciplinary studies focused on understanding human decision making, automated validation shifts the attention to replicating human decision making. Automated validation involves computational decision making, which generally follows different principles and relies on the models of human decision making. The study of recognition of human behaviours still obtains more attention than reproduction of human behaviour.

Obviously, validation with agents that replicate human decision-making processes completely differs from validation of agent-based systems. While the latter has been addressed in the literature for a rather long time, fewer publications addressed the methodological and computational issues of the former. The importance of automatically predicting and reproducing human decision-making is high in the development of intelligent and automated computer (e.g., games, recommender, assistance, robotics, driving, and security) systems. The overwhelming majority of the implemented human decisional models have been created for a particular (given) purpose, and it is difficult to extrapolate from these specific contexts. Agent technology is deemed the most potent enabler of equipping computational entities with individual or collective decision making capabilities (Grundy et al., 2005). In the literature, the most frequently discussed types of decision making agent are: (i) predictive, (ii) negotiating, (iii) problem solving, (iv) voting, (v) acting, (vi) cooperating, and (vii) security agents. However, due to the large variation of the applications and the specific expectations, there are no royal ways or generic solutions, only best practices (Mudengudi & Kakkasageri, 2022).

3. Synthetic validation agent using the resources of the active recommender framework

3.1. The reference process protocol as a knowledge bridge between the active recommender framework and the synthetic validation agent

The ARF's architecture includes seven computational mechanisms of which two were implemented in the form of testable prototypes (Tepjit, 2022). These implemented mechanisms are for real-time process monitoring and recommendation generation. Based on process monitoring, the ARF is able to detect unusual events (hesitation to choose or deviation from a purposeful design action). The core computational constituent and knowledge carrier of the ARF is the pre-defined reference process protocol (RPP). The RPP is a multi-optional computational model to (i) monitor the execution of design processes, (ii) control the selection of design actions, and (iii) identify procedural deviations/obstacles. In the RPP, the to-be-completed design actions are arranged into alternative design routes based on their specific objectives, their interfacing is based on the relationships of input and output data, and the design transformations realized by them. Accordingly, the alternative design routes are goal-directed feasible flows of design actions.

As a simple practical example, consider the design process of a computational mechanism for finding the best parking lot and parking a self-driving car there. Depending on the preferred input data (e.g., GPS location data, digital map data, scanned spatial data, etc.), the mechanism design process may include alternative design routes (different activity flows, that comprise different sequences of design

actions). For instance, if the process flow model (PFM) of the digital map data-based parking is used, then the first design action develops an algorithm for locating the car model on the map, the second design action develops an algorithm for finding a proper nearby parking lot, the third design action develops an algorithm for determining the optimal motion trajectory for the car to be parked, the fourth design action develops an algorithm for deriving the motion/steering control information for the car, and the last design action develops an algorithm to simulate the process of parking, before the actual control information is sent to the car and the parking is executed. Each of these design actions is mapped to (represented as) a computational design entity (DE).

Eventually, the RPP describes an entire design process as a composition of alternative design action flows, which constitute the stored PFMs. The design actions are captured (rendered) in the RPP as computational DEs. As elements of the computational representation of the PFMs, (i) preceding design entities, (ii) actually processed design entities, and (iii) succeeding design entities are differentiated. A design entity is a preceding one if its output data are directly used as input by one or more actually processed design entities. A design entity is a succeeding one if it receives input data from one or more processed design entities, and becomes an actually processed design entity in the computation. The method or methods by which the design actions (more precisely, the actually processed design entities) can be executed is/are assigned to the computational representation of every design entity.

The ARF is an intellectualized engineered system that combines knowledge from multiple sources. The two major sources of system knowledge are (i) the reasoning mechanism designers (RMDs) and (ii) the ARF developers (ARF-Ds) (Fig. 1). In addition, the ARF may acquire knowledge through task- or history-related machine learning. The overall explicit knowledge of the ARF-Ds, (E_k), is related to the transformation of human design knowledge into problem solving knowledge for the ARF and capturing this knowledge in a software system augmented with hardware components (e.g., sensors). A specific part of E_k is embedded in the ARF as a set of its explicit knowledge elements, (E_{ARF}). A subset of this, (E_r), is included in (i) the design entities, (ii) the design methods, and (iii) the decisional mechanisms of the RPP, respectively. Furthermore, a subset of E_r is mapped into the process flow models, PFMs (E_p) included in the RPP.

On the flip side, the explicit professional knowledge, (E_d), of RMDs is considered (Fig. 1). This includes formal and informal knowledge about design tasks, procedures, methods, tools, users, contexts, and experiences related to designing various families of ASRMs. This body of knowledge is augmented by the knowledge that the ARF system provides for the RMDs in the form of design concepts- and procedures-related knowledge. A specific body of knowledge is the tacit knowledge of the RMDs, which is difficult to capture and formalize. This is referred to as implicit knowledge, (E_{im}), in Fig. 1. The specific part of the design knowledge, which includes the knowledge elements shared by the RMDs and the ARF-Ds, is identified as (E_c). It is partly captured in the PFMs (E_p) and partly comprised by the other models of the RPP. The generated recommendations, which are synthesized based on the E_{ARF} and the E_r , convey discrete chunks of situation-dependent knowledge, (REC), to the RMDs. The proportion of the knowledge elements possessed by the synthetic validation agent (SVA) and by the RPP can in principle be quantified.

Figure 2 shows an example of visualization of an RPP as a circular knowledge graph. This representation exposes both the knowledge elements and their connectivity. The dots represent the design entities and the dashed lines represent the relationships among them. In association with Fig. 1, the area of knowledge elements carried by the RPP can be quantified by the number of coloured dots in Fig. 2. The red dots represent the common knowledge elements shared by the designer and the RPP. The yellow dots represent the concerned segments of the process flow models. The green dots are the knowledge elements shared in the recommendation by the designer. These together serve as the basis of decision-making by the SVA. The numbers associated with the bullets on the boundary circle are the identification numbers of the design entities representing the design actions in a design process.

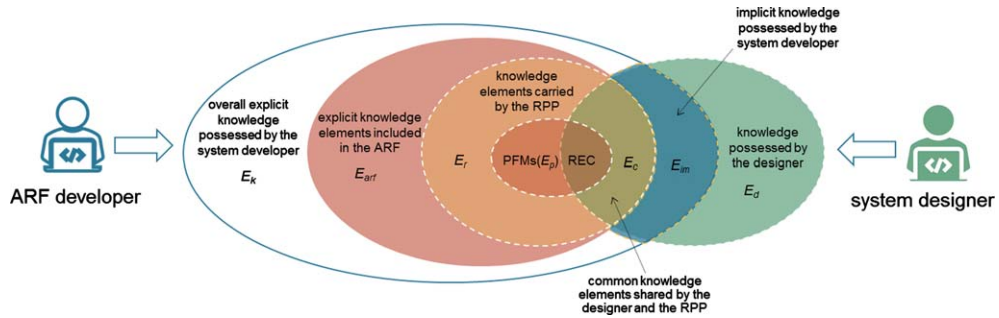


Fig. 1. Sources and forms of knowledge processed in the ARF.

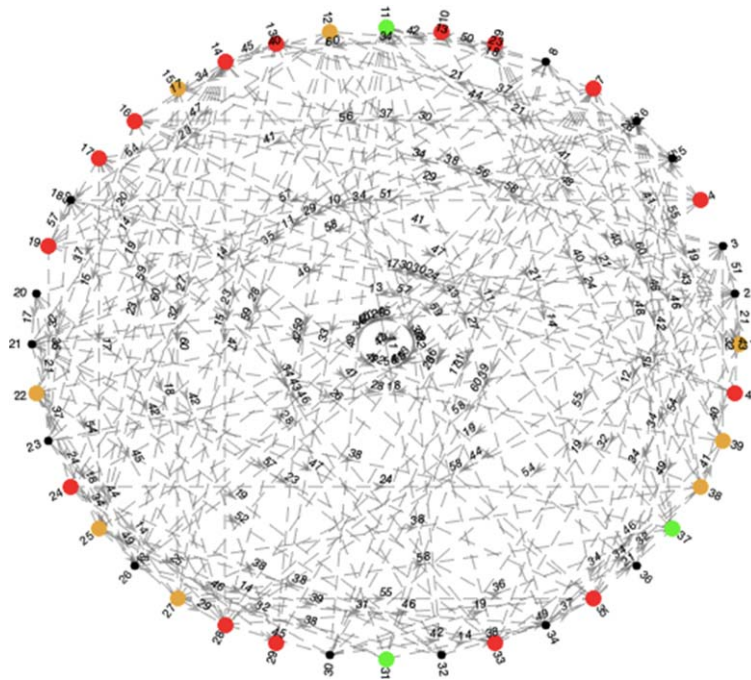


Fig. 2. Graphical representation of the connectivity of the knowledge elements included in the RPP.

The numbers associated with the lines inside the circle diagram are the identification numbers of the relations between the design entities (connectivity of the knowledge elements included in the RPP).

The RPP also serves as an imaginary bridge between the ARF and the SVA. On the one hand, the ARF uses the RPP to generate context-sensitive recommendations according to the outcome of monitoring the design process. On the other hand, the designer is supposed to have her/his own knowledge of the relevant design actions and the best execution of the concerned design process, but can receive recommendations in case of knowledge deficiency. When the designer is fully aware of a successful process, the knowledge operationalized by her/him at least equals the overall process knowledge captured in the RPP. If the designer has only limited knowledge about a successful completion of the design process, then the operationalized knowledge overlaps only with a reduced part of the knowledge captured in the RPP. This issue will be revisited in sub-sections 3.3 and 3.4.

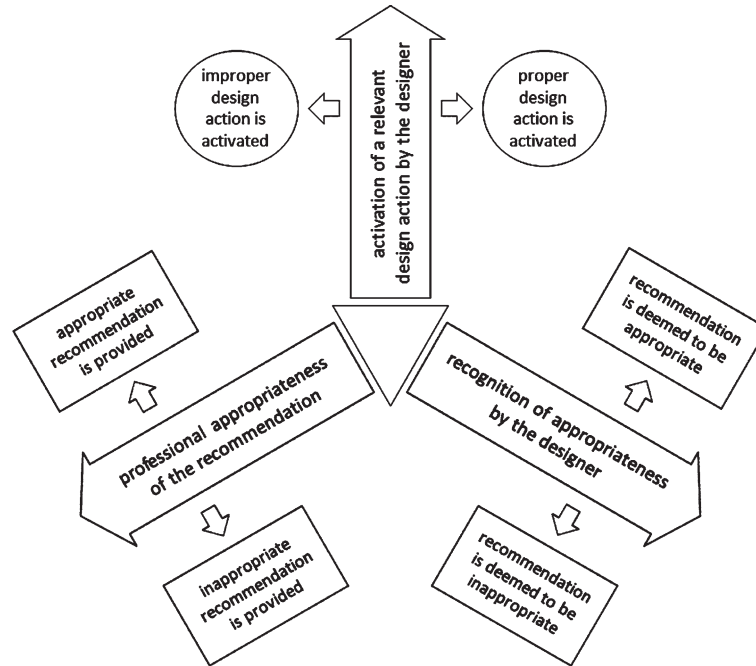


Fig. 3. Three aspects and decisional variables of common decisional behaviours.

3.2. Decisional logic: a simplified model of the decisional behaviour of humans

In the design process, the designer chooses a probable design action based on its (intuitively or rationally) assumed appropriateness and practicality. If no choice, or a wrong choice is made, then the information conveyed by a recommendation may be instrumental for the designer to arrive at a solution. However, the acceptance or rejection of the provided recommendation depends on the decisional behaviour (subjective judgement) of the designer. Therefore, a primary requirement for the SVA is to model the decisional behaviour of designers with sufficient fidelity. Towards this end, first of all, we need to model how designers may logically come to a decision on the acceptance or rejection of an obtained recommendation. In other words, the key question is what they consider at making a decision on a possible design action?

In general, the recommendation obtained from the ARF should be appropriate, but the recommendation generation mechanism is not based on any absolute knowledge and full context awareness, and there may be multiple choices with regard to the most promising design action. For the sake of practicality, let us assume that one appropriate recommendation is provided and focus on the possible decision of the designer. As shown in Fig. 3, it may be influenced by three interplaying decision variables: (i) the factuality of the appropriateness of the recommendation provided by the ARF, (ii) the view and decision of the designer on the given recommendation, and (iii) the way of getting to the next design action believed by the designer. The interpretation of these decisional variables can be seen in Fig. 3. The possible logical relationships of these decisional variables result in various decisional situations, as discussed below.

Mentioned above, the ARF generates individual recommendations based on a finite set of procedural and context information. For this reason, a recommendation may be not completely appropriate, and even inappropriate in complicated cases, for decision making. Interestingly, in spite of its trueness, a proper recommendation can be rejected and an improper recommendation can be accepted as true. The

Table 1
Decisional options of the designer

Aspects of evaluation		Recommendation provided by the ARF			
		Appropriate recommendation		Inappropriate recommendation	
		Reception by the designer			
		Accepted	Rejected	Accepted	Rejected
Decision on design action selected	Proper design action	Justified objective decision type I JOD-I (+,+,+)	Unjustified subjective decision type I USD-I (+,-,+)	Unjustified subjective decision type II USD-II (-,+,+)	Unjustified subjective decision type III USD-III (-,-,+)
	Improper design action selected	Incorrect subjective decision type I ISD-I (+,+,-)	Incorrect subjective decision type II ISD-II (+,-,-)	Incorrect subjective decision type III ISD-III (-,+,-)	Justified objective decision type II JOD-II (-,-,-)

recognition if the recommendation is appropriate or not in a given decisional situation is left on the designer's mental model, case knowledge, and subjective judgment. It must also be taken into account that, in the end, the designer may end up with the right design action based on her/his comprehension of the design task and methodological/procedural knowledge, and not only based exclusively on the received recommendation. This situation is referred to as an arbitrary proper selection (APS). The chance of an APS cannot be downgraded relative to a recommendation-based proper selection.

Based on the consideration of the above facts, a decisional behaviour model has been constructed. It is represented as a matrix in Table 1. The designer's decisional options can be generated by a systematic combination and evaluation of the above decisional variables. In the end, the decisional behaviour model captures eight decisional options concerning the utilization of a recommendation. The possible combinations of the eight options imply three classes of decisions: (i) justified objective decision (JOD); (ii) unjustified objective decision (UOD); and (iii) incorrect objective decision (IOD). JOD may be a positively justified objective decision (JOD-I) and a negatively justified objective decision (JOD-II). Both USD and ISD may have three types of decisional options according to the possible combination of the logical options. APSs are not at all considered in this decisional behaviour model because of their unpredictability.

Human decisional behaviour is captured in a relatively simple scheme according to the above logic of reasoning. A JOD-I (+,+,+) type decision means that an appropriate recommendation is received by the designer, who accepts it, and selects a proper design action based on the content of the recommendation, while a JOD-II (-,-,-) means that an inappropriate recommendation is received by the designer, who rejects it, but cannot find a proper design action without asking for other recommendation. A USD-I (+,-,+) means that an appropriate recommendation is received by the designer, who rejects it, but - contrary to this - finds a proper design action independent of the content of the recommendation. A USD-II (-,+,+) means that an inappropriate recommendation is received by the designer, who accepts it, but - for a certain reason - finds a proper design action independent of the content of the recommendation. A USD-III (-,-,+) means that an inappropriate recommendation is received by the designer, who rejects it, and - contrary to this - intuitively finds a proper design action independent of the content of the recommendation. An ISD-I (+,+,-) means that an appropriate recommendation is received by the designer, who accepts it, but - for a certain reason - selects an improper design action. An ISD-II (+,-,-) means that an appropriate recommendation is received by the designer, who rejects it, but - for

a certain reason - selects an improper design action. An ISD-III (-,+,-) means that an inappropriate recommendation is received by the designer, who accepts it, but - contrary to this - selects an improper design action.

Approaching it from the side of decisional rationality, a justified objective decision of type I or II happens when the belief of the decision maker and the observable facts meet (are congruent). This is also referred to as a truly-factual decision. An unjustified subjective decision of type I, II or III happens when the designer believes that something is not true in a given context, but in fact it is. This is also referred to as a para-factual decision. An incorrect subjective decision of type I, II or III happens when the designer believes that something is true in a given context, but in fact it is not. This is also referred to as a contra-factual decision. This set of options of the decision-making logic lent itself to a formal logic that was applied in computational implementation of the SVA, but it had to fulfil some other operational requirements that are discussed below.

3.3. Decisional knowledge: a reference process protocol as a multi-model construct

From a computational point of view, the RPP is a prescriptive model of design processes and integrates three constituents, formally:

$$RPP = TAM \cup PFM \cup DTM \quad (1)$$

where: TAM is a timed design actions model, PFM is a process flow model, DTM is a decision tree model, and \cup is symbolic union. Let α_i be a design action, DA a finite, non-zero set of design actions, and $\alpha_i \in DA$. Furthermore, let t_s be the start point in time of $\alpha_i \in DA$, t_e the end point in time of α_i , and m a finite, non-zero set of the alternative methods of computational execution of α_i . Let us denote the timed design actions as a_i^t . With these parameters, the TAM can be specified formally as:

$$TAM = \left\{ \alpha_i^t, t_s, t_e, m_j \left(P_{i,k}^{j,in}, P_{i,l}^{j,out} \right) \right\} \quad (2)$$

where: $P_{i,k}^{j,in}$ is a finite, non-zero set of k input parameter values related to a particular timed design action a_i^t and method m_j , $P_{i,l}^{j,out}$ is a finite, non-zero set of l output parameter values related to a particular timed design action a_i^t and method m_j , and $m_j \in \mathcal{M}$. For computation, the TAM is represented as a split (upper diagonal) matrix, which arranges the design actions a_i^t according to their temporal sequence. The TAM captures the input parameters and output parameters (computational variables) for each design action.

A PFM is a state-transition model of a design process, including a finite, non-zero set of the process flow elements, which represent the design actions a_i^t as computational transitions and the input and output states of the design actions represented by the (quantitatively or qualitatively evaluated) values of the design parameters (variables). For the purpose of computation, a PFM is represented as a Petri-net-like structure. It is a bipartite directed multigraph consisting of two types of nodes: (i) the set of transitions, associated with design transitions (changes), T_α , and (ii) the set of states of design actions, S_α . With these dispositions, a PFM is expressed formally as:

$$PFM = \{T_\alpha, S_\alpha, W, M\} \quad (3)$$

where: T_α is a finite, non-zero set of computational transitions related to design actions, α_i , S_α is a finite, non-zero set of process states, W is a finite, non-zero set of relations among the elements of the set T_α and the elements of the set S_α , represented as arcs of a bipartite directed multigraph process model, and M is a finite, non-zero set of markings. The PFM captures the pattern of the design actions and their expected outcomes.

A *DTM* is defined as a computational means to support computational decisions. In symbolic form:

$$DTM = \{w, \mathcal{M}_\alpha, R\} \quad (4)$$

where: α represents a design action, w is a finite set of decision variables, \mathcal{M} is a finite set of potential computational methods for an execution of a design action, and R is a finite set of decision rules. The decision tree is used to make a decision on the relevance of a computational transition, T_α , and of a computational method, $m \in \mathcal{M}$ related to design action $\alpha \in DA$. Computationally, a *DTM* uses a classifier algorithm to find a proper method for the computational execution of a design action.

In operation, the *SVA* has direct access to all pieces of information stored in the precompiled *RPPs*. It means that, in a sense, the *SVA* is aware of the progression achieved in the design process, as well as of the appropriateness of taking a particular design process flow, and choosing a design action and an execution method. Consequently, if an obstacle (a complication concerning the next step of the design activity flow) emerges, then it can be detected through the *RPP*. The computational interrogation of the *RPP* happens at the level of design entities. Figure 4 shows a fictional example of their interrelationships. The boxes drawn with dashed lines represent the design entities (i.e., the represented design actions). The contents of the *DPFs* are compiled from (i) the transitively connected (input-output data interfaced) design entities, (ii) the set of semantic rules that allow creating relationships between two or more design actions, and (iii) the set of the related decision points in the decision tree model. The green solid lines represent the connectivity of the knowledge elements in a particular design activity flow, consisting of alternative sequences of design actions/entities. The connectivity between the subsequent design entities is shown by the dashed arrows, their overall temporal relationships are indicated by their sequence, and the probabilistic nature of the relationships is expressed by the variables p_i . The circles and the shaded/rounded rectangles in the boxes represent the states and the transitions, respectively.

In principle, the *RPP* includes all alternative *DPFs* (i.e., a thorough specification of the alternatives for how to complete the design process of particular *ASRMs*). In order to make the possible design flows traceable within the *RPP*, the causal relationships of design actions are represented in the form of a *Bayesian network* (*BN*), a directed acyclic probabilistic graph. The *BN* includes a set of nodes (representing the design actions) and a set of directed edges between nodes (proper logical connections of the design actions in a *DAF*). Based on probabilistic reasoning, the *BN* provides support for making a decision about the best matching succeeding design actions. The possibility of having multiple usable design methods to execute a considered design action should also be taken into consideration. In fact, this explains why a decision tree was included in the implemented *RPP* to help select the best option.

3.4. Decisional probability: pseudo-randomization of the simulated decision making

As explained above, the *SVA* mimics the decisional behaviour of human designers concerning the utilization of the obtained recommendations. The output of the *SVA* is a data set representing the pattern/distribution of the simulated procedural decisions of the designers. This synthesized dataset is used in the separate validation process to validate the usefulness of individual recommendations in an application context. The basis of generating this dataset is consideration of the socially and emotionally influenced nature of human decision-making concerning the computationally generated recommendations. Capturing the pseudo-randomness implied features of the decisional behaviour was a challenging part of the research process. The following facts were taken into consideration in the pseudo-randomization of the simulated decision-making.

The real time process monitoring of the *ARF* provides information about the current state of the design process. This and the actual decisional situation are taken into account by the design support

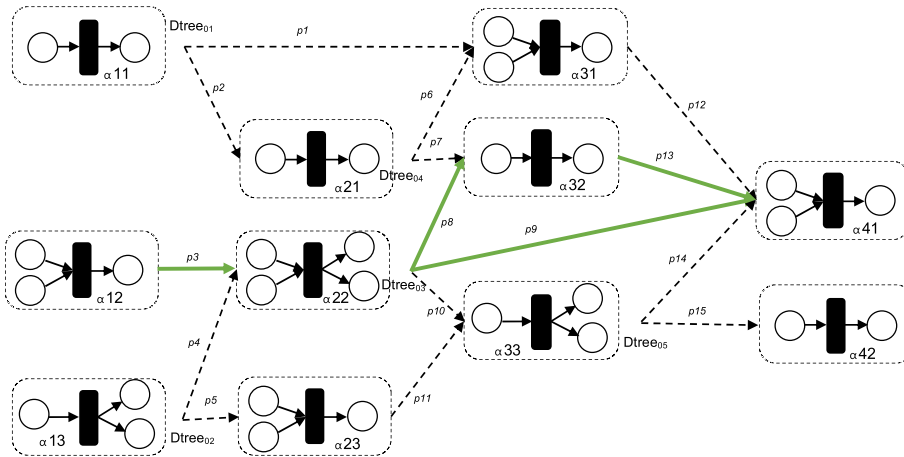


Fig. 4. Graphical representation of the connectivity of the knowledge elements of the reference process protocol.

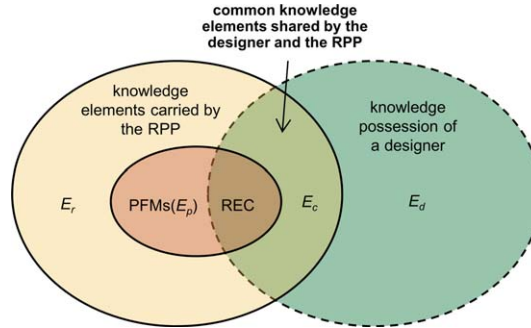


Fig. 5. The relation of the knowledge possessed by the designer and the knowledge elements processed in the RPP.

mechanism for generating recommendations. Both mechanisms rely on the information incorporated in the three above-discussed models of the RPP. However, the designer uses her/his own knowledge and selects the necessary design actions and methods based on it. The decisions made depend on (i) the knowledge the designer has about the problem at hand, (ii) the past experiences with the problem at hand, and (iii) the immediate and directly conceivable uncertainty, risk, and impact of the decision made. If the designer’s knowledge is a superset or a near-matching subset of the knowledge that is needed to solve the design tasks, then there is a higher probability of success.

In other words, in an ideal situation, the designer has the same knowledge as the body of knowledge contained in the RPP. This is visualized in Fig. 5. Consequently, we argue that the success depends on the proportion of the knowledge possessed by the designer and the knowledge included in the three models of the RPP. On the other hand, the designer’s implicit (tacit/informal) knowledge is considered not-sharable (or not-shared) knowledge. That is, the implicit knowledge shown in Fig. 1 is not considered at determining E_c . The number of emerging procedural hindrances and the total number of recommendations needed are (by-and-large) proportional to the shared knowledge. When it comes to accepting or rejecting a given recommendation, the relationship of these two bodies of knowledge plays a crucial role.

Based on the above assumptions, the concept of *decisional modes* (DMs) was introduced as an aspect of modelling decisional behaviour by the SVA. A decision mode is associated with the amount of possessed knowledge and interpreted as the decisional potential that an individual designer has

based on this knowledge (without sharing knowledge, communicating with others, or searching for any further knowledge). DMs help represent the experience level of designers (e.g., junior, skilled, proficient). Computationally, DMs can be quantified as the (actual or assumed) proportion of the knowledge shared by the designer and the RPP. The common knowledge, denoted by (c_k), is formally defined as follows:

$$E_c = E_r \cap E_d \quad (6)$$

$$c_k = n(E_c) / n(E_r) \quad (7)$$

where: E_c is a finite, non-empty set of the common design entities cognitively shared by the designer and the RPP, E_r is a finite, non-empty set of design entities included in the RPP, E_d is a finite, non-empty set of design actions which are supposed to be known by the designer, and c_k is the proportion common knowledge, $0 < c_k < 1$, $n(E_c) \in \mathbb{Z}$ is the total number of design entities shared, and $n(E_r) \in \mathbb{Z}$ is the total number of design entities included in the RPP, $n(E_r) > 0$.

In a general practical case, it is very unlikely that a 100 percent coincidence exists with regard to the knowledge of the designer and the knowledge contained in the RPP. It is also very unlikely that there is a 0 percent coincidence between them in a given project because, in this case, the designer does not have enough competence to deal with the problem. Therefore, the statistical significance of coincidence was considered through the introduction of the *statistical significance parameter*, p , to characterize the related probabilities. With this, the three (most likely) decisional modes, Δ_i , were defined as a function of c_k in the following manner:

$$\Delta_i = \begin{cases} \Delta_1, 0.05 < c_k \leq 0.25 \\ \Delta_2, 0.25 < c_k \leq 0.75 \\ \Delta_3, 0.75 < c_k \leq 0.95 \end{cases} \quad (8)$$

where:

- Δ_1 : In the case of a less competent designer, the proportion of the common knowledge, c_k , is assumed to be between 0.05–0.25. Accordingly, the designer possesses insufficient knowledge to judge the appropriateness of the received recommendation, and she/he has no other options to continue the design process. In this decision mode, there is a high probability that the recommendation will be accepted. The probability will decrease slightly when the ratio of the shared knowledge increases.
- Δ_2 : In the case of an intermediary designer, the proportion of the common knowledge, c_k , is assumed to be between 0.25–0.75. The designer may recognize if the recommendation is appropriate to eliminate the obstacle in the design process, but she/he may also have other options based on her/his experiences with the to-be-completed design process. In this decision mode, the designer may hesitate to accept a recommendation that she/he is not familiar with. Using a familiar design action and method may seem to be a better choice for the designer. Therefore, the probability of accepting the received recommendation is getting lower.
- Δ_3 : In the case of a highly competent designer, the proportion of the common knowledge, c_k , is assumed to be between 0.75–0.95. The designer possesses the necessary and sufficient knowledge to complete the design process and to assess the appropriateness of the incidentally received recommendation. In this decision mode, the designer also recognizes how to operationalize the recommendation. Consequently, there is a high probability of accepting the recommendation.

Concerning Equation 8, two other decisional modes would be Δ_0 , that is $0 < c_k \leq 0.05$, and Δ_4 , that is $0.95 < c_k \leq 1$. However, due to the statistical significance of their occurrence, it is unnecessary to consider these two modes in practical cases. Notwithstanding, we introduced three formulas to capture the patterns of decisional behaviours, or more specifically, to express: (i) the probability of the acceptance of a recommendation, $p(aR)$, (ii) the probability of shared knowledge, $p(s_k)$, and (iii) the probability of knowing the knowledge elements included in the recommendation by the designer, $p(f_k)$. These formulas are as follows:

$$p(aR) = f(s_k, f_k) \quad (9)$$

$$p(s_k) = n(E_p \cap E_c) / n(E_p \cup E_c) \quad (10)$$

$$p(f_k) = n(E_p \cap E_c) / n(E_c) \quad (11)$$

where: $p(aR_i)$ is the probability of the acceptance of a recommendation, E_p is a finite, non-empty set of design entities included in the design process or in a concerned segment of it, and E_c is as specified by Equation (6). The function $p(aR)$ reflects the possible deviation of the probability function of the shared knowledge, $p(s_k)$, from the probability function of the knowledge elements known by the designer, $p(f_k)$. Mathematically:

$$p(aR) = 1 - \alpha * (\exp(\beta * p(s_k)) - \exp(\gamma * p(f_k))) \quad (12)$$

where: α is the expected probability of accepting a recommendation, and $0 \leq \alpha \leq 1$, β is a factor (coefficient) determining the acceptance level, and γ is the expected acceptance level, and $\gamma < \beta < 0$.

Taking the possible decisional situations and the probability formulas into consideration, a so-called *decisional model* was constructed. Visualized in Fig. 6, the exponential model captures the correlation between the acceptance probability of the recommendations and the probability of shared knowledge with regard to the three decisional modes. Conceptually, the decisional model is based on the assumptions that concern the knowledge shared by the agent-simulated designer and the ARF. The exponential model captures the non-linear relationship between the probability of acceptance of the recommendation and the shared knowledge concerning the (defined) decision modes. However, this is one constituent used to define the decisional behaviours of an SVA. This model is yet not sufficient to compute the decisional behaviours reproduced by the SVA, because they are also influenced the E_c , which is quantified as proportion of the number of design entities (actions) that are known by the designer and the number of those known by the RPP. The knowledge elements carried by the recommendations, (REC), do not influence this. The numeric evaluation of the decisional model provides a pattern of the decisional behaviour concerning the acceptance or rejection of recommendations.

Another issue is the tendency of decision making of the designer with regard to the properness or improperness of the recommendations. The tendency of decision making was specified by the following conditions:

$$tdm_i = \begin{cases} 1, & n(E_x \cap E_c) \geq 1 \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where: tdm_i is the decision tendency of the designer, and $tdm_i \leftarrow \{0, 1\}$, where (1) means a proper recommendation, and (0) means an improper recommendation, E_x is a finite, non-zero set of design entities that are included in the extended design action flow, and E_c is a finite, non-zero set of design entities (actions) related to the design process, which are supposed to be known by the designer. The principle of prognostic reasoning was used to predict the possible distribution of the decision options under the specified conditions (Oh et al., 2013). To determine the values of the above defined

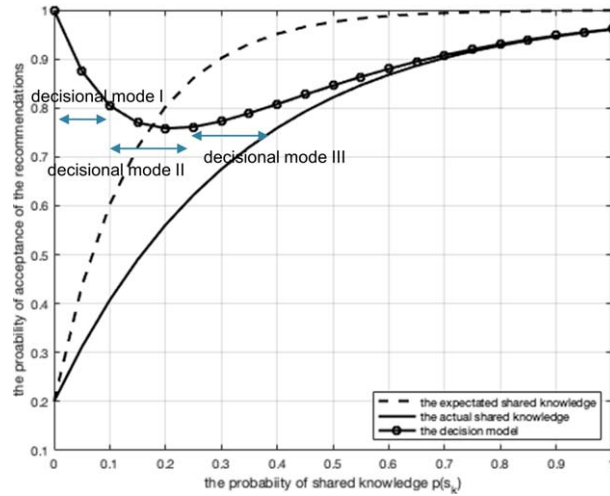


Fig. 6. The decisional model used for the construction of the SVA and the captured relations.

coefficients (factors) of the reference decision model, we set up design scenarios and numerically analysed the outcomes. Furthermore, to scrutinize the rationality and the reliability of the developed decisional model, a comprehensive logical examination was completed by doing a large number of simulations.

4. Functionality and testing of the synthetic validation agent in application

4.1. Functionality of the implemented synthetic validation agent

The SVA was developed (i) to model the decisional behaviour of RM designers in terms of recognition of the appropriateness of the proposed recommendations, (ii) to characterize the decisional behaviour of all involved RM designers, and (iii) to generate robust datasets concerning the various decisional behaviour options. The synthesized datasets were used in the research project to validate the usefulness of the individual recommendations in various design action flows. Thus, the overall function of the SVA was formulated as a simulation of the designers' decisional behaviour. This overall function of the SVA was decomposed into two main functions: (i) simulate decisional behaviours, and (ii) generate validation dataset. The main function (i) was further decomposed to three sub-functions: (a) F1.1. - simulate the acceptance probability, (b) F1.2. - predict the features of decisional options, and (c) F1.3. - simulate decision tendency. The main function (ii) was decomposed also to three sub-functions: (a) F2.1. - generate a data model to represent variation of obstacles in the design process, (b) F2.2. - enact the model to generate recommendations, and (c) F2.3. - predict the patterns of decisional options.

The sub-functions were transferred to one or more computational algorithms. The algorithms needed for the implementation of these sub-functions are listed in Fig. 7. Further details about the computational realization can be found in (Tepjit, 2022). The objective of testing the SVA was the proper implementation of the conceptualized functionality. The software testing included the checking if (i) the specified functional requirements are fulfilled, and (ii) the outputs of the related algorithms are according to the expectations.

The functions F1.1 and F2.3 have been considered as the critical functions. The former should be tested to see if the patterns of decision modes are logically aligned to the assumptions or not. The latter should be tested because it provides the patterns of decision options according to the different decision

functions		needed algorithms
F1.0: simulate decisional behaviours of the designer		
F1.1	A1.01:	generate data model to represent the common knowledge of the designer
	A1.02:	calculate the probability of shared knowledge elements
	A1.03:	calculate the probability of known knowledge elements
	A1.04:	calculate the probability of acceptance of the recommendations
F1.2	A1.05:	predict the features of decision behaviours
F1.3	A1.06:	identify the follow up design entities
	A1.07:	simulate decisional tendency concerning the design process
F2.0: generate a validation dataset		
F2.1	A2.01:	generate data model to represent variation of obstacles in the design process
	A2.02:	generate the recommendations
F2.3	A2.06:	classify the patterns of decisional behaviours
	A2.07:	calculate the probability of patterns of decisional behaviours

Fig. 7. Implementation of the sub-functions by algorithms.

modes. However, it must be mentioned, other functions are also required to provide input data to the successive functions. The operationalization of the SVA provides the data for the overall validation of the usefulness of recommendations. The expected outcomes are the correlations between the decision modes and the usefulness of recommendation. In the following sub-sections, the objectives and results of computational testing of four sub-functions are presented.

4.2. Testing sub-function F1.1

Concerning this sub-function, the requirement was that the non-linear relationships of the acceptance probability of the recommendations and the proportion of the common knowledge should be correctly reflected by the decision model. To check the fulfilment of this requirement, three scenarios were tested, in which different numbers of design entities, 50, 100, and 200, were included in the RPP and the effect of the three decisional modes were investigated. The results showed characteristic trends. In decision mode I (i.e., in the range from $c_k = 0.05$ to $c_k = 0.25$), the probability of acceptance decreased dramatically when the proportion of common knowledge increased. In decision mode II, (i.e., in the range from $c_k = 0.25$ to $c_k = 0.75$), the probability of acceptance was slightly decreased and reached the lowest value at $c_k = 0.5$. After that, the trend of the acceptance probability showed a positive relationship pattern with the proportion of common knowledge. In decision mode III, (i.e., in the range from $c_k = 0.75$ to $c_k = 0.95$), the acceptance probability steadily increased as the proportion of common knowledge of the designer was supposed to increase. Shown in Fig. 8, the results of this computational experiment indicated that the computed patterns of the acceptance levels were consistent with the expectations in the case of each scenario.

4.3. Testing sub-function F1.2

The basic requirement for sub-function F1.2 was to have the highest possible performance in terms of an aggregated probability of the acceptance of appropriate recommendations or the rejection of inappropriate recommendations. To determine the probability of the features of the decisional options, three independent variables were used in the testing scenarios: (i) the total number of entities included

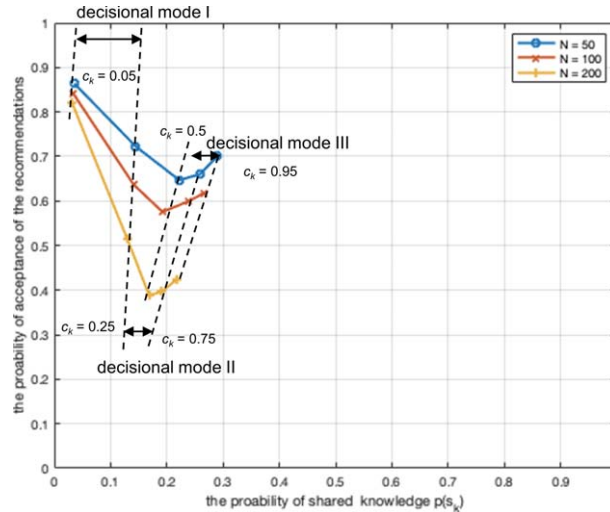


Fig. 8. The patterns of the recommendation acceptance probability in relation with the probability of shared knowledge.

in the RPP, (ii) the accuracy rate of the recommendation generation, and (iii) the proportion of common knowledge. Actually, the first two variables were regarded as the constant parameters in the test, and, as such, were set to 100 entities and to 75 % of the accuracy rate, respectively. We assumed that the accuracy rate represented the probability of the appropriate recommendations. The third variable was set to discrete values of 0.05, 0.25, 0.5, 0.75 and 0.95, respectively, representing the five levels of common knowledge.

The aggregated probability of the four features (accepted-appropriateness, rejected-appropriateness, accepted-inappropriateness, and rejected-inappropriateness) of the decision options were numerically simulated for all scenarios. The results shown in Fig. 9 indicate that the occurrence of the different patterns (features) of the decision options happened according to their aggregated probabilities. Mimicking a particular type of designer, the SVA intended to accept all recommendations in the case of decision mode I. When a 75 % accuracy rate was applied, the proportion of the ‘accepted-appropriate’ and the ‘accepted-inappropriate’ decisions was 0.75 and 0.25, respectively. Likewise, the acceptance probabilities of the recommendations in decision mode II reflect the assumed shared knowledge. At $c_k = 0.95$, the acceptance probabilities complied with decision mode III. This allowed us to conclude that the requirement posed for $F_{1,2}$ was fulfilled.

4.4. Testing sub-function $F_{1,3}$

The requirement for sub-function $F_{1,3}$ was providing correct information about the aggregated probability of selecting the proper/improper recommendations by the SVA. To determine the probability of the properness of the recommendations, three independent variables were taken into account: (i) the total number of design entities included in the RPP, (ii) the proportion of the shared design entities, and (iii) the process flow model with the highest joint probability value. The joint probability distribution of the process flow models was determined by considering three of the case-related recommendations and N number of succeeding design entities.

The investigation focused on the correlations with the aggregated probability of the proper recommendations. As in the previous test, the proportion of the assumed common knowledge was set to the five levels (i.e., 0.05, 0.25, 0.5, 0.75, and 0.95). The assumed-to-be-known knowledge elements of

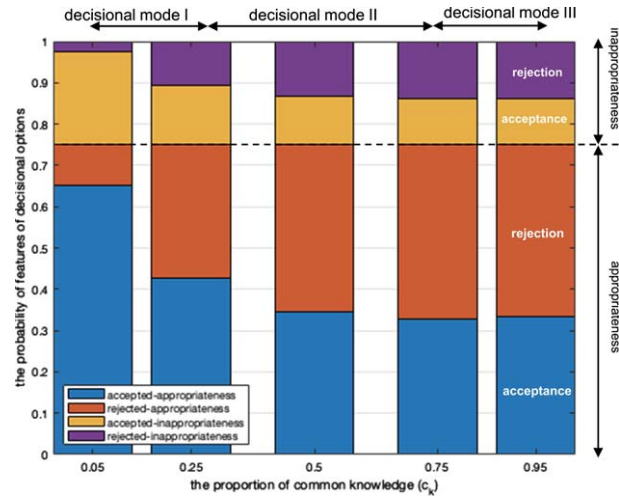


Fig. 9. The feature patterns of the decisional options according to the levels of the common knowledge.

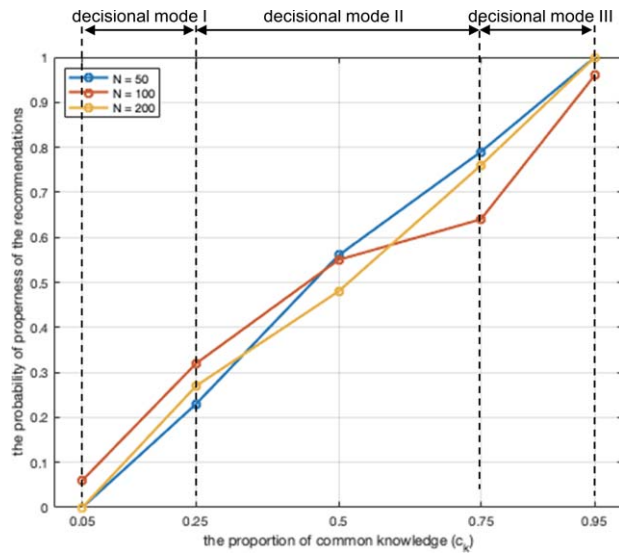


Fig. 10. Correlations between the probabilities of the properness of the recommendations and the proportion of common knowledge.

the SVA were generated randomly by varying the proportion of common knowledge. Three scenarios, including different numbers (namely: 50, 100, and 200) of design entities in the RPP, were tested. For each scenario, 100 recommendations were generated by simulation runs. The obtained results concerned the aggregated probability of the combined decisional tendency of the SVA for the different proportions of the common knowledge. They are presented in Fig. 10 for each of the three scenarios. It can be noticed that there are strong positive relationships between the proportion of common knowledge and the probability of recognizing the properness of the recommendations. In fact, the strong positive relationships represent linear trends for each decisional mode and each scenario. That is, the two algorithms used for realizing F1.3 produced the output as per expectation.

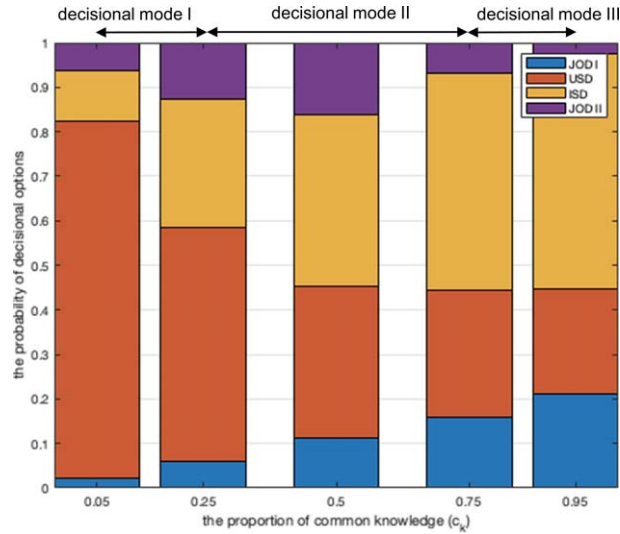


Fig. 11. The distribution patterns of the decisional options according to the proportion of the common knowledge.

4.5. Testing sub-function F2.33

The requirement for the realization of sub-function F2.33 was to make a reliable prediction for the probability of the four decisional options concerning the recommendations. The two algorithms developed to implement sub-function F2.33 had to classify the actual pattern of decision options. Towards this end, the classification had to combine all the data that were needed for the characterization of the decisional behaviours. These include: (i) the data on the recognition of the appropriateness of recommendations, (ii) the data on the features of the decision options, and (iii) the data on the properness of the recommendations. Like in the preceding tests, five scenarios were defined with different proportions of the assumed common knowledge (0.05, 0.25, 0.5, 0.75 and 0.95, respectively).

When the shared knowledge was set to low in the simulation (case of $c_k = 0.05$), the outcomes were according to decision mode I. That means, all recommendations were accepted, but the SVA (meaning: the simulated designer) had insufficient knowledge to evaluate the appropriateness of the recommendations. Thus, the pattern of the decisional options displayed a high proportion of unjustified subjective decisions. When the shared knowledge was set to ordinary (case of $c_k = 0.25$ to $c_k = 0.75$), the patterns of the decisional options were according to decision mode II. Finally, when the shared knowledge was set to high (case of $c_k = 0.95$), the pattern followed decision mode III. Figure 11 displays the results obtained for the probabilities of the decision options. The bars indicate the distributions for each of the five proportionate levels of common knowledge. These results confirmed that the realized algorithms could fulfil the requirement posed to the sub-function F2.33.

5. Reflections and conclusions

The need for conceptualization, implementation, and testing of a synthetic validation agent (SVA) was raised by the recent pandemic, which prevented involving groups of human designers in the recommendation testing process. The fact of the matter is that the concept of SVA was introduced as the surrogate of designers taking part in observational research studies and focus group sessions. Consequently, the reported research had four practical goals: (i) development of the logical fundamentals and computational resources for the SVA, (ii) modelling the decisional behaviour of designers, (iii)

computational implementation of the functionality of the SVA, and (iv) application of the SVA in data generation for validation of the usefulness of recommendations.

As our literature study explored, the most frequently discussed types of decision-making agent are: (i) predictive, (ii) negotiating, (iii) problem solving, (iv) voting, (v) acting, (vi) cooperating, and (vii) security agents. Agent technology has also been studied as a framework for modelling and as an evolving enabler of computational realization of software entities with individual or collective decision-making capabilities. However, using a synthetic validation agent for (i) modelling decisional behaviours of reasoning mechanism designers, and (ii) generating probabilistic predictive data is a novel issue and has not been addressed extensively yet.

Our conclusion is that smart agent technologies have a high potential to replace human participants in repetitive assessment processes even under regular circumstances. Such programmable surrogates can mimic (i) creative activities; (ii) decision-making actions; and (ii) social behaviour of human designers. Current research knowledge allows achieving high-fidelity multi-feature surrogates. Dedicated, agent-centred validation methodologies and methods need to be developed as well as methods and tools for testing the operation of synthetic validation agents.

Understanding and replication of human reasoning and decision-making processes are the starting points in the development of an SVA. Social science, cognitive psychology, software engineering, marketing, and many other disciplines have investigated the cognitive processes associated with making choices and decisions. Both conceptual behavioural frameworks and computational behavioural models have been proposed. Behavioural frameworks (purposefully arranged and interrelated concepts) apply abstractions to capture the essence of the phenomena observable in the real-world. Computational behavioural models are realized to replicate somewhat uncertain and unexplained decision mechanisms in application contexts.

The development process of the SVA was challenging due to the complexity of the natural cognitive mechanisms, the strong dependence on intellectual and affective factors, the vague and non-linear nature, and the limitations related to prediction and forecasting. No exact logical procedures or explicit mathematical equations can be constructed to capture all constituents, influencing factors, and characteristics. On the other hand, the probabilistic nature of these should be taken into account in modelling.

Based on the above considerations, the SVA was based on a probabilistic decisional model that quantifies decisional options according to the assumed decisional conditions and tendencies. The three key concepts underlying the SVA are (i) decisional logic, (ii) decisional knowledge, and (iii) decisional probability. These together enable generation of reliable data about the decisional behaviours of human designers concerning the obtained recommendations. The completed tests proved the above assumption, though the feedback on the goals/actions and the adaptations in the decisional behaviour has not been considered.

The SVA is in a direct informational relationship with the reference process protocol (RPP), which is the principal means of the intellectualization of the active recommender framework. The RPP includes multiple design process flows that are purposeful arrangements of design entities and design methods. The shared knowledge, quantified in terms of the proportion of the design actions/entities included in the RPP and known by the designers, played an important role in the simulation of the probable decisional behaviours and in the compilation of the datasets for the validation of the usefulness of the individual recommendations. A key issue is how to determine the actual proportion of the common knowledge based on empirical facts, rather than on logical and experimental assumptions. In this context, other methods could be considered in future research (Ayal & Hochman, 2009).

As a viable alternative of the probabilistic logical process modelling-based approach of SVA development, deep learning- or transfer learning-based approaches might be used. Like a probabilistic decisional model, trained deep neural network models could be used as surrogates of human designers

and to generate data on decisional behaviour for validation of the usefulness of the ARF generated recommendations in various situations. It is also an interesting research question if the self-validation hypothesis could be extended to SVAs? (Petty et al., 2002).

References

- Asadollahi, S., Rafe, V., & Rafeh, R. (2009). Towards automated software verification and validation. In: *Proceedings of the 2009 International Conference on Computer Technology and Development*, Vol. 1, IEEE, 206-210.
- Ayal, S., & Hochman, G.U. (2009). Ignorance or integration: The cognitive processes underlying choice behavior. *Journal of Behavioral Decision Making*, 22(4), 455-474.
- Balke, T., & Gilbert, N. (2014). How do agents make decisions? – A survey. *Journal of Artificial Societies and Social Simulation*, 17(4), 13-43.
- Burciu, A., & Hapenciu, C.V. (2010). Non-rational thinking in the decision-making process. In: *Proceedings of the European Conference on Intellectual Capital*, 2, 152-160.
- Caire, G., Cossentino, M., Negri, A., Poggi, A., & Turci, P. (2004). Multi-agent systems implementation and testing. In: *Proceedings of the Agent Technology to Agent Implementation Symposium*, 1-10.
- Choi, J., & Choi, B. (2002). Test agent system. *International Journal of Software Engineering and Knowledge Engineering*, 12(03), 269-290.
- Chung, H.M., & Silver, M.S. (1992). Rule-based expert systems and linear models: An empirical comparison of learning-by-examples methods. *Decision Sciences*, 23(3), 687-707.
- Creyer, E.H., Bettman, J.R., & Payne, J.W. (1990). The impact of accuracy and effort feedback and goals on adaptive decision behavior. *Journal of Behavioral Decision Making*, 3(1), 1-16.
- Dastani, M., Hulstijn, J., & van der Torre, L. (2001). BDI and QDT: A comparison based on classical decision theory. In: *Proceedings of the AAAI Spring Symposium on Game Theoretic and Decision Theoretic Agents*, 16-26.
- Dorer, K. (2004). Extended behavior networks for behavior selection in dynamic and continuous domains. In: *Proceedings of the ECAI Workshop on Agents in Dynamic Domains*, Valencia, Spain, 1-6.
- Engel, A. (2010). Verification, validation, and testing of engineered systems (Vol. 73). John Wiley & Sons, Hoboken, N.J.
- Friess, N., Crawford, H., & Aycok, J. (2009). A multi-agent approach to testing anti-spam software. In: *Proceedings of the 20th International Workshop on Database and Expert Systems Application*, IEEE, 38-42.
- Frolik, J., Abdelrahman, M., & Kandasamy, P. (2001). A confidence-based approach to the self-validation, fusion and reconstruction of quasi-redundant sensor data. *IEEE Transactions on Instrumentation and Measurement*, 50(6), 1761-1769.
- Gibson, W., & Powell-Evans, K. (1998). Validation fundamentals: How to, what to, when to validate. CRC Press. Boca Raton, 1-248.
- Grundy, J., & Ding, G. (2002). Automatic validation of deployed J2EE components using aspects. In: *Proceedings of the 17th International Conference on Automated Software Engineering*, IEEE, 47-56.
- Grundy, J., Ding, G., & Hosking, J. (2005). Deployed software component testing using dynamic validation agents. *Journal of Systems and Software*, 74(1), 5-14.
- Guerini, M., & Moneta, A. (2017). A method for agent-based models validation. *Journal of Economic Dynamics and Control*, 82, 125-141.
- Huang, J., Wu, W., Zhang, Z., & Chen, Y. (2020). A human decision-making behavior model for human-robot interaction in multi-robot systems. *IEEE Access*, 8, 197853-197862.
- Jager, W., & Janssen, M. (2012). An updated conceptual framework for integrated modeling of human decision making: The Consumat II. In: *Proceedings of the Workshop Complexity in the Real World@ECCS*, 1-18.
- Jain, S., & Patel, A. (2021). Semantic contextual reasoning to provide human behavior. Retrieved from <https://arxiv.org/abs/2103.10694>.
- Kant, J.D., & Thiriot, S. (2006). Modeling one human decision maker with a multi-agent system: The CODAGE approach. In: *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, 50-57.
- Kim, C.N., Yang, K.H., & Kim, J. (2008). Human decision-making behavior and modeling effects. *Decision Support Systems*, 45(3), 517-527.
- Klügl, F. (2008). A validation methodology for agent-based simulations. In: *Proceedings of the Symposium on Applied Computing*, ACM, 39-43.
- Kumar, A., & Goyal, V. (2012). EARCT - Environment for automated rank based continuous agent testing. *International Journal of Computing*, 11(3), 180-190.
- Kumaresan, P.P., Frasher, M., & Enoiu, E.P. (2020). Agent-based software testing: A definition and systematic mapping study. In: *Proceedings of the 20th International Conference on Software Quality, Reliability and Security Companion*, IEEE, 24-31.
- Kwasniewski L. (2009). On practical problems with verification and validation of computational models. *Archives of Civil Engineering*, 55(3), 323-346.
- Maes, P. (1989). The dynamics of action selection. In: *Proceedings of the International Joint Conference on Artificial Intelligence*, 991-997.
- Manzoor, U., Irfan, J., & Nefti, S. (2011). Autonomous agents for testing and verification of softwares after deployment over network. In: *Proceedings of the 2011 World Congress on Internet Security*, IEEE, 36-41.
- Meng, Y., Qasem, S.N., & Shokri, M. (2020). Dimension reduction of machine learning-based forecasting models employing principal component analysis. *Mathematics*, 8(8), 1233, pp. 1-15.

- Moghadam, M.H. (2019). Machine learning-assisted performance testing. In: *Proceedings of the 27th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ACM, 1187-1189.
- Moghadam, M.H., Hamidi, G., Borg, M., Saadatmand, M., Bohlin, M., Lisper, B., & Potena, P. (2021). Performance testing using a smart reinforcement learning-driven test agent. arXiv preprint arXiv:2104.12893.
- Monschein, D., Mazkatli, M., Heinrich, R., & Koziolok, A. (2021). Enabling consistency between software artefacts for software adaption and evolution. In: *Proceedings of the 18th International Conference on Software Architecture*, IEEE, 1-12.
- Mudengudi, S.S., & Kakkasageri, M.S. (2022). Cooperative agent-based location validation for vehicular clouds. In: *IoT and Analytics for Sensor Networks*. Springer, Singapore, 171-181.
- Mutanu, L., & Kotonya, G. (2018). A self-learning approach for validation of runtime adaptation in service-oriented systems. *Service Oriented Computing and Applications*, 12(1), 11-24.
- Nguyen, C.D., Perini, A., & Tonella, P. (2008). eCAT: A tool for automating test cases generation and execution in testing multi-agent systems (demo paper). In: *Proceedings of AAMAS-08*, 1669-1670.
- Oh, J., Meneguzzi, F., Sycara, K., & Norman, T.J. (2013). Prognostic normative reasoning. *Engineering Applications of Artificial Intelligence*, 26(2), 863-872.
- Parker, A.M., & Fischhoff, B. (2005). Decision-making competence: External validation through an individual-differences approach. *Journal of Behavioral Decision Making*, 18(1), 1-27.
- Petty, R.E., Briñol, P., & Tormala, Z.L. (2002). Thought confidence as a determinant of persuasion: The self-validation hypothesis. *Journal of Personality and Social Psychology*, 82(5), 722-741.
- Raghavan, A., & Baras, J.S. (2019). Non-commutative probability models in human decision making: Binary hypothesis testing. *IFAC-PapersOnLine*, 51(34), 47-52.
- Ramspek, C.L., Jager, K.J., Dekker, F.W., Zoccali, C., & van Diepen, M. (2021). External validation of prognostic models: What, why, how, when and where? *Clinical Kidney Journal*, 14(1), 49-58.
- Robinson, S. (1997). Simulation model verification and validation: Increasing the users' confidence. In: *Proceedings of the 29th Winter Conference on Simulation*, 53-59.
- Robinson, S. (2003). Modelling human decision-making. In: *Proceedings of the 17th European Simulation Multiconference*, 448-455.
- Rodríguez, N.D., Cuéllar, M.P., Lilius, J., & Calvo-Flores, M.D. (2014). A survey on ontologies for human behavior recognition. *ACM Computing Surveys*, 46(4), 1-33.
- Rosenfeld, A., & Kraus, S. (2018). Predicting human decision-making: From prediction to action. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Morgan & Claypool, 1-150.
- Rothwell, P.M. (2005). External validity of randomised controlled trials: "To whom do the results of this trial apply?". *The Lancet*, 365(9453), 82-93.
- Ryan, M.J., & Wheatcraft, L.S. (2017). On the use of the terms verification and validation. In: *Proceedings of the INCOSE International Symposium*, 27(1), 1277-1290.
- Shahar, N., Hauser, T.U., Moutoussis, M., Moran, R., Keramati, M., NSPN Consortium, & Dolan, R.J. (2019). Improving the reliability of model-based decision-making estimates in the two-stage decision task with reaction-times and drift-diffusion modeling. *PLOS Computational Biology*, 15(2), e1006803.
- Silveira, F.R., Campos, G.A., & Cortes, M.I. (2013). Rational agents for the test of rational agents. *IEEE Latin America Transactions*, 11(1), 651-657.
- Steyerberg, E.W., Bleeker, S.E., Moll, H.A., Grobbee, D.E., & Moons, K.G. (2003). Internal and external validation of predictive models: A simulation study of bias and precision in small samples. *Journal of Clinical Epidemiology*, 56(5), 441-447.
- Steyerberg, E.W., Harrell, F.E., Borsboom, G.J., Eijkemans, M.J., Vergouwe, Y., & Habbema, J.D. (2001). Internal validation of predictive models: Efficiency of some procedures for logistic regression analysis. *Journal of Clinical Epidemiology*, 54(8), 774-781.
- Tepjit, S. (2022). Designing an active recommender framework to support the development of reasoning mechanisms for smart cyber-physical systems. *Ph.D. Dissertation*, Delft University of Technology, Delft, the Netherlands, 1-313.
- Tepjit, S., Horváth, I., & Rusák, Z. (2019). The state of framework development for implementing reasoning mechanisms in smart cyber-physical systems: A literature review. *Journal of Computational Design and Engineering*, 6(4), 527-541.
- Thacker, B. H., Doebbling, S. W., Hemez, F. M., Anderson, M. C., Pepin, J. E., & Rodriguez, E. A. (2004). Concepts of model verification and validation. *Technical Report, LA-14167 TRN: US0500431*, Los Alamos National Lab., Los Alamos, NM, 1-36.
- Thiriot, S., & Kant, J.D. (2006). A multi-agent cognitive framework to model human decision making under bounded rationality. In: *Proceedings of the International Conference on Behavioural Economics and Economic Psychology*, Paris, 1-12.
- Tversky, A., & Kahneman, D. (2015). Causal schemas in judgments under uncertainty. *Progress in Social Psychology*, 1, 49-72.
- van Duinen, R., Filatova, T., Jager, W., & van der Veen, A. (2016). Going beyond perfect rationality: drought risk, economic choices and the influence of social networks. *The Annals of Regional Science*, 57(2), 335-369.
- Wainwright, J., & Mulligan, M. (2004). Modelling human decision-making. In: *Environmental Modelling: Finding Simplicity in Complexity*, John Wiley & Sons, Ltd., 225-244.
- Wu, W., Huang, J., & Zhang, Z. (2020). Human decision-making behavior modeling for human multi-robot interaction system. *IFAC-PapersOnLine*, it 53(2), 10202-10207.
- Yang, Y., Jiang, Y., Zuo, Z., Wang, Y., Sun, H., Lu, H.,... & Xu, B. (2019). Automatic self-validation for code coverage profilers. In: *Proceedings of the 34th International Conference on Automated Software Engineering*, IEEE, 79-90.
- Yin, C., & McKay, A. (2018). Model verification and validation strategies and methods: An application case study. In: *Proceedings of the 8th International Symposium on Computational Intelligence and Industrial Applications*, 02-06 November, 2018, Tengzhou, China, 1-6.

- Yu, C.W., & Choi, J.Y. (2005). Behavior decision model based on emotion and dynamic personality. In: *Proceedings of the International Conference on Control, Automation and Systems*, 2-5 June 2005, Gyeong Gi, Korea, 101-106.
- Zha, X.F., Lim, S.Y., & Lu, W.F. (2003). A knowledge intensive multi-agent system for cooperative/collaborative assembly modeling and process planning. *Journal of Integrated Design and Process Science*, 7(1), 99-122.
- Zhang, Z., Russwinkel, N., & Prezenski, S. (2018). Modeling individual strategies in dynamic decision-making with ACT-R: A task toward decision-making assistance in HCI. *Procedia Computer Science* 145, 668-674.

Author biographies

Dr. Imre Horváth is an emeritus professor of the Faculty of Industrial Design Engineering, Delft University of Technology, the Netherlands. In the last years, his research group focused on research, development, and education of smart cyber-physical system design, with special attention to cognitive engineering. Prof. Horváth is also interested in systematic design research methodologies. He was the promotor of more than 25 Ph.D. students. He was first author or co-author of more than 460 publications. His scientific work was recognized by five best paper awards. He has a wide range of society memberships and contribution. He is past chair of the Executive Committee of the CIE Division of ASME. Since 2011, he is a fellow of ASME. He is member of the Royal Dutch Institute of Engineers. He received honorary doctor titles from two universities, and the Pahl-Beitz ICONNN award for internationally outstanding contribution to design science and education. He was distinguished with the Lifetime Achievement Award by the ASME's CIE Division in 2019. He has served several international journals as editor. He was the initiator of the series of International Tools and Methods of Competitive Engineering (TMCE) Symposia. His current research interests are in various philosophical, methodological, and computational aspects of smart product, system, and service design, as well as in synthetic knowledge science and development of self-adaptive systems.

Dr. Sirasak Tepjit holds a Ph.D. from the Faculty of Industrial Design Engineering, Delft University of Technology, the Netherlands. His dissertation concerns the conceptualization and design of an intellectualized design-support tool for the development of reasoning mechanisms for smart cyber-physical systems. Currently, he works as a lecturer in Production and Robotics Engineering at the Faculty of Engineering, King Mongkut's University of Technology North Bangkok, Thailand. His research interests are in various areas of systems design and engineering, including design methodologies, knowledge engineering, smart engineering systems, and decision-support systems.