A Framework for Monitoring Evolution and its Drivers in Training Simulators

EBBA THORA HVANNBERG, University of Iceland

Software systems evolve with societal, business and technological changes. Because of these changes, socio-technical systems need to adapt to new situations that were unknown at the time of design. Good knowledge of software system evolution can help with that adaption. Although the evolution of software systems has been broadly debated, little research has been conducted on the specific genre of software systems and even less empirical research has been performed on the evolution of interactive software. We propose a three-factor framework which consists of identifying the changes during the evolution of training simulators, the drivers for those changes and how the changes effect innovation and robustness of the training simulators. In reviewing the literature on training simulators, we argue for this framework. The contribution of this paper is a framework that can be used to carry out empirical studies on the evolution of training simulators.

Interaction Science Key Words: Training simulators, Crisis Management, Fidelity, Innovation, Robustness, Evolution, Sociotechnical systems

DOI: 10.24982/jois.1813019.002

1 INTRODUCTION

Software systems evolve with societal, business and technological changes. Software technologies are said to evolve in a Darwinian manner by co-evolving with human culture [1]. Societal changes, including the migration of people and workers, poverty, health and education, can influence how software systems evolve. Business operations and their marketplace can be drivers of the evolution of software systems. Furthermore, innovations in technologies for building software systems influence their evolution. Because of these changes, today's sociotechnical systems need to adapt to new situations that were unknown at the time of design [2]; good knowledge of interactive software system evolution can help with that adaptation. Sociotechnical systems cover hardware, software, personal and community aspects [3] and when we refer to interactive software systems in this paper, we are not limiting the discussion to the software and its users. An example of a sociotechnical system is a training simulator for managing the response to large scale accidents such as those involving aircraft or trains. It concerns the hardware used, the software developed, the trainers and trainees using the system, and the community surrounding the professions and stakeholders. The development of such training simulators is costly. Training simulators can be long-lived up to several decades and thus undergo various changes in the course of use. Examples of these are changes because of improved work processes that are discovered during training exercises or real response, or demand for higher fidelity as trainees, for example firefighters training to extinguish fires.

Researchers have debated whether there exist such things as revolutionary products, or inventions, or if all are evolutionary [4, p.86], created in small steps visible only to the designer but invisible to the consumer [5, p.2]. Vicente [6] distinguished between revolutionary design and evolutionary design problems. Whereas in the former, the constraints decided on in past versions can change, in the latter, the constraints imposed by earlier generations of designs remain unchanged and must be accounted for by new developments. The processes of revolutionary design are better known than those of evolutionary design. In evolutionary design, design decisions made in earlier versions may affect the work analysis of the new versions and, thus, new requirements can be difficult to satisfy [6, pp.134-135].

While changes in requirements are continuous [7, 8], developers strive to minimize changes to systems to keep costs down [9]. One way to lower cost is to foresee the need for changes by developing adaptable systems. This also applies to sociotechnical systems where workers are encouraged to finish a design within a constraint-based approach (see Vicente [6, p.124] citing [10]).

The evolvability of a system is a quality characteristic that captures the ease of further system development. To be evolvable, interactive systems need to be adaptable to changing requirements and contexts. Nehaniv, Hewitt, Christianson and Wernick [11] tentatively defined evolvability as the "capacity to vary robustly and adaptively over time or generations in digital and natural systems". Evolvability was addressed by the software community as early as 1987 by Brooks [12] (as cited in Boehm [13]), who suggested meeting the software challenge with evolutionary methods, e.g., growing software instead of building it. Laws of software evolution that describe change, complexity, self-regulation, conservation of stability and familiarity and declining quality have been proposed to explain how software evolves [14]. Another strand of research has investigated whether mutational robustness, … meaning unchanged behavior of software when undergoing random mutations, can contribute to evolvability [15]. While some focus has been on the impact of change on software quality, less research has been on the impact of evolution on innovation, e.g. in HCI (Human Computer Interaction), where innovation is a desirable impact since it will help increase value for the developing organization [16].

Studying the evolution of interactive systems may be much more complex than studying the evolution of software with limited interactions. Loomes and Nehaniv [17] have noted the implicit assumption that there is one interface between a system and its users and that design problems are formulated by referring to a single, presupposed system, instead of viewing a set of interfaces for the system for a variety of diverse users. People interact with different representations of the system and each representation is a different object. Because people and their actions change over time and the variation in situations affects people's behavior, the set of drivers' changes. Thus, the interface design problem becomes one of responding to the evaluation of a series of interactions between people and representations of systems.

One aspect of managing changes encourages monitoring the drivers of evolution over time. Several drivers of change in software systems that have a long lifespan have been identified. After interviewing engineers and functional managers during site visits in a wide range of industries, Fricke, Gebhard, Negele and Igenbergs [7] presented the following eight categories of changes and their causes: needs and requirements changes because of technological evolution, competitors and customers; feedback and complaints from, for example, customers may be causes for change; complexity limiting feasibility and usability may need to be reduced; low degree of knowledge and experience in innovation technology may hinder their application; changes may spur more changes because of a network of components; insufficient communication and coordination between different organizational units or persons can lead to changes; requirements of time to market and lack of discipline in making decisions can be causes for changes. From the above list we see that the causes for change can be external factors such as customers or markets and internal factors such as complexity or knowledge of developers. Eckert, Clarkson and Zanker [18] call these causes *initiated* changes and *emergent* changes, respectively. These categories of changes and their rationales are quite broad, which calls for further research in order to understand and conceptualize them in terms of specific domains.

Although the evolution of software and interactive systems has been broadly debated, little empirical research has been conducted on the evolution of the specific genre of interactive systems, e.g. training simulators. Empirical software engineering research is important [19] and to facilitate research in software evolution, software tools have been developed to ensure that research studies are reproducible and extensible [20]. For maintaining a good quality of empirical studies it has been recommended that studies be based on previous research results [19]. To create a foundation for empirical research on the evolution of a training simulator, we present and discuss a framework that we have created after a literature review where we have focused on training simulators, especially for crisis management training. We emphasize four aspects of change of training simulators: fitness to practice, fidelity, evaluation and transfer of knowledge. In the introduction to the framework we justify this selection. This paper investigates how these four aspects can contribute to the understanding of the evolution of simulators, especially identifying the drivers of change and their impact on innovation and robustness.

2 BACKGROUND

2.1 Innovation and Robustness

Change, as an activity, includes cultures of inquiry and actions. Design is one of the first of many traditions of inquiry and actions [21] and as a strong component of building interactive systems it comprises creativity and innovation. Innovation is different from creativity in that it entails putting an idea or creation in use [22]. When understanding the role of innovation in the evolution of interactive systems, it may be useful to consider the view of Nelson and Stolterman [21] on design which states that if design is to mean progress instead of mere change, it does not merely include a strategy of having a vision of what needs to be built but needs to start with a client's expression of where he intends to go, his or her expression of desiderata. Thus, intention is not only about setting a vision of where to go but determining the direction to which we need to go to get there. It is not about evolution that happens by chance or accident but is motivated by intention. Learning how this intention motivates evolution resulting in innovation is one part of studying how interactive systems adapt to societal, business and technological changes. Similarly, researching this process, Usher studied how inventions get created, and he rejected the idea that they happen by accident, e.g. that it was no accident that a bicycle mechanism developed into an automobile, but that inventions needed insight in forms of new relationships and that that only happens when the mind is conditioned within the framework of the problem to be solved [23].

In achieving evolvability, researchers have investigated the application of biologically inspired concepts to implement real-world software systems. Evolution in biological systems captures the happenstance process of natural selection, which includes a cycle of the generation of random solutions, the selection of the best solutions, and random mutations to solutions that survive [24]. In biological systems, scientists study the concept of robustness and its relationship with innovation, complexity and degeneracy [25]. For example, researchers have found that complexity increases to improve robustness, degeneracy contributes to innovation, and that degeneracy is a precondition for evolvability and a more effective source of robustness [25]. Lee [26] concurs but goes further in stating that paradigm shifts in engineering occur because of crises in complexity and opportunity. Analogies to biological evolution have also been drawn in design [27] and architecture [4]. Lee [1] debates whether software evolves or is the result of top-down intelligent design and concludes that software co-evolves with cultural artefacts (programming languages, tools and practices) or techno-species; for instance he takes as an example that Wikipedia is an entity that evolves collaboratively with human cognition and culture. Although parallels are often drawn among software evolution, biological evolution and evolutionary computation, software evolution and maintenance present different modes of variability and of descent with modification than do biological or evolutionary computation since we do not know what an individual in a software system is nor what is the equivalence of a species or gene, the unit of heritable material [11].

The issue of robustness can be seen in engineering design when researchers have investigated the impact of change, which has been called change propagation, meaning change initiates new changes. Eckert, Clarkson and Zanker [18] have described three different absorption-propagation behaviors of systems in complex engineering domains. So-called constants are unaffected by change, though this is rare. They neither absorb changes nor cause new changes. Absorbers can absorb more change than they cause. The third type of behavior of engineering systems or components is comprised of carriers that absorb a similar number of changes as they themselves cause. The fourth type generates more changes than they absorb and increases the complexity of the change problem. Kelly [28] modelled a similar concept in her change model, a filter which describes whether a change in the environment results in change to the software. Although we have highlighted innovation and robustness, not everyone agrees that they play a larger role than other factors. For example, Lee [26] noted that craftsmanship and aesthetics may be as important as innovation and evolving technologies. The characteristics of innovation and robustness are not unique for systems in isolation but can be seen in HCI where users demand robust interactive systems that conform to their habits and that allow them to develop new habits [29] with new work or innovative technologies.

To better motivate the discussion on the relationship between innovation and robustness in technological systems, we show in Fig. 1 a hypothetical evolution of the degree of innovation and robustness of a software system. It is well known that in the beginning of technology evolution, innovation is high but then levels out [30]. As far as we could see there is little research on the evolution of robustness in software systems but a systematic review shows that more research is needed, especially of defining robustness as requirements [31]. At the onset of development, innovation increases, and robustness is steady, and it is easy to add features to a system. The system is relatively small, and developers have a good overview of it. After a while the robustness decreases, e.g. because of increased complexity, and in responding to the situation, efforts are spent on increasing the robustness of the system such that innovation remains steady. After having made the system more robust, innovation can again increase.

2.2 Technological evolution of software and interactive systems

Researchers have studied how software systems change at the micro and macro levels. Examples of the former are when changes to a system's components are studied over a relatively short period [28], and examples of the latter are when changes in technologies are studied over a long time period, even decades [32]. Kelly [28] studied internal and external factors that motivate

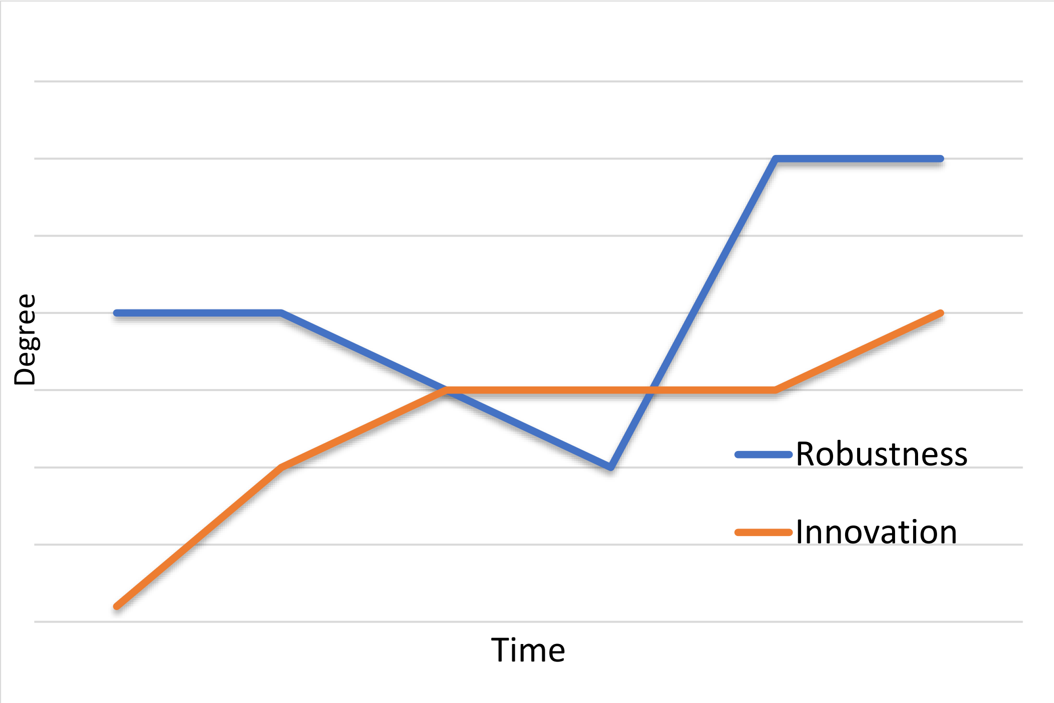

Fig. 1. Hypothetical impact of changes on Robustness and Innovation.

change. Typically, internal factors are related to the design or complexity of the software system and are measured to understand the need to simplify the system. Examples of external factors motivating changes include developers and users directly involved in the software development as well as those who are not involved but who set development policies. Kelly's motivation was to understand how to successfully change systems, and she concluded that the software architecture design and software development group characteristics played leading roles in the successful evolution of software. Fernandes, Henriques, Silva and Moss [8] also researched drivers of change and found that requests for change from users accounted for 15% of the changes over six years of development of complex aerospace systems at Rolls-Royce. External and internal drivers of change are measured not only in software engineering but also in engineering design. Two indices, the generational variety index and the coupling index, are used to design an architecture for variety [33]. The objective of these indices is to help standardize as many components as possible so that they remain unchanged throughout multiple generations and to modularize the remaining components so that their changes do not affect other components [33]. At the macro level, Ishii [32] provided an overview of the evolution of tangible user interfaces over ten years by looking at several applications. The lack of change in user interfaces of commercial systems motivated Ishii [32] to conduct the study, but he hoped to contribute to dynamic interfaces that integrate sensing and display into digital/physical material. The four studies of Kelly [28], Fernandes, Henriques, Silva and Moss [8], Martin and Ishii [33] and Ishii [32] are different. The first two analyze triggers for changes and attempt to learn how to make a system robust to change through either design or an improved process. The third develops measurements for change and uses them to develop a robust architecture that is unlikely to change. The fourth concentrates on analyzing the evolution of technologies that could influence innovation in applications.

While the systems being investigated in the above four studies are of specific domains, the results were not necessarily specific to the domain, and in the case of Kelly [28], for example, it is claimed that the research method and the results can be generalized to other domains than scientific applications. We maintain that it is necessary to develop a framework of evolution considering the genre of interactive systems and specifically training simulators. In the remainder of this section we discuss previous work in the former genre and in the following section we provide a background on training simulators.

Evolution and evolvability have been studied within the discipline of HCI. Arias, Eden, Fischer, Gorman and Scharff [34] have addressed the challenging problem of developing open human-computer systems which provide opportunities for changes to a system during its lifetime, as opposed to closed systems in which all functionality and qualities are fixed when the system is designed. They address the problem from the user perspective based on several principles: software systems must evolve, allowing users to make incremental changes to the core functionality; software systems must evolve at the hands of the users, e.g., through end-user programming or end-user development [35]; and interactive systems must be designed for evolution [34]. Invariably, users' needs must co-evolve with systems, systems influence users and vice versa, a situation which has motivated research in the area. Carroll and Rosson [36] described a task artefact co-evolutionary cycle, suggesting that a new artefact called for a new task, which called for a new artefact and so on. Fischer, Giaccardi, Ye, Sutcliffe and Mehandjiev [37] proposed the seeding, evolutionary growth, reseeding process model, which encourages designers to describe their designs as meta-designs, thus giving users the freedom to be creative designers instead of passive users. Others have proposed co-evolution between technological changes and organizational environments [38], using the results of activity theory.

2.3 Training simulator evolution

For decades, training has taken place with the aid of software simulators. The origin of training simulators can be traced back to centuries before they were implemented by means of IT, such as in military training [39] and surgery [40]. Furthermore, training simulators for air traffic control have a long history [41] (see also [42], as cited in [43]). Simulations that are set up to train individuals have been termed gaming, where gaming always involves individuals as decision makers [44]. Trainees use such simulators to train a variety of skills, including physical skills, knowledge, strategy formation and, following them, tactics and communication.

Several systematic reviews have been conducted on training simulators that allow individuals or groups to train skills or knowledge in areas such as health care [45], military gaming [39] and business gaming [46]. Smith [39] reviewed the history of military gaming and divided the eras into the stone age, paper age, mathematical age, computer age and personal age. During the stone age, military leaders manipulated a small physical copy of the battlefield by using sand tables with abstract icons to represent soldiers and units in battle. During the paper age, strategy board games were composed of wood or paper. During a pre-computer age, called the mathematical age, where calculators were used, dating to 1948-1953, the interest was in high-fidelity training that resulted in improved mathematical precision and reproducibility of results. In this era, the representation of the game, i.e., the board or the physical medium, did not change. A notable divide between training simulators and games was that improved precision came at the cost of playability of the simulators, which game players sought for entertainment. The computer age brought additional features, such increased game size, improved tactics, distributed gaming, customized views for players and attractive graphics. The personal gaming age brought wider accessibility to computer power for training and gaming. Smith [39] emphasizes six core technologies in gaming systems: 3D engines that create visualizations that stimulate players; graphical user interfaces (GUIs); physical models for the effects of movements, engagements, interactions and sensors for real-world accuracy; artificial intelligence to create an adaptable experience that adjusts the game; networking for multiplayer games; and persistent worlds to store the states of virtual worlds over extended periods. Faria, Hutchinson, Wellington and Gold [46] proposed seven key dimensions to understand the impact of technological changes on the effectiveness of business games: realism is another word for fidelity, i.e., how close the simulation is to real life; accessibility is about the accessibility of learners to simulation games; compatibility is a dimension measuring the compatibility between old and new technologies; flexibility and scale are related to allowing the trainer to change parameters of the game or add or delete models of the simulation and allowing many trainees to participate; simplicity of use is about the ease of playing the game, understanding the results and ease of determining how to improve performance; decision support in the form of numerical data analysis or trends to help learners make decisions; and communication within a team. Examination of the relationships among the main technologies suggested by Smith [39] and the attributes proposed by Faria, Hutchinson, Wellington and Gold [46] indicates strong ties between most of the components. As an extension of the above discussion on technologies, we present major technological breakthroughs that may have affected simulators by means of a few examples of training simulators (see Table 1).

Noting the wide variety and purpose of gaming simulations, Kriz [47] suggested two main categories and three subcategories of gaming simulation applications. The analytical sciences perspective can be described by a set of applications that are used as scenarios to empirically test, justify and develop theories in specific domains. The other is a science of design perspective that emphasizes the usability of simulation games that are evaluated in practical contexts. The latter category includes serious games. This category, which is the main concern here, is further divided into three subcategories. The first emphasizes the learning of the individual, the second focuses on policy making and the collective learning and support of real decision making within groups of actors and stakeholders, and the third includes gaming simulation applications that help organizational (re)design for the innovation and improvement of a system. An example of the last subcategory is the design and evaluation of a digital medical management training simulator [48] because the debriefing of a training exercise can be used for reflection on cooperation strategies in the organizations of the participants. Another example of a simulator in this category is one for training crisis management. Medical staff, rescue teams, police force and airline personnel must regularly train for and practice their roles in the management of airline crisis incidents. The effectiveness of the response to these incidents is paramount and involves training specialist skills, e.g., medical triaging, providing resources, including medical supplies, skilled personnel and transport, forming a strategy and following a plan.

Table 1. Technology evolution and training simulators

Decade	**Changes**	**Training simulator examples**
1960s	Mathematical models, Human factors, Graphics, Programming	
1970s	Software Engineering, Human-computer interaction, 3D graphics	
1980s	Personal computing, Video games, Artificial intelligence (knowledge based)	Radar training simulator [49]
1990s	Networking, Web, Multimedia, Gaming, 3D modelling, Virtual reality	Sigmoidoscopy simulator training [50]
2000s	GPUs, Mobile phones, Sensors, Robotics	Aircraft simulator [51]
2010s	Augmented reality, 3D printing, Artificial intelligence	Crisis management simulation, Neurosurgical simulator [40]

Communication between parties and keeping track of resources and casualties are additional skills that are trained. Training occurs via exercises of various sizes and forms. Small exercises can have five to ten trainees, and large-scale exercises can have up to 150 trainees. A simple form of a low-fidelity exercise, a desktop exercise, occurs around a table for training the strategic transport of resources, but in a high-fidelity exercise, trainees train on and around an airfield with a simulator airplane and actors playing the role of passengers.

From the above account of the evolution of software systems and evolution of training simulators specifically and specific characteristics of simulators, we argue that it will be useful to suggest a framework for the evolution of training simulators, a domain-specific genre of sociotechnical systems.

3 FOUR ASPECTS OF CHANGE IN TRAINING SIMULATORS

3.1 Introduction to the framework

The framework suggested and argued for in this paper includes three factors: the drivers of change, what changes, and the impact of these changes on the degree of innovation and robustness. A skeleton of the framework is depicted in Fig. 2, but we will complete it at the end of the paper. The following subsections describe the four aspects of what changes and discuss the impact of change. Section 4 explicates the drivers for change. We will assess the impact of changes qualitatively, but it is not our intention to present metrics for the degree of innovation or robustness. The second factor, what changes in training simulators, includes four aspects: fitness

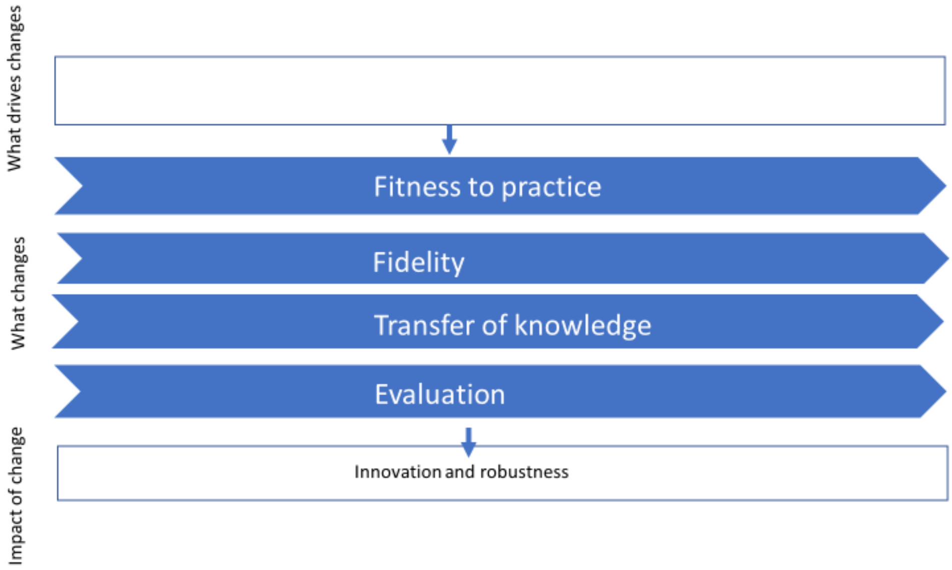

Fig. 2. Outline of a framework for monitoring change and drivers of evolution.

to practice, fidelity, evaluation and transfer of knowledge. A training simulator allows trainees to train for work, processes and communication, which are not stagnant factors. The fidelity of training simulators has been researched extensively [52], e.g., with respect to their effectiveness. Evaluation is an essential part of training simulator development that is conducted to improve the usability or pedagogy of the training simulator. A fourth aspect that has been less investigated is how knowledge can be transferred between technology domains in the development of training simulators, such as gaming to training [53] or from one domain to another, e.g., from surgery training to crisis management training. These aspects are more specific than those of Fricke, Gebhard, Negele and Igenbergs [7] mentioned earlier. Fitness to practice is classified as the changing needs and requirements of customers. Fidelity is also related to needs, specifically because of competitors, customers and technological evolution. Evaluation can be classified as feedbacks and complaints. The fourth aspect, transfer of knowledge, is not mentioned explicitly by Fricke, Gebhard, Negele and Igenbergs [7]. Some of their other types of changes are internal to an industrial system, such as change effecting change, complexity or because of organizational processes such as communication, employees' knowledge of innovation and decision discipline.

3.2 Fitness to practice

When developing a training simulator for training crisis management, instead of studying crisis management as a series of snapshots, it may be important to research how work evolves along temporal and spatial dimensions. Kuutti and Bannon [54] noted the importance of a practice perspective that examines historical processes and performances instead of qualitative observational studies in situ. Norros, Savioja and Koskinen [2] suggested the practice perspective as the unit for the analysis of human factor development. We have adopted this view and posited that, as work evolves, the training simulator's fitness to practice will be affected.

The core of any training simulator must be the work that the trainee must learn. A trainer must design several scenarios for a training exercise. A scenario that allows a trainee to meet a training goal consists of a series of events that are injected into the scenario, either manually or automatically, that the trainee must respond to. Among the characteristics of crisis management are that the work changes over time and is variable between sites, e.g., between airports. Response plans, roles, transportation and equipment resources change over time. Training simulators attempt to accommodate the variability in training needs between sites. This variability is accounted for through the generalization of scenarios that are instantiated in each case. In the remainder of this subsection, we provide examples where such generalizations are applied, as well as other examples of innovations that can help to increase the robustness of training simulators.

One example of such generalization is the study of Rudinsky and Hvannberg [55], who looked at crisis management across different response systems and incidents to identify typical tasks. The study assessed incident response systems in three countries, covering two aircraft incidents, a bomb threat and a train crash. Extensive material was collected from site visits, end-user workshops and observations of training exercises. Five types of models were created as part of the contextual design work models [56], which were subsequently merged to create an overall model. One of the conclusions of this work was that similarities were clear at the level of crisis management organization and command, but differences prevailed in procedural details. The drivers of such abstractions are economical, i.e., it is expensive to develop software for each new training scenario and marketing environment because large airport service providers want to enjoy the flexibility of training various scenarios.

Planning crisis management training for employees requires a large library of scenarios to satisfy the demands for enough variability and complexity. To satisfy the requirements for flexible, adaptive and creative skills, the Variable Uncertainty Framework [57] was developed to create scenarios for training with parameters for situational complexity, number of events occurring simultaneously, and randomness of events. Field, Rankin, Pal, Eriksson and Wong [57] coupled the Variable Uncertainty Framework with the Four Component Instructional Design (4C/ID), which considers four key components in the learning process for complex skills: training whole tasks, supportive information, just-in-time information, and part-task practice. To ensure realistic training scenarios in crisis management, the scenarios not only need to be variable but also need to evoke different emotions, especially surprising events that may startle trainees. Barnett, Wong, Adderley and Smith [58] suggested the generation of unexpected events, response and startle. As an alternative to large exercises with extensive preparations and full day execution including debriefing, part-task practice allows shorter, limited exercises with a few stakeholders in a brief period, e.g., in free time available between tasks. Thus, the focus has been on using part-task scenarios in addition to whole-task scenarios.

We have described example frameworks for generating scenarios, but an important innovation in the development of training simulators is the automatic generation of scenarios to save resources [59]. Martin, Schatz, Bowers, Hughes, Fowlkes and Nicholson [59] described scenario generation at the conceptual level. The inputs to such generation should be training goals, trainees' capabilities and mission briefs. The inputs can be obtained from human trainers or algorithms that generate the input. One output of scenario generation includes contexts such as terrain, weather and resources, which are domain dependent. The second output of the scenario generation includes components to support training effectiveness, e.g., embedded triggers or events that provide opportunities for training. Training should have clearly defined goals and offer scenario variety, psychological fidelity so that the trainee finds the simulation believable, and a range of complexity. The generation of scenarios is multifaceted, and development efforts have been mostly at the research level (see, e.g., Praiwattana and El Rhalibi [60] for a review). One example is the generation of scenarios in crisis management training via Bayesian network methods that model the causality between the key events and training objectives of the scenario, where human-designed scenarios are used as the seeds of an algorithm [61]. One key feature of Bayesian networks is their graphical representations that allow different stakeholders in the training to view them and supply knowledge from different areas [62]. Other options for scenario generation are being investigated, such as the procedural modelling approach that uses internal rules and symbol sets to represent the elements being modelled [59] and a data-driven scenario generation that uses neural networks [63].

An after-action review following a real-life exercise of crisis response showed that work processes required improvement, for example in maintaining the counts of casualties for different stations, and that these work processes could be supported with IT [64]. In the after-action review, managers suggested the use of software for improving delegation and prioritization on the scene, improving primary triage and casualty counting, and improving the maintenance and availability of resources [64]. Furthermore, in some cases, simulator training has spurred IT innovation to help to keep track of casualties and monitor how they flow between different stations. As noted by Kuutti and Bannon [54], one goal in HCI is to understand how practices are changing due to the introduction of IT. In advocating the turn to practice in HCI research, Kuutti and Bannon [54] asked how and why these transformations of new practices occur and how they can be supported. Similarly, Vicente [6] defined formative approaches, focusing on technical and organizational requirements that need to be fulfilled if a device like IT is going to support work effectively. These requirements will inform the design, which in turn will give the worker constraints within which he/she will perform the work.

Separating concerns in conceptual modelling of work may aid the understanding of how work and its context drive the evolution of training simulators. These concerns will evolve at different paces and separating them will help developers and trainers understand them and track their evolution. For example, human competencies may evolve more slowly than management and organization, which may evolve more slowly than work, which in turn may evolve more slowly than technologies. Different conceptual frameworks exist to analyze work and its context, which may be useful in understanding the separation of concerns (see, e.g., PACT [65], Cognitive Work Analysis [6], Contextual Design [56] and Six facets of domain engineering [66]). Hvannberg and Rudinsky [67] reviewed conceptual modelling that was performed for a training simulator for crisis management training. They analyzed which components had been extracted and examined whether these components could be found in four different analytical methodologies. Although considerable overlap among the methodologies exists, each has some differences.

Since scenario design is resource demanding, advances in artificial or computational intelligence are likely to drive scenario generation, but the ability to model work and its context will certainly help in understanding the basis of intelligence. Furthermore, understanding which concerns evolve faster and which develop more slowly could help to decide which parts of the scenario are efficient to generate and which can be manually designed.

3.3 Fidelity

The second aspect worth monitoring for change in training simulators is fidelity. Originally, fidelity in the context of simulators was defined as the accuracy of the simulator's imitation of the operational equipment, environment and tasks [68]. The concept has since developed and has been viewed from three perspectives: physical, functional and psychological [69]. Physical fidelity refers to the degree to which a simulation imitates the real-life environment with respect to sensual experiences, i.e., visual, aural and haptic experiences [53, 70]. Functional fidelity is the degree to which the simulation accurately reconstructs the real-life environment and its functions, focusing on issues such as operational knowledge and response options. Psychological fidelity is the degree to which the simulation imitates the psychological engagement that trainees experience. These are general definitions, but a need for domain-specific levels of simulation fidelity for training skills has been noted [71].

In determining the degree of fidelity in training simulators, designers and customers weigh the transferability of training [53] and its effectiveness against cost. In the domain of air traffic control, the industry-wide belief is that trainees receive better training with increasing fidelity [72]. However, studies have shown that lower-fidelity simulation can provide effective training [72]. Despite the importance of these trade-offs, cost analysis has been inadequately researched [73]. Dahlstrom, Dekker, Van Winsen and Nyce [72], with reference to Dennis and Harris [74]. Also, Lee [75] has claimed that higher fidelity in the aviation industry is mostly technology driven. For example, innovative technologies, such as 3D graphics, have resulted in higher fidelity. This improvement has partially originated in games where gamers desire to experience ever more realistic and imaginative scenarios. Pagulayan, Keeker, Wixon, Romero and Fuller [76] noted that the constant demand for novelty has been a strong incentive for developers to innovate, at the expense of less consistency and increased difficulty for novice players. This characteristic is one of the reasons to investigate the evolution of the fidelity of training simulators and to study its drivers.

The term fidelity has been debated. One difficulty is that a benchmark of truth or goodness is difficult to establish [26]. Lee [26] takes the example from Plato's allegory of the cave, which suggests that human understanding of what is real is always imperfect. Another caveat is that fidelity may depend not on the artefact but the richness of the experience of the user [77]. Hamstra, Brydges, Hatala, Zendejas and Cook [78] questioned the usefulness of the term fidelity because it is difficult to define, as evidence from the aviation industry and the military and health professions has shown. They recommended replacing the term fidelity with specific terms for describing physical resemblance and functional task alignment and focusing on methods that enhance the transfer of learning. They further suggested that when trainees see discrepancies between the simulator and the physical model, the trainees will suspend disbelief in the simulator for a moment because there may be issues that are not relevant to the task that they are training for and they realize that the training is important. The issue is not limited to training simulators but concerns the nature of authenticity in learning [79], presenting realistic scenarios to the learner or persuading him/her that the situation is authentic [80].

In the following, we provide two examples of fidelity from crisis management training simulations. The success of simulator training relies on how well the simulation manages to provide learning feedback to trainees. Several forms of feedback exist in terms of the learning context and the system context. Druzhinina, Hvannberg and Halldorsdottir [81] presented an overview of different forms of feedback. Examples from the learning context can be positive, negative, corrective, motivational, unidirectional, multidirectional, peer-review and oral feedback. Feedback in the system context includes visual/graphical, textual, haptic, sound, immediate and delayed. The researchers investigated the fidelity of feedback by comparing a Table Top exercise (TTex) with a Real-Life exercise (RLex) and the requirements specification of a Virtual Reality Simulator (VRS). Haptic feedback was equally absent in all three environments: RLex, TTex and VRS. Some visual feedback delivery was contradictory to the real-life exercise, and the visual representations differed. The results showed that in terms of functional fidelity, major deviations were observed between the practice of the VRS and RLex. The instructor can be prevented from intervening to achieve higher functional fidelity of feedback in simulator training. Delayed feedback in the VRS could be improved with the implementation of an optional post-training meeting. Such post-training sessions would allow peers to learn from mistakes made during crisis response. Research on the third type of fidelity, psychological fidelity, was not found during the analysis of different training forms. Neither negative nor positive forms of feedback could be found in TTex or RLex.

The second example we give from crisis management training simulators concerns soundscapes and communication. Sounds from fire engines, helicopters, fire and people can be heard at the scene of crisis management. Including these sounds in a training simulator can contribute to higher fidelity. Since communication between firefighters, police, the airport and health workers is paramount, such communication should be performed in noisy conditions in a training simulator [82].

The discussion of fidelity assumes that the original model is a physical model. Examples of medical training include patients, hospitals and equipment, and in crisis management, the physical model includes transportation, an airfield, airplanes, passengers or casualties. However, the original model is increasingly an IT system or a combination of an IT system and a physical model. Examples of air traffic control include radar systems, flight trackers and communication systems. Examples of crisis management include communication devices, such as Tetra, mobile phones, and resource trackers implemented with IT. Technical factors supporting information management were reviewed in an analysis of the output of an after-action review [64]. The results showed that there were concerns related to the technical support of information management with the Internet, software and communication infrastructure, as needed for crisis response. The IT used in managing crises requires the training simulator to be closely integrated with the crisis management system itself. A simulator's ability to imitate IT systems can affect physical and functional fidelity. To increase fidelity, in some cases, air traffic controller simulators are integrated with operation systems. Augmented reality, that is, an overlay of virtual reality on physical objects or physical simulations, such as laparoscopy equipment or mannequins, is one potential innovation to help to overcome this challenge in some disciplines, e.g., surgery [83].

Regarding the fidelity of training simulators, several innovations have been observed, including 3D graphics of a virtual environment and face-to-face communication in proximity and soundscapes. As mentioned above, little innovation in providing training feedback has been reported in terms of training simulators for crisis management. The high fidelity of a training simulator that is implemented by integration with operational systems may increase the simulator's robustness since updates in the user interfaces of operation systems do not need to be replicated in a simulator.

3.4 Evaluations

Evaluation is vital in the development of a training simulator to improve its pedagogy and usability. Thorough evaluation schemes exist to assess validity [84], and outcomes of evaluations include opportunities for improvement [85]. In a systematic review of the multi-user virtual world of health education, Liaw, Carpio, Lau, Tan, Lim and Goh [86] noted that all the studies were short-term in nature and recommended that studies are needed for longer-term evaluation and cost analysis.

A systematic evaluation scheme, consisting of several benchmarks, to assess the validity of training simulators exists. These benchmarks include face validity, where an expert evaluates whether the measure assesses what it is supposed to measure; content validity, which measures whether the content covers what the trainee is to learn; construct validity, which assesses whether the performance of experts is different from that of novices; concurrent validity, where the simulator is compared to other training instruments; and predictive validity, which measures whether the performance scores in the simulator predict performance scores in reality [84].

Wang, DeMaria, Goldberg and Katz [45] reviewed serious games used for training health professionals. A list they collected consisting of 42 serious games included 16 training simulations where the situations were sufficiently realistic for skills to be trained. Seven of those were evaluated. Four of seven showed improved performance, three showed no significant effect, and three showed other factors, such as high scores for realism and content, ease of use and usefulness. The results indicated that only one of the 42 studies assessed validity beyond gain in trainee knowledge or skills, which is in accordance with previous research [87]. The question of what motivates the evaluations remains. A second question is to what extent these evaluations can direct designers, trainers and trainees, and educators towards more effective designs of simulators and training instruments. Koivisto, Haavisto, Niemi, Haho, Nylund and Multisilta [88] studied iterative cycles of designing, testing and refining a simulation game. The study included iterative cycles in collaboration with researchers, educators, students and game designers and showed that realistic patient scenarios are important for learning via game-based simulation and that authentic representation supported real-life experience. Some of the conclusions of the study are that resources could have been saved had students been involved earlier in the design process and that experts from many fields must be engaged in the design so that the training simulator is appropriate for the profession being trained. Of the 42 serious games included in Wang, DeMaria, Goldberg and Katz [45] systematic review, 19 included medical experts, three included trainees on the development team, and two included educationalists. In six cases, the development team and the technical resources were not specified in the source. Considering the requirement that training simulators result in improved performance, the fact that so few educationalists and trainees are involved in the development is surprising.

Virtual environments must allow many users to exist in the same environment [89], and collaborative virtual environments are more challenging to evaluate than single-user environments [90]. A systematic review of the validity of serious games for medical education and surgical skills training found that 10 of 17 serious games had multiplayer functions. Eight of the 17 serious games underwent validity testing, but only one of these had multiplayer functions. The results of a randomized controlled trial proved concurrent validity, i.e., some improvement was observed for both conditions, i.e., with and without a virtual reality serious game [87].

Various methods have been developed to assess the usability of a virtual environment [90]. Usability has not been addressed specifically in many systematic reviews [45, 52, 83, 87, 91] in the medical domain. An exception is a systematic study including 18 papers by Liaw, Carpio, Lau, Tan, Lim and Goh [86], who found that most papers reviewed learners' reactions. Six of the studies reported hindrances, such as lack of accessibility, and usability issues, including difficulty with communication, identity confusion and difficulty in navigating avatars.

A usability evaluation of collaboration during crisis management training in a simulator implementing a soundscape has been conducted [92]. A collaborative scenario for two users playing the role of coordinators in the field and in a response-center was prepared. Several methods were used to probe usability problems or mistakes in conducting crisis management activities. The results showed that the functional and physical fidelity of the simulator was less than planned. Additionally, the training instruments, i.e., the training activities and the roles played, were sometimes inadequate. Several usability problems indicated low fidelity of the training simulator. Physical fidelity was low in some cases, e.g., when a user met the avatar representing his/her partner and, knowing his partner, did not recognize the avatar. However, this scenario is a situation of suspension of disbelief [78], which means that users are able to accept a simulation, i.e., ignore its medium [89], and the trainees could continue their training without being substantially affected by the low physical fidelity. Users needed some time to become familiar with the communication and to switch from the radio metaphor to face-to-face communication within the simulator. Overall, the communication was efficient, but some information did not reach the recipient due to inattention, which may have been caused by the busy conditions or noise. The number of problems attributed to participants following a script indicated that one of the greatest challenges in conducting scripted evaluations is to not disturb the users and to motivate the users to respond to situations instead of thinking that they can write their own script. This characteristic has also been noticed when participants participate in real-life exercises; hence, the problem is not unique to the virtual training environment. Based on these studies, we conclude that researchers cannot study the fidelity of a training simulator in isolation but only in the context of other training instruments, including the exercises given, the actors and the roles they play. Innovations in evaluation methodologies for pedagogy and usability evolve along with the technologies, e.g., augmented reality, and the demand for validity and usability. Most likely, the evaluation methodologies designed for innovative technologies will evolve at a slower rate than the innovations themselves.

3.5 Transfer of new knowledge

During the lifetime of a system, new knowledge discovered in a related discipline could be adopted. The transfer of knowledge can occur from one domain to another through adoption, adaptation or transformation with the purpose of making further use of existing knowledge [93]. Malerba [94] proposed a sectoral system of innovation framework to help understanding of how innovation occurs in terms of knowledge, actors, networks and institutions. The framework was inspired by evolutionary theory and the innovation system approach. Hvannberg [93] used this approach to analyze the components of crisis management. An analysis of case studies of how two evaluation methods were transferred to crisis management training simulators resulted in a process model describing the transfer of methodological knowledge by applying the sectoral system of an innovation framework. Other examples of the transfer of knowledge can be found in the work of Rudinsky and Hvannberg [95], who investigated the transfer of knowledge from games to training simulators for voice communication in crisis management training. An empirical study of voice communication in training simulators derived statements and verified them in the literature on multiplayer games. Approximately two-thirds of statements on voice communication proved coherent in both domains. The work resulted in design guidelines for a virtual environment for crisis management training.

Wiberg [96], citing Pagulayan, Keeker, Wixon, Romero and Fuller [76], noted that games have been a driving force of HCI development. Some concern exists in transferring entertainment properties into military applications [39] or adopting technologies that originate from other industries. The concern is that these training simulators, whether military warfare, crisis management, air traffic control or medical surgery, are serious businesses that may not have any resemblance to the competition, excitement and fun of games. Nonetheless, as Shepherd and Bleasdale-Shepherd [97] show, potential gains can be achieved by transferring knowledge from video games to other domains, e.g., virtual geographic environments, via design-by-adaptation. Other research has shown that the transfer of knowledge from an aerospace application to cruise control in cars has potentially compromised safety [98]. Motivated by the cost of training [99], considerable development and research on training simulators in health care and surgery has been reported. In a reflection on the design approaches of virtual environments, Sutcliffe, Poullis, Gregoriades, Katsouri, Tzanavari and Herakleous [100] cited examples of domain-specific design approaches but noted that domain-specific experiences are difficult to generalize so that they can be specialized in other domains. Hvannberg, Halldorsdottir and Rudinsky [101] reviewed how heuristic guidelines for virtual environments are used and found that some designers use specialized heuristics according to their needs, e.g., application domain, but researchers do not agree on how generic or specialized the guidelines should be. Thus, the question remains of how transferrable development knowledge is to training simulators in other domains of practice.

4 DRIVERS OF CHANGE IN TRAINING SIMULATORS

In this section, we explicate the drivers of change and thus complete the framework of the evolution of training simulators. The framework resulting from the analysis is presented in Fig. 3.

Trainees' and trainers' demand for realism: Demand for realism has affected fidelity in simulators for centuries. Although much debated, it is likely that the trainees' and trainers' demand for realism will continue to drive change. One must be careful in concluding what or who the main driver for technological change is. It may well be that the driver of change is the technology company or designers convincing trainees and trainers to adopt new technology, or as Loomes and Nehaniv [17] gave examples of a techno-scientist's mechanism in giving arguments for adopting a technology "I want it, why don't you?" [102]. Companies selling the product could say "Everyone will move to a virtual environment for crisis management training". Another argument is "If you make a short detour through our methods…" Marketing could say that they know that producing high physical fidelity is expensive but claim that once the investment is made, the benefits will be obtained. Functional fidelity is essential, physical fidelity is debated although it is stated to be important [103], and psychological fidelity may be yet to come. The contest between innovation and robustness, which appears as a need to deliver performance, is clear.

New incidents of work and innovative technologies used as part of work: Since trainees train in a field that changes, whether crisis management, air traffic control or health care, new incidents of work appear and drive the need for new training. Even training itself can influence work. The use and evolution of the technological implementation of work is a driver of change in

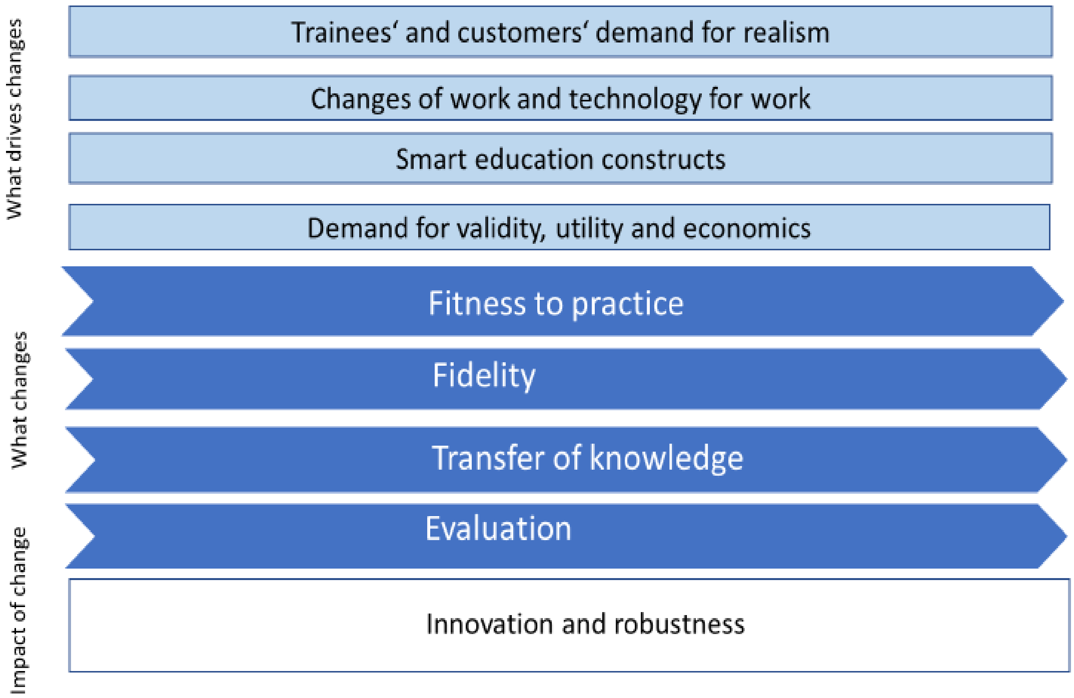

Fig. 3. A framework for monitoring change and drivers of evolution.

training simulators. This characteristic may require the integration of training simulators and working systems to a further extent than we have already seen. Otherwise, there is a risk that training simulators will never catch up with the technology platform or the context in which the training occurs.

Evaluation outcomes: usability, utility and economics: The literature has expressed a need for thorough evaluations and called for the validation of training simulators. Much is at stake and it must be confirmed whether training simulators are better than other types of training and whether training simulators improve trainees' performance. Additionally, the results of these evaluations should be used as evidence and a driver for change in simulator design [104]. However, given the small number of results on the long-term effect of evaluations, i.e., the so-called downstream effects [105], it remains to be seen whether this fact holds.

The discussion of fidelity includes a debate about its effectiveness and economics. The argument is that low fidelity is more economical than high fidelity and just as effective. In fields such as surgery where much is at stake, a lack of training can increase patient morbidity and mortality and be financially expensive as well [99]. The development of training simulators with the latest technologies and thorough evaluation of the system, user and education levels is also expensive. With uncertainty in the cost of software development of training simulators and predictive validity, it may be difficult to convince buyers to invest in such large systems.

Smart education constructs: Although early researchers noted the need for a simulator that was not scripted and the requirement for a game or a simulator with a free flow of tasks and collaboration, little coverage of these topics exists in the literature, except as basic research. Artificial intelligence could be a larger driver of change in the future than it has been in the past, for example by allowing simulators to respond more dynamically over time to trainees' actions and their evolving expertise. Part of this change is allowing training simulators to provide feedback to trainees and allowing trainees to reflect on their learning. Advances in simulators that study the effectiveness of operations could help air traffic control training simulators. Examples of such simulators have studied the effectiveness of using artificial intelligence to optimize the task load of air traffic controllers [106]. This research could be exploited in the variable task loads of trainees in training scenarios, which could synergize well with the development of variable uncertainty scenarios [57].

5 DISCUSSION

One of the aims of this work was to analyze changes, drivers of change and the impact of change for a domain-specific interactive system. While the labels of the three factors of the framework reflect current work on change and evolution and are generic, some of their components may be domain specific. For example, fitness to practice may not be important in all interactive systems since not all systems are for implementing evolving work, e.g. games. However, fidelity may be an important factor to monitor for games. It may be a topic for further research to see if some domains are more open to transfer of knowledge than others. Research shows that evaluation of interactive systems is domain-specific [101, 107, 108]. Reviewing the drivers of change, some of them are specific to a domain such as smart education constructs, albeit a large domain. The demand for validity and economics may vary between domains but may also be dependent on the size and cost of the interactive system. Thus, while the individual components may vary between domains, the labels of the three factors of the framework could be applied to other genres.

Kelly [28] proposed a three-tier change model consisting of five knowledge domains that can drive change, three change filters to determine how strongly or weakly the change is advocated, and the third tier is modification of the code. Two of the five knowledge domains in the change model, physical and operational knowledge domain, have overlaps with the framework presented in this paper. The model is a general one, but it is recognized that the five knowledge domains may apply differently to different types of applications, e.g. military application software or medical imaging software, and that the systems may respond differently to changes from the knowledge domains. The conclusion of Kelly's study was that architectural design of the software and the characteristics of the software development group played a major role in the successful evolution of the software. Evolution of software architecture has been the focus of much research, e.g. for real-time systems [109]. There is also work on the effect of evolution of software systems on complexity [110] which affects robustness of the system. Guzman, El-Haliby and Bruegge [111] analyzed how users asked for changes in apps and may be useful for understanding drivers of change. While such general conclusions are useful and could be a basis for understanding the evolution of domain-specific interactive systems, they do not explain to a software developing organization how the evolution of a domain-specific interactive system takes place. Research has investigated the challenges in evolution of domain-specific systems, e.g. automated production systems. In this domain changes in hardware occurred during the extended lifetime of the software and customer needs evolved. One of the major challenges is that it is a variant-rich system where many variations are a part of a product family [112].

A second aim of this work was to suggest a framework that could provide a basis for an empirical study of evolution. In future work it will be interesting to see if it will show that a software developing organization may be more interested in monitoring other aspects of change and subsequently find that there are other drivers for change.

6 CONCLUSION

The contribution of this paper is a framework of drivers for change, which changes are addressed, and how these changes affect the innovation and robustness of training simulators. We identified several drivers of change: trainees' and trainers' demand for realism; new incidents of work and innovative technologies used in work; evaluation outcomes, including usability, utility and economics; and smart educational constructs. Unlike other similar studies on the evolution of systems, this study focused on a specific domain. The aim of the framework is to use it as a foundation for empirical study on how training simulators evolve.

A study like this cannot be exhaustive but is only an attempt to understand the drivers of change. For example, artificial intelligence has only been briefly included in this account of training simulators. Early research efforts in the eighties applied expert systems in training simulators to improve conceptual fidelity [113]. The past decade has seen major advances in artificial intelligence, including agent technology that uses artificial intelligence to coordinate collaborative behavior and that can support human and systems behavior in complex operations such as crisis management. However, it takes time before such advances show up in training simulators in operation [114].

Detailed studies of the changes in software applications have been reported. Although we have found discussion of evolution, co-evolution and adaptation in the HCI literature, we have not seen detailed studies on change parallel to those in the software development literature. By studying factors likely to be drivers of change, we will be better prepared to conduct such studies. The dynamicity of training simulators and their influence on the need for training itself and the work to be trained make this a challenging research topic.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable and constructive comments.

Received December 2018; revised April 2019; accepted June 2019

REFERENCES

[1] Lee, E. A. Is software the result of top-down intelligent design or evolution? *Commun. ACM*, 61, 9 (2018), 34-36. https://doi.org/10.1145/3213763

[2] Norros, L., Savioja, P. and Koskinen, H. *Core-task Design: A Practice-Theory Approach to Human Factors*. Morgan & Claypool, 2015.

[3] Whitworth, B. and Ahmad, A. *The social design of technical systems: Building technologies for communities*. In: J. M. Carroll and J. Kjeldskov(eds). *Encyclopedia of Human-Computer Interaction.* 2013. Interaction Design Foundation. Retrieved from https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/socio-technical-system-design

[4] Steadman, P. *The Evolution of Designs: Biological Analogy in Architecture and the Applied Arts*. Routledge, 2008.

[5] Eger, A. O. and Ehlhardt, H. *On the Origin of Products: The Evolution of Product Innovation and Design*. Cambridge University Press, 2018.

[6] Vicente, K. J. *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-based Work*. CRC Press, 1999.

[7] Fricke, E., Gebhard, B., Negele, H. and Igenbergs, E. Coping with changes: Causes, findings, and strategies. *Systems Engineering*, 3, 4 (2000), 169-179. https://doi.org/10.1002/1520-6858(2000)3:4<169::AID-SYS1>3.0.CO;2-W

[8] Fernandes, J., Henriques, E., Silva, A. and Moss, M. A. Requirements change in complex technical systems: An empirical study of root causes. *Research in Engineering Design*, 26, 1 (2015), 37-55. https://doi.org/10.1007/s00163-014-0183-7

[9] Jarratt, T., Eckert, C., Caldwell, N. and Clarkson, P. Engineering change: An overview and perspective on the literature. *Research in Engineering Design*, 22, 2 (2011), 103-124. https://doi.org/10.1007/s00163-010-0097-y

[10] Rasmussen, J. and Goodstein, L. Decision support in supervisory control of high-risk industrial systems. *Automatica*, 23, 5 (1987), 663-671. https://doi.org/10.1016/0005-1098(87)90064-1

[11] Nehaniv, C. L., Hewitt, J., Christianson, B. and Wernick, P. What software evolution and biological evolution don't have in common. In *2006 Second International IEEE Workshop on Software Evolvability (SE'06)*. Philadelphia, Pensylvania, 2006. IEEE. https://doi.org/10.1109/SOFTWARE-EVOLVABILITY.2006.18

[12] Brooks, F. P. No Silver Bullet Essence and Accidents of Software Engineering. *Computer*, 20, 4 (1987), 10-19. https://doi.org/10.1109/MC.1987.1663532

[13] Boehm, B. A view of 20th and 21st century software engineering. In *28th International Conference on Software Engineering (ICSE '06)*. 2006. ACM. https://doi.org/10.1145/1134285.1134288

[14] Lehman, M. M., Ramil, J. F., Wernick, P. D., Perry, D. E. and Turski, W. M. Metrics and laws of software evolution-The nineties view. In *Fourth International Software Metrics Symposium*. 1997. IEEE. https://doi.org/10.1109/METRIC.1997.637156

[15] Schulte, E., Fry, Z. P., Fast, E., Weimer, W. and Forrest, S. Software mutational robustness. *Genetic Programming and Evolvable Machines*, 15, 3 (2014), 281-312. https://doi.org/10.1007/s10710-013-9195-8

[16] Frohlich, D. M. and Sarvas, R. HCI and innovation. *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (2011), 713-728. https://doi.org/10.1145/1979742.1979670

[17] Loomes, M. and Nehaniv, C. L. *Fact and artifact: Reification and drift in the history and growth of interactive software systems*. In: M. Benyon, N. C.L. and D. K.(eds). *Cognitive Technology: Instruments of Mind.* Lecture Notes in Computer Science, vol. 2117. CT, 2001. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44617-6_3

[18] Eckert, C., Clarkson, P. J. and Zanker, W. Change and customisation in complex engineering domains. *Research in Engineering Design*, 15, 1 (2004), 1-21. https://doi.org/10.1007/s00163-003-0031-7

[19] Sjoberg, D. I. K., Dyba, T. and Jorgensen, M. The future of empirical methods in software engineering research. *2007 Future of Software Engineering* (2007), 358-378. https://doi.org/10.1109/fose.2007.30

[20] Dit, B., Moritz, E., Linares-Vásquez, M., Poshyvanyk, D. and Cleland-Huang, J. Supporting and accelerating reproducible empirical research in software evolution and maintenance using TraceLab Component Library. *Empirical Software Engineering*, 20, 5 (2015), 1198-1236. https://doi.org/10.1007/s10664-014-9339-3

[21] Nelson, H. G. and Stolterman, E. *The design way: Intentional change in an unpredictable world: Foundations and fundamentals of design competence*. Educational Technology, 2003.

[22] Zhang, W., Zhang, Q. and Song, M. How do individual-level factors affect the creative Solution formation process of teams? *Creativity and Innovation Management*, 24, 3 (2015), 508-524. https://doi.org/10.1111/caim.12127

[23] Ruttan, V. W. Usher and Schumpeter on invention, innovation, and technological change. *The Quarterly Journal of Economics*, 73, 4 (1959), 596-606. https://doi.org/10.2307/1884305
[24] Bongard, J. Biologically Inspired Computing. *Computer*, 42, 4 (2009), 95-98. https://doi.org/10.1109/MC.2009.104
[25] Whitacre, J. and Bender, A. Degeneracy: A design principle for achieving robustness and evolvability. *Journal of Theoretical Biology*, 263, 1 (2010), 143-153. https://doi.org/10.1016/j.jtbi.2009.11.008
[26] Lee, E. A. *Plato and the Nerd: The Creative Partnership of Humans and Technology*. MIT Press, 2017.
[27] Simon, H. A. *The Sciences of the Artificial*. MIT Press, 1996.
[28] Kelly, D. Determining factors that affect long-term evolution in scientific application software. *Journal of Systems and Software*, 82, 5 (2009), 851-861. https://doi.org/10.1016/j.jss.2008.11.846
[29] Hornbæk, K. and Frokjaer, E. Usability inspection by metaphors of human thinking compared to heuristic evaluation. *International Journal of Human-Computer Interaction*, 17, 3 (2004), 357-374. https://doi.org/10.1207/s15327590ijhc1703_4
[30] Roberts, E. B. What we've learned: Managing invention and innovation. *Research-Technology Management*, 31, 1 (1988), 11-29. https://doi.org/10.1080/08956308.1988.11670497
[31] Shahrokni, A. and Feldt, R. A systematic review of software robustness. *Information and Software Technology*, 55, 1 (2013), 1-17. https://doi.org/10.1016/j.infsof.2012.06.002
[32] Ishii, H. The tangible user interface and its evolution. *Commun. ACM*, 51, 6 (2008), 32-36. https://doi.org/10.1145/1349026.1349034
[33] Martin, M. V. and Ishii, K. Design for variety: Developing standardized and modularized product platform architectures. *Research in Engineering Design*, 13, 4 (2002), 213-235. https://doi.org/10.1007/s00163-002-0020-2
[34] Arias, E., Eden, H., Fischer, G., Gorman, A. and Scharff, E. Transcending the individual human mind—creating shared understanding through collaborative design. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7, 1 (2000), 84-113. https://doi.org/10.1145/344949.345015
[35] Costabile, M. F., Fogli, D., Mussio, P. and Piccinno, A. *End-user development: The software shaping workshop approach*. In: H. Lieberman, F. Paternò and V. Wulf(eds). *End User Development.* Human-Computer Interaction Series, vol. 9. Dordrecht, 2006. Springer Netherlands. https://doi.org/10.1007/1-4020-5386-X_9
[36] Carroll, J. M. and Rosson, M. B. Getting around the task-artifact cycle: How to make claims and design by scenario. *ACM Transactions on Information Systems (TOIS)*, 10, 2 (1992), 181-212. https://doi.org/10.1145/146802.146834
[37] Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A. G. and Mehandjiev, N. Meta-design: A manifesto for end-user development. *Communications of the ACM*, 47, 9 (2004), 33-37. https://doi.org/10.1145/1015864.1015884
[38] Bourguin, G., Derycke, A. and Tarby, J.-C. *Beyond the Interface: Co-evolution Inside Interactive Systems — A Proposal Founded on Activity Theory*. In: A. Blandford, J. Vanderdonckt and P. Gray(eds). London, 2001. Springer London. https://doi.org/10.1007/978-1-4471-0353-0_18
[39] Smith, R. The Long History of Gaming in Military Training. *Simulation & Gaming*, 41, 1 (2010), 6-19. https://doi.org/10.1177/1046878109334330
[40] Rehder, R., Abd-El-Barr, M., Hooten, K., Weinstock, P., Madsen, J. R. and Cohen, A. R. The role of simulation in neurosurgery. *Child's Nervous System*, 32, 1 (2016), 43-54. https://doi.org/10.1007/s00381-015-2923-z
[41] Buckley, E. P., DeBaryshe, B. D., Hitchner, N. and Kohn, P. *Methods and Measurements in Real-time Air Traffic Control System Simulation*. Federal Aviation Administration Technical Center, Atlantic City, NJ, 1983. Retrieved from https://apps.dtic.mil/dtic/tr/fulltext/u2/a193533.pdf
[42] Buckley, E. P., O'Connor, W. F. and Beebe, T. *A Comparative Analysis of Individual and System Performance Indices for the Air Traffic Control System*. Department of Transportation, Federal Aviation Administration, National Aviation Facilities Experimental Center, Systems Research and Development Service, Atlantic City, NJ, 1969. Retrieved from https://apps.dtic.mil/docs/citations/AD0710795
[43] Wickens, C. D., Mavor, A. S. and McGee, J. P. *Flight to the Future: Human factors in Air Traffic Control*. National Research Council, National Academies Press, 1997.
[44] Shubik, M. Bibliography on simulation, gaming, artificial intelligence and allied topics. *Journal of the American Statistical Association*, 55, 292 (1960), 736-751. https://doi.org/10.2307/2281597
[45] Wang, R., DeMaria, S. J., Goldberg, A. and Katz, D. A systematic review of serious games in training health care professionals. *Simulation in Healthcare*, 11, 1 (2016), 41-51. https://doi.org/10.1097/sih.0000000000000118
[46] Faria, A. J., Hutchinson, D., Wellington, W. J. and Gold, S. Developments in business gaming: A review of the past 40 years. *Simulation & Gaming*, 40, 4 (2008), 464-487. https://doi.org/10.1177/1046878108327585
[47] Kriz, W. C. Types of gaming simulation applications. *Simulation & Gaming*, 48, 1 (2017), 3-7. https://doi.org/10.1177/1046878117689860
[48] Rybing, J., Prytz, E., Hornwall, J., Nilsson, H., Jonson, C.-O. and Bang, M. Designing a digital medical management training simulator using distributed cognition theory. *Simulation & Gaming*, 48, 1 (2016), 131-152. https://doi.org/10.1177/1046878116676511
[49] Boone, J. O., Van Buskirk, L. and Steen, J. *The Federal Aviation Administration's Radar Training Facility and Employee Selection and Training*. Federal Aviation Administration Washington DC Office of Aviation Medicine 1980. Retrieved from https://apps.dtic.mil/dtic/tr/fulltext/u2/a093027.pdf
[50] Tuggy, M. L. Virtual reality flexible sigmoidoscopy simulator training: impact on resident performance. *The Journal of the American Board of Family Practice*, 11, 6 (1998), 426-433. https://doi.org/10.3122/jabfm.11.6.426
[51] Craighead, J., Murphy, R., Burke, J. and Goldiez, B. A robot simulator classification system for hri. In *Collaborative Technologies and Systems, 2007. CTS 2007*. Orlando, Florida, 2007. IEEE. https://doi.org/10.1109/CTS.2007.4621743
[52] Yuan, H. B., Williams, B. A., Fang, J. B. and Ye, Q. H. A systematic review of selected evidence on improving knowledge and skills through high-fidelity simulation. *Nurse Education Today*, 32, 3 (2012), 294-298. https://doi.org/10.1016/j.nedt.2011.07.010

[53] Alexander, A. L., Brunyé, T., Sidman, J. and Weil, S. A. *From Gaming to Training: A Review of Studies on Fidelity, Immersion, Presence, and Buy-in and their Effects on Transfer in PC-based Simulations and Games*. 2005. Retrieved from https://pdfs.semanticscholar.org/3475/e2b1a4093661fe4c82ef9b41fab549f3ea35.pdf
[54] Kuutti, K. and Bannon, L. J. The turn to practice in HCI: towards a research agenda. *Proceedings of the 32nd Annual ACM Conference on Human factors in Computing Systems* (2014), 3543-3552. https://doi.org/10.1145/2556288.2557111
[55] Rudinsky, J. and Hvannberg, E. T. Consolidating requirements analysis models for a crisis management training simulator. In *8th International Conference on Information Systems for Crisis Response and Management: From Early-Warning Systems to Preparedness and Training, ISCRAM 2011*. Lisbon, 2011. ISCRAM. Retrieved from http://idl.iscram.org/files/rudinsky/2011/897_Rudinsky+Hvannberg2011.pdf
[56] Beyer, H. and Holtzblatt, K. *Contextual Design: Defining Customer-Centered Systems*. Elsevier, 1997.
[57] Field, J., Rankin, A., Pal, J. v. d., Eriksson, H. and Wong, W. Variable uncertainty: Scenario design for training adaptive and flexible skills. *Proceedings of the 29th Annual European Conference on Cognitive Ergonomics* (2011), 27-34. https://doi.org/10.1145/2074712.2074719
[58] Barnett, J., Wong, B. W., Adderley, R. and Smith, M. *SPURS: A framework towards scenario planning for unexpected events, response, and startle using research, horror films, and video games*. In: H. Chaudet, L. Pellegrin and N. Bonnardel(eds). *International Conference on Naturalistic Decision Making* Marseille, France 2013. Arpege Science publishing. Retrieved from https://www.effectivecommand.org/Content/docs/ReferenceArticles/Eleventh%20International%20Conference%20on%20Naturalistic%20Decision%20Making%20(2013).pdf
[59] Martin, G., Schatz, S., Bowers, C., Hughes, C. E., Fowlkes, J. and Nicholson, D. Automatic scenario generation through procedural modeling for scenario-based training. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 53, 26 (2009), 1949-1953. https://doi.org/10.1177/154193120905302615
[60] Praiwattana, P. and El Rhalibi, A. *Survey: Development and analysis of a games-based crisis scenario generation system*. In: A. El Rhalibi, F. Tian, Z. Pan and B. Liu(eds). *International Conference on Technologies for E-Learning and Digital Entertainment.* Lecture Notes in Computer Science, vol. 9654. 2016. Springer. https://doi.org/10.1007/978-3-319-40259-8_8
[61] Grois, E., Hsu, W. H., Voloshin, M. and Wilkins, D. C. Bayesian network models for generation of crisis management training scenarios. In *The Tenth Conference on Innovative Applications of Artificial Intelligence* Madison, Wisconsin 1998. The Association for Advancement of Artificial Intelligence https://www.aaai.org/Papers/IAAI/1998/IAAI98-015.pdf
[62] Carmona, G., Varela-Ortega, C. and Bromley, J. The use of participatory object-oriented bayesian networks and agro-economic models for groundwater management in spain. *Water Resources Management*, 25, 5 (2011), 1509-1524. https://doi.org/10.1007/s11269-010-9757-y
[63] Luo, L., Yin, H., Cai, W., Zhong, J. and Lees, M. Design and evaluation of a data-driven scenario generation framework for game-based training. *IEEE Transactions on Computational Intelligence and AI in Games*, 9, 3 (2017), 213-226. https://doi.org/10.1109/TCIAIG.2016.2541168
[64] Halldorsdottir, G. and Hvannberg, E. Þ. *An Analysis of A Plane Crash Training Event after Action Reviews* University of Iceland, Reykjavik, 2011. Retrieved from https://opinvisindi.is/handle/20.500.11815/1148
[65] Benyon, D. and Macaulay, C. Scenarios and the HCI-SE design problem. *Interacting with Computers*, 14, 4 (2002), 397-405. https://doi.org/10.1016/S0953-5438(02)00007-3
[66] Bjørner, D. *Software Engineering 3: Domains, Requirements, and Software Design*. Springer Science & Business Media, 2006.
[67] Hvannberg, E. T. and Rudinsky, J. *Crisis management training: Techniques for eliciting and describing requirements and early designs across different incident types*. In: C. Stephanidis(eds). *Universal Access in Human-Computer Interaction.* Lecture Notes in Computer Science, vol. 6768. Berlin, Heidelberg, 2011. Springer. https://doi.org/10.1007/978-3-642-21657-2_24
[68] Hays, R. T. *Simulator Fidelity: A Concept Paper*. Army Research Inst. for The Behavioral and Social Sciences, Alexandria, VA. , 1980. Retrieved from https://apps.dtic.mil/docs/citations/ADA101475
[69] Toups, Z. O., Kerne, A. and Hamilton, W. A. The Team Coordination Game: Zero-fidelity simulation abstracted from fire emergency response practice. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 18, 4 (2011), 23. https://doi.org/10.1145/2063231.2063237
[70] Hays, R. T. and Singer, M. J. *Simulation fidelity as an organizing concept.* In: H. R.T. and S. M.J.(eds). *Simulation Fidelity in Training System Design. Recent Research in Psychology.* 1989. Springer.
[71] Dow, C. and Histon, J. An air traffic control simulation fidelity definition and categorization system. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 58, 1 (2014), 92-96. https://doi.org/10.1177/1541931214581020
[72] Dahlstrom, N., Dekker, S., Van Winsen, R. and Nyce, J. Fidelity and validity of simulator training. *Theoretical Issues in Ergonomics Science*, 10, 4 (2009), 305-314. https://doi.org/10.1080/14639220802368864
[73] Zendejas, B., Wang, A. T., Brydges, R., Hamstra, S. J. and Cook, D. A. Cost: The missing outcome in simulation-based medical education research: A systematic review. *Surgery*, 153, 2 (2013), 160-176. https://doi.org/10.1016/j.surg.2012.06.025
[74] Dennis, K. A. and Harris, D. Computer-based simulation as an adjunct to ab initio flight training. *International Journal of Aviation Psychology*, 8, 3 (1998), 261. https://doi.org/10.1207/s15327108ijap0803_6
[75] Lee, A. T. *Flight Simulation: Virtual Environments in Aviation*. Routledge, 2017.
[76] Pagulayan, R. J., Keeker, K., Wixon, D., Romero, R. L. and Fuller, T. *User-centered Design in Games*. CRC Press Boca Raton, FL, 2002.
[77] Buxton, B. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, 2010.

[78] Hamstra, S. J., Brydges, R., Hatala, R., Zendejas, B. and Cook, D. A. Reconsidering fidelity in simulation-based training. *Academic Medicine*, 89, 3 (2014), 387-392. https://doi.org/10.1097/acm.0000000000000130
[79] Herrington, J., Reeves, T. C. and Oliver, R. *A Guide to Authentic E-Learning*. Routledge, 2009.
[80] Petraglia, J. *Reality by Design: The Rhetoric and Technology of Authenticity in Education*. Routledge, 1998.
[81] Druzhinina, O., Hvannberg, E. T. and Halldorsdottir, G. Feedback fidelities in three different types of crisis management training environments. *International Journal of Sociotechnology and Knowledge Development (IJSKD)*, 5, 2 (2013), 45-62. https://doi.org/10.4018/jskd.2013040104
[82] Rudinsky, J., Hvannberg, E. T., Helgason, A. A. and Petursson, P. B. Designing soundscapes of virtual environments for crisis management training. *Proceedings of the Designing Interactive Systems Conference* (2012), 689-692. https://doi.org/10.1145/2317956.2318059
[83] Barsom, E. Z., Graafland, M. and Schijven, M. P. Systematic review on the effectiveness of augmented reality applications in medical training. *Surgical Endoscopy*, 30, 10 (2016), 4174-4183. https://doi.org/10.1007/s00464-016-4800-6
[84] Gallagher, A., Ritter, E. and Satava, R. Fundamental principles of validation, and reliability: Rigorous science for the assessment of surgical education and training. *Surgical Endoscopy and other Interventional Techniques*, 17, 10 (2003), 1525-1529. https://doi.org/10.1007/s00464-003-0035-4
[85] Harrington, C. M., Kavanagh, D. O., Quinlan, J. F., Ryan, D., Dicker, P., O'Keeffe, D., Traynor, O. and Tierney, S. Development and evaluation of a trauma decision-making simulator in Oculus virtual reality. *The American Journal of Surgery*, 215, 1 (2018), 42-47. https://doi.org/10.1016/j.amjsurg.2017.02.011
[86] Liaw, S. Y., Carpio, G. A. C., Lau, Y., Tan, S. C., Lim, W. S. and Goh, P. S. Multiuser virtual worlds in healthcare education: A systematic review. *Nurse Education Today*, 65 (2018), 136-149. https://doi.org/10.1016/j.nedt.2018.01.006
[87] Graafland, M., Schraagen, J. M. and Schijven, M. P. Systematic review of serious games for medical education and surgical skills training. *British Journal of Surgery*, 99, 10 (2012), 1322-1330. https://doi.org/10.1002/bjs.8819
[88] Koivisto, J. M., Haavisto, E., Niemi, H., Haho, P., Nylund, S. and Multisilta, J. Design principles for simulation games for learning clinical reasoning: A design-based research approach. *Nurse Education Today*, 60 (Jan 2018), 114-120. https://doi.org/10.1016/j.nedt.2017.10.002
[89] Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R. V. and Hart, J. C. The CAVE: Audio visual experience automatic virtual environment. *Commun. ACM*, 35, 6 (1992), 64-72. https://doi.org/10.1145/129888.129892
[90] Bowman, D. A., Gabbard, J. L. and Hix, D. A survey of usability evaluation in virtual environments: Classification and comparison of methods. *Presence: Teleoper. Virtual Environ.*, 11, 4 (2002), 404-424. https://doi.org/10.1162/105474602760204309
[91] Gorbanev, I., Agudelo-Londoño, S., González, R. A., Cortes, A., Pomares, A., Delgadillo, V., Yepes, F. J. and Muñoz, Ó. A systematic review of serious games in medical education: Quality of evidence and pedagogical strategy. *Medical Education Online*, 23, 1 (2018), 1438718. https://doi.org/10.1080/10872981.2018.1438718
[92] Hvannberg, E. T., Halldorsdottir, G. and Rudinsky, J. *The usefulness of method-resources for evaluating a collaborative training simulator*. In: J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque and M. Winckler(eds). *Human-Computer Interaction – INTERACT 2015*. Bamberg, 2015. Springer International Publishing. https://doi.org/10.1007/978-3-319-22668-2_9
[93] Hvannberg, E. T. Identifying and explicating knowledge on method transfer: A sectoral system of innovation approach. *Universal Access in the Information Society*, 14, 2 (2015), 187-202. https://doi.org/10.1007/s10209-013-0340-1
[94] Malerba, F. Sectoral systems of innovation: A framework for linking innovation to the knowledge base, structure and dynamics of sectors. *Economics of Innovation and New Technology*, 14, 1-2 (2005), 63-82. https://doi.org/10.1080/1043859042000228688
[95] Rudinsky, J. and Hvannberg, E. T. Transferability of voice communication in games to virtual teams training for crisis management. *International Journal of Sociotechnology and Knowledge Development (IJSKD)*, 9, 1 (2017), 1-25. https://doi.org/10.4018/IJSKD.2017010101
[96] Wiberg, C. Game-inspired architecture and architecture-inspired games. *Interactions*, 25, 2 (2018), 68-70. https://doi.org/10.1145/3177814
[97] Shepherd, I. D. H. and Bleasdale-Shepherd, I. D. The design-by-adaptation approach to universal access: Learning from videogame technology. *Universal Access in the Information Society*, 10, 3 (2011), 319-336. https://doi.org/10.1007/s10209-010-0204-x
[98] Harris, D. and Harris, F. J. Evaluating the transfer of technology between application domains: A critical evaluation of the human component in the system. *Technology in Society*, 26, 4 (2004), 551-565. https://doi.org/10.1016/j.techsoc.2004.08.003
[99] Jabbour, N. and Snyderman, C. H. The economics of surgical simulation. *Otolaryngologic Clinics of North America*, 50, 5 (2017), 1029-1036. https://doi.org/10.1016/j.otc.2017.05.012
[100] Sutcliffe, A. G., Poullis, C., Gregoriades, A., Katsouri, I., Tzanavari, A. and Herakleous, K. Reflecting on the design process for virtual reality applications. *International Journal of Human–Computer Interaction* (2018), 1-12. https://doi.org/10.1080/10447318.2018.1443898
[101] Hvannberg, E. T., Halldorsdottir, G. and Rudinsky, J. Exploitation of heuristics for virtual environments. *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design* (2012), 308-317. https://doi.org/10.1145/2399016.2399065
[102] Latour, B. *Science in Action: How to follow Scientists and Engineers through Society*. Harvard University Press, 1987.
[103] Borodzicz, E. P. The missing ingredient is the value of flexibility. *Simulation & Gaming*, 35, 3 (2004), 414-426. https://doi.org/10.1177/1046878104266221

[104] Cook, D. A. and Hatala, R. Validation of educational assessments: A primer for simulation and beyond. *Advances in Simulation*, 1, 31 (2016), 1-12. https://doi.org/10.1186/s41077-016-0033-y
[105] Law, E. L.-C. Evaluating the downstream utility of user tests and examining the developer effect: A case study. *International Journal of Human-Computer Interaction*, 21, 2 (2006), 147-172. https://doi.org/10.1207/s15327590ijhc2102_3
[106] Amin, R., Tang, J., Ellejmi, M., Kirby, S. and Abbass, H. A. Trading-off simulation fidelity and optimization accuracy in air-traffic experiments using differential evolution. *2014 IEEE Congress on Evolutionary Computation (CEC)* (2014), 475-482. https://doi.org/10.1109/CEC.2014.6900573
[107] Følstad, A. Work-domain experts as evaluators: Usability inspection of domain-specific work-support systems. *International Journal of Human–Computer Interaction*, 22, 3 (2007), 217-245. https://doi.org/10.1080/10447310709336963
[108] Alrobaea, R. *Generating a Domain-Specific Inspection Method through an Adaptive Framework*. University of East Anglia, 2015. Retrieved from https://ueaeprints.uea.ac.uk/59678/
[109] Hassan, A., Queudet, A. and Oussalah, M. *Evolution style: framework for dynamic evolution of real-time software architecture*. In: B. Tekinerdogan, U. Zdun and A. Babar(eds). *Software Architecture. ECSA 2016.* Lecture Notes in Computer Science, vol. 9839. 2016. Springer, Cham. https://doi.org/10.1007/978-3-319-48992-6_12
[110] Alenezi, M. and Almustafa, K. Empirical analysis of the complexity evolution in open-source software systems. *International Journal of Hybrid Information Technology*, 8, 2 (2015), 257-266. https://doi.org/10.14257/ijhit.2015.8.2.24
[111] Guzman, E., El-Haliby, M. and Bruegge, B. Ensemble Methods for App Review Classification: An Approach for Software Evolution (N). *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (2015), 771-776. https://doi.org/10.1109/ASE.2015.88
[112] Vogel-Heuser, B., Fay, A., Schaefer, I. and Tichy, M. Evolution of software in automated production systems: Challenges and research directions. *Journal of Systems and Software*, 110 (2015), 54-84. https://doi.org/10.1016/j.jss.2015.08.026
[113] Hollan, J. D., Hutchins, E. L. and Weitzman, L. STEAMER: An interactive inspectable simulation-based training system. *AI Magazine*, 5, 2 (1984), 15-27. https://doi.org/10.1609/aimag.v5i2.434
[114] Fiedrich, F. and Burghardt, P. Agent-based systems for disaster management. *Commun. ACM*, 50, 3 (2007), 41-42. https://doi.org/10.1145/1226736.1226763