

Evaluating the Usability of Authoring Environments for Serious Games

Simulation & Gaming
2017, Vol. 48(4) 553–578
© The Author(s) 2017
Reprints and permissions:
sagepub.com/journalsPermissions.nav
DOI: 10.1177/1046878117705249
journals.sagepub.com/home/sag



Aad Slootmaker¹, Hans Hummel¹, and Rob Koper¹

Abstract

Background. The EMERGO method and online platform enable the development and delivery of **scenario-based serious games** that foster students to acquire professional competence. One of the main goals of the platform is to provide a user-friendly **authoring environment** for creating virtual environments where students can perform authentic tasks.

Aim. We present the findings of an in-depth qualitative case study of the platform's **authoring environment** and compare our findings on usability with those found for comparable environments in literature.

Method. We carried out semi-structured interviews, with two experienced game developers who have authored a game for higher education, and a literature review of comparable environments.

Findings. The analysis shows that the **usability** of the **authoring environment** is problematic, especially regarding **understandability** and **learnability**, which is in line with findings of comparable environments. Other findings are that authoring is well integrated with the EMERGO method and that **functionality** and **reliability** of the authoring environment are valued.

Practical implications. The lessons learned are presented in the form of general **guidelines** to improve the **understandability** and **learnability** of **authoring environments** for **serious games**.

Keywords

authoring environment, development process, functionality, game design research, learnability, operability, reliability, scenario-based, serious games, understandability, usability

¹Open University of the Netherlands, Heerlen, The Netherlands

Corresponding Author:

Aad Slootmaker, Open University of the Netherlands, Heerlen, The Netherlands.

Email: aad.slootmaker@ou.nl

Serious games (SGs) are considered to provide powerful and attractive ways of acquiring professional competences. However, their use is still limited, because their technical requirements are high, they are: difficult to customize for the educational process, difficult to support, and, the field lacks standards for SG design (Klemke, van Rosmalen, Ternier, & Westera, 2015). In addition, the field also lacks good architecture for SG development (Nadolski, Hummel, Slootmaker, & Van der Vegt, 2012) and simpler authoring tools (Arnab et al., 2012). In the previous decade, our institution experienced similar needs and developed their Efficient Method for Experiential Education EMERGO (a Dutch acronym meaning Efficient Method for Experiential Education), including an online platform for SG development (Nadolski et al., 2007). It intends to both simplify and better support the development and delivery of scenario-based SGs. In this kind of game, students acquire professional competences in complex problem spaces that mimic real-world situations. The scenario describes the problem space and how it should adapt to the students' actions. The platform's authoring environment is used to convert the scenario into platform content.

Evaluations of the platform (Nadolski et al., 2007; Slootmaker, Kurvers, Hummel, & Koper, 2014) show that educators, after having received some instruction, can author most platform components independently and with ease. However, the authoring environment has not yet been evaluated in detail. We believe a deeper understanding is relevant, for both practice and research.

This type of research is part of the field of game design research, as Kultima (2015) examines and discusses it, stating that understanding game studies as design research could deepen our understanding of game design.

This article presents the findings of an in-depth qualitative case study on the usability of the authoring environment and its integration with the EMERGO method. We compare the usability findings with those found in literature for comparable environments. We present the lessons learned in the form of general guidelines to improve the understandability and learnability of authoring environments for SGs.

We first give some information on game design research, game authoring, usability, and comparable studies in the section *Background*. In the *EMERGO* section, we present the EMERGO method and the development, authoring, and debriefing of EMERGO games. In the *Method* section, we explain the method that is followed in order to arrive at our findings in the *Findings* section, in which we also provide practical guidelines for improving the understandability and learnability of authoring environments for serious games. Finally, in the *Conclusion and discussion* section, we present the main conclusions to be drawn from this study.

Background

Game Design Research

Although *game design* is the most popular keyword in game research papers, there is no explicit reflection on notions of *design* and *design research* (Kultima, 2015). In addition, game studies have focused on the *game* and the *player* but not on the *context*

that involves the *design, designer, process, and practice* of the game. According to Kultima, trying to understand game studies as design research would potentially improve our understanding of game design and bridge the epistemic gap between practice and science. She therefore encourages taking design research as theoretical background for future game studies. For possible theoretical background to be utilized, the author refers to Cross (2007), who distinguishes three areas of design research that are based on, respectively, people, process, and products: design epistemology (study of designers' professional theories), design praxeology (study of the practices and processes of design), and design phenomenology (study of the form and configuration of artifacts). Kultima (2015) argues that the multidisciplinary domain of game studies has been unsuccessful in addressing the latter two areas of design research.

To facilitate future studies and understanding of game design practice, Kultima and Sandovar (2016) proposed a framework of game design values. Their framework utilizes three design theory frameworks from architectural and industrial design: Lawson's Guiding Principles, Schön's Appreciative Systems, and Holm's Designers' Distinctive Design Values. All three frameworks indicate that designers never start from scratch but already have their own motivations, their own reasons for wanting to design, and their own sets of beliefs, values, and attitudes. The author divides game design values into nine categories: (1) Value of Player Centricism, (2) Casual Game Design Values, (3) Traditional Game Design Values, (4) Societal Impact and Cultural Values, (5) Value of Artistic Expression, Innovation, and Experimentation, (6) Values of Production and Creation Process, (7) Ludological Values, (8) Values of Independency, and (9) Commercial Values. The first three categories are more oriented toward players and involve values like usability and playability, flexibility and simplicity, and immersion, challenge, and competition. The fourth category is oriented toward society and culture and involves values like ethics and morality and cultural diversity and tradition. Categories five through eight are more oriented toward game developers and involve values like visual design and aesthetics, development as a challenge, collaboration, value of teamwork, value of game mechanics, and autonomy and artistic freedom. The last category is oriented toward business and involves values like economic success. The author states that, if viewed from a general perspective, there is no single design value that is more important than another. However, this case-based research focuses on game developers and more specifically on the game authoring process that can be classified under the category Values of Production and Creation Process.

Authoring Leisure Games

Current video games are often complex, immersive games that are developed through large and costly projects and involve many specialized developers who use specific development tools. The flexibility, productivity, and usability of these dedicated tools are decisive success factors. Game designs are complex when they entail many game elements to be classified under four categories: story (narrative), aesthetics (look and feel), technology (materials and interactions), and mechanics (fostering game rules and interactivity) (Schell, 2008).

The development of complex games requires comprehensive and dedicated development tools. Hartson and Pyla (2012) identified *two types of system complexity* that may influence usability: (1) *interaction complexity*, which is related to the intricacy or elaborateness of user actions, including cognitive load; and (2) *work domain complexity*, which is related to the degree of intricacy and the technical nature of the corresponding field of work (e.g., game development). Systems with high interaction and high work domain complexity are more likely to have low usability. This is in line with Oja (2010), who stated that usability is even more critical for complex software development.

No single definition for usability currently exists. Nielsen (1993) defined usability by its quality of five components: (1) learnability (for novice users), (2) efficiency (amount of time to accomplish task), (3) memorability (for frequent users), (4) errors (number, severity, recoverability), and (5) satisfaction (pleasantness). ISO/IEC (2011) defined usability as the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. A more recent concept is user experience (UX), which involves the effects of usability factors, usefulness factors (how useful a tool is for a task), and emotional impact (broader than Nielsen's satisfaction) and strongly depends on the context of a usage by a particular user (Hartson & Pyla, 2012). Not all UX aspects are equally important for every type of user. Paakkanen (2014) finds that experienced video game developers find effectiveness (usefulness) most important. For novice users, this probably would be the ease of use (usability).

Popular video game development tools, like Unity3D, Unreal Engine 4, and Cry Engine, have a high interaction and work domain complexity. They include many sophisticated editors (e.g., AI editors) that all contribute to their complexity and steep learning curve. Patrasitidecha (2014) found that, out of 16 3D mobile game engines, Unity3D is easiest to learn but has a relatively low usability, which most likely is due to Unity3D's high complexity.

Popular console and web-based editors, like Super Mario Maker and Scratch, have low interaction and work domain complexity. Super Mario Maker is a very user-friendly editor for the Wii U console. Its interaction complexity is low, because games can be easily created by dragging and dropping objects from a tool palette. Objects can even be combined to get a new object that shows combined behavior. Its work domain complexity is also low, because the number of objects and possible manipulations on it are limited, and objects have built-in behavior, so no entry of game rules is needed. This low complexity contributes to its high usability. Scratch is a massively used, low threshold web-based game editor. Its interaction complexity is low, because it makes use of blocks that can be dragged and dropped and connected to each other to create the game flow and even to create the game rules. Its work domain complexity is also low, because the number of different blocks is limited. Again, low complexity contributes to high usability.

Besides interaction and work domain complexity, design decisions may also influence tool usability. Murray (1999, 2004, 2015) investigated design trade-offs for authoring tools. Increasing the flexibility (the ability to author a diversity of game types), breadth (of the domains supported), or depth (of the models to author) of a tool

usually comes at the cost of usability. In addition, learnability and productivity are often in conflict, because simplicity for novice users means less powerful features that foster productivity for experienced users.

Conditions during development and implementation of a tool may also influence its usability. What is the budget? What is the time schedule? Are there great risks involved? Are the right people with the right skills available? What is the expected number of authors? In case of setbacks or a small number of authors, the priority will likely be to deliver a tool that works, so usability aspects, efficiency, and satisfaction will be at the expense of effectiveness.

Authoring Serious Games

Unlike leisure games, serious games should support learning. Dede (2009), Clark and Mayer (2011), and Thillainathan and Leimeister (2014) stated that the learner should be in control and that learning should be situated and authentic, possibly based on a didactical model or approach and should support transfer of learned skills. To be able to give relevant support, guidance, and feedback, the game should keep progress of and assess the learner and should adapt to the learner's learning strategies and skills. The game may assess the learner on performance, emotion, motivation, and on personality aspects and may adapt to the learner on the micro and macro levels (Kickmeier-Rust & Albert, 2010, 2012; Kickmeier-Rust, Mattheiss, Steiner, & Albert, 2011). Micro-level adaptation is embedded in the game flow and leads to, e.g., giving the learner advice or feedback or motivating or urging him. Macro-level adaptation leads to adjustments of either the game flow or the game's pace or intensity.

Murray (2004, 2015) did some important work on the *complexity of authoring tools* and stated that the complexity level of an authoring tool should match the complexity capacity of its user. He identified four types of complexity. *Interface and tool complexity* is related to the number of editor features and components. The more features and components, the more difficult it is to manage and combine them. *Object complexity* is related to the number of abstract concepts whose definitions and uses are not obvious. For instance, it is more difficult to understand and explain an abstract concept like feedback, when compared to a more concrete object like an image. *Structural complexity* is related to the number of complex structures of linked objects. Creating and maintaining such structures is cognitively challenging. *Dynamic complexity* is related to the number of laws, rules, mechanisms, or influences that contribute to change, which may lead to many possible student paths that are difficult to test and debug.

In addition, Murray identifies *five possible types of users with different complexity capacity*. *Teachers* have a low complexity capacity, so they cannot be expected to use complex authoring tools. This is in line with Theodosiou and Karasavvidis (2015), who found that student teachers struggle to incorporate critical game elements and have major difficulties in connecting game elements effectively. *Domain experts and content developers* have a medium complexity capacity, though they may have little practical or theoretical knowledge of pedagogy. *Instructional designers and learning theorists* have a medium complexity capacity too, though they may not have the time

to dedicate to a steep tool learning curve. *Knowledge engineers and game developers* have a medium to high complexity capacity, because they are trained for representing knowledge in a computationally usable fashion. *Computer scientists and software developers* have a high complexity capacity, because they are used to design and debug structural and procedural models. Only the last two types of users can be expected to manage sophisticated authoring tasks.

Related Work

Other authors also evaluated the usability of authoring environments for SGs. Mehm, Göbel, and Steinmetz (2012) evaluated STORYTEC, an authoring tool that integrates the work of game designers, pedagogues, artists, and domain experts into one unified authoring tool. Most usability aspects were found to be average. Only so-called *self-descriptiveness* (the dialogue should make clear what the user should do next) (ISO, 2006), which relates to understandability, and error tolerance, which relates to operability, were rated lower than average.

Van Est, Poelman, and Bidarra (2011) evaluated SHAI, a (prototypical) scenario editor for simulation games that enables instructors to arrange scenario building blocks to match individual trainees' needs and to make real-time adjustments. Usability was found to be poor. Shortcomings relate to understandability ('options are too complex', 'graphics are unclear', 'large scenarios are difficult to comprehend'), learnability ('better descriptions are needed'), operability ('keyboard shortcuts are missing'), and user interface aesthetics ('better presentation is needed').

Marchiori et al. (2012) evaluated WEEV, a method and system for educational adventure game authoring, and identified many usability problems related to understandability ('part of the system is complex to use', 'example games are needed to understand the purpose of the system') and learnability ('a guided tutorial is needed to help novice users').

Gaeta et al. (2014) evaluated an authoring tool for the creation of stories that support learning in an emergency context and found usability to be relatively low.

EMERGO

We developed the EMERGO method and online platform (Nadolski et al., 2007) to simplify and better support the development and delivery of scenario-based SGs. In this kind of game, learners are confronted with realistic, ill-defined problems, often allowing multiple solutions and requiring application of necessary methodologies or tools and collaboration with fellow learners (Westera, Nadolski, Hummel, & Wopereis, 2008). The platform's authoring environment offers 22 components that support different (didactical) functions that should be present in scenario-based SGs. In addition, the platform offers environments to play the developed games, to monitor students, and to manage users and game runs. EMERGO has been used to develop 24 games for all kinds of disciplines. It supports the acquisition of four out of five capability types, as defined by Gagné (1985): intellectual skills, cognitive strategies, verbal

information, and attitudes. Motor skills are not (yet) supported. The online platform is Open Source and is available on SourceForge (EMERGO, 2013).

Earlier and more superficial evaluations of the authoring environment show that educators, after receiving some instruction, could use most components independently and with ease (Nadolski et al., 2007; Slootmaker et al., 2014). One component could not be used independently, and two components were not easy to use. This makes us question why the usability of some components is lower than of others, how we could improve this, how this is related to component complexity, and if it is related to the conversion of the scenario into game content. The research goal of this case study is to evaluate in detail the usability of the EMERGO authoring environment and integration of authoring with the EMERGO method. We expect that the usability evaluation will enable us to derive some guidelines for increasing the usability of the environment.

This case study addresses two areas of design research that are still underrepresented in game studies (Kultima, 2015): design praxeology (i.e., the development process using the EMERGO method) and design phenomenology (i.e., the EMERGO authoring environment). The study deals with the game design value category *Values of Production and Creation Process*, as proposed by Kultima and Sandovar (2016), which contains values like *Technological advancement*, *Development as a challenge*, *Collaboration and value of teamwork*, and *Open source ideology*.

In this article, we use usability as defined by ISO/IEC 25010:2011 (ISO/IEC, 2011). It is one out of eight software quality characteristics and is defined as the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. The other characteristics are functionality, reliability, performance efficiency, compatibility, security, maintainability, and portability. Usability is further subdivided into six aspects: understandability, learnability, operability, user error protection, user interface aesthetics, and accessibility. Note that, for better readability, we replace the ISO/IEC characteristic *functional suitability* with functionality and the characteristic *appropriateness recognizability* with understandability.

In the following sections, we will describe development, authoring, and debriefing for EMERGO games and the EMERGO authoring environment itself.

Developing EMERGO Games

The use of the EMERGO authoring environment is embedded in the EMERGO method. This method comprises five phases and (although based on ADDIE) recommends using iterations, like a Unified Process approach with cycles that prevent overspending and minimizes risks or failures.

During the analysis phase, the development team formulates answers to a list of standard questions. Answers are used as input for a global description of the game that includes learning goals and competencies to achieve. During the design phase, the method supports the team in the creative process of writing a scenario in three steps. First, the team formulates which activities have to be accomplished, why, when, where, and in what order, if needed. Activities are formulated as location plans, using

the template *Where the student will...<description of the activity>*. Second, the team identifies: (i) with whom activities must be done and with what materials and tooling, (ii) when activities are completed and how this is assessed, and (iii) which feedback is given and when, and in what form and by whom. Third, the team describes each activity exhaustively in terms of its required materials and tooling. In this stage, it becomes evident whether materials are already available or still need to be developed and whether the scenario can be realized with available platform components, or if it needs new components or even a new game skin. During the development phase, the authoring environment is used to convert the scenario and materials into game content. If needed, new components or skin are developed, film recordings are made, and other materials like documents or images are developed. During the implementation phase, the game is deployed to students and educators (for monitoring), and during the evaluation phase, the game is evaluated.

An EMERGO development team consists of content matter experts, educational technologists, interaction designers, and ICT developers. Educational technologists and content matter experts write the global description and scenario of the game and involve other team members to check for feasibility. Interaction designers and ICT developers develop graphical assets and, if needed, new components or skin. If film recordings are needed, the team temporarily is reinforced with cameramen, actors or experts, and video editors. Initially, our objective was for educational technologists and content matter experts to do all authoring, but actual practice shows that game script authoring is mostly too complicated and is then done by an ICT developer, which is in line with Murray (2015), who states that the latter user type has a higher complexity capacity than the former.

Authoring EMERGO Games

In a typical EMERGO game, the student acts as a PC (Playing Character) and enters an authentic environment, where he works as a trainee. He can navigate to different locations, where he finds NPCs (Non-Playing Characters), like his supervisor, colleagues, experts, or specialists, or can attend interviews or meetings (see Figure 1). In the environment, he has a tablet with apps, e.g., a task overview, a resources app, an (in-game) email app, or an app to conduct tests. He also has a memo recorder, to record interesting parts of interviews and meetings, and a notepad to make contextualized notes.

The student gets tasks from his supervisor or other NPCs, either in person or by mail. He can be assessed on every action he performs, e.g., which interviews he attends, which questions he asks, which resources he consults, or which mail he sends. In addition, he can be assessed, e.g., using tests that enable measuring foreknowledge and performance. Depending on his actions, the game can adapt the environment at the micro level, e.g., by sending mail, showing an alert, changing an NPC reaction, releasing new resources or new interview questions, or changing an answer to a question, or at the macro level, e.g., by providing new or alternative tasks. The student gets feedback on his performance by NPCs in person, by mail, as screen text, or in tests. This



Figure 1. A patient being interviewed. At the bottom left, we see the tablet, and at the bottom right, we see the memo recorder and notepad.

feedback can incorporate mail attachments or the release of resources, such as worked-out examples or expert reports. The student gets navigation support through alerts, e.g., reminders for meetings or instructions for where to go next.

The EMERGO authoring environment offers 22 components to realize the above kind of scenarios (see Table 1).

The components support eight different (didactical) functions: present and adapt environment (E), assign tasks and provide overview (T), present knowledge (K), assess learner (A), provide feedback (F), support processing of information (P), support collaboration (C), and support navigation (N). Note that one component may serve several functions and that one function may involve several components. For instance, the *conversations* component can be used to assign a task, to present knowledge, or to provide feedback. And the *script* component should assess the learner to trigger the *conversations* component to give the right feedback.

The components allow much freedom in the way the environment is presented, how tasks are assigned, if they must be executed in a certain order, how they are assessed, and how feedback is provided and thus support a wide range of game scenarios. Of course, the metaphor of an environment with locations and the available components put constraints on the end form of the game, but this partially can be overcome by adding new components or a new game skin.

Based on Murray's (2004, 2015) four tool-complexity types, we identified three complexity levels: low, medium, and high. This allowed us to relate components' complexity to the complexity capacity of its user (Murray, 2015). Low complexity components have a low interface, object, and structural and dynamic complexity. They either have only a few configuration options or an obvious and simple data model without dynamics. Medium complexity components have a medium interface, object, and/or structural complexity, but a low dynamic complexity. The *navigation*, *conversations*, *email*, *assessments*, and *items* components comply with this condition. They all have

Table 1. EMERGO Components and Their Possible Functions and Complexity.

Component	Description	Functions	Complexity
Navigation	Enable spatial navigation through the game	E	Medium
Conversations	Enable communication with NPC's using video or text	ETKF	Medium
Notepad	Enable making contextualized notes	EP	Low
Memo recorder	Enable recording of conversations	EP	Low
Alerts	Provide popup texts	EFN	Low
Notifications	Provide (accumulated) embedded texts	EFN	Low
Scores	Provide score overview	EF	Low
Profile	Enable sharing profile with PC's	EC	Low
Chat	Enable communication with PC's	EC	Low
Tablet	Enable choosing apps	E	Low
Tasks	Provide task (completion) overview. App	ET	Low
Resources	Enable consulting resources. App	EKF	Low
Email	Enable communication with NPC's and between PC's. App	ETKFC	Medium
Assessments	Enable conducting tests. App	EAF	Medium
Logbook	Provide overview of notes. App	EP	Low
Memo player	Enable playing back of recordings. App	EP	Low
Google maps	Enable inspecting maps with markers. App	EK	Low
Directing	Enable analyzing communication between NPC's. App	EP	Low
Game manual	Provide help on game interface. App	EN	Low
Items	Define questions to be used in tests	EAF	Medium
States	Define states to be used by the Script or Scores component	A	Low
Script	Define rules to assess the learner and adapt the game at the micro and macro levels	ETKAFPCN	High

a medium object complexity, because they include abstract concepts whose definitions and uses are not obvious. In addition, the *navigation* component has a medium interface complexity, because it includes a larger number of authoring features, and the *navigation*, *conversations*, and *items* components have a medium structural complexity, because they include complex structures of linked objects. High complexity components have a medium or high interface, object, structural, and dynamic complexity. Only one component, the *script* component, complies with this condition. It has a medium interface and a high object complexity, and depending on the number of game

rules and their interrelations in the game scenario, it may have a high structural and/or dynamic complexity.

Depending on their individual complexity capacity, content matter experts will use low or medium complexity components that involve knowledge presentation. Educational technologists will mostly use medium complexity components that involve task assignment, assessment, and feedback. ICT developers will use high complexity components, although some educational technologists may also consider using them.

Object, structural, and dynamic complexities are related to the SG domain and therefore are difficult to influence. However, interface complexity is related to the way user tasks are translated into usable interfaces and therefore leaves room for improvement, which is the motive for our usability evaluation.

Debriefing EMERGO Games

The EMERGO platform supports the design of in-game and post-game debriefing for the reflection on and the sharing of the game experience to turn it into learning (Crookall, 2010).

For the design of in-game reflection on the game experience, game authors can use components that support giving feedback or processing of information. The *conversations* component can be used for reflection on a task with a supervisor or reflection on the domain with an expert. The *resources* component can be used to provide additional reflection materials. The *email* component can be used for asking the student to send in a reflection document like a report or for commenting on a student's task process or outcomes by NPCs or PCs (fellow students or educators), including attachments like expert outcomes that can be used for reflection. If an educator has a PC role, he can support or moderate students' reflection during the game. He can even do this by impersonating an NPC. The *assessments* component can be used to reflect on learning, the *logbook* to reflect on notes made, and the *memo player* and *directing* component to reflect on NPC communication, e.g., a patient interview.

For the design of in-game sharing of the game experience, game authors can use the *email* and *chat* components that support communication with fellow students or educators.

For post-game reflection on and the sharing of the game experience, educators can organize debriefing sessions with students where all student data can be used as input to foster discussion. Educators can use the platform or ask administrators to provide overviews of students' performance and to reflect on developed games.

The EMERGO Authoring Environment

One of our main goals was to develop a user-friendly, reliable, and stable authoring environment that would enable efficient development of scenario-based SGs by offering a set of common components. We set up functional and non-functional requirements (see Appendix 1) that laid the foundation for the structure and the working of the environment, which comprises four pages (see Figure 2).

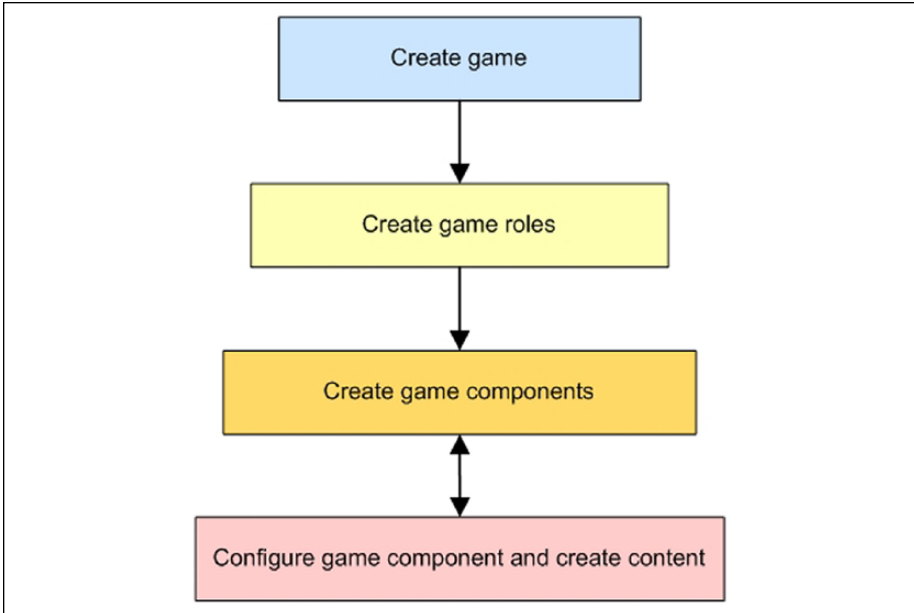


Figure 2. Authoring process.

The *games* page shows an overview of games and allows CRUD (Create, Read, Update, Delete) operations on and import and export of games.

The *game roles* page shows an overview of game roles for a chosen game and allows CRUD operations on game roles. A game role can be either a PC or an NPC.

The *game components* page shows an overview of game components for a chosen game and allows CRUD operations on and import, export, and copying of game components. Most components allow instantiation of multiple game components, which enables thematically arranging content, e.g., one *conversations* component per interviewee.

The *game component content* page (see Figure 3) shows the game component configuration and content and allows CRUD operations on and drag, drop, and copying of game content. Configuration implies setting initial properties – e.g., is the game component initially present? Content is presented as a tree and authored using a single game component content editor whose working is determined by the component definition that defines possible content elements (e.g., locations or backgrounds), their hierarchy, which content can be entered, and their properties. A content element is edited in a pop-up dialogue that validates entered content. The *script* component is authored the same way. Script conditions are added as root tree items and will be triggered by property changes that are initiated by student actions, timers, or script actions. Script actions are children of a script condition and are executed if the condition is triggered. They change properties that may adapt the player environment or the game script itself.

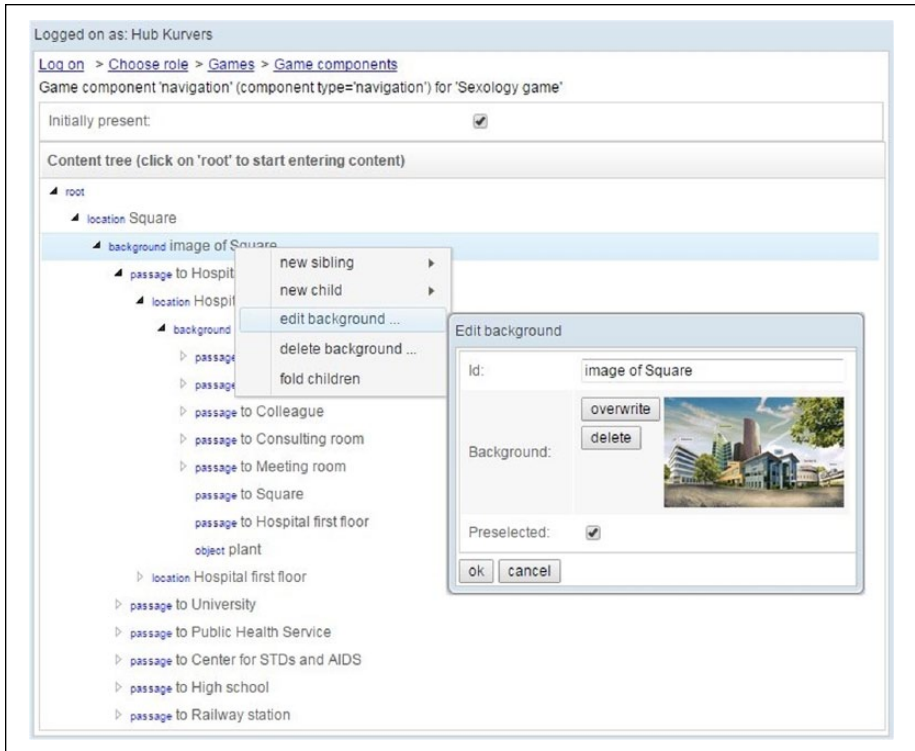


Figure 3. Game component content page for the Navigation component.

To enter all content, authors will switch between the *game components* page and the *game component content* page. Game and game components can be previewed and tested in the player environment.

Method

As subject for our evaluation, we choose the development of a game on Sexology, one of the most recently developed games. It is a typical example of an EMERGO game, and it uses most EMERGO components, 14 out of 22 available.

Participants

Two experienced male game developers, who developed several other EMERGO games before, completed the game authoring. The first author is an educational technologist (without any technical background), who also lead the project and wrote the scenario. The second author is an ICT developer, who also developed new game components. The educational technologist authored the *conversations*, *notepad*, *alerts*,

tablet, resources, email, and logbook component. The ICT developer authored the *navigation, memo recorder, tasks, memo player, directing, game manual, and script* component.

Data Collection Method

As data collection method, we chose semi-structured interviews (Bryman, 2012). Strengths of this method are that it has a high validity, because interviewees are able to talk about something in detail and depth, and a high flexibility, because it allows complex questions and issues to be discussed. Weaknesses of this method are that it is not very reliable, because it is difficult to exactly repeat an interview, and that the findings are difficult to generalize. We found other data collection methods, like questionnaires, observation studies, think- or talk-aloud protocols, focus groups, automated collection of heat maps, or a combination of methods, not appropriate. Questionnaires, observation studies, and think- or talk-aloud protocols give too little detail and depth and are less flexible. Focus groups are more suitable for larger groups and bear the risk that opinions are not expressed equally. Automated collection of heat maps is not possible, because the EMERGO platform does not log the authoring process. A combination of methods is not appropriate, because the aforementioned methods are not appropriate.

We prepared the interviews by setting up an interview guide with themes and related questions (see Appendix 1). The themes are: (1) the author's general impression of the authoring environment, (2) the requirements for the environment, (3) the components used for the Sexology game, and (4) the development process.

Procedure

The same interviewer interviewed the two game authors separately. Both interviews were conducted about a year and a half after the game was developed, lasted about two and a half hours, and were recorded with consent. The spoken language was native, so interviewees could better express themselves. Interviewee and interviewer together walked through the pages of the authoring environment and the 14 EMERGO components to recall working with it in a natural setting.

For our data analysis, we first used the interview recordings to make notes per interviewee and per theme. Second, we identified issues and counted related remarks. Third, we related these issues to usability aspects and other ISO/IEC software quality characteristics and to aspects of the development process. As a last step, we collected suggested improvements to be able to set up general usability guidelines.

Findings

We present the findings related to our original evaluation goal, which was to evaluate the usability of the authoring environment and the integration of authoring in the EMERGO method. As interviewees also made remarks related to other software

quality characteristics and the development process itself, we present these findings as well. A general finding is that interviewees' remarks do not contradict each other. We end with general usability guidelines for authoring environments for SGs.

Usability of the Authoring Environment

We give a general impression per interviewee that is composed of their striking literal remarks.

The educational technologist. 'If you see it for the first time, you think: 'What on Earth do I have to do here?' However, things fairly quickly become clear if you get some peer support, and are somewhat familiar in scenario writing. You don't have to be an ICT developer, but it is good to get an idea of the different layers you can distinguish in authoring, where switching properties on and off is close to programming. If you get deeper, it conceptually becomes more complicated. If you work with it somewhat longer, almost all components are a piece of cake, except scripting. Then it becomes Spartan, because it is not always intuitive.'

The ICT developer. 'You actually see a somewhat empty environment. It has a *new* button, but further you see little information, especially for someone who knows nothing about it. You get no location or context-specific help. You might build in that you can get some explanation on every screen, on the purpose, what you can do exactly, and how you might proceed. If you understand the editor, it works fine. However, entering game script requires concentration to prevent errors.'

Interviewees made remarks that can be related to three out of six usability aspects, namely *understandability*, *learnability*, and *operability*. No remarks can be related to user error protection, user interface aesthetics, or accessibility. Both interviewees identified most issues. ISO/IEC defines understandability as the degree to which users can recognize whether a product or system is appropriate for their needs, learnability as the degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk, and satisfaction in a specified context of use, and operability as the degree to which a product or system has attributes that make it easy to operate and control.

The understandability of the authoring environment is problematic. Although interviewees find the distribution in pages and navigation through them obvious, they quite often find the used terminology unclear and not fitting their expectations (16 remarks) and find it unclear as to why, when, and where certain options are present, or why two options offer the same functionality (25 remarks). Less problematic is that interviewees miss examples of scenarios, games, and game components (three remarks).

The learnability of the authoring environment is problematic. Interviewees miss on-screen guidance and clear instruction on all pages and pop-up dialogues (17 remarks) and miss information on didactics and use of the components, their mutual

dependencies, and the order of entering component content (10 remarks). For a large part, missing guidance, instruction, and information can be found in a comprehensive authoring manual, but it is partly outdated, and searching in a manual for the right help is laborious.

The operability of the authoring environment is somewhat problematic. Interviewees mention that available components are not filtered on the chosen game skin (one remark), file names cannot contain special characters (two remarks), objects cannot be easily positioned on the screen (one remark), and the preview option does not always function as expected (two remarks). However, interviewees like drag and drop and the uniform input control for URLs and files.

Interviewees made no remarks about the *notepad*, *memo recorder*, *tablet*, *logbook*, and *memo player* components, probably because these components only need some configuration and no authoring of game content.

We give an impression per interviewee and authored component that is composed of their striking literal remarks.

The educational technologist. 'It is a lot of work, but not really complicated. After some time, it is no longer a problem to work with. It cannot be made easier. You just have to copy/paste from Word and work very precisely.' (*Conversations*). 'No problem, I understood everything.' (*Alerts*). No problem at all. The preview option is indispensable.' (*Resources*). 'I had some trouble to understand the distinction between sent and received mails. I further had no problems.' (*Email*).

The ICT developer. 'Part of authoring does not seem logical or is unclear, unless you imagine it visually. However, for a large part authoring is straightforward, once you know how it functions and what it stands for.' (*Navigation*). 'No problems, pretty simple.' (*Tasks*). 'Explanation is missing.' (*Directing*). 'It is somewhat difficult to understand the distinction between the three types of resources. The rest is obvious.' (*Game manual*). 'It is not entirely clear what every property stands for and some menu options are unclear. You can implement different solutions for the same problem. This all made authoring more difficult, especially if I had not used the component for a while. In itself I found authoring convenient.' (*Script*).

The Development Process and the Integration of Authoring

Interviewees made remarks that can be related to the development team, the EMERGO design and development phase, and the transition between these two phases. The educational technologist identified most issues.

Both interviewees find the development team very important. The educational technologist states that you need a good role distribution ('What I did really suits my role. '), the right people ('The quality of the content matter expert is almost decisive. '), and a good organization of such a project.

Team members should complement each other and should consult each other to get the best result ('Together you find solutions that you would not find separately. ').

During the authoring phase, both authors were in close contact, so they could efficiently work together to make things work and fix bugs.

For the design and development phase, the educational technologist states that he wrote the scenario without taking the available components into account very much. However, novice developers should know how to deal with the components beforehand, otherwise they get into trouble. If they are expected to author only one game, they should author only low complexity components.

Both interviewees have their opinions about the moment of transition between the design and development phase. The educational technologist states that authoring normally starts when the scenario is finished, but that you could start earlier if the storyline is clear and you know which components you need, at the risk of time-consuming adjustments in case of scenario changes. In addition, the ICT developer states that it also depends on someone's preference. One person likes to write everything down, while the other likes to try things out early.

We do not identify any problems related to the integration of authoring in the EMERGO method. Interviewees are positive about the process of converting the scenario to game content and are satisfied with the efficient way of authoring together. The ICT developer could quite easily extract script conditions and actions from the scenario, because the scenario is structured so that this is possible.

Other Software Quality Characteristics

Interviewees made remarks that can be related to two out of seven other software quality characteristics, namely *functionality* and *reliability*. No remarks can be related to performance efficiency, compatibility, security, maintainability, and portability. The ICT developer identified most issues. ISO/IEC defines functionality as the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions and reliability as the degree to which a system, product, or component performs specified functions under specified conditions for a specified period.

The functionality of the authoring environment is valued. Both interviewees stated that the preview and test options are essential and indispensable. They find it convenient that you can easily change the content of a game in exploitation. The ICT developer is positive about the flexibility of the Script component.

The reliability of the authoring environment is valued. Interviewees find saving content to be reliable, although drag and drop sometimes results in partial loss of data.

General Usability Guidelines for Authoring Environments for Serious Games

Just as with the EMERGO authoring environment, other authoring environments for SGs seem to have usability problems related to understandability and learnability (see section *Related work*). To prevent such problems, we present general guidelines that are based on improvements suggested by the interviewees.

Guidelines to improve understandability:

Offer an intuitive user interface so authors can more easily do what they want. An intuitive interface might be different for different kinds of authors. The educational technologist: 'It is not always intuitive. For students, we now have an interface, whereby they only have to think about what they want to do, not how to do it. Can we improve the usability for authors in the same way? Probably you should make different kinds of screens for different kinds of authors.'

Offer two levels of input, basic for novices and advanced for experts, so novice authors are shielded from unnecessary complexity. The ICT developer: 'Better first present mandatory input controls and then optional input controls.'

Offer examples of scenarios, games, and game components and how they relate to each other, so authors better understand what to do. The educational technologist: 'If I would be a novice developer, I would like to see some example scenarios that highlight where and when certain components are used.'

Offer a preview option to preview entered content at any time, so authors better understand what they are doing. The educational technologist: 'The preview option helped me to understand the working of the authoring environment.'

Use clear terminology fitting user expectations. The educational technologist: 'You have not yet managed to name the properties in a way that I understand exactly what will happen.'

Guidelines to improve learnability:

Offer clear instruction and wizards, so authors are guided during the authoring process. The ICT developer: 'Context sensitive help is needed when entering game content. You also could use wizards.'

Offer information on didactics and use of components, so novice authors can make a quick start. The ICT developer: 'You could give a short description why and how to use a component.'

Conclusions and Discussion

Our research goal was to evaluate in detail the usability of the EMERGO authoring environment and integration of authoring in the EMERGO method for serious game development. The case-based research addresses two areas of design research, as distinguished by Cross (2007) – design praxeology and design phenomenology – and falls into the game design value category *Values of Production and Creation Process*, as proposed by Kultima and Sandovar (2016).

We found understandability and learnability of the authoring environment to be problematic and operability to be somewhat problematic. On the one hand, this is caused by a lack of guidance and support but on the other hand, it is caused by the complexity of the domain and the environment itself. The first problem originates from the initial development of the EMERGO platform, when the priority was to build a player environment for students without enough capacity available to invest in the usability of the authoring environment. The second problem is related to tool complexity (Murray, 2004, 2015). Complex learning requires complex scenarios that need a

powerful environment with a lot of functionality and freedom, which may lead to lower usability. So power and flexibility of the authoring environment are both a strength and weakness.

We found authoring to be well integrated in the EMERGO method. However, our evaluation method did not include detecting the opposite, because we did not ask a question that specifically addressed this issue. Limitations of the EMERGO method might be that it only supports one type of game (scenario-based). However, if the scenario does not involve motor skills and can be realized with the available components, other types of games (like point-and-click adventure games) might also be supported. In addition, the method does not dictate but rather offers tooling to support design and development. The way the scenario is set up (using location plans) leaves plenty of room for creativity. Interviewees also do not mention any problems that can be related to the separation between design and development phases. This might be because they do not know better, but it also has to do with efficient development. Interviewees stated that if development starts when the scenario is not mature enough, time-consuming adjustments are needed in case of scenario changes. Of course, the authoring environment might be used in a more creative and agile manner, by skipping the scenario and switching between generating content, entering content, and previewing.

We found functionality and reliability of the environment to be valued.

Our findings do not include all usability aspects and all software quality characteristics. We think this is not due to our evaluation method but to the fact that interviewees did not mention related remarks, either because they had no problems with it or it was not relevant in their context of use. For instance, they will have had to deal with the aspects *user error protection* and *user interface aesthetics*, but they made no related remarks. And the characteristics *maintainability* and *portability* are not relevant in their context of use, because they relate to the EMERGO platform itself, not to developed games.

Besides the limited scope on certain usability aspects and software quality characteristics, our evaluation has other limitations. The data obtained are based on the development of only one game by only two authors who did not use all EMERGO components. However, we have strong indications that our findings are generic for the development of all EMERGO and similar complex learning games, because of following reasons. First, the developed game contains a didactical scenario that is representative of a typical EMERGO game. Second, we focused on the development process by eliciting knowledge from informants that were strongly connected to it. Third, the facts that these authors / informants had already developed and authored EMERGO games before, come from different backgrounds (different world views), and have used other components as well, make it very probable that their remarks are generic for other EMERGO games and components as well. Fourth, our findings are in line with more superficial findings we collected with other authors in two previous studies (Nadolski et al., 2007; Slootmaker et al., 2014). Fifth, all components are authored by one editor that uses common input controls, so components that are not evaluated are also indirectly partly evaluated. We do not claim that our findings are applicable to development of serious games in general, especially when these do not contain specific references to learning features.

Our method might not be reliable, since we only interviewed two authors who may have given desired answers. However, they are the most experienced and most recent authors, and we think that they were honest, also because they criticized the authoring environment a lot. The fact that the interviews were conducted long after the game was developed is a clear limitation. However, the authors have still used the authoring environment after the game was developed, and walking through the environment helped them recall their memories and led to a very detailed narrative.

We are not able to generalize our findings to specific learning features, such as assessment and feedback, because we did not raise them during the interviews. Also, the authoring environment has no single components that deal with learning features, like other environments (Kickmeier-Rust & Albert, 2010), but instead requires the cooperation of several components to support learning.

We presented some general usability guidelines for authoring environments for serious games. These guidelines include providing examples and didactic advice that might direct authors in a particular style of game, which may reduce the chances of other creative solutions. However, novice authors need examples and advice, and good and varied examples might also feed creativity. Another risk is that authors might get overfed by all the instruction and information or even do not pay attention to it. However, this probably depends on the type of person. There are people who prefer guidance and others who prefer trial and error. All authors should be served.

Most guidelines seem obvious but can easily be neglected under time pressure or due to other causes. Further, the guidelines seem to be so general that they may also be applicable to other types of authoring environments.

As a follow-up of this evaluation, we plan to impose the usability guidelines on the EMERGO authoring environment. In a future study, we will evaluate the environment again to see if the guidelines indeed cater to better usability. It would be interesting to investigate if a more graphical or block-based interface would be an improvement.

Since the development of the game used for this evaluation, the EMERGO platform has been extended with new functionality. For instance, the platform now supports more adventure-like games, like a game developed by Westera, Sloomaker, and Kurvers (2014). We recently integrated the use of the webcam to record students who counsel virtual patients. These recordings are used for in-game peer feedback and are discussed in a post-game debriefing session. We are currently working on a game for an introductory course on Psychology. This game also will be used for research purposes. We will simplify the rollout of games to different experimental groups.

We have plans to integrate an external service to analyze quality of reports and to add components that support collaboration. We already developed two games that use online collaboration (Hummel, Geerts, Sloomaker, Kuipers, & Westera, 2013; Hummel et al., 2010). We will use this experience to add new components for rating, voting, and negotiation.

Appendix I

Interview Guide

General impression

- What is your general impression of the authoring environment?
- Is it easy to use?
- Is it clear enough?
- Is its subdivision in screens straightforward?
- Is the navigation through screens straightforward?

Requirements (F=Functional, N=Non-functional)

Create and edit games (F)

- Did you create games yourself?
- If so, is editing of games straightforward?
- Did you encounter any problems editing games?
- Did you have any trouble with the concept of games?

Create and edit game roles (F)

- Did you add game roles?
- If so, did you add PCs or NPCs or both?
- Was editing of game roles straightforward, or did you encounter any problems?
- Did you have any trouble with the concept of game roles?

Select and edit game components (F)

- Did you have a good overview of components available beforehand?
- If not, was it difficult to get this overview?
- Did you have any trouble choosing the right components for your game, starting from the scenario?
- Was it obvious to you in which order components should be filled with content?
- Did you miss some functionality or components?
- Could you map your scenario easily to the available components?
- If not, were you able to implement the missing functionality using other components?
- Did you have any trouble with the game component editor in general?
- Did you have any trouble with the concept of the game component?

Developing a game together (F)

- What are your experiences with working together on the same game content?
- Could this process be improved?

Previewing a game or game component (F)

- Did you use the preview option?
- If so, did you use it to preview the game or just a single game component?
- For which game components did you use it?

- Did you have any difficulty using this option?
- Is it straightforward?
- Could it be improved?

Testing a game (F)

- Did you use the preview option for testing?
- If so, did you use it to test the game in total or just a single game component?
- For which game components did you use it?
- Did you use the option to create multiple preview items corresponding to multiple starting points within the game?
- Did you have any difficulty using this option?
- Is it straightforward?
- Could it be improved?

Import or export a game (F)

- Did you copy, import, or export games?
- What are your experiences with it?
- Did you have trouble with it?
- Could it be improved?

Import or export a game component (F)

- Did you copy, import, or export game components?
- If so, which game components?
- What are your experiences with it?
- Did you have trouble with it?
- Could it be improved?

Reliability and stability (N)

- Did you have any technical problems entering game content?
- Did you lose any entered data due to technical problems?
- How could we improve reliability and stability?

Delivering and updating games (N)

- Did you adjust the game or any game components while students were already playing?
- Did you experience any problems?
- Could it be improved?

General questions for components

- Did you use this component?
- Did you have any trouble using the component?
- What did you miss working with the component?
- How could we improve the component?

- Did you use the preview option for this component?
- What turned out to be handy when using the component?
- Do you feel comfortable using the component?

Development process

No questions prepared.

Acknowledgment

We would like to thank Henk van den Brink and Hub Kurvers of the Open University of the Netherlands for participating in the interviews.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

References

- Arnab, S., Berta, R., Earp, J., de Freitas, S., Popescu, M., Romero, M., ... Usart, M. (2012). Framing the adoption of serious games in formal education. *Electronic Journal of e-Learning, 10*(2), 159-171.
- Bryman, A. (2012). *Social research methods* (4th ed.). Oxford, England: Oxford University Press.
- Clark, R. C., & Mayer, R. E. (2011). *e-Learning and the science of instruction*. New York, NY: Wiley-Blackwell. doi:10.1002/9781118255971
- Crookall, D. (2010). Serious games, debriefing, and simulation/gaming as a discipline. *Simulation & Gaming, 41*(6), 898-920. doi:10.1177/1046878110390784
- Cross, N. (2007). From a design science to a design discipline: Understanding designerly ways of knowing and thinking. In R. Michel (Ed.), *Design research now: Essays and selected projects* (pp. 41-54). Basel, Switzerland: Birkhäuser Publishing. doi:10.1007/978-3-7643-8472-2_3
- Dede, C. (2009). Immersive interfaces for engagement and learning. *Science, 323*(5910), 66-69. doi:10.1126/science.1167311
- EMERGO. (2013). *EMERGO project page*. Retrieved from <http://sourceforge.net/projects/emergo/>
- Gaeta, M., Loia, V., Mangione, G. R., Orciuoli, F., Ritrovato, P., & Salerno, S. (2014). A methodology and an authoring tool for creating Complex Learning Objects to support interactive storytelling. *Computers in Human Behavior, 31*, 620-637. doi:10.1016/j.chb.2013.07.011
- Gagné, R. (1985). *The conditions of learning and theory of instruction*. New York, NY: Holt, Rinehart and Winston.
- Hartson, R., & Pyla, S. (2012). *The UX book: Process and guidelines for ensuring a quality user experience*. New York, NY: Elsevier. doi:10.1145/2347696.2347722

- Hummel, H., Geerts, W., Sloomaker, A., Kuipers, D., & Westera, W. (2013). Collaboration scripts for mastership skills: Online game about classroom dilemmas in teacher education. *Interactive Learning Environments*, 23(6), 670-682. doi:10.1080/10494820.2013.789063
- Hummel, H. G. K., Van Houcke, J., Nadolski, R. J., Van der Hiele, T., Kurvers, H., & Löhr, A. (2010). Scripted collaboration in serious gaming for complex learning: Effects of multiple perspectives when acquiring water management skills. *British Journal of Educational Technology*, 42(6), 1029-1041. doi:10.1111/j.1467-8535.2010.01122.x
- ISO. (2006). *Ergonomics of human-system interaction – Part 110: Dialogue principles* (ISO 9241-110:2006). International Organization for Standardization. Retrieved from <https://www.iso.org/standard/38009.html>
- ISO/IEC. (2011). *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models* (ISO/IEC 25010:2011). International Organization for Standardization/International Electrotechnical Commission. Retrieved from <https://www.iso.org/standard/35733.html>
- Kickmeier-Rust, M. D., & Albert, D. (2010). Micro-adaptivity: Protecting immersion in didactically adaptive digital educational games. *Journal of Computer Assisted Learning*, 26(2), 95-105. doi:10.1111/j.1365-2729.2009.00332.x
- Kickmeier-Rust, M. D., & Albert, D. (2012). Educationally adaptive: Balancing serious games. *International Journal of Computer Science in Sport*, 11(1), 1-10.
- Kickmeier-Rust, M. D., Mattheiss, E., Steiner, C., & Albert, D. (2011). A psycho-pedagogical framework for multi-adaptive educational games. *International Journal of Game-Based Learning*, 1(1), 45-58. doi:10.4018/ijgbl.2011010104
- Klemke, R., van Rosmalen, P., Ternier, S., & Westera, W. (2015). Keep it simple: Lowering the barrier for authoring serious games. *Simulation & Gaming*, 46(1), 40-67. doi:10.1177/1046878115591249
- Kultima, A. (2015). Game design research. In *Proceedings of the 19th International Academic Mindtrek Conference* (pp. 18-25). New York, NY: Association for Computing Machinery. doi:10.1145/2818187.2818300
- Kultima, A., & Sandovar, A. (2016). Game design values. In *Proceedings of the 20th International Academic Mindtrek Conference* (pp. 350-357). New York, NY: Association for Computing Machinery. doi:10.1145/2994310.2994362
- Marchiori, E. J., Torrente, J., del Blanco, Á., Moreno-Ger, P., Sancho, P., & Fernández-Manjón, B. (2012). A narrative metaphor to facilitate educational game authoring. *Computers & Education*, 58(1), 590-599. doi:10.1016/j.compedu.2011.09.017
- Mehm, F., Göbel, S., & Steinmetz, R. (2012). Authoring of serious adventure games in StoryTec. In D. Hutchison (Ed.), *Lecture Notes in Computer Science: Vol. 7516. E-learning and games for training, education, health and sports* (pp. 144-154). doi:10.1007/978-3-642-33466-5_16
- Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 10, 98-129.
- Murray, T. (2004). Design tradeoffs in usability and power for advanced educational software authoring tools. *Educational Technology*, 44(5), 10-16.
- Murray, T. (2015). Coordinating the complexity of tools, tasks, and users: On theory-based approaches to authoring tool usability. *International Journal of Artificial Intelligence in Education*, 26(1), 37-71. doi:10.1007/s40593-015-0076-6
- Nadolski, R. J., Hummel, H. G. K., Sloomaker, A., & Van der Vegt, W. (2012). Architectures for developing multiuser, immersive learning scenarios. *Simulation & Gaming*, 43(6), 825-852. doi:10.1177/1046878112443323

- Nadolski, R. J., Hummel, H. G. K., Van den Brink, H. J., Hoefakker, R. E., Slootmaker, A., Kurvers, H. J., & Storm, J. (2007). EMERGO: A methodology and toolkit for developing serious games in higher education. *Simulation & Gaming, 39*(3), 338-352. doi:10.1177/1046878108319278
- Nielsen, J. (1993). *Usability engineering*. New York, NY: Elsevier.
- Oja, M.-K. (2010). Designing for collaboration: Improving usability of complex software systems. In *Proceedings of the 28th International Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 152-158). New York, NY: Association for Computing Machinery. doi:10.1145/1753846.1754059
- Paakkanen, V. (2014). *User experience of game development environments: Can making games be as fun as playing them?* (Master's thesis). Aalto University, Espoo, Finland.
- Patrasitidecha, A. (2014). *Comparison and evaluation of 3D mobile game engines* (Master's thesis). Chalmers University of Technology, University of Gothenburg, Sweden.
- Schell, J. (2008). *The art of game design: A book of lenses* (2nd ed.). CRC Press. doi:10.1201/b17723
- Slootmaker, A., Kurvers, H., Hummel, H., & Koper, R. (2014). Developing scenario-based serious games for complex cognitive skills acquisition: Design, development and evaluation of the EMERGO platform. *Journal of Universal Computer Science, 20*(4), 561-582.
- Theodosiou, S., & Karasavvidis, I. (2015). Serious games design: A mapping of the problems novice game designers experience in designing games. *Journal of e-Learning and Knowledge Society, 11*(3), 133-148.
- Thillainathan, N., & Leimeister, J. M. (2014). Serious game development for educators—A serious game logic and structure modeling language. In *Proceedings of the 6th International Conference on Education and New Learning Technologies* (pp. 1196-1206). Valencia, Spain: International Academy of Technology, Education and Development.
- Van Est, C., Poelman, R., & Bidarra, R. (2011). High-level scenario editing for serious games. In *Proceedings of the International Conference on Computer Graphics Theory and Applications* (pp. 339-346). SciTePress. doi:10.5220/0003374503390346
- Westera, W., Nadolski, R. J., Hummel, H. G. K., & Wopereis, I. G. J. H. (2008). Serious games for higher education: A framework for reducing design complexity. *Journal of Computer Assisted Learning, 24*(5), 420-432. doi:10.1111/j.1365-2729.2008.00279.x
- Westera, W., Slootmaker, A., & Kurvers, H. (2014). The playground game: Inquiry-based learning about research methods and statistics. *Proceedings of the 8th European Conference on Games Based Learning, 2*, 620-627.

Author Biographies

Aad Slootmaker is ICT developer and PhD candidate at the Faculty of Psychology and Educational Sciences of the Open University of the Netherlands. He has experience in the design and development of software within Technology-Enhanced Learning projects. His research focus lies on development and evaluation methods for serious games.

Contact: aad.slootmaker@ou.nl.

Hans Hummel is an associate professor of Technology-Enhanced Learning at the Welten Institute (Research Centre for Learning, Teaching and Technology) at the Open University of the Netherlands. His research focus lies on innovative media for acquiring workplace-based professional skills such as serious games.

Contact: hans.hummel@ou.nl.

Rob Koper is a distinguished (university) professor at the Open University of the Netherlands with a specific focus on the innovation of (higher) education. Before this appointment he was the Dean of the Centre for Learning Sciences and Technologies (celstec). In his personal research he is interested in the social and cognitive aspects of human learning, the effective use of ICTs to support human learning and the improvement of educational institutions to facilitate better teaching and learning. He has written more than four hundred scientific contributions in journals and books, and has developed many different ICT systems to support learning and teaching.

Contact: rob.koper@ou.nl.