Personalized Support, Guidance, and Feedback by Embedded Assessment and Reasoning: What We Can Learn from Educational Computer Games

Michael D. Kickmeier-Rust, Dietrich Albert

HAL Id: hal-01055486
https://inria.hal.science/hal-01055486

Submitted on 12 Aug 2014

Personalized Support, Guidance, and Feedback by Embedded Assessment and Reasoning: What we can Learn from Educational Computer Games

Michael D. Kickmeier-Rust and Dietrich Albert

Department of Psychology, University of Graz
Brueckenkopfgasse 1, 8020 Graz, Austria
{michael.kickmeier; dietrich.albert}@uni-graz.at

Abstract. Software that intelligently interprets the goals and needs of its users on the basis of their behaviors without interrupting the work flow and consequently disturbing concentration and software that can support the users in a personalized, smart, yet unostentatious way is a desirable vision, for sure. One attempt to such support system was Microsoft's famous paperclip. The underlying logic, unfortunately, was rather simple and the users did not accept the feature very well. This paper introduces a psychologically and formally sound approach to a non-invasive, hidden assessment of very specific needs of the users as well as their competencies and corresponding tailored support and feedback. The approach was developed in the context of adaptive digital educational games and is based on the concepts of Competence-based Knowledge Space Theory as well as that of Problem Spaces. The purpose of this paper is to broaden the concept and elucidate a possible bridge from computer games to regular software tools.

Keywords: Embedded Assessment; Micro Adaptation; Support Methods, Feedback, User Model

1 Introduction

An important trend in the area of learning technologies is game-based learning. Computer games are a tremendously successful, popular, and attractive genre. A substantial number of young people spend many hours a week playing computer games, and these games are often the preferred play. Thus, using the motivational potential of computer games for educational purposes may open new horizons for educational technology and is challenging educators and developers. The very nature of utilising (computer) games for education is that playing games is one of the most natural forms of learning. Children learn to talk by playing with sounds and learn collaboration and strategic thinking when playing role playing games such as *Cowboys and Indians*. Immersive digital educational games (DEG) fall back to that origin and are a highly promising approach that makes learning more engaging, satisfying, inspiring, and maybe even more effective (see [1]). The major strengths of DEGs are generally observed at a high level of intrinsic motivation to play and

proceed in the game, a meaningful yet rich and appealing learning context, immediate feedback, and a high level of interactivity, challenge, and competition. Some researchers even argue that the exposure to "twitch speed" computer games, MTV, and the Internet has altered cognitive processes, emphasising specific cognitive aspects while de-emphasising others [2]. Thus, the so-called "digital natives" may require different, possibly non-conventional, educational approaches. Current challenges in the area of DEGs are seen in, for example finding an appropriate balance between gaming and learning activities (e.g., [3]), finding an appropriate balance between challenges through the game and abilities of the learner (e.g., [4]), convincingly embedding educational objectives in a game scenario, particularly when declarative knowledge is concerned (cf. [5]), or managing the extensive costs of developing high quality games (e.g., [3]).

In closer examination, many of those challenges are very similar to the challenges of interaction principles and interaction design with "regular" software". The digital natives, Marc Prensky [2] was talking about in the context of DEGs, are also users of regular software products (office tools, web browsers, websites, communication tools, etc.) and demand very specific – maybe unusual – features. Moreover, we are facing a large group of users who are highly inexperienced and sceptical towards the use of computer devices of all kinds, for example the elder generation. Examples of requirements might be balanced and intuitive multi-tasking possibilities, an embedded and personalized support with complex functionalities, integrating various tools for various purposes in common and usable devices, or managing high development and support costs.

In the focus of the present paper is to address a major challenge for research in the context of translating ideas from game-based learning to the usability of regular software products. In DEGs it is all about an individual (learning) experience in order to reach educational effectiveness and maintain fun, immersion, flow experience[1], and the motivation to play and, therefore, to learn. Thus, meeting individual preferences, interests, abilities, and goals is key to successful game-based learning. Such claim, of course, is also a highly desirable asset of regular software. In the following sections we will present a formal cognitive approach for an non-invasive, embedded assessment of psycho-pedagogical aspects such s motivational state or learning progress and corresponding personalized support by subtle feedback and support, which is currently implemented in learning adventure games.

2 Intelligent Adaptation and Personalization in DEGs

To meet the aforementioned claims, DEGs are supposed autonomously adapt to the learner along a variety of axes, for example, prior knowledge, learning progress, motivational states, gaming preferences, and psycho-pedagogical implications. Generally speaking, adaptive approaches to technology-enhanced learning contest the one-size-fits-all approach of traditional learning environments and the attempt to

---

[1] According to Mihaly Csikszentmihalyi [6], flow refers to a highly immersed experience when a person is engaged in a mental and/or physical activity to a level where this person loses track of time and the outside world and when performance in this activity is optimal.

tailor the learning environment according to individual needs and preferences. The spectrum of approaches, methods, frameworks, and applications is quite broad [7] there are basically three major concepts:

- *Adaptive presentation*: adjusting the look and feel of a learning environment according to individual preferences or needs; for example, different colour schemes, layouts, or amount of functionality;
- *Adaptive curriculum sequencing:* providing the learner with learning tailored to individual preferences, goals, learning styles, or prior knowledge;
- *Adaptive problem solving support*: providing the learning with feedback, hints, or solutions in the course of problem solving processes.

Most of these methods and frameworks for adaptation and personalisation were developed in the context of conventional e-learning. The underlying concepts and ideas are currently extended and adjusted to the requirements of the rich virtual gaming worlds, particularly to maintain an immersive gaming experience and high levels of motivation, curiosity, and flow experience [8].

A method, which is highly interesting for interaction design in general, is an approach to non-invasive assessment of knowledge and learning progress in the open virtual worlds of computer games and a corresponding adaptation by personalised psycho-pedagogical interventions. The approach, labelled *micro adaptivity*, was developed in the context of *80Days* (www.eightydays.eu), a multidisciplinary research and development project. The project had the ambitious goal of utilising the advantages of computer games and their design fundamentals for educational purposes and addressing specific disadvantages of game-based learning. Within the project, a methodology for the successful design of educational games was established, and a game demonstrator was developed based on a state-of-the-art 3D adventure game (see Fig. 3 for some impressions of the game).

3 Adaptation on the Micro Level

A primary task for (game-based) adaptive mechanisms is to guide and support the user in accomplishing specific goals, for example, informing the user, intervening when misconceptions or errors occur or the work/learning progress is unsatisfactory, hinting, or providing the user with appropriate feedback. In addition, tasks are motivating, maintaining immersion, and personalising the environment according to the preferences and needs of the user. Accomplishing this goal requires a theoretical and technological approach that enables the system to assess cognitive states (e.g., *competence states* or *motivational states*), (learning) progress, possible misconceptions, or undirected/unsuccessful problem solving strategies. A distinct characteristic of adaptive DEGs is that gathering the necessary information from the user cannot occur in a conventional form (e.g., by test items, questions, or tasks). A DEG requires an assessment that does not harm motivation, immersion, flow experience, or the game's storyline [8]. The very basis of micro adaptivity is a formal psychological model for interpreting the behaviour within the virtual environment.

To realise the vision of a non-invasive support of users in DEGs (i.e., embedded assessment and subsequent educational interventions), we combine the *Competence-*

based Knowledge Space Theory (CbKST), which is a formal set-theoretic psychological approach to human learning and development that has been successfully utilised in conventional adaptive e-learning, with cognitive theories of problem solving. This theory provides a detailed domain model that includes a set of meaningful *competence states* as well as a set of possible *learning paths*. Problem solving theories, in turn, provide a set of possible *problem solving states* as well as possible problem solving paths. While in an DEG (at least in most types of DEGs) complex problem solving is an important mechanisms of self-directed, constructivist learning, for regular software we can assume that the user's goal is to accomplish a specific task or to solve a specific problem that, in turn, requires performing a sequence of specific action and interactions with the software. To give a very simple example, the task might be to highlight a certain word of a text in an office software. The steps to solve this problem are (i) taking the mouse and highlighting to word by setting the cursor at the beginning or the end of the word, pushing the left mouse button and by moving the mouse with pressed mouse button along the word and , finally, by releasing the mouse button. When the word is selected, (ii) the user has to press the correct highlighting symbol in the tool bar with the left mouse button.

3.1 Competence-based Knowledge Space Theory

In essence, CbKST [9, 10] originates from the *Knowledge Space Theory* (KST) established by Jean-Paul Doignon and Jean-Claude Falmagne [11, 12], which is a well-elaborated formal cognitive framework for covering the relationships between problems (e.g., tasks or test items). It provides a basis for structuring a domain of knowledge and for representing the knowledge based on *prerequisite relations* (Fig. 1, left panel), in the sense that one task is easier than another or that it is likely mastered before another. While KST focuses only on performance/behaviour (e.g., solving a test item), CbKST introduces a separation of observable performance and latent, unobservable competencies, which determine the performance. Essentially, CbKST assumes a finite set of competencies and a prerequisite relation between those competencies. As mentioned, a prerequisite relation states that a competency *a* (e.g., multiplying two positive integers) is a prerequisite to acquiring competency *b* (e.g., dividing two positive integers); if a person has *b*, we can assume that the person also has *a*. Due to the prerequisite relations between the competencies, not all subsets of competencies are possible *competence states* (e.g., it is highly unlikely that a person can multiply but not add numbers). The collection of possible competence states corresponding to a prerequisite relation is called *competence structure* (Fig. 1, right panel). While in the learning/development context we likely find logical prerequisites (e.g., to add two integers is a logical prerequisite of multiplying two integers), in the context of using software tools the prerequisite are likely not so obvious (e.g., to know how to change the font of a text can be considered a prerequisite for knowing how to make a table). Thus far, the structural model focuses on latent, unobservable competencies. By utilising mapping functions, the latent competencies are assigned to a set of tasks/test items/actions relevant for a given domain, which induces a *performance structure*, the collection of all possible *performance states*. Learning or
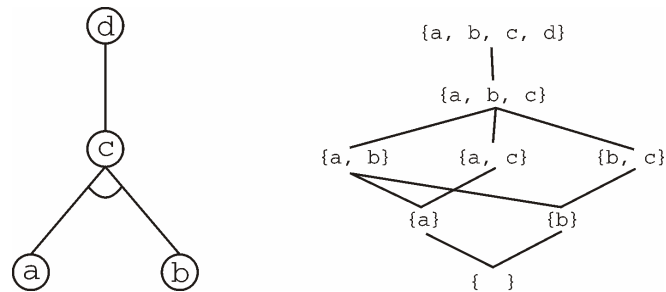
Fig 1. The left panel shows a prototypical prerequisite function; the right panel illustrates the corresponding competence structure.

development is not seen as a linear course, equal for all learners; rather, learning follows one of a set of individual learning paths.

Translated to the context of accomplishing tasks with a software tool, this approach enables to define the competencies necessary to use the tool in a very detailed and precise way. By the so-called competence states, the collection of competencies a specific user has, we can determine which task s/he is able to accomplish and what kind of information and support is required to solve other tasks. In addition to the structural hypothesis lying the competence structures, we need an equally formal idea of problem solving as a process. In this work, we rely on a formal theory of the human problem solving process, the theory of Newell and Simon ([13]; see also [14] for a conclusive overview).

3.2   A Theory of Human Problem Solving

[13] argued that problem solving is dynamic, rule driven process. The actions a person performs can be seen as a function of memory operations, control processes, and rules. The very basis of this approach is to decompose a problem or situation (you may think about all possible states of the *Tower of Hanoi* problem) into a *problem space*, a collection of all possible and meaningful *problem solution states*, objects relevant for a problem, and transition rules, which specify how admissible transitions from one to another problem solving state can occur. In so far, the approach of Newell and Simon is highly compatible with the structural approach to learning and development of CbKST. Consequently, the Tower of Hanoi's *problem space* would include all states where the rules "a larger disk cannot top a smaller" and "all disks must be on one of the pegs" are not violated (see Fig. 2) Likewise, we establish a problem space for specific tasks to be performed with a software tool.

3.3   Merging Competence Structures and Problem Spaces

To provide the system with holistic information about the competence domain (i.e., the competence structure and the basis of a set of tasks to be accomplished with a
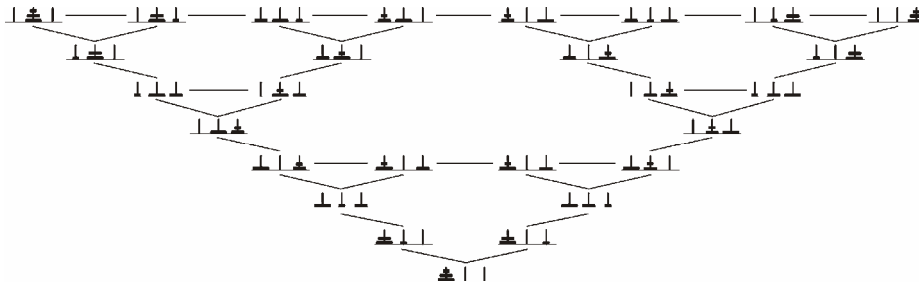
Fig 3. Prototypical problem space for the "Tower of Hanoi" problem.

given tool) and the possible actions within the problem solving processes (i.e., the problem spaces for the set of possible tasks), both must be linked together.

On a very fine grained level of competencies, with a large number of possible tasks, and with a large degree of freedom for the user, unfortunately, we would end up with vast competence structures and problem spaces, which could not be handled and computed in real time. In the context of games, we introduced the concept of *game states*, which we want to term now *system state*. These are meaningful accumulations of specific states in the virtual environment (e.g., when specific objects are assembled to a machine correctly or in terms of regular software, for example, when a set of options such as font, size or colour are set appropriately for a specific task). This accumulation is now a substrate of the problem space, which can be linked to the user's available and unavailable competencies. There are two options to do so; first, a *deterministic linkage*, and second, a *probabilistic linkage*. The former means that each system state is associated with a specific set of competencies that the user has and a specific set of competencies that the user lacks. The latter means that a numeric value that describes the strength of belief that a specific set of competencies is available or lacking is assigned to each position category. Independent of the linkage type, a *utility value* is assigned to each state to provide the game with information about "how appropriate" or "how inappropriate" a system state is for a specific task.

3.4 Updating Competence State Probabilities

In a next step it is necessary to transfer the assumptions of available and lacking competencies to entire competence states and the likelihood of those states. To identify a user's initial (i.e., at first use of the software) competence state, we assume an initial probability distribution over the competence structure. There exist an almost arbitrary number of possibilities to obtain such initial probability distribution. The simplest form would be a uniform distribution in which each competence state has the same initial probability. The selection of the right method strongly depends on the given information about the user's competencies and prior knowledge as well as the user's goals.

With each action the user performs, the system updates the probability distribution over the competence states, where updating rules define the way in which the probabilities are updated in a specific situation [12]. A simple method is to increase the probabilities of all competence states that include competencies that are (either definitely or likely) available when realising the corresponding position category. Conversely, we can decrease the probabilities of those competence states that include competencies that are unavailable. On this basis, we are continuously approaching every action and every realised position category with a clearer interpretation of the learner's competence state. Although single interpretations may not be convincing, with an increasing number of actions, certain and most often similar competence states become increasingly clear. These procedures are now the basis for supporting the user in a personalized, suitable, and hopefully in a non annoying manner.

4 Adaptive Support and Feedback

The basic idea of the micro adaptivity concept is to support the user with suitable guidance and support and with informative feedback. We have described so far, how we think that the system can gather a holistic picture of what the user knows, in terms of which competencies s/he has, and at which point of the problem space the user is on his/her way to accomplish a specific task (having the idea in mind that accomplishing a task can be interpreted as solving a problem in a given, well-defined and rule-based setting). The next logical step is to equip the system with a set of responses that can be triggered if necessary. Bearing in mind the game-based learning context, we can distinguish following types of response:

- *Competence activation interventions* may be applied if a user is stuck in some area of the problem space and some competencies are not used even though the system assumes that the user possesses them. For example, the system can make a specific function (e.g., a tool bar button) more visible.
- *Competence acquisition interventions* may be applied in situations when the system concludes that the user lacks certain competencies. For example, the system informs the learner about specific features of the software and how to use them.
- *Feedback* may be utilised to provide the user with information about the work progress. For example, the software could inform the user if s/he has used the software in an efficient way.
- *Assessment clarification interventions* may be applied, for example, in the form of a query if the user's actions provide contradicting support for the assumption of a certain competence state. In addition, such type of intervention maybe used to determine which task a user wants to accomplish.

Such kind of support and guidance proved to be effective in the learning context [15], more importantly in the context, recent research showed that such interventions could improve immediate goal achievement [16].

5 Leering at the Games: 80Days

5.1 Architecture

The concept of micro adaptive assessment and interventions has been developed in the context of the ELEKTRA project and its successor 80Days. In 80Days, prototype game was developed to demonstrate and evaluate the concept of micro adaptivity. The architecture consists of four modules, (1) the game itself, quite traditionally, is created using a high-end gaming engine, (2) a *skill assessment engine* updates the competence state probabilities, and the resulting information regarding the learner's competence state and its changes are then forwarded to (3) an *educational reasoner*, the pedagogical part of micro adaptivity. Based on a set of pedagogical rules and meta-rules as well as learning objectives, this engine provides recommendations on adaptive interventions to the adaptation realisation module. This, in turn, maps the abstractly formulated educational recommendations onto more concrete game recommendations. In this mapping process, data on game elements and information on previously given recommendations are considered. The necessary information for the assessment-intervention loop is stored (4) in an *OWL* [17], which allows the aforementioned engines to extract not only information, but also the relationships among the information from the ontology.

5.2 The Game

We have developed a learning adventure that is supposed to teach geography for the age group of 12 to 14 years. The curriculum includes, for example, knowledge about the planet Earth such countries or cities but also aspects such as longitude or latitude. In the game the learner takes the role of a boy or a girl (depending on the learners' gender) at the age of 14. The story starts from an extraordinary event; a space ship is landing in the backyard and an alien named Feon appears. Feon turns out to be a friendly alien, being an alien scout who has to collect information about foreign planets, in this case planet Earth. The learner accompanies Feon and is having fun with flying a space ship and exploring interesting places on Earth. Feon creates a report about the Earth and its geographical features. This is accomplished by the player by means of flying to different destinations on Earth, exploring them, and collecting and acquiring geographical knowledge. The goal is to send the Earth report as a sort of travelogue about Earth to Feon's mother ship. At a certain point in the game, however, the player makes a horrible discovery; the aliens are not really friendly but collect all the information about Earth to conquer the planet, lately. This discovery reveals the "real" goal of the game: The player has to save the planet and the only way to do it is to draw the right conclusion from the traitorous Earth report. The subject matter is embedded in hat story and learning occurs at various events in the game. From a pedagogical point of view, learning occurs by receiving information (e.g., seeing/reading something in the game or hearing something from Feon or other game characters), problem solving (e.g., reducing the negative impacts of a flood by appropriate "terra-forming"), or imitation (e.g., watching other game characters and learning from their behaviours). Screenshots are shown in Fig. 3.

Fig 3. Screenshots from 80Days' game demonstrator.

6 Conclusion and Outlook

The idea of adaptation on the micro level is to monitor a learner's behaviour within an adaptive DEG and to provide him/her with appropriate and tailored support and feedback. In the context of the 80Days project we implement the corresponding functionalities in a compelling demonstrator game. This game was subject of experimental evaluations which showed that the envisaged kind of "subliminal" or "hidden" assessment not only works in terms of assessment accuracy but also that the corresponding interventions lead to superior learning performance, motivation, and satisfaction with the game in comparison to no adaptive interventions or inappropriate interventions [18].

In the present paper we raised the idea that such non-invasive assessment method and especially the underlying logic might be successfully applied ion the context of regular software as well. While the famous and unsuccessful even annoying paper clip assistant of Microsoft's Office was a good idea in its essence, the underlying logic was not smart enough to convince most of its users. If we can transfer the psychologically sound frameworks of CBKST in combination with problem spaces we can equip a software system with a "deeper understanding of the user and his/her needs – at least in comparison to other support methods.

Of course, this approach is still in its infancy and must be elaborated in future work. Exactly here lies the major purpose of this paper, that is, introducing the concept to a community broader than that of game-based learning, to encourage discussions and to broader the concept for a variety of applications.

Acknowledgments

References

1. de Freitas, S.: Learning in immersive worlds. A review of game-based learning (2006). Retrieved August 28, 2007 from http://www.jisc.ac.uk/media/documents/programmes/ elearning_innovation/gaming%20report_v3.3.pdf
2. Prensky, M.: Digital game-based learning. New York: McGraw-Hill (2006)
3. Van Eck, R.: Digital game-based learning. It's not just the digital natives who are restless. Educause Review, March/April 2006
4. Kickmeier-Rust, M. D., Peirce, N., Conlan, O., Schwarz, D., Verpoorten, D., & Albert, D.: Immersive digital games: Interfaces for next-generation e-learning. In C. Stephanidis (Ed.), Universal Access in HCI, Part III, HCII 2007, LNCS 4556 (pp. 647–656). Berlin: Springer (2007)
5. Kafai, Y. B.: Playing and making games for learning. Games and Culture, 1(1), 36-40 (2006)
6. Csikszentmihalyi, M.: Flow: The psychology of optimal experience. New York: Harper and Row (1990)
7. De Bra, P. M. E.: Adaptive hypermedia, In H. H. Adelsberger, Kinshuk, J. M. Pawlowski, & D. Sampson, (Eds.), Handbook on Information Technologies for Education and Training (pp. 29-46), Berlin: Springer (2008)
8. Kickmeier-Rust, M. D., Albert, D., Hockemeyer, C., & Augustin, T.: Not breaking the narrative: Individualized Competence Assessment in Educational Games. In Proceedings of the European Conference on Games-based Learning (ECGBL), October 25-26, 2007, Paisley, Scotland (2007)
9. Albert, D., & Lukas, J. (Eds.): Knowledge spaces: Theories, empirical research, and applications. Mahwah, NJ: Lawrence Erlbaum Associates (1999)
10. Korossy, K.: Extending the theory of knowledge spaces: A competence-performance-approach. Zeitschrift für Psychologie, 205, 53-82 (1997)
11. Doignon, J.-P., & Falmagne, J.-C.: Spaces for the assessment of knowledge. International Journal of Man-Machine Studies, 23, 175-196 (1985)
12. Doignon, J.-P., & Falmagne, J-C.: Knowledge spaces. Berlin: Springer (1999)
13. Newell, A. & Simon, H.: Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall (1972)
14. Newell, A.: Unified Theories of Cognition. Cambridge, MA: Harvard University Press (1990)
15. Azevedo, R., & Bernard, R.M.: A meta-analysis of the effects of feedback on computer-based instruction. Journal of Educational Computing Research, 13, 111-127 (1995)
16. Tan, J., Biswas, G., & Schwartz, D.: Feedback for metacognitive support in learning by teaching environments. In Proceedings of the 28th Annual Meeting of the Cognitive Science Society, Vancouver, Canada, pp. 828-833 (2006)
17. Kickmeier-Rust, M. D., & Albert, D.: The ELEKTRA ontology model: A learner-centered approach to resource description. Advances in Web Based Learning – ICWL 2007 (pp. 78-89). Lecture Notes in Computer Science, 4823/2008. Berlin: Springer (2008).
18. Kickmeier-Rust, M. D., Marte, B., Linek, S. B., Lalonde, T., & Albert, D.: The effects of individualized feedback in digital educational games. In T. Conolly & M. Stansfield (Eds.), Proceedings of the 2nd European Conference on Games Based Learning (pp. 227-236), October 16-17, 2008, Barcelona, Spain. Reading, UK: Academic Publishing Limited (2008)