An Ontology for Integrating Didactics into a Serious Training Game

Marieke Peeters[1,2,4], Karel van den Bosch[2], John-Jules Ch. Meyer[1,2], and Mark A. Neerincx[2,3]

[1] Information and Computing Sciences, Utrecht University
[2] Training & Performance Innovations, TNO
[3] Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
[4] Corresponding author: m.m.m.peeters@uu.nl

Abstract. Serious games offer high potential for immersive, effective, and autonomous training. However, research has shown that trainees need guidance and structure during training. This could be achieved by means of well-chosen scenarios and targeted adaptations of the storyline based on didactic considerations. This paper discusses some of the challenges posed in adaptive game design. Additionally, the paper outlines the design rationale behind the adaptive game architecture for training (AGAT). An ontology is proposed that serves as a foundation and knowledge base for a system able to orchestrate the game's storyline in a didactically desirable fashion. The ontology's use is versatile: it supports requirements elicitation and refinement, results in traceable underlying assumptions and design choices, and provides the knowledge base used by the system itself. The architecture is illustrated by means of a case study. Future work focuses on the development of a generic set of procedural rules to operate on this ontology and generate user-tailored didactically-driven adaptive scenario content.

Keywords:
ontology, adaptive educational game, game design, instruction, didactics, requirements analysis, situated cognitive engineering, scenario-based training, serious games

1 Introduction

The growing demand for autonomous training has led to an increase in research on intelligent instructional systems, such as serious games [21]. Serious games are designed to offer trainees the opportunity to develop their skills and knowledge in a meaningful and practical, yet virtual, training setting. They carefully balance fun, knowledge transfer, and reality to provide the trainee a meaningful, immersive, and motivating learning experience. A substantial amount of research regarding serious games has involved the use of intelligent agents to control the characters in the storyline, examples of which are the Intelligent Story Architecture for Training (ISAT) [11,10], IN-TALE [22], and Thespian [25].

The use of intelligent agents to control important non-player characters (NPCs) in the scenario allows for training opportunities in the absence of an instructor and without the need for team members being present. For a long time, research has focused on generating believable and adaptive NPC behavior, however, Yannakakis (2012) claims that NPC AI is almost solved [31]. More interesting, novel challenges revolve around the knowledge about users, tailoring the game to those users, and techniques to control automated content generation.

The need for personalization is even more important to the development of *serious* games. In order to warrant the didactical training value, the trainee requires guidance and structured learning content, comparable to following a personalized curriculum guided by a personal coach [8]. Of course, personalization should be grounded in didactical principles, derived from efficacious training forms (e.g., scenario-based training [23]) and instructional design (e.g., 4C/ID [27]) [12,17,18]. Such an approach combines the immersion and appeal of serious games with the structure offered by intelligent tutoring systems [15,28]. Instructional theory promotes a balance of challenge and ability, and the provision of meaningful scenarios to the trainee. But the question remains how these principles should be embedded in AEG design.

Personalization means that the system knows and interprets the trainee's performance and adapts the training exercise to match this performance using didactical strategies [19]. Examples of didactical strategies are the iterative offline selection of suitable training objectives and topics, the online delivery of variable amounts of support, or adjustments in the pace of training. In addition to the training exercise, the adaptation itself can be altered to fit the trainee's personal preferences or learning style. Note that the trainee is not the only person involved in training with personal preferences or styles; the instructor may also have a preferred didactic strategy. Such requirements for customization and personalization form a serious challenge for AEG design [24]. Preferably, AEG design separates these requirements in an early stage and tackles them in a modular way to promote code reusability over domains, trainees, and instructors.

This paper presents work on an adaptive educational game architecture that fosters reusability of its components and has a strong foundation in didactical principles and instruction theory. Section 2 describes relevant related work in the field of adaptive educational games. Section 3 outlines the reasons for and method of developing an AEG ontology, along with a high/level presentation of the resulting ontology and its intended use. The ideas presented in the paper are discussed in Section 4.

2 Related Work on Adaptive Games

In the past years several researchers started focusing on player-centric adaptive games. As Zook and Riedl (2012) point out, there are two aspects to user-adaptive game design: challenge tailoring (CT) and challenge contextualization (CC) [32]. CT refers to online as well as offline dynamic difficulty adjustment: reasoning about scenario content on a didactic level. This reasoning is based on didactical principles or strategies leading to decisions about appropriate learning topics and levels of challenge or support. CT requires high resolution player profiles [9], usually specifying the player's skill proficiencies. In contrast, CC refers to the construction of the game world and events that set up the selected learning objective and challenge in the actual game environment. It deals with the reality and believability of the trainee's learning experience.

As mentioned above, challenge tailoring refers to offline as well as online adaptivity. Aa promising development in offline challenge tailoring is procedural content generation (also mentioned by Yannakakis (2012) [31]) controlled by semantic modeling techniques [5]. By embedding and interpreting higher level semantic annotations in virtual objects and agents, the content generation process can be constrained to create meaningful and realistic content that matches the learner's profile [1,26]. Such higher level constraints can then be fulfilled by equipping objects with the capacity to provide services in the game [6]. For instance, in the CT stage, the scenario is prepared offline by generating a set of constraints that delineate the learning goal (e.g., treat a thermal lesion) and the level of challenge (e.g., beginner). The scenario generator then collects a set of annotated objects that offer the services required to fulfill those constraints (e.g., a hot object, a victim, a water tap, and a first aid kit). A straightforward method to manage online adaptivity is the use of dynamic world elements, such as NPCs and dynamic objects, enhanced with didactically meaningful behavior variations or variable characteristics. In the case of the example, the victim could have two behavior variations, one in which the victim is calm, and one in which the victim is panicking. The ALIGN system [20], for example, uses annotated adaptive elements to enable online adaptivity. It incorporates personalised didactics into a serious game, while separating the pedagogical principles from the game, thereby making it reusable. Peeters et al. (2011) [18] and Westra et al (2010) [30] used scripted NPCs that were able to perform different behavior variations, thereby enabling online scenario adaptation.

2.1 Context of Previous Work by the Authors

In a previous paper, the situated Cognitive Engineering (sCE) method [13] led to the specification of a set of design principles and a high level AEG architecture: the Adaptive Game Architecture for Training (AGAT). The combination of knowledge from different fields (e.g., game research, intelligent tutoring systems, instructional theory, and educational psychology) resulted in an initial requirements baseline ([R1]-[R5]) for an AEG. Each requirement is founded in a set of measurable and testable claims ([C1.1]–[C5.2]), each of which is grounded in literature research and expert knowledge. The interested reader is referred to [19] for the details on the underlying literature review that lead to these claims and requirements.

[R1] Match scenarios to the trainee's skill level (offline).
  [C1.1] Presenting scenarios in order of increasing complexity and matching them to the trainee's level of experience prevents cognitive overload.
[R2] Adjust the support level during task performance (online).

[C2.1] Adjusting the level of challenge to match the trainee's skill level fosters flow and high levels of motivation.

[R3] Generate authentic scenarios.

[C3.1] Authentic training tasks foster transfer.

[C3.2] Authentic training tasks foster intrinsic motivation.

[C3.3] Engaging in authentic training tasks fosters immersion, and thereby flow and motivation.

[R4] Generate a wide variety of adaptive scenarios.

[C4.1] This will foster transfer and the development of generic solutions.

[R5] Provide feedback about the task performance during the scenarios.

[C5.1] This will foster self-efficacy.

[C5.2] This will foster a better understanding of the task domain.

A first experimental evaluation of our prototype was conducted during which domain experts rated video fragments of adaptive and non-adaptive scenarios in terms of learning value. This study revealed that online adjustments ([R2]) of the support level significantly improve the quality of training [18], validating further research on the development of our architecture.

As the research project progressed an additional technical requirement was added to this list:

[R6] Promote reusability over domains, trainees, and instructors.

3 Adaptive Game Architecture for Training

The requirements mentioned in Subsection 2.1 form the foundation for our AEG architecture. Two important notions led to the design presented below. First of all, as mentioned in the previous section, there are two stages in difficulty adjustment, offline as well as online: 1) challenge tailoring, and 2) challenge contextualization. Second, domains, world content, teaching strategies, and trainees may change over time, and the system should offer ways to handle such changes through reusable components. This requires a clear format for new information and generic procedures able to handle that format. For instance, the system needs to know about the concept of didactical strategies and use this knowledge by employing those strategies using a generic method, instead of employing hard-coded, implicit didactical strategies. We propose the use of an ontology to specify relevant information about the user, the didactical strategies, and the domain. In addition, generic procedural rules are designed to use this information and generate constraints on the procedural content generation process. The training scenarios are first generated on a didactical level (offline) and are then contextualized using semantically annotated objects. In turn, these objects are able to perform several behavior variations to enable online adaptivity.

The rest of this section describes the first part of the architecture, the ontology, which describes the knowledge areas that are characteristic and relevant to AEG design: the task domain, the trainee, the available didactic strategies, the instructor's personal touch, the game world, and the system's design. First, the concept of ontologies is explained in Subsection 3.1. Thereafter, it presents and exemplifies the resulting ontology in Subsection 3.2.

3.1 The Need for an Ontology

The motivation to create an ontology that defines all concepts related to AEG design was twofold: 1) by creating an ontology, the system's specification is refined, since it forces the developer to build a solid argument and plan for each functionality, and 2) the ontology contributes to the desire of building a modular system that consists of generic rules imputed by exchangable (formalized) knowledge bases. An ontology represents the basic concepts relevant to the system's operations, along with their attributes and interrelations, thereby modeling a domain of knowledge. The use of an ontology is beneficial to the design of adaptive systems; it supports a shared understanding of the system's concepts and interrelations [3,4], but also the early refinement and testing of the system's requirements [16]. However, as explained earlier, our main interest in using an ontology is that it can serve as a knowledge base for the system to rely on, that allows for reusability and easy modification [2].

Related Ontologies. Kickmeier-Rust and Albert (2008) agree that serious games should balance challenge and ability to promote flow and motivation [7]. Their ELEKTRA ontology resembles parts of our ontology areas, the most important resemblance being the distinction between task performance and skill proficiency. This distinction is important since it abstracts away from the task, defining the learning content as a higher level ability and understanding. This makes it possible to separate the performance data from the task domain, since the skills to be developed overarch several domains.

The ontology by Van Welie et al. (1998) has served as the starting point for our task domain ontology area [29]. It defines tasks as activities performed to reach a certain goal, and possibly, there are multiple ways to reach it. The goal of a task is a specific state that is reached after successful execution of the task. Since tasks can be performed by a group of people in dynamic environments, Van Welie et al.'s ontology takes roles and events into account.

Ontology Engineering Method. The ontology described below was created using an iterative 4-step process, derived from Noy and McGuiness' Ontology 101 (2001) [14]. It uses First Aid Training as the application domain for clarification purposes. The four steps used during the creation of the ontology were:

(1) Specify all the terms relevant to the requirements.
(2) Identify the important properties of the terms specified in Step 1.
(3) Define the relations between the terms.
(4) Create domain-specific instances for all of the terms by applying the ontology to the training domain.

The ontology specification process is iterative; each step results in new knowledge about the quality and sufficiency of the terms, attributes, relations and their definitions identified in previous iterations. The main reason for iterative refinement is that the ontology as a whole needs to be cohesive and consistent.

3.2 The AEG Ontology - Description

The ontology serves to answer questions like 'What will the system teach, and to whom?', 'What strategies can the system use to teach?', 'What narrative elements can the system use to contextualize the learning content?', and 'What higher level design and system constructs does the system use?'. Various sources of information, e.g., observations, interviews, and literature research, were used to answer these questions, meaning that the ontology also serves a purpose of theory development. The analysis resulted in an ontology consisting of 6 main areas: 'Task domain', 'Trainee', 'Didactics', 'Instructor', 'World', and 'System'.

- Task Domain - this ontology area refers to concepts involved in the task execution, such as 'Task', 'Role', 'Objective', etc. This ontology area was based on work by van Welie et al. (1998) [29].
- Trainee - these concepts specify all the required knowledge to reason about the trainee and his/her progress during training, e.g., 'Performance', 'Skill', and 'Motivation'.
- Didactics - this area includes concepts referring to instructional features of the system, examples of which are 'Support Level', 'Feedback', and 'Cognitive Load'.
- Instructor - these concepts deal with the interaction between the system and the *instructor*, and include concepts such as 'Didactic Strategy' and 'Scenario Compilation'.
- World - the concepts in this area refer to all concepts relevant to the game world. It includes concepts such as 'Object', 'Agent', and 'Event'.
- System - this area contains concepts that refer to higher level abstractions and to terms relevant to the initial design architecture, specifying, for example, 'Task Model', 'System Component' and 'Intelligent Agent'.

The ontology has been implemented in Protégé frames 3.5 alpha [5]. In the near future it will be reviewed by experts and thoroughly checked for consistency. Due to space limitations the complete ontology is not discussed here. However, the ontology and its use are illustrated by means of a case study in the next subsection.

An Illustrative Case Study This section illustrates the intended architecture and the use for the ontology with a case study.

Jeremy has trouble connecting the topics presented during different training sessions

[5] http://protege.stanford.edu

over time. In a previous session he received instructions on how to deal with non-cooperative patients. Today he will receive instruction on the diagnosis of burns.

To properly teach Jeremy how to perform First Aid, the system will need to meet the specified requirements. For this example we shall discuss just one of them: (1) choose a scenario that is appropriate for the learning goal. To meet this requirement, the system needs to have a proper 'understanding' of what it means. To do that it must have knowledge about the meaning of the concepts in the requirement. This knowledge is available in the ontology.

First of all, the system needs to know what a *scenario* is. The ontology defines scenarios on a semantic level: scenarios refer to a set of tasks, and contain an intitial world state, and possibly a sequence of necessary events. This intitial world state is then defined as a set of specified objects and agents situated in some environmental setting. The ontology also specifies *tasks* and how they should be decomposed and/or performed in the Task ontology area. The storytelling elements, such as settings, objects, and characters along with their actions and the effects thereof, all belong to the World ontology area. Objects are embedded with additional information about their use within the task domain as well as their didactical purposes (e.g., difficulty levels).

Second, the system should recognize the concept *learning goal*, which is defined as an objective that Jeremy should achieve with respect to his *skill* development. To derive Jeremy's learning goal, the system relies on the knowledge collected in Jeremy's current *skill graph*. The system can now use this knowledge to derive an appropriate learning goal for Jeremy:

'Generalize task procedures over contexts.'

To generate a scenario that fits the learning goal, the system must know how to match scenarios to learning goals. However, a learning goal refers to skill development, whereas a scenario refers to a task performance embedded in a storyline. This requires a relation between the concepts 'task' and 'skill'. As mentioned in the discussion of the ELEKTRA ontology, tasks are specified to rely on a (set of) skill(s). This allows for the system to produce a scenario that matches Jeremy's skill set.

It becomes clear, that there are still a lot of concepts mentioned in this example that need further specification before the system is actually able to reason about them. For now, we will leave this example. The AEG ontology, however, covers a lot more than the content discussed in the example above.

4 Discussion

This paper discusses the design of an adaptive game architecture that promotes reusability and a proper didactical foundation. It proposes the use of an ontology as a knowledge base, combined with a set of generic procedural rules that operate on this ontology. The ontology specifies all the relevant concepts, their attributes, and their interrelations.

The ontology has several purposes: 1) it serves as a vocabulary to use during consultations with stakeholders and domain experts, 2) it explicates choices made in the design, thereby making them traceable, 3) it can be checked for consistency and coherence, 4) it leads to early refinement and testing of the requirements, and 5) it serves as a declarative knowledge base for the system, promoting reuse of the generic procedural rules on interchangeable knowledge bases regarding for instance different training domains, didactic strategies, and virtual worlds.

The result of our research is a coherent and consistent ontology, forming a solid knowledge base that is useful to the stakeholders, developers, and the system itself. Moreover, the ontology led to a refinement of the requirements, newly discovered requirements, and a way to warrant the system's robustness. The ontology has been implemented and is currently checked for consistency. In short notice, the ontology will be verified by domain experts.

Future work focuses on the further development of a generic procedural rule base that uses the ontology to impute the rules' variables and produce user-tailored, domain specific, and adaptive training scenarios. This process will result in an even more detailed refinement of the system's requirements. Once the ontology and the reasoning rules are finished, an (agent-based) environment will be connected to the system, so the system can be evaluated on its requirements by testing their corresponding claims. The results of that test will lead to a further refinement of the architecture and system design.

Adaptive serious games have mainly focused on the maintenance of the storyline and believability of the characters, lacking didactical principles to adjust the storyline in favor of the learning goals. Alternatively, intelligent tutoring systems rely on didactical principles and result in structured learning content, but often only apply to well-defined training domains, such as computer programming, mathematics, and physics. Bridging the gap between these two research areas would result in highly engaging and effective autonomous training opportunities, however this requires a holistic view on game design: the development of adaptive systems is labour-intensive and reusability of (parts of) the game is important for the future of serious games.

Preferably, an AEG architecture uses several ontologies to draw its knowledge from: one referring to the training domain, one referring to the game world elements, one referring to the trainee, and one referring to the didactic strategies it can use to select and alter training scenarios. By combining semantic modeling, procedural content generation techniques, and adaptive storytelling elements, games may not just become adaptive, but modular and reusable as well.

References

1. Bidarra, R., de Kraker, K., Smelik, R., Tutenel, T.: Integrating semantics and procedural generation: key enabling factors for declarative modeling of virtual worlds. In: Proceedings of the FOCUS K3D Conference on Semantic 3D Media and Content. pp. 51–55 (2010)
2. Bouamrane, M.M., Rector, A., Hurrell, M.: Ontology-driven adaptive medical information collection system. In: An, A., Matwin, S., Ras, Z., Slezak, D. (eds.) Foundations of Intelligent Systems. Lecture Notes in Computer Science, vol. 4994, pp. 574–584. Springer Berlin / Heidelberg (2008)

3. Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What are ontologies, and why do we need them? IEEE Intelligent Systems 14(1), 20–26 (1999)
4. Falbo, R.d.A., Guizzardi, G., Duarte, K.C.: An ontological approach to domain engineering. In: Proceedings of the 14th international conference on Software engineering and knowledge engineering. pp. 351–358. SEKE '02, ACM (2002)
5. Hullett, K., Mateas, M.: Scenario generation for emergency rescue training games. In: Proceedings of the 4th International Conference on Foundations of Digital Games. pp. 99–106. FDG '09, ACM (2009)
6. Kessing, J., Tutenel, T., Bidarra, R.: Services in game worlds: A semantic approach to improve object interaction. In: Proceedings of the 8th International Conference on Entertainment Computing. pp. 276–281. ICEC '09, Springer-Verlag (2009)
7. Kickmeier-Rust, M., Albert, D.: The elektra ontology model: A learner-centered approach to resource description. In: Leung, H., Li, F., Lau, R., Li, Q. (eds.) Advances in Web Based Learning ICWL 2007. Lecture Notes in Computer Science, vol. 4823, pp. 78–89. Springer Berlin / Heidelberg (2008)
8. Kirschner, P., Sweller, J., Clark, R.: Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. Educational psychologist 41(2), 75–86 (2006)
9. Lopes, R., Bidarra, R.: Adaptivity challenges in games and simulations: A survey. IEEE Transactions on Computational Intelligence and AI in Games 3(2), 85–99 (2011)
10. Magerko, B., Wray, R., Holt, L., Stensrud, B.: Customizing interactive training through individualized content and increased engagement. In: The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC). vol. 2005. NTSA (2005)
11. Magerko, B., Laird, J.E., Assanie, M., Kerfoot, A., Stokes, D.: Ai characters and directors for interactive computer games. In: Proceedings of the 16th conference on Innovative applications of artifical intelligence. IAAI'04, vol. 1001, pp. 877–883. AAAI Press (2004)
12. Merrill, M.D.: First principles of instruction. Educational Technology Research and Development 50(3), 43–59 (2002)
13. Neerincx, M., Lindenberg, J.: Situated cognitive engineering for complex task environments. In: Schraagen, J., Militello, L., Ormerod, T., Lipshitz, R. (eds.) Naturalistic Decision Making and Macrocognition, pp. 373–390. Aldershot, UK: Ashgate Publishing Limited (2008)
14. Noy, N., McGuinness, D.: Ontology development 101: A guide to creating your first ontology. Tech. rep., Knowledge Systems Laboratory, Stanford University (2001)
15. Ohlsson, S.: Some principles of intelligent tutoring. Instructional Science 14(3), 293–326 (1986)
16. Omoronyia, I., Sindre, G., Stålhane, T., Biffl, S., Moser, T., Sunindyo, W.: A domain ontology building process for guiding requirements elicitation. In: Requirements Engineering: Foundation for Software Quality. pp. 188–202. Springer (2010)
17. Oser, R., Cannon-Bowers, J., Salas, E., Dwyer, D.: Enhancing human performance in technology-rich environments: guidelines for scenario-based training. Human Technology Interaction in Complex Systems 9, 175–202 (1999)
18. Peeters, M., Bosch, K.v.d., Meyer, J.J., Neerincx, M.: Scenario-based training: Director's cut. In: Proceedings of the 15th Int. Conf. on AIED. vol. 15, pp. 264–271 (2011)

19. Peeters, M., Bosch, K.v.d., Meyer, J.J., Neerincx, M.: Situated cognitive engineering: the requirements and design of directed scenario-based training. In: Proceedings of the 5th Int. Conf. on ACHI (2012)
20. Peirce, N., Conlan, O., Wade, V.: Adaptive educational games: Providing non-invasive personalised learning experiences. In: Proceedings of the 2008 Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning. pp. 28–35. DIGITEL '08, IEEE Computer Society (2008)
21. Rieber, L.: Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. Educational Technology Research & Development 44(2), 43–58 (1996)
22. Riedl, M., Stern, A., Dini, D., Alderman, J.: Dynamic experience management in virtual worlds for entertainment, education, and training. International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning 4(2), 23–42 (2008)
23. Salas, E., Priest, H.A., Wilson, K.A., Adler, A.B.: Scenario-based training: Improving military mission performance and adaptability. In: Britt, T., Adler, A., Castro, C. (eds.) Military Life: The Psychology of Serving in Peace and Combat, pp. 32–53. Praeger Publishers (2006)
24. Salehie, M., Tahvildari, L.: Self-adaptive software: Landscape and research challenges. ACM Transactions on Autonomous and Adaptive Systems (TAAS) 4(2), 14:1–14:42 (2009)
25. Si, M., Marsella, S., Pynadath, D.: Directorial control in a decision-theoretic framework for interactive narrative. In: Interactive Storytelling. pp. 221–233. Springer (2009)
26. Smelik, R., Tutenel, T., de Kraker, K., Bidarra, R.: A declarative approach to procedural modeling of virtual worlds. Computers & Graphics 35(2), 352–363 (2011)
27. Van Merriënboer, J.: Training complex cognitive skills: A four-component instructional design model for technical training. Educational Technology Publications (1997)
28. VanLehn, K.: The behavior of tutoring systems. International Journal of Artificial Intelligence in Education 16(3), 227–265 (2006)
29. Welie, M.V., Veer, G.C.V.D., Elins, A.: An ontology for task world models. In: Proceedings of the 5th International Eurographics Workshop on Design Specification and Verification of Interactive Systems. vol. 98, pp. 3–5. Citeseer (1998)
30. Westra, J., Dignum, F., Dignum, V.: Keeping the trainee on track. In: Computational Intelligence and Games (CIG), 2010 IEEE Symposium on. pp. 450–457. IEEE (2010)
31. Yannakakis, G.: Game ai revisited. In: Proceedings of the 9th conference on Computing Frontiers. pp. 285–292. ACM (2012)
32. Zook, A., Riedl, M.: A temporal data-driven player model for dynamic difficulty adjustment. In: Proceedings of the Conference on Artificial Intelligence and Interactive Digital Entertainment. AAAI (2012, to appear)