1-1-2012

A Meta-model for Developing Simulation Games in Higher Education and Professional Development Training

K. M. L. Cooper
University of Texas at Dallas

C. Shaun Longstreet
Marquette University, christopher.longstreet@marquette.edu

A Meta-Model For Developing Simulation Games In Higher Education And Professional Development Training

C. Shaun Longstreet
Center for Teaching and Learning, Marquette University
Milwaukee, WI
Kendra Cooper
Department of Computer Science, The University of Texas at Dallas
Richardson, Tx

Abstract: Games and simulations can be powerful educational tools for higher education and professional development training. At the same time, the labor and technical costs of development is sufficiently high that simulation games tend to be rather static and course or program specific. The main contribution of this paper is an effort to define a new meta-model for educational simulation games that is transferable to a broad range of disciplines and technical training fields. A meta-model provides a formal structure upon which similar categories of game play components are tied to specific learning objectives and easily updatable knowledge content while offering an infinite combination of playing experiences and maintaining consistent pedagogical standards. To illustrate, this paper draws from the application of our meta-model as used in developing a simulation game for software engineering education. Preliminary validation results are presented, in which the meta-model is used as a foundation in the architecture for a game development platform.

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

SECTION I.

Introduction

Educators and trainers from different institutions and programs often face similar challenges: they must teach a significant level of specialized, rapidly evolving knowledge that often requires quick and cost-effective updates. At the same time, effective educators work to increase student time-on-task, encourage participation from a broad range of students of different backgrounds and various levels of preparedness, while working to develop critical skills needed by industry. Simulations are useful educational tools because they encourage students to be active, self-paced, learners in a computer-generated reality wherein they receive rapid feedback on their progress as they interface within the virtual learning reality. Likewise, serious games have significant pedagogical potential as they provide immersive, engaging and fun environments that require deep thinking and complex problem solving within a construct of overcoming obstacles and challenges.[1] Simulation games create an on-line environment that provide environments where learners can apply and develop required knowledge and skills in an open, yet directed environment, with student progress required towards specific, measurable, learning outcomes.

This project developed out of a prototype simulation game for higher education science and engineering courses. As the simulation game developed, student developers rotated with each passing semester. To help us implement the game engine, we initially established a project glossary, which provided a common definition for terms for different development teams (computer science student teams that turned over every semester). For example, the term "assessment" has diverse meanings to educators, game engineers, and computer scientists. Although the restricted glossary of terms was helpful, the value of a meta-model to integrate the concepts and vocabularies of these three distinct disciplines became quite apparent as we proceeded to define the game play requirements, game engine requirements, and game engine architecture. From the educational perspective, we needed to associate game challenges, which required applying course content in a variety of simulated real world scenarios and tracking game outcomes, to the Software Engineering Body of Knowledge (SWEBOK) guide.[2]

In software development, a meta-model is a semantic construct that rigorously defines a collection of elemental building blocks and the rules that tie their interplay together. An ontology is a more abstract semantic construct that defines a vocabulary of terms and a grammar; the grammar constrains the well-formed formulae.[3] While many

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

educational games and simulation programs have been developed, there is little work available on re-usable meta-models or ontologies (cf.[4,5] which provide researchers with a general framework for analyzing games). As a first step, we are defining a meta-model. A meta-model lends itself to better verification via prototyping and the development of mini-simulation garnes, because it is less abstract than an ontology.[6,7] In the future, we contend that a more fully developed ontology could be a helpful contribution to educational simulation game development and an interesting research direction to pursue.

In this paper, we outline a meta-model for educational simulation games that is transferable to a broad range of disciplines and technical training fields. Building on an initial presentation of our research position,[8] the metamodel is described using examples from our SimSYS project. This meta-model serves several purposes. First, it provides a structure upon which similar categories of game-play components may be designed across a greater variety of subjects, providing a wide variety of playing experiences effectively and efficiently, while maintaining consistent pedagogical standards. Second, it allows both faculty and game designers to create tailored games easily by toggling particular elements of the game components that best suits progress towards course learning objectives. Third, the metamodel creates a consistent discourse for education simulation games for disciplines that may be applied across accreditation jurisdictions.

The approach we have selected to validate the metamodel has multiple steps. As a first step, we are utilizing the meta-model in the research and development of our game development platform (one project, one research team). An overview of the architecture for the platform is presented; the meta-model foundation is emphasized in the description. In the future, the validation needs to continue by applying it on additional projects, both internally and externally.

The structure of this paper is as follows. Section II presents the meta-model for simulation games; we use a running example from our SE simulation game for illustration purposes. The meta-model validation work to date is presented in Section III. Related work is described briefly in Section IV; Section V outlines conclusions and future work.

SECTION II.

Developing a Meta-model for a Educational Simulation Game (MMesg)

A. Components of MM_{ESG}

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

Our proposed simulation-game meta-model keeps in mind a holistic approach to gaming and learning. It establishes that, in addition to being engaging and entertaining, educational simulation games are immersive learning environments that embed specific learning objectives into a set of challenges expressing particular skills as classified by a learning taxonomy.

B. External Entities

Our meta-model considers two external entities. First, the game engages student players who function as the game Protagonist. An introduction to the simulation game requires the player to choose her or his character as part of an orientation to the game environment and the game-play fundamentals. Afterwards, the player must navigate a series of game challenges that, if completed successfully, will allow the player to defeat her or his computer-played NPC rival and progress into the next faculty-designed challenge. The specific challenges within the game are designed to reinforce and demonstrate content knowledge, application, analysis and evaluation. As a simulation game modeled with real-world parameters, the challenges will develop transferable skills required by programs and/or international accreditation agencies while providing continual, immediate, feedback to the student player.

The second external entity is the instructor and/or game developer. Instructor developers establish the simulation game parameters: by toggling contexts, characters and challenges they create a student learning environment through the use of our proposed templates. Templates support the modularization of game specifications; they are an established best practice in SE/Sys engineering processes. Game designers can instantiate the templates using a WYSHIWYG UI. While the student player receives feedback during game play, the instructor will also receive feedback on student player(s') progress and abilities, thereby adjusting future class and/or game simulation content.

C. Simulation Game Elements

One of the SimSYS example games is a third person simulation game wherein a student-player takes on the role of a budding Software Engineer who aspires to become a team leader for a software development company. During the game, the player must interact with non-player characters in a variety of situations in which she or he can advance only by successfully navigating a series of challenges that reflect real-world software engineering management issues.

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

1) Character

As noted above, the Character class includes the Protagonist (student-player), but also includes a variety of non-player characters (NPCs). For example, a primary NPC is the Antagonist, who provides personified adversity for the player in a more interactive manner. Each NPC has a profile, a set of variable attributes consisting of: name and title; game function (e.g. antagonist, director, constructor, interlocutor); role vis-à-vis the player (e.g. employer, client, colleague, friend, lab partner); relevant technical skills (i.e. degrees, talents); experience (i.e. level of effectiveness); and; charisma (i.e. ability to affect other NPCs).

In addition, NPCs can have pertinent demographic information for encouraging student thought about studying and working in a diverse social environment (e.g. gender/ethnicity/nationality). NPCs may be assigned a frequency of presence to the student-player to encourage or limit the level of assistance/challenge that that NPC offers.

In our example simulation game, the player character takes on the function of protagonist in the role of a Software Engineering student, who progresses through the game challenges acquiring rewards (i.e. being hired, acquiring certificates, awards, promotions) and/or penalties (not advancing, losing prestige). The non-player characters are organized by their function in the simulation. The primary non-player characters in our simulation game are: UR Boss, Dr. Ima Coder, and Nim Esis. UR Boss is a Director character; his role is the player's supervisor at the company SoftiCorp. Dr. Ima Coder is an Interlocutor character; her role is the player's software engineering instructor at GUI College and engages the player with questions during the dialog challenge. Nim Esis is an Antagonist; he is the CEO's nephew and his role is the student-player's peer in the classroom, then a co-worker at Softi-Corp, where they are both later hired (provided the student player progresses through the initial classroom challenges). Thus, the student-player and Nim Esis compete throughout the game. The main non-player characters are autonomous. The employee characters have specific software engineering titles. For example, the character Andrika Baker is a Junior software designer, who depending on the situation can be a team member, a director, or an interlocutor. In the end, the instructor designer can name and toggle a role and function for any number of NPCs as appropriate for the learning objectives and challenges at hand.

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

2) Context

The Context class establishes the simulation environment for the player/protagonist. It provides a grammar for the simulation environment and game play. The Context class contains the elements that create an immersive world within which students can interface. The Context class includes the scenario (e.g. an office, a hospital), the game situation (e.g. a mission briefing or a confrontation); a place within the game story arc, transitions and goals (e.g. moving from junior to senior staff, graduating from college) and provides the framework for the challenges in the game (e.g. final exam, a company project, or a road block). The interplay between the player/protagonist within the plot context and the non-player characters is designed to help students better understand course content from multiple perspectives, as well as reinforcing the effects of good and bad decision-making.

For example, in our SE simulation game, one location is a classroom in which the player competes in a content knowledge challenge against Nim Esis, wherein Dr Ima Coder asks a series of questions about Software Engineering in order to move on to the career fair. The player must also answer follow-up questions to demonstrate higher level cognitive processes such as application or analysis. Upon answering required number of questions correctly and defeating Nim Esis, the player can then move on to the next challenge.

3) Mechanics

The last class within the traditional game simulation elements is Mechanics. The Mechanics class provides the fundamental, low-level game resources such as characters, music, and graphics. It also supports maintaining the state of the game including the player's progress (where they are in the game) and their assessment (how well they are doing in the game, e.g. points). This class supports the Context class and the Challenge class.

D. Educational Game Elements

To facilitate the pedagogical base for rapid development of simulation games across curricula and training programs we have separated the game functionality (i.e. the raw simulation components) from the knowledge content and student assessment via specific modalities of game challenges (Figure 1). This allows faculty and trainer designers to input and switch out content quickly from course to course and, depending on student progress,

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

from class to class. That is, instructor designers could toggle a laboratory or a crime scene environment for the simulation context, and then input chemistry- or forensic-specific content in the game challenges that can vary from course section to course section.

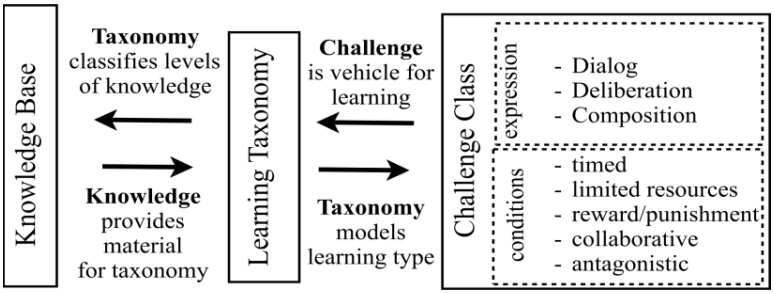

Figure 1. Meta-model Educational Game Elements

1) Knowledge Base

Every discipline or program has an information or knowledge set of which students must demonstrate some level of competency by the end of their education. In software engineering, international standards determine competency. The SE2004 guide organizes the Software Engineering Education Knowledge (SEEK) into three levels: knowledge areas; units; topics. SEEK defines ten knowledge areas, representing generally recognized sub-disciplines of software engineering. It then parses smaller units of knowledge from the ten areas, about which students need various levels of knowledge and skills competency.

2) Learning Taxonomy

The Learning class (taxonomy) represents the skills and abilities reflected in the simulation game learning objectives. For our meta-model, we use Bloom's taxonomy as the types of skills that student players must demonstrate a level of competency in the simulation game challenges.[7] For each knowledge base topic, one or several of Bloom taxonomy levels is assigned, indicating what skill level graduates should possess: knowledge (remembering learned material), comprehension (understanding information and the meaning of course content), application (ability to use learned material in new and concrete situations), synthesis and analysis (seeing the connections and components of larger systems), and evaluation (being able to assess the value of varied and variable information within specific contexts and requirements).

Game challenges align with the Learning class taxonomy. Instructors can use different challenges to accentuate different levels of knowledge. When a faculty member

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

would like to assess content retention, then he or she would toggle a Dialog challenge asking students to demonstrate knowing a term's definition. Likewise, more complex challenges that require demonstrating higher skill sets within the simulation game, provide opportunities for increased knowledge and skills building.

3) Challenge

The Challenge class holds different formative educational activities that are tied to specific, faculty-defined learning outcomes. As it is the linchpin of the educational simulation game, we developed additional modalities onto the Challenge class to allow for a wider range of possible learning opportunities. Challenges in our MM_{ESG} represent traditional, formative educational opportunities. Challenges are populated by non-playing characters (NPCs) who function in a variety of roles to communicate, assist or complicate the challenge for the student protagonist. The challenges are situated in a plotline wherein the student tries to advance. This narrative context, with competition against non-player adversaries, makes challenges fun and imaginative, rather than a typical on-line multiple choice test or written assignment.

The Challenge class is divided into two subcategories, Expression and Conditions. There are three Expression forms by which the game's assessment takes place: Dialog, Deliberation and Composition. There are four Conditions forms that affect limitations and motivations for student players in their engaging with the Challenges: Timed, Limited Resources, Rewards/Punishment and competition with NPC Antagonist character.

In the Dialog Challenge, an Interlocutor NPC asks the student player a line of questions, that require content knowledge at first but then some demonstration of understanding and possible applications of that knowledge. This encourages a Socratic approach to learning and demonstrating mastery of course content. Primarily, the Dialog Challenge appeals to lower levels of a learning taxonomy such as Bloom's (i.e. knowledge, understanding, application).[5] In our SimSYS simulation game, a simplified example is as follows:

> ***Question 1***
> *Concept: Design*
> *SWEBOK: General Design Concepts*
> *SWEBOK: Context of Software Design*
> *Level of Difficulty: Average Text: What does a design specification describe?*
> ***Quiz answer 1***

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

A specification of what will be implemented
Evaluation: INCORRECT
Feedback: No, I'm afraid not. A requirements specification defines what will be implemented.
Quiz answer 2
Text: A specification of how the software will be implemented.
Evaluation: CORRECT
Feedback: Yes, you're correct! You are on your way to earning your recommendation!

Another challenge is Deliberation, in which the player must pull apart a problem cluster provided in a Director character's briefing. In our Software Engineering game, for example, after a briefing on a new secret project, the player is given a set of resources that she or he must drag onto a simulated GANTT to establish a plan for completing a project under budget and on time. The Deliberation Challenge requires students to demonstrate higher skills sets related to higher levels of thinking beyond knowledge, understanding and application. Deliberation requires synthesis and analysis where students must demonstrate how components of a larger piece function together or are related within a greater whole.

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

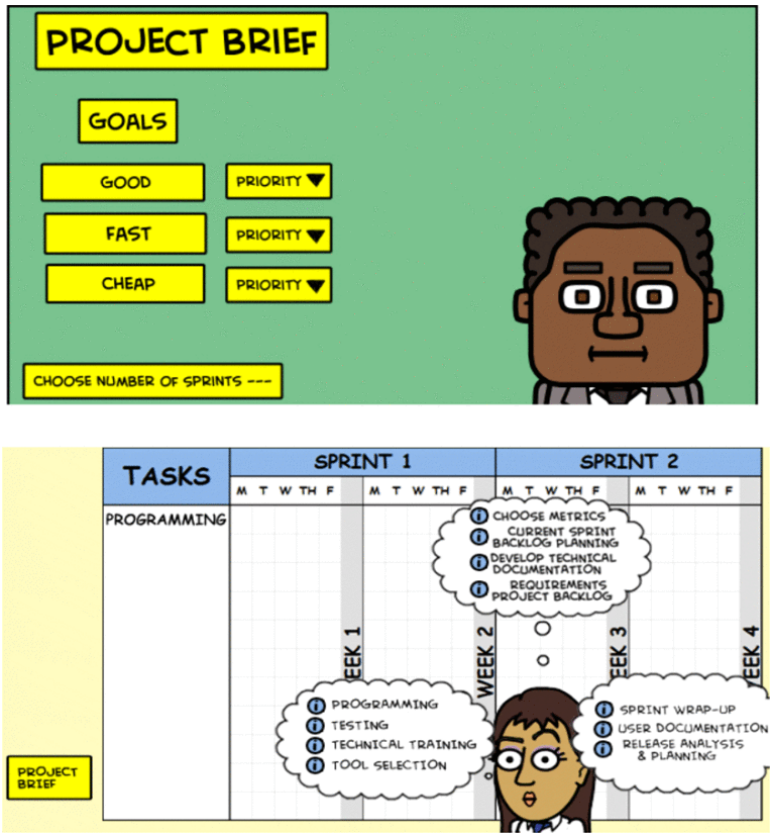

Figure 2. Screenshots of a Deliberation Challenge

The third Challenge Expression is Composition. This challenge also appeals to higher levels of skills application, wherein the student is tasked to assess larger pools of information for its relevance and merit in a more complex constellation of application and consequences. An example from our simulation game: the Director NPC requires the student player to review a set of very diverse résumés from a corporate pool to create a team that could best complete a specific project outlined by that NPC. That is, the SE student must assess several variables, in addition to having a solid grasp of particular content pieces, to successfully evaluate pertinent and, at times fuzzy, information from the pool of NPC résumés that can best address the needs to complete the project framed by the Challenge.

As noted, Challenges may have Conditions (Figure 1) set upon them as a means to motivate the player. For example, players can compete against time or the NPC Antagonist character. They may also need to play with strategically limited resources or a combination of limited conditions. For example, in a dialog challenge, the player might compete against an NPC antagonist under a time limit to answer each question. On the other hand, that

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

same challenge might be free of any conditions and a player is only taking a self-test to check her or his comprehension or recall ability.

SECTION III.

validating the meta-model: SimSYS Game Development Platform

The approach selected for validating the meta-model is to use it as a foundation for the SimSYS GDP prototype development effort. Here, we present the architecture of the SimSYS GDP; the discussion of its components reflect the application of the meta-model. For emphasis, elements in the meta-model are presented in bold, italic font. As the prototype progresses, errors in the meta-model can be identified and corrected.

The SimSYS **GDP** is a means to investigate research issues in intelligent, semi-automated game generation, automated player assessment and automated game adaptation, in addition to agent-oriented game play engines (Figure 3). The architecture for the platform is based on the repository style. This style consists of a repository of data with controlled access and a collection of knowledge sources that directly interact with the repository (i.e. they are clients of the repository). Here, the repository is a collection of domain dependent and domain independent sub-repositories. The domain dependent sub-repository stores game scripts, game play data, and re-usable game assets such as collections of domain specific ***Challenges*** (e.g., quiz questions/answers/feedback about software engineering, mathematics, physics, and so on) that are used in the games. The challenge content also captures relationships to ***Body of Knowledge*** standards (e.g., SWEBOK), ***Taxonomy of Learning*** standards (e.g., Bloom's) and the level of difficulty. Re-usable domain independent game assets, such as general sound effects or common graphic images, are stored in a separate sub-repository.

The knowledge sources for the platform are a collection of modular components, which can interact indirectly via changes to the repository. The modules include Game Generation, Player Assessment, Game Adaptation, and a Game Play Engine. These modules embody knowledge of related standards, including body of knowledge, accreditation, and certification. The Game Generation module provides an intelligent approach to semi-automatically generate games for a domain. A wizard interface prompts the user to enter parameters for the game they need such as domain, topics, depth, level of difficulty (target players), length, and plan for use in the curriculum. The Player Assessment module provides an unobtrusive means to collect game play data. The approach is configurable, to make it applicable to a wide variety of games. The data collected include the number of

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

correct answers, incorrect answers, length of time to answer a question, number of times the player changes their answer. The data are collected as the game is played and stored in the repository for subsequent analysis; summary reports are automatically generated for the instructor. The Game Adaptation module analyzes the game play data to determine if the game is too easy, too hard, or at a good level for each individual player; game play data for collections of players is also analyzed in this module. This module makes adaptation recommendations and chooses questions to present to the player at the next level of the game. The Game Play Engine module reads in the game script (XML); the player interacts with this module to play a game. Traditional game elements are supported in this module, including the ***Characters, Context***, and ***Mechanics***. To provide a more interesting game the game play engine uses an agent-oriented approach, which makes the nonplayer characters more autonomous. The intent is that a nonplayer character can behave differently from game to game.

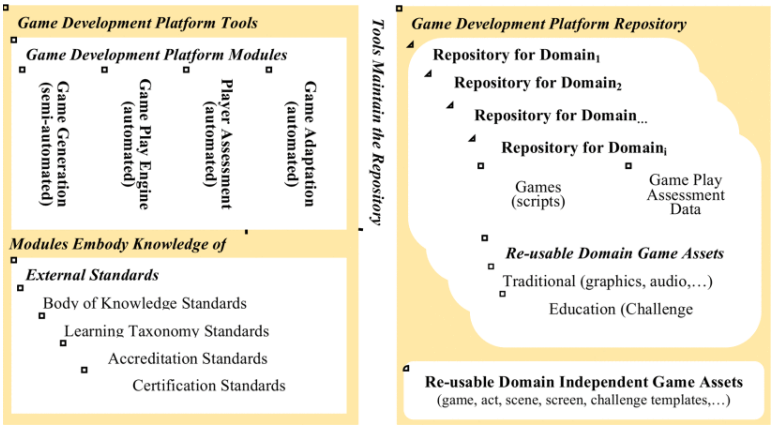

Figure 3. Realizing the Meta-model in the SimSYS Game Development Platform Architecture

SECTION IV.

Related Work

The Game Ontology Project (GOP) is developing a game ontology that captures the structural elements of games and the relationships between them.[2] The top level of the ontology consists of five elements: interface, rules, goals, entities, and entity manipulation. The five elements are hierarchically decomposed and maintained on a website.[5] The *interface* is how the player interacts with the game and how the game communicates to the player. The *rules* define what can or can't be done in a game; they lay down the framework, or model, within which the game shall take place. *Goals* are the objectives or conditions

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

that define success in the game. The assessment of the player is included in the decomposition of this element. *Entities* are the objects within the game that the player manages, modifies or interacts with at some level. This definition includes objects that are not controlled by the player. Finally, *entity manipulation* encompasses the alteration of the game made either by the player or by in-game entities. Entity manipulation thus refers to the actions or verbs that can be performed by the player and by in-game entities.

Each ontology entry consists of a title or name, a description of the element, examples of games that embody the element, a parent element, related child elements, and part elements (elements related by the part-of relation). The website provides links to explore the text-based ontology; a visual (graphic) representation is not available.

SECTION V.

Conclusion and Future Work

A significant consequence of our work to extend the state-of-the-art research in SE education was, in part, proposing a game development platform using an interdisciplinary perspective (from that of educational theory, game research, and of software engineering). As work developed, we recognized the utility of creating a meta-model wherein a simulation game could be effectively and efficiently developed, not just from development team to development team, but from different knowledge-based disciplines as well. To summarize our preliminary results, the development of the meta-model has shown to be useful already as it has begun to streamline the development of our education-game development process. Because of the student developers rotation from semester to semester, having the meta-model allows for more rapid transitioning of teams as well as a much faster grasping of the projects' fundamental purpose and direction. At the same time, the meta-model has also led to a more refined construction of game challenges that could be traced to SWEBOK classifications.

Regarding our current limitations, we recognize validation of the model is limited to our preliminary prototyping work (one research team, one project). In the future, additional validation is needed, both by continuing our own project and by external educational game researchers. Currently the meta-model does not yet consider multiplayer functionality and character behavior is rather limited in that the meta-model does not outline considerations for player choice of expression, character movement or in-scene object manipulation. We plan to explore these issues as future work.

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.

The meta-model serves to help faculty rapidly develop consistent, connected, but also varied and easily adaptable simulation game challenges. As such, it has great potential for exploring the adoption and implementation of a game-based curriculum, wherein whole programs, courses and curricula could be tied to game-play aimed towards cultivating sets of knowledge and skills acquisition.

In conclusion, the development of MM_{ESG} as a seed for an educational simulation game ontology would allow developers to create myriad challenges for student player protagonists in which a shared body of conceptualizations could be developed across institutions while maintaining international accreditation standards.

References

[1]Gee, J.P., 2003. What video games have to teach us about learning and literacy, Macmillan, USA.

[2]David Michael and Sande Chen, Serious Games: Games That Educate, Train, and Inform, Course Technology PTR, 2006

[3]Alain Abran, James W. Moore and Pierre Bourque Guide to the Software Engineering Body of Knowledge (SWEBOK), 2004 Version by IEEE Computer Society, Appendix D.

[4]M. Fernndez, A. Gomez-Perez, and N. Juristo, 1997. "METHONTOLOGY: from ontological art towards ontological engineering." in: Symposium on Ontological Engineering of AAAI. Stanford, CA.

[5]Jose P. Zagal, Michael Mateas, Clara Fernandez-Vara, Brian Hochhalter, Nolan Lichti - "Towards an Ontological Language for Game Analysis", Proceedings of the Digital Interactive Games Research Association Conference (DiGRA) (2007), Tokyo, Japan, pp. 516-523.

[6]Game Ontology Project webpage, available at http://www.gameontology.com.

[7]T. Gruber. Towards Principles for the Design of Ontologies for Knowledge Sharing. International Journal of Human Computer Studies 43(5/6), pages 907-928, 1995.

[8]Guarino, N.: Formal Ontology in Information Systems. Proceedings of FOIS'98, IOS Press, Amsterdam. (1998) pp. 3-15

[9]Building Core Ontologies. White Paper of the DELOS Working Group on Ontology Harmonization. April 2003.

[10]Shaun Longstreet and Kendra M.L. Cooper, "Developing a Meta-Model for Serious Games in Higher Education," Proceedings of the International Conference on Advanced Learning Technologies (ICALT) (2012), Rome Italy.

[11]A. Tull, T. Smith, and K. Cooper, "Towards an Agent-Oriented Framework for Serious Games, Architecting with Behavioural Software Agents", International Conference on Simulation and Modeling Methodologies, Technologies, and Applications (Simultech)(2011), Amsterdam, Netherlands.

[12]F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, Pattern-Oriented Software Architecture Volume 1: A System of Patterns, Wiley, 1996.

[13]Anderson, L. & Krathwohl, D. A. (2001) Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives New York: Longman.

2012 17th International Conference on Computer Games (CGAMES), (2012): pg. 39-44. DOI.