

A task annotation model for SandBox Serious Games

Francesco Bellotti, Riccardo Berta, Alessandro De Gloria, and Ludovica Primavera

Abstract— SGs represent a great potential for education since they can join an instructional value with the appealing language of digital natives. Searching for methodologies to support effective authoring of Serious Games (SGs), we have abstracted a model consisting of a 3D geo-referenced Virtual World (VW) where knowledge is implemented also through tasks that are disseminated across the environment.

The model is designed to support authors to focus on defining the tasks' content and annotation (e.g. relevance to various topics, difficulty, position in the VW, typology of supported learning styles, interrelation with other tasks and events, etc.), without being asked to do game scripting, which typically involves hard-coding events, actions, locations and contents of a specific game. Task annotation allows reusing tasks also in different games. Moreover, different game designers can exploit repositories of tasks and build educational games specifying a VW, and its missions as sequences of tasks that meet learning strategy criteria that can be expressed through the proposed framework. This model opens new perspectives to pedagogical experts and game designers to develop runtime game engines able to provide players with adaptive contents. The paper presents the user, task and learning strategy models in detail and an example of application.

Keywords— Serious Games, technology enhanced education, user modeling, task based learning, learning spaces.

I. INTRODUCTION

The research community is paying a growing attention on how to support learning through computer games, given their appeal on a wide audience, their capacity to keep the player's attention for long time spans and their ability to provide realistic simulations. Serious Gaming (SGs) is a specialization of computer gaming that addresses educational objectives and has developed a number of high-level products in several application domains [1]. In this field, there is a clear need for supporting pedagogical authors with methodologies and tools that can support them in providing effective learning experiences [2].

Exploring this challenge, we have analyzed a number of successful SGs (e.g. [3, 4, 5, 6, 7, 8]) and observed that there is a class – the Sand Box Serious Games (SBSGs), with a counterpart also in successful pure entertainment games such as Grand Theft Auto¹ and Oblivion² - that tends to provide

players with suited knowledge structures for investigating a specific educational domain. SBSGs lend themselves well to be defined through an abstract model for facilitating authors in creating adaptive contents. The abstract model consists of 3D a geo-referenced Virtual World (VW), where knowledge is implemented also through tasks disseminated throughout the environment. Tasks embody units of knowledge that have to be discovered by the player in the environment and tackled in order to progress in the game. This supports a goal-oriented VW exploration and a problem-based interaction with contextualized, specific and focused items (e.g. quizzes, mini-games, conversations with Virtual Humans) that are aimed at constructing meaning, building lasting memories and deepening understanding of the touched item(s).

Simple tasks can be realized as instances of configurable software templates that can be easily created by pedagogical authors by simply inserting domain-specific contents, without any need for programming knowledge [9]. This allows creating a wide basis of tasks - also exploiting the User Generated Contents trend that is now popular in TV and multimedia. The subsequent point concerns how to deliver these tasks to the user in a game.

In general, two aspects are fundamental when designing tasks: the content and their delivery strategy (i.e. when, where and how they become available to the player or are directly assigned to the player). The model we propose involves annotation of tasks, so that they can be re-used also in different games and their delivery can be managed dynamically at runtime by the game engine, matching the actual profile of the user and the educational strategy of the game designer. Our proposal intends to decouple the two aspects. In our view, a domain expert pedagogical author has to focus on defining the tasks' content and annotation (i.e. definition of metadata such as relevance to various topics, difficulty, position in the VW, typology of supported learning styles, interrelation with other tasks and events etc.), while a game designer specifies the requirements for a delivery strategy according to her educational and entertainment objectives. Annotated tasks are put in a repository and can then be exploited by various game designers who define the requirements (i.e. the delivery strategy) for their game. It will be the responsibility of a Computational Intelligence (CI) module embedded in the runtime game engine to learn the strategy and determine the most suited sequencing of tasks according to the actual user profile.

A general description of the whole system, with particular

This work was supported in part by the European Union under the Culture 2000 programme (Travel in Europe project) and by Regione Liguria (SeaGame project).

Authors are with the Dept. of Electronics and Biophysical Engineering, DIBE, University of Genova, Via Opera Pia 11a, 16145, Genoa, Italy (phone: +39-010-3532795; e-mail: {franz, berta, adg, primavera}@elios.unige.it)

¹ <http://www.rockstargames.com/sanandreas/>

² <http://www.elderscrolls.com>

attention to the runtime CI module - namely, the Experience Engine (EE) – and a computational application example, is provided in [10]. This paper, instead, presents the details of how we model tasks and users and allow the game designer to specify the delivery strategy. The paper also describes a simple application in an SG dedicated to the promotion of the cultural heritage.

II. BACKGROUND AND STATE OF THE ART

Adaptive learning has been deeply investigated in the eLearning research, which has yielded to the development of Intelligent Tutoring Systems (ITS) [11, 12] and Adaptive Hypermedia [13]. For SGs, a great challenge consists in combining agency, which is a key feature of compelling state of the art commercial games, with effective teaching strategies [14]. This is typically tackled through author's scripting. The game scripting is very effective in game design in order to provide highly dramatic and/or entertaining experiences, but cannot be adapted easily to fit the learner's profile and needs [15]. Our proposal intends to exploit and specify the SBSG model in order to enhance authoring for entertainment and education, with the aim to make interaction with educational contents appealing to a wide audience - potentially, the present audience of current gamers.

In literature, starting from the Aptitude-Treatment Interaction (ATI) concept [16], the pedagogical research has developed the notion of adapting the form of instruction to the student features. This research field has focused on the identification of the salient characteristics that can be used to anticipate differences between users. The particular configuration of these characteristics for a user is called his learning style [17]. It is assumed that the more the form of instruction is relevant for a user's learning style the more efficiently he learns. We adopt the concept of Learning Space to model the interaction between the learner and the tasks in a SG. The Learning Space idea provides a quantitative model for learning, positioning each player's profile in a space defined by the following dimensions: visual, auditory, reading and kinesthetic learning modalities.

For the entertainment part, a commonly accepted user's enjoyment model does not exist yet [18]. However, recent psychological research has focused on the identification of what can engage people in playing videogames. The main results are the three principles of Malone theory (challenge, fantasy and curiosity) [19] and the theory of Flow [20], which has been applied also to the videogames sector, with the GameFlow model [21, 22, 23, 24, 25]. Other theoretical models include disposition theory [26], attitude [27] and transportation theory [28].

The framework we propose keeps into account the concept of Zone of Proximal Development (ZPD), which is the difference between the actual player level and the level of potential development [29]. Our proposal of user model and task model annotation allows a game designer to provide the runtime managing system with the information needed to dynamically select contents that fit in the player's ZPD.

III. AN ABSTRACTION MODEL OF SERIOUS GAMES

We propose to model an SBSG as an exploration field where the player is given missions from the system and each mission consists of a sequence of specific, contextualized tasks embedded in the VW environment. The EE module is the responsible to define at runtime the most appropriate sequence of tasks for a mission based on the matching between the user profile and the task profiles and according to the educational delivery strategy specified by the game designer. It is important to notice that, differently from some commercial Sand Box Videogames (e.g. Oblivion), when a mission's task sequence is decided by the EE, the player has to follow it without any branching possibilities.

This model can be schematized as a simple tree structure (Fig. 1) that can be easily handled by the author. Leaf nodes represent the basic units of the gameplay, the tasks. Internal nodes can be mapped the game concept of missions, objectives and the overall game. The model can account for any number of levels.

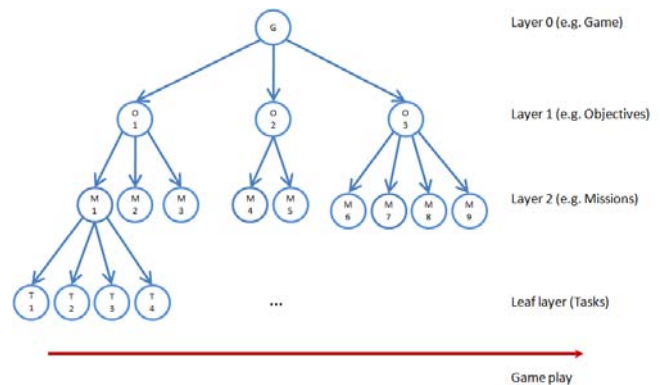


Figure 1. Tree representation of a game modeled as an aggregation of tasks. Tasks correspond to the units that constitute a mission. Missions can be further hierarchically structured. Time (the gameplay) flows left to right.

The gameplay corresponds to a depth-first traversal of the tree. Starting from the root, the game introduces the first objective, mission and task. When all the first mission's tasks have been accomplished, the second mission is triggered, which leads to the exploration of all its tasks. When all the first objective's missions are finished, the next objective is played, and so on until a complete exploration of the tree or a premature end of the game.

The requirements for the decision policy (i.e. the EE's strategy for delivering the tasks) are parameters of the model as well. When designing a SBSG, the game designer will have to specify the delivery strategy requirements (possibly also families of delivery strategies, that may be chosen at runtime by the user or a tutor as they may prefer putting an emphasis on different aspects, such as education with respect to entertainment).

In the following, we present an implementation of the models of the tasks, user and delivery strategy that is based on our studies and field experience. Such models may be easily upgraded in order to account for different parameters.

A. Model of the tasks

A task is characterized by its author-defined value in the parameters that we present in table I.

TABLE I
TASK MODEL PARAMETERS

| Name | Symbol | Description |
|-----------------------------|-----------------------|---|
| Type | T_{Ti} | Tasks could be of very various types. For instance, a task could be a puzzle, an action mini-game. The author defines the list of task types available in her SG. A task can be of one single type or a combination of several types, in different proportions. For a task, the sum of such proportions is normalized to 1: $\sum_{i=0}^{nTypes} T_{Ti}=1$. nTypes is the total number of available types. |
| Entert. Value | T_E | The entertainment value of the task, normalized. |
| Quality Val. | T_Q | The quality value normalized between 0 and 1 |
| Length | T_L | Expected length, normalized between 0 and 1 |
| Skill Relevance | T_{SRi} | The author pre-defines a list of subjects/skills relevant to the SG – we generically call them skills. The list may come from a domain ontology (e.g. LOM [30]). Every task covers (i.e. is useful to develop knowledge on) one or more of such skills, in different proportions. For a single task, the sum of such proportions is normalized to 1. |
| Skill Benefits | T_{SBij} | For each one of the Skills covered by the task, the author specifies the expected benefit for a player performing that task. This is a 0-1 normalized curve, which is function of the player's ability level in that subject/skill (the player's ability is quantized on a 0-1 scale, as described in the user model). In the symbol, i refers to the ability, j to the user ability's level. |
| Difficulty | T_D | A difficulty value normalized between 0 and 1. This value can also be estimated as a function of the benefits for each covered subject (each subject weighted with the relevant coverage value). $T_D = \sum_{i=0}^{nSkills} T_{SRi} \sum_{j=0}^{nLevels} j * T_{SBij}$ The function estimates as difficult those tasks whose benefits are greater for skilled users. |
| Difficulty Adaptation Range | T_{DAR} | In order to increase the system flexibility and scalability, some tasks typologies have a difficulty-adaptation range. For instance, a multiple-answer quiz can adapt its difficulty by adding and/or cutting some possible choices, or increasing/decreasing the allowed response time. The corresponding adaptation range can be automatically computed by the system. |
| Covered Learning Styles | T_{LSi} | The author pre-defines a list of learning styles relevant to the SG - we describe learning styles in the user modeling subsection. Every task supports one or more styles, in a different proportion. The sum of such proportions is normalized to 1. |
| Dependences Place | T_{Dep} T_{Pi} | The list of tasks on which this task depends. The x,y coordinates in the SG's Virtual World where this task is placed (if relevant) |
| Interactivity Level | T_{IL} | The interactivity level of the task, normalized. For instance, questions typically have a low T_{IL} value, while action mini-games have high T_{IL} . |

All these values (also including entertainment and quality) are estimated and set by the author. For instance, some tasks may be funny but have little educational value and viceversa.

The author can leave some task parameters as adaptive. This means that the system can update their value at runtime, according to proper heuristics that evaluate the actual game

play. For instance, the skill benefits of a task could be dropped if players tend to perform easily well in that task and badly in a subsequent dependent task. Estimating entertainment without requiring explicit user feedback is more difficult. But entertainment can be estimated considering the task interruption rates and times [10], according to the Flow theory [25] and the ZPD [29].

We have modeled some parameters in a possibly simplistic way. For instance, the entertainment and quality factors are absolute, thus independent of any user's feature. If necessary, the model can be upgraded.

B. Model of the user

An important feature of the model concerns the position of the player in a Learning Space having dimensions such as: visual, auditory, reading and kinesthetic learning modalities. The player position is defined by his strength in the different modalities, while the task position is defined by its support for the different modalities. Our user and task models support Learning Spaces through the corresponding parameters (Learning Style Need and Covered Learning Styles).

TABLE II
USER MODEL PARAMETERS

| Name | Symbol | Description |
|--------------------------------|-------------|--|
| Skill Level | U_{SL_i} | The 0-1 normalized and quantized ability level in each of the skills/subjects defined by the author as relevant to the SG. |
| Navigation Ability Level | U_{NAL_i} | The normalized ability level of the user in navigating in the SG's VW. This parameter is used in combination with the placement of the tasks in order to choose an adequate distance between a task and the subsequent one (adequate considering the exploration of the VW as a challenge). |
| Task Type Need/Preference | U_{TT_Ni} | For each task type, how much it is judged by the author as needed by the player or how much it is preferred by the player (the author has to specify which criterion to use). The total of the preferences/needs must sum to 1: $\sum_{i=0}^{nTaskTypes} U_{TTNi}=1$. These values represent the target (ideal) probability distribution for the task type in a sequence. |
| Skill Need/Preference | U_{S_Ni} | As above, but considering covered skills instead of task types. |
| Learning Style Need/Preference | U_{LS_Ni} | As above, but considering learning styles. |

Considering the need/preferences distributions of the last two rows of the Tab. 2, the author can specify a *variety* factor, in a 0-1 scale. A high variety level means that the actual distribution should be smoothed in order to deliver a generally wider variety of choices than as strictly specified in her needs/preferences. The smoothing law is linear.

$$x' = x - variety * (x - x_e)$$

Where x' is the smoothed value, x is the original value and x_h is the value for the equi-probable distribution.

A value 0 variety means that the actual target distribution should be the one defined by the author/player. A value 1 variety means that all the task types become equally probable.

Fig. 2 (a) shows a sample task type probability distribution, while (b) shows the distribution obtained after applying a 0.5 valued *variety* factor.

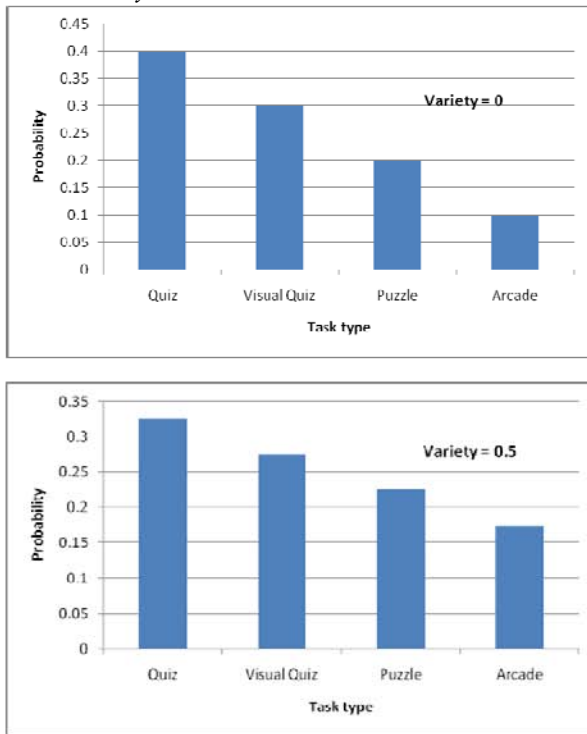


Figure 2. (a) A sample task type target probability distribution. (b) The new distribution obtained after applying the variety factor.

A couple of user model parameters depend on each task. These parameters suppose that a system records the history of the user among game sessions.

TABLE III
USER-TASK PARAMETERS

| Name | Symbol | Description |
|-----------------|--------|---|
| Number of Times | UT_n | Number of times the task T has been done by the user U |
| Performance | UT_p | Average performance of the user in the task. Performance is normalized 0-1. |

C. Model of the delivery strategy

The above defined model of users and tasks allows a quantitative evaluation of several items/factors that can be considered by the author when assessing the value of a sequence of tasks to constitute a mission.

The delivery strategy requirements can thus be simply expressed by the author by defining the parameters of a linear *cost function*. Each additive term of the cost function represents a *Cost Item* (CIs) of the analyzed sequence. For each item, the author specifies the sequence's *reference features* (values, curves, distributions - we detail them in the following list). Every item of the cost function penalizes, with a weight defined by the author, the deviation of the analyzed/assessed sequence from such reference features.

$$cost(sequence) = \sum_{i=0}^{nCIs} weight_i * CI_i(sequence)$$

All the CIs are in the range 0-1, with the exception of the inter-task dependence failure, since it is important that the author can strongly penalize the insertion of a task that does not meet a dependency requirement. Also, tasks are play-once - they cannot appear more than once in a sequence. This hard constraint is checked at runtime by the system.

We suppose that the sequences' length is fixed. In game terms, this means that the number of tasks for a mission is pre-determined. Different missions can have different lengths.

Here follows a detailed description of the CIs.

1) Difficulty Curve

The author specifies a *reference difficulty curve* for the tasks in a sequence. The curve values are specified relative to a base level. This level has to be determined at runtime by the game engine averaging the player's current skill ability levels weighted over the target needed skills. This is typical in personalized education, where the teacher starts from the learner's level and then tries to draw her to ever more difficult tasks/concepts.

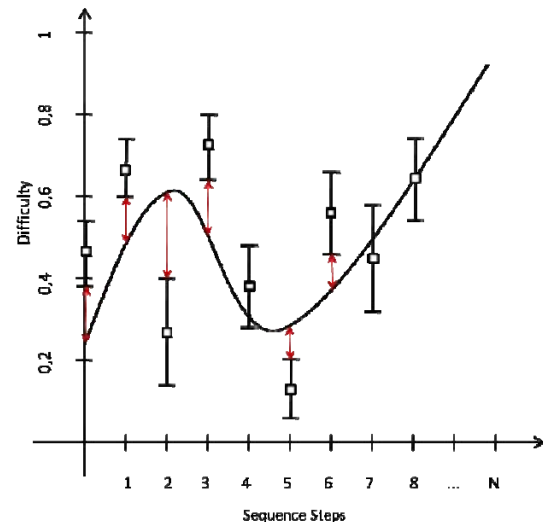


Figure 3. An example of a reference difficulty curve. Tasks are represented as squares, each one with its difficulty adaptation range. The deviation for each task in the sequence is represented as a bidirectional arrow.

The *Difficulty Curve* CI penalizes the deviation of each task in the analyzed sequence from the relevant value in the reference curve (the *target* value). Difficulty-adaptation ranges are considered. This means that the deviation is computed from the target difficulty value to the closest extreme of the difficulty adaptation range. If a task difficulty can be adapted to the target difficulty (i.e. the target difficulty at the given step is within the task's difficulty adaptation range), the deviation is 0. Fig. 3 gives an example of a reference difficulty curve, showing also the tasks in the sequence, their difficulty adaptation range and deviation from the reference value.

The total deviation (or distance) of the sequence is:

$$dev = \frac{1}{seqLength} \sum_{a=0}^{seqLength} |T_{Dad}^a - D^a|$$

Where $seqLength$ is the length of the sequence, the superscript indicates the a^{th} task in the sequence. D^a is the difficulty level stated by the author for the a^{th} task. Dad indicates the adapted difficulty of the task:

$$\begin{aligned} T_{Dad}^a &= D^a \text{ if } T_D^a - T_{DAR} \leq D^a \leq T_D^a + T_{DAR} \\ T_{Dad}^a &= T_D^a - T_{DAR} \text{ if } D^a < T_D^a - T_{DAR} \\ T_{Dad}^a &= T_D^a + T_{DAR} \text{ if } D^a > T_D^a + T_{DAR} \end{aligned}$$

The CI is quantitatively defined as a linear function of the deviation. The author specifies a threshold deviation value over which the cost saturates to 1. This, combined with the fact that the cost is 0 when dev is 0, determines the linear function.

2) Navigation difficulty curve

This term express the difficulty (estimated through the distance) for a player to reach a destination (task n) from the starting point (task n-1). The author thus specifies a reference curve (as for the difficulty) also for the geographical distance between every couple of subsequent tasks. The total deviation of the sequence is:

$$dev = \frac{1}{seqLength} \sum_{a=1}^{seqLength} |T_{Dist}^a - Dist^a|$$

$Dist^a$ is the reference distance stated by the author from the a-1th to the a^{th} task. T_{Dist}^a is the actual distance from the a-1th to the a^{th} task in the analyzed sequence.

The CI is defined by the author as a linear function of the deviation with saturation, as described in the case of the *Difficulty Curve*.

3) Inter-task dependences

The author specifies the penalty for a task if it is not preceded in the sequence by a task on which it depends. A cost may also be assigned to a task if it does not depend on any of the previous tasks. This aims at favoring the composition of sequences of interrelated tasks, thus promoting discovery paths constituted by complex chains of consequential items, which typically supports the building of knowledge structures. The total deviation of the sequence is the sum of the dependence costs at each step:

$$dev = \sum_{a=0}^{seqLength} C_{dep}^a$$

$$\text{where } C_{Dep}^a = C_{NoDep} * Nd^a + C_{FailedDep} * Fd^a .$$

C_{NoDep} is the cost defined by the author to penalize the absence of dependences from previous tasks. Nd^a evaluates to 1 if the a^{th} task has no dependences, 0 otherwise.

$C_{FailedDep}$ is the cost that penalizes the absence, in the previous tasks of the sequence, of a task on which the a^{th} task depends. Fd^a evaluates to 1 if the a^{th} task has a dependency failure (i.e. there is at least one task on which the a^{th} task depends that has not been inserted in the sequence before the a^{th} position), 0 otherwise.

Differently from all the other CIs, the dependence cost is

not normalized to 1, since it is important that the author can strongly penalize the case of inserting a task that does not meet the dependency requirement (i.e. $C_{FailedDep}$ can be $\gg 1$).

4) Task type distribution

The author specifies the needed (preferred) distribution of the task types for a sequence. This CI penalizes deviations of the analyzed sequence from this target distribution. The total deviation of the sequence is:

$$dev = \sum_{i=0}^{nTaskTypes} \left| U_{TT_N_i} - \frac{1}{seqLength} \sum_{a=0}^{seqLength} T_{T_i}^a \right|$$

Where $U_{TT_N_i}$ is the user's task type need/preference of the i^{th} task type, after being corrected with the *variety* factor (the formula is in III.B).

The CI is defined as a linear function of the deviation, with saturation over a threshold, so that the cost is limited between 0 and 1.

5) Learning style distribution

The author specifies the needed distribution of the learning types supported by the tasks in the sequence. The CI penalizes deviations of the analyzed sequence from this target distribution. The total deviation of the sequence is:

$$dev = \sum_{i=0}^{nLearnStyles} \left| U_{LS_N_i} - \frac{1}{seqLength} \sum_{a=0}^{seqLength} T_{LS_i}^a \right|$$

Where $U_{LS_N_i}$ is the user's need/preference value of the i^{th} learning style, after being corrected with the variety factor.

$T_{LS_i}^a$ represents the relevance of the i^{th} type of learning style for the a^{th} task.

The cost is defined as a linear function of the deviation, with saturation over a threshold.

6) Covered skills distribution

The author specifies the needed/preferred distribution of the skill/subjects covered by the tasks in the sequence. The CI penalizes deviations of the analyzed sequence from this target distribution. The total deviation of the sequence is:

$$dev = \sum_{i=0}^{nSkills} \left| U_{S_N_i} - \frac{1}{seqLength} \sum_{a=0}^{seqLength} T_{S_i}^a \right|$$

Where $U_{LS_N_i}$ is the user's skill need/preference of the i^{th} skill, after being corrected with the variety factor.

$T_{S_i}^a$ represents the relevance of the i^{th} skill for the a^{th} task.

The cost is defined as a linear function of the deviation, with saturation.

7) Trend for entertainment and Trend for quality

The author can choose between an increasing, decreasing or alternate trend for the entertainment value. This cost is measured considering the difference in value (thus not absolute values) of all the subsequent couples of tasks. For instance, an alternate trend requires a continue alternation of

increments and decrements. A penalty occurs if, when a local rise is expected, the next task has an entertainment value lower than the previous one.

The total deviation of the sequence is the sum of the entertainment trend costs at each step of the sequence (starting from the second task):

$$\text{dev} = \frac{1}{\text{seqLength}} \sum_{a=1}^{\text{seqLength}} C_{\text{Etrend}}^a$$

For instance, in the case of an increasing trend, if $T_E^{a-1} < T_E^a$, $C_{\text{Etrend}}^a = 0$. Otherwise, C_{Etrend}^a is a linear function of $T_E^{a-1} - T_E^a$.

The cost is defined as a linear function of the deviation.

A similar concept applies to the trend for quality.

8) Entertainment threshold and Quality threshold

A minimum entertainment value can be set. Tasks with lower entertainment values are penalized

The total deviation of the sequence is the sum of the costs at each step of the sequence:

$$\text{dev} = \frac{1}{\text{seqLength}} \sum_{a=0}^{\text{seqLength}} C_E^a$$

Where the Entertainment Cost for the a^{th} task $C_E^a = 0$ if $T_E^a > ME$. ME is the Minimum Entertainment threshold. Otherwise, C_E^a is a linear function of $ME - T_E^a$.

The cost is defined as a linear function of the deviation, with saturation.

A similar concept applies for the quality threshold.

9) Performance curve

The author specifies a reference curve, as for the difficulty curve. In this case, the evaluated value is the previous performance of the player in the analyzed tasks. Typically, we expect that the author will specify a decreasing trend, leaving harder tasks (tasks in which the player historically performs worse) towards the end.

The total deviation (or distance) of the sequence is:

$$\text{dev} = \frac{1}{\text{seqLength}} \sum_{a=0}^{\text{seqLength}} |UT_p^a - P^a|$$

Where UT_p^a indicates the average user performance in the a^{th} task in the sequence. P^a is the performance level desired by the author for the a^{th} task.

The cost is defined as a linear function of the deviation, with saturation.

10) Skill Benefit threshold

The author specifies a threshold over which the skill acquisition benefit of a task for the user should be. The benefit is computed as the weighted sum of the benefits of the task for each one of the covered skills, considering the user ability level at each skill.

The total deviation of the sequence is the sum of the costs at each step of the sequence:

$$\text{dev} = \frac{1}{\text{seqLength}} \sum_{a=0}^{\text{seqLength}} C_{\text{SB}}^a$$

Where the Skill Benefit Cost for the a^{th} task $C_{\text{SB}}^a = 0$ if $UT_{\text{SB}}^a > MSB$. MSB is the Minimum Skill Benefit threshold. Otherwise, C_{SB}^a is a linear function of $(MSB - UT_{\text{SB}}^a)$, with saturation.

UT_{SB}^a is the benefit for the user of task a . It is computed as the weighted sum - over all the skills covered by the task - of the benefits expected for each skill given the user's level in that skill.

$$UT_{\text{SB}}^a = \sum_{i=0}^{\text{nSkills}} T_{\text{SR}_i}^a * T_{\text{SB}_i, U_{\text{SL}_i}}^a$$

The cost is defined as a linear function of the deviation, with saturation.

As an extension, the author may specify different thresholds for different skill types. Also, the cost can be split in various parts (each one grouping one or more skills), so that the author can give more weight to such skills.

11) Repetition

A frequent assignment of the same task - to the same player in different game sessions - is penalized. The total deviation of the sequence is the sum of the repetition costs at each step of the sequence:

$$\text{dev} = \frac{1}{\text{seqLength}} \sum_{a=0}^{\text{seqLength}} C_R^a$$

Where the Repetition Cost C_R^a for the generic a^{th} task (it is independent of its position in the sequence) is computed as a linear function of the number of times a task has already been performed by the user in her game history. The function evaluates to 0 for the minimum of the UT_n values ($\min_{\text{allTasks } T} \{UT_n\}$) and 1 for the $\max_{\text{allTasks } T} \{UT_n\}$. Thus, the C_R^a cost penalizes tasks that have already been performed more times by the current user

The cost is defined as a linear function of the deviation, with saturation.

IV. SAMPLE APPLICATION: TRAVEL IN EUROPE

We have implemented the overall SBSG model described in this paper in an SG for maritime safety for youngsters, SeaGame [31], and in the Travel in Europe (TiE) platform [32].

The core of the implementation consists in the runtime management module, namely, the Experience Engine (EE), that we built atop of the Torque commercial Game Engine³ [10]. The current EE version learns the delivery strategy specified by the author through a Genetic Computation algorithm that continuously matches the user and task profiles, using the cost items described in subsection III.C. The EE can operate also in absence of the specification of the delivery strategy cost function, continuously adapting the game flow

³ www.garagegames.com

without aiming at the achievement of target knowledge levels predefined by the author. This operation mode is implemented through Reinforcement Learning and exploits the user and task model defined in III.A and III.B [10].

The TiE platform is designed to promote the knowledge of the European cultural heritage to a wide audience using the paradigm of multiplayer SGs. This architecture aims at supporting the project's long-term idea of having a European map with an ever growing number of cities. We are currently developing significant city examples from several EU countries participating in the project (Genoa, Strasbourg, Prague, Cluj, Maribor, Plovdiv, Tomar, Arousa Norte, etc).

The first game realization based on the TiE platform is a cultural treasure hunt across the Europe. Fig. 4 shows a snapshot from the Genoa settings.



Figure 4. A snapshot from the TiE 3D reconstruction of Genoa

The target of the player is to visit a certain number of cities (each one representing an objective, in the terms introduced in fig. 1), and in each town the player has one or more missions to accomplish. The mission is time-limited and is characterized by a number of general questions, which the player should consider while exploring the city. In order to answer to these questions, the player's avatar explores the faithfully reconstructed urban environments in search of the places indicated in the mission. At each target place, the player looks for 3D icons, that are linked to important Point of Interests (PoIs), like palaces and churches. Each 3D icon triggers a task session through which the user can virtually manipulate pieces of the artistic heritage and face quizzes concerning the history, art and culture of that particular PoI. Each task is selected by the dynamic delivery CI module of the game engine by matching the user model and the task models according to the pedagogical requirements expressed by the game designer. During the visit, the player freely chooses where to go, so orientation in the city and finding the right path to the next PoI are major challenges - the city is a sort of challenging labyrinth for the player. Fig. 5 shows an example of a simple task about the façade of Palazzo Ducale in the Genoa city center. The task is played in front of the virtual reconstruction of the building, which invites players to find the wrong details in a picture by observing the right

virtual reconstruction. Accomplishment of a mission is decided at the end of the city exploration. There is a city-level final trial with a sort of millionaire game on the city, with quizzes that are related with the (visit-driving) mission's questions. Accomplishment of a mission rewards the player with a city-prize (e.g. a picture, a symbol), that can be conserved in the player's repository.



Figure 5. A snapshot of a tasks (an instance of the *FindTheWrongDetails* micro-game template) about Genoa's Palazzo Ducale, in the TiE treasure hunt.

The structure of TiE is as in fig. 1 and the high-level plot is a treasure hunt game, which is quite simple and schematic. But it represents a format that is scalable and flexible in terms of contents that can be cost-effectively developed and inserted also by third parties and also in a User Generated Content (UGC) perspective (e.g. with a software toolkit that supports an easy visual creation of instances from task templates).

Also, a simple, easy to understand structure – coupled with minigames created as instances of templates whose interaction modalities can be easily learned once by a player and applied in several different content cases - poses little overload to the player, who can effectively focus his attention on the contents. Of course, it is fundamental that the 3D settings, game mechanics and interactions modalities are similar to those of state of the art games, in order to be appealing to players.

V. CONCLUSIONS AND FUTURE WORKS

SGs represent a great potential for education since they can join an instructional value with the appealing language of digital natives [33]. Searching for methodologies to support effective authoring of SGs, we have abstracted a model consisting of 3D a geo-referenced VW where knowledge is implemented also through tasks that are disseminated across the environment.

The model is designed to support authors to focus on defining the tasks' content and annotation (e.g. relevance to various topics, difficulty, position in the VW, typology of supported learning styles, interrelation with other tasks and events, etc.), without being asked to do game scripting, which typically involves hard-coding events, actions, locations and contents of a specific game. Task annotation allows reusing tasks also in

different games. This is important in a User Generated Contents perspective, with involvement of user communities, which is particularly important for a long-term viability. Moreover, different game designers can exploit repositories of tasks and build educational games specifying a VW and the task delivery strategy criteria for the missions, that will be learnt at runtime by a CI engine. This model opens new perspectives to pedagogical experts and game designers to provide players with adaptive contents.

We have implemented the overall SBSG model described in this paper in a couple of SGs devoted to maritime education and promotion of the cultural heritage. We now need extensive user testing in order to assess and analyze the support of the proposed framework for effective learning in a real entertainment context. This will enable us to answer some key questions and upgrade the system accordingly. Open issues concern an extensive validation of the user and task models and of the delivery strategy cost function, and the speed of the runtime user adaptation process. However, it is important that the developed games are able to attract users, similarly to commercial videogames, even if they deal with educational topics [31]. This is an important target of educational games, that aim at extending learning opportunities, particularly in attracting a demographic traditionally averse to pursuing cultural activities.

Planned work includes also the provision of a standardized interface to support interoperability of the framework with Learning Objects [34], that are widely used in current Learning Management Systems (LMSs).

Research lines that we consider important for the future include development of ever more meaningful classes of configurable task templates and plot types that allow to insert task and missions in easily configurable, yet highly compelling and motivating adventure types.

REFERENCES

- [1] M. Zyda, "From Visual Simulation to Virtual Reality to Games", *IEEE Computer*, vol. 38, no. 9, pp. 25-31, 2005
- [2] F.L. Greitzer, O.A. Kuchar, and K. Huston, "Cognitive Science Implications for Enhancing Training Effectiveness in a Serious Gaming Context", *ACM J. Educational Resources in Computing*, vol. 7, no. 3, August 2007.
- [3] M. Zyda, A. Mayberry, C. Wardynsky, R. Shilling, and M. Davis, "The MOVES Institute's America's army operations game", in *Proc. 2003 ACM Symposium on Interactive 3D Graphics*, Monterey, CA, 2003.
- [4] H. Kelly, K. Howell, E. Glinert, L. Holding, C. Swain, A. Burrowbridge, and M. Roper, "How to build serious games," *Commun. ACM*, vol. 50, no. 7, pp. 44-49, Jul. 2007.
- [5] A. Sliney, D. Murphy, "JDoc: A Serious Game for Medical Learning," in *Proc. International Conference on Advances in Computer-Human Interaction*, pp. 131-136, 2008
- [6] *United Nations' Food Force Game*, available www.food-force.com
- [7] M. J. Mayo, "Games for science and engineering education", *Commun.ACM*, vol. 50, no. 7, pp. 30-35, Jul. 2007.
- [8] D. Michael, and S. Chen, *Serious Games: Games that education, train, and inform*, Thompson Course Technology PTR, 2006.
- [9] F. Bellotti, R. Berta, A. De Gloria, V. Zappi, *Exploring Gaming Mechanisms to Enhance Knowledge Acquisition in Virtual Worlds*, in Proc. 3rd International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA 2008), Athens, Greece, 10-12 September 2008.
- [10] F. Bellotti, R. Berta, A. De Gloria, L. Primavera, *Adaptive Experience Engine for Serious Games*, under review for IEEE Transactions on Computational Intelligence and AI in Games
- [11] V. J. Shute, J. Psozka, J. *Intelligent Tutoring Systems: Past, Present and Future*. In D. Jonassen (ed.) *Handbook of Research on Educational Communications and Technology*. Scholastic Publications, 2006
- [12] E. Wenger, *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. San Francisco: Morgan Kaufmann, 1987
- [13] P. Brusilovsky, *Adaptive Hypermedia*, User Modeling and User-Adapted Interaction, vol. 11, pp. 87-110, 2001.
- [14] S. W. McQuiggan, J.P. Rowe, S. Lee, J. C. Lester. *Story-Based Learning: The Impact of Narrative on Learning Experiences and Outcomes*. In *Intelligent Tutoring Systems*, Springer, 2008
- [15] W.L. Johnson, S. Marcella, Vilhjálmsón. *Serious Games for Language Learning: How Much Game, How Much AI?* Conference on Artificial Intelligence in Education Conference (AIED), 2005.
- [16] L. J. Cronbach, *Analysis of Covariance in Nonrandomized Experiments: Parameters Affecting Bias*, Available from Stanford Evaluation Consortium, 1977
- [17] A. J. Kolb, and D. A. Kolb, "Learning styles and learning spaces: Enhancing experiential learning in higher education", *Academy of Management Learning & Education*, vol. 4, no. 2, pp. 193-212, 2005.
- [18] F. Melo and I. Ribeiro, "Q-learning with linear function approximation," in *Proc. 20th Conf. Learning Theory*, 2007, pp. 308-322.
- [19] R. Bernhaupt, M. Eckschlager, and M. Tscheligi, "Methods for evaluating games: how to measure usability and user experience in games?", in *Proc of the international Conference on Advances in Computer Entertainment Technology*, Salzburg, Austria, June 13 - 15, 2007
- [20] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*, New York: Harper & Row, 1990.
- [21] N. Lennart and L. Craig, *Flow and immersion in first-person shooters: measuring the player's gameplay experience*, in Future Play '08: Proc.s of the 2008 Conference on Future Play, pp. 81--88, Toronto, Canada
- [22] N. Beume, H. Danielsiek, C. Eichhorn, B. Naujoks, M. Preuss, K. Stiller and S. Wessing, *Measuring flow as concept for detecting game fun in the Pac-Man game*, Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, June 1-6, 2008, Hong Kong, China
- [23] G. Tesauro, "Temporal difference learning and TD-Gammon", *Communications of the ACM*, vol. 38, no. 3, 1995.
- [24] J. Chen, "Flow in games (and everything else)", *Commun. ACM*, vol. 50, no. 4, Apr. 2007.
- [25] B. Cowley, D. Charles, M. Black, and R. Hickey, "Toward an understanding of flow in video games", *Comput. Entertain*, vol. 6, no. 2, Jul. 2008
- [26] Raney, A. *Expanding disposition theory: Reconsidering character liking, moral evaluations, and enjoyment*, *Communication Theory*. 14, 4 (2004), 348-369.
- [27] R. Nabi, M. Krcmar, *Conceptualising media enjoyment as attitude: implications for mass media effects research*. *Communication Theory*, vol. 4, no. 14, 2004.
- [28] M. Green, T. Brock, G. Kaufman, *Understanding media enjoyment: The role of transportation into narrative worlds*. *Communication Theory*, vol. 14, no. 4, 2004
- [29] Vygotsky, L. *Play and its role in the mental development of the child*. In J. Bruner, A. Jolly, & K. Sylva, (Eds.) *Play- Its Role in Development and Evolution*, pp. 537-554. New York: Basic Books. 1976.
- [30] Learning Object Metadata Working Group – <http://ltsc.ieee.org/wg12/>
- [31] Bellotti, F., Berta, R., De Gloria, A., Primavera, L. (2009). Enhancing the Educational Value of Videogames, *ACM Computers in Entertainment*, Vol. 7, No.2
- [32] F. Bellotti, R. Berta, A. De Gloria, L. Primavera, *Designing Virtual Worlds for Cultural Heritage*, ENTER 2009, Amsterdam, Netherlands, January 2009.
- [33] Bennett, S., Maton, K. & Kervin, L. The 'Digital Natives' Debate: A Critical Review of the Evidence, *British Journal of Educational Technology*, Vol. 39 No. 5, 2008.
- [34] Hodgins, H. W. The future of learning objects. In D. A. Wiley (Ed.), *The Instructional Use of Learning objects*, 2000.