Adaptivity Challenges in Games and Simulations: A Survey

Ricardo Lopes and Rafael Bidarra

***Abstract*—In computer games and simulations, content is often rather static and rigid. As a result, its prescripted nature can lead to predictable and impersonal gameplay, while alienating unconventional players. Adaptivity in games has therefore been recently proposed to overcome these shortcomings and make games more challenging and appealing. In this paper, we survey present research on game adaptivity, identifying, and discussing the main challenges, and pointing out some of the most promising directions ahead. We first survey the *purposes* of adaptivity, as the principles that could steer an adaptation and generation engine. From this perspective, we proceed to thoroughly discuss adaptivity's *targets* and *methods*. Current advances and successes in this emerging field point to many yet unexplored research opportunities. Among them, we discuss the use of gameplay expectations, learning preferences, and assessment data in the integrated adaptation of game worlds, scenarios, and quests. We conclude that, among other methods, procedural content generation and semantic modeling can powerfully combine to create offline customized content and online adjustments to game worlds, scenarios, and quests. These and other promising methods, deserving ample research efforts, can therefore, be expected to significantly contribute towards making games and simulations even more unpredictable, effective, and fun.**

***Index Terms*—Adaptive games, declarative modeling, online adaptivity, player assessment, player modeling, procedural content generation, semantic modeling.**

I. Introduction

TYPICALLY, when most commercial games are shipped, their gameplay has been prescripted. The same happens with simulations, which generally use game technology to emulate reality and training conditions. In both cases, game content, rules, narratives, and environments are created during development, mostly as *static* elements with which a *dynamic* player will interact. Designing such predefined content is standard because it allows games and simulations to remain robust, testable and controllable. As a result of such rigidity, game outcomes can be more easily anticipated by players, since all possible interactions are bounded by such static elements. Even worse, if players can predict certain outcomes, their progress can be often achieved by repeatedly exploiting a successful strategy.

Manuscript received October 05, 2010; revised February 28, 2011; accepted May 02, 2011. Date of publication May 12, 2011; date of current version June 15, 2011. This work was supported by the Portuguese Foundation for Science and Technology under Grant SFRH/BD/62463/2009.

R. Lopes and R. Bidarra are with the Computer Graphics Group, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: r.lopes@tudelft.nl; r.bidarra@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCIAIG.2011.2152841

In an attempt to account for player individuality, games often include minor variations that depend on players profiling themselves. For example, by customizing the difficulty level or choosing time constraints, players are classifying themselves as one of the available predefined low-resolution stereotypes, e.g., beginners or experts. However, this discrete approach implies that such games might fail in appealing to players who do not know how to profile themselves or who do not identify themselves with any of the available classifications.

Static game content and its predefined variations, based on low-resolution profiles, all lead to games and simulations that can be played in an impersonal, predictable and inflexible fashion and that can fail to appeal to broader audiences.

For games with purposes other than entertainment, such as serious games and simulations, these problems can become more acute. Players who need to capture or practice a certain skill, all have different learning abilities and training needs. However, serious games and simulations typically do not take such a high-resolution player individuality into account. Current *ad-hoc* and stereotyped training conditions can induce players to mostly perform the same exercises in the same conditions, adding little value to the learning process. This lack of player individuality can also affect the replay value of such games, since nothing new or different can be experienced in consecutive game sessions.

To solve the above shortcomings, many researchers agree that serious games and simulations have to become more challenging, unpredictable and player-centric, to be fully embraced as an effective way of knowledge transfer [1], [2]. Several other researchers claim that entertainment games should also address these issues, by catering the gaming experience to the individual user, being more responsive to different player types and their individual needs and adapting themselves to better fit the players [3]–[5].

Player-centered game adaptivity can help accomplishing the above goals. Dynamically adjusting game elements, according to the individual performance of the player (i.e., personal gameplay), can contribute to make the game experience more unique and personal. Consider the example of a driving simulation where a player is monitored as speeding more than desired. An adaptive game could adjust the city environment to discourage this behavior. Examples could be either increasing the number of speed bumps, traffic radars or police patrols or generating more curved roads, stoplights, or crosswalks, depending on the player's experience and personality.

In this paper, we survey the present state of adaptivity in games and simulations, identify the main challenges ahead and discuss possible research directions to tackle them. Fig. 1 lays

1943-068X/$26.00 © 2011 IEEE

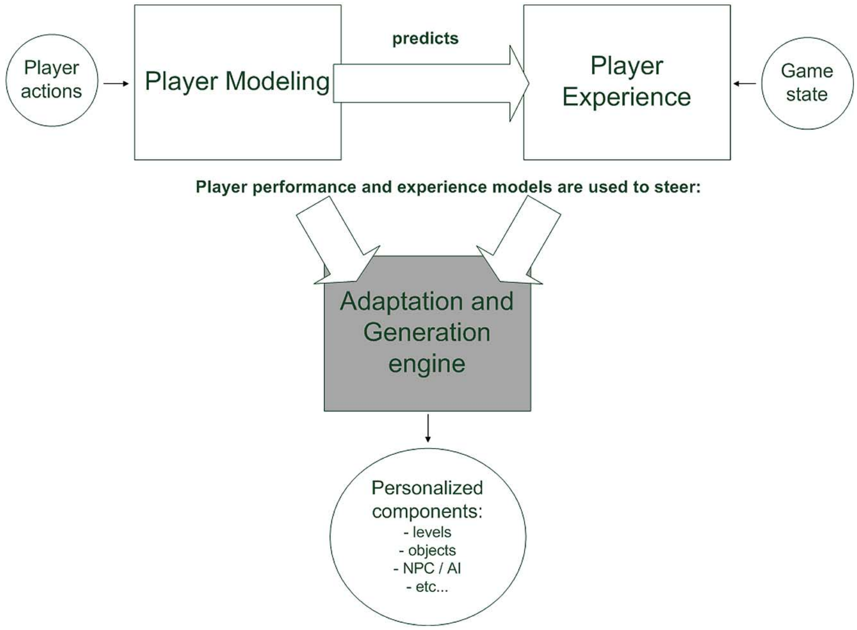

Fig. 1. Overview of game adaptivity architectural principles: player and experience modeling steer adaptation and generation of personalized game components.

out the architectural principles that drive research on adaptive games. These principles were already latent in the preliminary proposals of Houlette [6], Charles [3] and Magerko [5], as well as in the vast majority of the research that followed. In essence, game logs, recording the players' performance, are used to create models of players' actions, preferences, or personality. Given a game state, these models assess and predict the players desired experience for the next game state. Models for the player experience and performance are then used to steer an adaptation and generation engine, which adjusts the appropriate game components to better fit both.

This survey discusses game adaptivity research from an adaptation and generation perspective. We strongly focus on what (*targets*) and how (*methods*) adaptation and generation engines can or could adapt. In contrast, the *purposes* of adaptivity are surveyed from a generic perspective, independent of player modeling and player experience prediction. We discuss the principles that could, ultimately, serve as input and steer adaptation and generation. An in-depth analysis of their implementation through player modeling and experience prediction techniques will therefore not be considered here. Player modeling principles have been already discussed by several researchers [6]–[8], and experience prediction has been recently surveyed by Yannakakis and Togelius [9].

This article is structured as follows: in Section II we look at the *purposes* for adapting, by analyzing what is being presently done in steering adaptivity in games and simulations. In Section III we focus on adaptivity *targets*, surveying standard adaptive game components (e.g., nonplaying characters). In Sections IV and V, we survey and discuss, respectively, offline and online *methods* which can be used to adapt game content.

II. Steering Adaptivity

In games and simulations, adaptivity can be used to better suit the game to a dynamic element, for example, the skills of a player, the size of a team or the physical environment in which the game is played. As highlighted in Section I, this article focuses on player-centered adaptivity, i.e., adjustments which improve the individual player experience. For adaptivity to achieve this goal, it needs to be steered by some purpose that game developers can identify, measure and influence.

Knowledge on this steering purpose will determine how adaptation algorithms decide what, when and how to adjust. For taking this decision, algorithms should identify: 1) what triggers the need for adjustments; and 2) what should be adjusted. For example, if difficulty adjustment is the steering purpose, an adaptive game needs to recognize that consecutive failures may be a sign of high difficulty. It also needs to know concrete in-game ways of affecting the difficulty level. Understanding and choosing what to use to steer adaptivity is both an essential step and a major challenge, required to ensure that game adjustments induce the personalized player experience, on the desired way (e.g., adjusting difficulty in the previous example).

In this section, we survey adaptivity's purposes, i.e., the generic principles that support player modeling and experience prediction and steer game adaptation methods.

A. Entertainment Games

For entertainment games, fun is the fundamental purpose. There are many different theories for explaining how to achieve this largely subjective emotion but, so far, adaptive games have typically only been considering one dimension to engage fun: *challenge*. This usually means that the difficulty of performing game tasks must be in balance with the skills of the player, avoiding undesirable "too easy" or "too hard" situations. The methods, algorithms and games analyzed in Sections III and V are mainly steered by this purpose.

However, some promising work has already been done in different directions. Yannakakis and Hallam [10] propose a

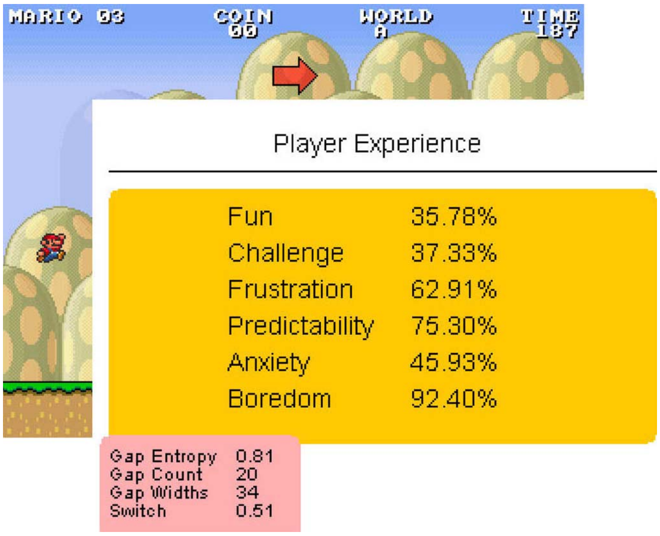

Fig. 2. Pedersen *et al.* [11]: predicted player emotions from a gameplay session.

methodology for adapting games on the Playware physical interactive platform. The authors explore control of user satisfaction rather than game difficulty, and their testbed is a "bug" (tile) stepping game for children. To model player satisfaction, the authors identify curiosity (the spatial diversity of bugs) and challenge (pace with which bugs appear and disappear) as the main factors. Furthermore, Pedersen *et al.* [11] built quantitative models that predict the player experience in a platform game, to be used in generating levels that are adjusted to those predictions. These models can predict gameplay as being: fun, challenging, boring, frustrating, predictable, or anxious. Fig. 2 illustrates an example where, after a gameplay session, the system predicts what emotions were experienced by the player.

These approaches show that there is room for going beyond *challenge* as a motivation for steering adaptivity. Magerko [5] argues that players have widely different reasons for playing and that adaptive games should capture and use them, focusing on the players main interests and adapting to match their motivations for playing. Both methods above show promising results in capturing, as Magerko proposed, various dimensions, beyond challenge, as useful indicators of players' motivations for playing. Curiosity, boredom, frustration, predictability, or anxiety are powerful features that extend beyond fun or challenge. They can allow for more detailed and flexible mechanisms of steering adaptation and generation.

Affective computing and advances in facial, motion and physiology detection can have an important role in steering adaptivity as well. When applied to games, these technologies have the potential to identify the affective states players experience. A better understanding of these can allow for more effectiveness and higher resolution in choosing and designing adaptivity purposes. For example, if a game can recognize and detect laughter, heart beats or body motion, fun can be further analyzed down to humor, excitement or physical interaction.

Recent research has been done in this direction, through the recognition of steering purposes as challenge [12], boredom, engagement and anxiety [13] and enjoyment [14] in adaptive games, using player physiology detection technology (e.g., electrocardiograms, galvanic skin response, electroencephalograms, palmar temperature sensors). A more in-depth discussion of the relation between affective computing, physiology detection technology, and adaptive games is out of scope here and can be found in [9].

B. Serious Games and Simulations

Serious games and simulations can have more purposes, other than fun. For example, they may aim at providing educational or training experiences, where players are required to achieve learning goals in supervised (and sometimes collaborative) environments. In this context, the motivation for steering adaptivity becomes clearer: improve the effectiveness of the knowledge transfer between the game and its players.

Traditionally, to steer adaptivity, research in serious games and simulations has been using a similar approach as in entertainment games: finding a balance between the player's skills and the game challenge level. Reaching this balance remains relevant for serious games and simulations, since it is a straightforward way of simplifying all types of learning goals and styles.

In serious games, the learning component strongly influences many design decisions. For example, the design philosophy of serious games needs to constantly balance *play* (or entertainment) with *meaning* (knowledge transfer) and a strong sense of *reality* [15]. Therefore, in serious games, adapting to specific skills is more important than to the global notion of difficulty or challenge. The learning goals to achieve are usually strongly coupled with the gradual personal improvement of a skill set, most of the time, one skill at a time. As so, in this domain, adaptive games have specialized (and usually *ad hoc*) approaches, where game components are adjusted to encourage training a specific skill. Adaptivity is steered by a specific skill players need to learn in a particular moment, and influenced by their personal proficiency.

Westra *et al.* [16], Peirce *et al.* [17], Magerko *et al.* [18] (all further analyzed in Section V) and Niehaus and Riedl [19] (discussed in Section III), all propose *personal skill proficiency* as the steering purpose for their adaptivity mechanisms. Another skill-oriented adaptive simulation was proposed by Johnson *et al.* [20], where individual language skills are modeled, determining how a virtual tutor offers guidance to the player. Lane *et al.* [21] also use a virtual tutor which, constrained by the player's past actions, gives feedback towards a set of skill-based training goals. Martin *et al.* [22] automatically generate scenarios for serious games using training objectives as the main requirements for generation. Although players are not modeled, these training objectives are also a list of specific tasks (or skills), appropriate for the domain of the game, e.g., hit a distant target using an artillery unit.

Some interesting research has been done beyond pure skill modeling and considering other aspects of the learning process. Research on the Crystal Island narrative-centred learning game demonstrates that supervised machine learning can be used to recognize players' affective states [23] or model their knowledge [24]. However, future work stills needs to be addressed to apply the recognized data to the adaptation of game content. On a different direction, Magerko *et al.* [25] identify learning styles (e.g., explorer, achiever) in users of an educational game; they then adapt the game to better fit players who have those learning styles, to better acquire the desired knowledge. This research shows that steering adaptivity in serious games and simulations can extend further beyond specific skill modeling, to focus on other important features of the player's learning mental process.

C. Assessment in Serious Games and Simulations

Apart from their purposes, serious games have another differentiating aspect: assessment. Measuring, discussing and reasoning on the gameplay effectiveness is specially important in the simulations domain, since it can lead to reflection and therefore improved learning. However, in this context, assessment has seldom been considered in academic research. In particular, there is no work on combining game adaptivity with assessment. Chen and Michael [26] have already identified the main challenges that assessment in serious games is facing, namely affecting and improving player experience. The authors suggest that log information and teachers/instructors knowledge should be fully explored and, in some way, incorporated back in the game, to guide its course.

So far, research in assessment for serious games has been mainly centered on after action review (AAR) methods. Still, some results already demonstrate that the direction identified by Chen contains a lot of potential. Lampton *et al.* [27] propose an AAR system for military simulations where trainees and trainers assess exercises together. An interesting result was that participants developed innovative ways to use AAR, not only for assessing past behavior, but also for planning new future training exercises. Raybourn [28] proposes a design method for creating training simulations that promote player communication, in-game performance feedback, and sharing of strategies. The author focuses on using in-game and AAR assessment information to create an emergent domain culture that could allow the cocreation of future game scenarios.

Some recent research is already incorporating performance logged data to control virtual participants in AAR sessions. Lane *et al.* [21] proposed a virtual reflective tutor that, given the history of player actions, is able to automatically assess their performance and even conduct an interactive deep reasoning AAR with the player. Core *et al.* [29] and Gomboc *et al.* [30] proposed explainable AI, a game log based system where AAR participants can directly question virtual characters about their in-game actions, goals and even motivations behind those.

All these results show that assessment information can be better explored, and that assessment game data could be used in future scenarios (as suggested by Chen). This logged data is even being already incorporated to steer postgame virtual entities (in this case, in AAR sessions).

D. Discussion

With respect to our initial definition of adaptivity's steering purposes, entertainment games and serious games/simulations still form two cases far apart. Both entail valid research challenges that are now discussed.

In entertainment games, some approaches are already being explored beyond the traditional dynamic difficulty adjustment mechanism. A major challenge still lies in exploring even further and materializing Magerko's [5] vision. To adapt better and more, there is a stronger need to capture and be guided by the real reasons why people play. These reasons can be captured by the characteristics and affective states of the gameplay that players expect to experience and be immersed in. For example, a player whose motivations for engaging in a first person shooter (FPS) game are to engage in a specific level of a stressful, scary, but low pace experience.

For steering adaptivity with this kind of high-resolution gameplay expectations, a number of challenges still need to be addressed, including: 1) capturing player expectations; 2) quantifying these to appropriate measurable levels; and 3) processing them to the correct game adjustments. Both 1) and 2) are player modeling and experience prediction challenges, while 3) is an adaptation and generation problem. Another major challenge in this direction lies in supporting these mechanisms in a game domain independent fashion, so they can be reused and consolidated.

Serious games and simulations are a different case. Due to their specific learning/training aim, many specialized approaches can adapt the game to provide opportunities to develop the most needed skills, at the appropriate proficiency level. However, some research shows that there is a need to better account for player individuality. Besides the case of learning styles based adaptivity [25], Rowe *et al.* [31] also evidence this. The authors investigate individual differences in gameplay and learning during student interactions with an educational game. They conclude that learning preferences (student's background knowledge and interests) are strongly coupled to the gameplay style (e.g., objects used, content read) and need to be considered in game design.

The challenge in steering adaptive serious games and simulations still remains in reaching further beyond skill modeling. Magerko *et al.* 's learning styles approach points in a promising direction where a broader array of learning preferences can guide adaptivity. Besides styles, this scope of learning preferences can include, among other, player's learning background, interests (as highlighted by Rowe *et al.*), gameplay [25] or even instructors/teachers assessment about players. The same challenges discussed before for gameplay expectations still apply for future work on modeling these learning preferences.

Assessment of past performances can also play a role in adaptive simulations. In this domain, there is typically plenty of valuable information emerging from game logs and AAR sessions. This information is currently far from being fully explored by the game itself. Game logs usually offer an enormous amount of unstructured game data that is therefore difficult to interpret and use. Moreover, while AAR information emerges from trainees and their instructors, it is typically not incorporated back in the game. Using this information as a source to steer adaptivity, as it currently steers virtual entities, seems a promising, unexplored area. Interesting research opportunities exist in using assessment information to, for example, regenerate "try again" game missions, adapted and focused on what the players failed on the previous session. So, offering an adapted regenerated "game session" could simultaneously allow a better understanding of what went wrong, and better opportunities to succeed.

TABLE I
CLASSIFICATION OF SURVEYED WORK ACCORDING TO ADAPTIVE COMPONENTS AND INDUSTRY/ACADEMIA DOMAINS

	Commercial games	**Academic research**
Game worlds	[37]	[46], [73], [74], [75], [76]
Mechanics	[33]	[10], [25], [76]
AI / NPC	[34], [35]	[16], [17], [69], [70], [71], [72], [76]
Narratives	[36], [37], [39]	[8], [41], [42], [43], [44]
Scenarios/quests		[18], [19], [61], [62], [63]

Fig. 3. Scene from *Max Payne*, by Remedy Entertainment [33].

III. ADAPTIVE GAME COMPONENTS

After discussing adaptivity's *purposes* in the previous section, we now turn our attention to its *target* recipients. Potentially, all components that are considered at game development can become adaptive. In fact, dynamically adjusting: 1) game worlds and its objects; 2) gameplay mechanics; 3) nonplaying characters (NPC) and AI; 4) game narratives; and 5) game scenarios and quests, all can contribute to offer an individualized gameplay experience. Table I illustrates how surveyed work is distributed according to game components and domain.

Gameplay mechanics, i.e., how game elements can work, including actions like running or shooting [32], have already been made adaptive, in commercial games. In *Max Payne* [33] (illustrated in Fig. 3), a mechanism unknown to players altered the level of mechanics like player aim assistance, according to individual skills (thus adjusting shooting difficulty).

Traditionally, adaptivity has been mostly researched and applied within the *AI domain*, specifically towards NPCs, since behavioral adaptation is a strong means of displaying intelligent behavior. In *Mario Kart Wii* [34], rubber band AI techniques are used to increase the opponent NPC abilities when the player performs too well. *Pro Evolution Soccer 08*[35] introduced Teamvision, an adaptive AI opponent system that changes its tactics and strategy to suit the player style and explore his weaknesses. In academia, and as identified in Table I, several techniques have been proposed to support adaptive AI that recognizes the player actions and responds by adjusting NPC behavior. Also, academic research on AI adaptation focuses strongly on the pedagogical serious games domain, due to the extensive use of NPCs in learning contexts (e.g., tutors). Several of the techniques surveyed in Section V are applied to AI adaptation, both in entertainment and serious games.

Adaptivity has also been applied to *game narratives*, both in the commercial and academic domains. Games can become more personal when the progressing narrative builds up in a unique fashion, fitting players' behavior. Valve's *Left 4 Dead* series [36], [37] introduced procedural narrative as a technique to generate sequences of events, adapted to the pace and behavior of the player. An AI Director analyzes players behavior (e.g., if they were particularly challenged by one kind of enemy) and adds subsequent events (e.g., spawning that enemy). According to Valve [38], this mechanism serves as a story-telling device (at least, in simple narrative domains as most FPS games are) because players can experience some notion of intentionality on the opponents' side. *Heavy Rain* [39] is an interactive drama game that focuses on personal gameplay, where all the specific decisions each player takes are analyzed, in a more complex way than before, to determine the narrative and outcomes of the game.

In academia, there is a strong interest in interactive narratives, story-based experiences which typically use game technology, both for entertainment or pedagogical purposes. Roberts and Isbell [40] have recently surveyed interactive narratives and drama management systems, identifying, among other aspects, their adaptive capabilities. The following paragraph presents an overview of these systems. For a more detailed discussion, e.g., on concerns on the use of centralized manager agents versus multiagent networks, Roberts and Isbell's survey is recommended.

Barber and Kudenko [41] researched the generation of dilemma-based interactive narratives. A model of player behavior under specific dilemmas is used to estimate and select difficult dilemmas, which a planner weaves together to form a story. Mott and Lester [42] use a dynamic decision network as a planner for creating interactive narratives. The decision network contains nodes for the players goals, experiences and relationships, thus influencing decision making. In Sharma *et al.* drama management system [43], player preferences are determined by an explicit case-based player model, derived from the behavior of earlier players. This model guides generation towards stories that fit those preferences. Fairclough [44] also uses a case-based approach, but to synthesize stories from a knowledge base, constrained by the player's evolving relationship with NPCs. Finally, Thue *et al.* [8] present an interactive narrative generation system which models the player's style according to five predetermined player types. Story events are annotated with their appeal for each player type and are selected accordingly for inclusion in the narrative.

Game scenarios and quests only recently started to become a target of adaptivity research. Game scenarios and quests both describe the flow of events and actions within a game but they are primarily used, respectively, in simulations and entertainment games. Generation of personalized game quests is already being researched and is discussed in detail in Section IV-B.

As for game scenarios, they highlight the importance of adaptivity in the simulations domain. Niehaus and Riedl have recently proposed a methodology for adapting game scenarios to suit players' learning goals [19]. A scenario adaptor adds, removes or replaces abstract game events, guided by a mapping between a world domain knowledge base (i.e., the dynamics of the simulation world events) and a lifelong learner model, which tracks a learner and chooses the next training objectives that will help him advance. Earlier research from Magerko *et al.* [18] also adapts game scenarios, and it will be discussed in detail in Section V.

As for results on adaptive *game worlds*, they are very scarce. The only example we found of game world adaptivity is in the commercial game *Left 4 Dead 2* [37]. According to the developers, the layout of certain sections of levels is dependent on the player's performance [45] (a graveyard with a simpler layout for underachieving players is presented as an example). Being a recent commercial release, the game's publishers have not yet disclosed any technical details and the reach of this adaptivity mechanism, in terms of accounting for how much of the content is static or dynamic. It is therefore still unknown which player modeling or procedural content generation techniques are used, if at all. As for academic research, several projects, mainly aimed at player modeling and difficulty adjustment techniques, are focusing on adapting 2-D game level structures, as discussed in Section V. These simple game levels are still far from being compared to the complexity and richness of modern game worlds. Although not fully adaptive, Charbitat [46] (further analyzed in Section IV) is the only example we are aware of where procedural generation of complex game worlds is somehow influenced by the player performance.

A. Discussion

Compared to all the above game components, adaptive *game worlds* and *scenarios* (or quests) are still lacking in broad, consolidated and integrated research focus. Modern game worlds have been made partially adaptive in the commercial domain. *Left 4 Dead 2* shows the potential in game world adaptivity but, in the absence of more detailed information regarding the extent of what is adapted, should be regarded, in academia, as an example to improve on. As for game scenarios, research has been more fruitful. Niehaus and Riedl [19], and Magerko *et al.* [18] are good examples of the advances achieved so far. However, they still evidence research opportunities to: 1) reach beyond skill-driven adaptivity (as discussed in Section II); and 2) integrate scenario with world adaptation/generation.

The importance of these two components, particularly if they are integrated, can be explained through their definitions. Game worlds are the virtual environments within which gameplay occurs, its geometry, geography, layout, and objects. Game scenarios are the framework for the global progression within a game level, its initial settings and the logical flow of events and actions that follow [47]. As such, the fulfillment of a game scenario within a game world defines and characterizes the majority of the player experience. Integrated world and scenario adaptivity seems therefore more likely to solve the shortcomings identified in Section I, certainly offering more possibilities for affecting player experience than the current proposals for adapting game mechanics, NPC or narratives. The example in Section I illustrates some of this unexplored potential, by influencing both the game world (generation of more speed bumps) and the game scenario (instantiation of more police patrols). Integrated approaches can go even further by having game world and scenario adaptation influence each other, in simple or more complex ways (e.g., if police patrols are sufficient, no speed bumps needed).

Currently, game worlds are created during the design stage, prior to game release. In that process, games and simulations occasionally use procedural generation algorithms to automatically create some of the game world elements, with techniques widely researched in academia (like the ones surveyed in Section IV). As for game scenarios, they are typically created during the programming stage, when scripts and code define the flow of events for the game. A major challenge in automatically authoring game worlds and scenarios, as in fact with all game content, lies in delaying its generation until the game is running. This challenge is essential for adaptivity, since the creation of content that is adjusted to players relies on analyzing their in-game performance. There are two main *methods* to tackle the challenge of supporting adaptive game worlds and scenarios, through delayed authoring: 1) offline (pregame) customized generation; and 2) online (i.e., in-game) adaptivity. In the next two sections, we will survey the present state of research on each of these topics, and how they confirm that adaptive game worlds and game scenarios raise very promising and challenging research questions.

IV. OFFLINE ADAPTIVITY: CUSTOMIZED CONTENT GENERATION

Offline adaptivity implies that adjustments are made considering player-dependent data, but prior to initiating any gameplay. The typical example of its application would be the processing of player data and game adjustments during the loading stage of a game session. Therefore, offline adaptivity involves mainly a generation challenge, i.e., an emerging creation process.

Automatic content generation can therefore play a significant role in offline adaptivity. Research results in this field are particularly promising towards *customized content generation*, a *method* for the automatic creation of virtual game worlds, adjusted to better suit players. We believe the same principles can be extended and applied to what occurs within these worlds, i.e., to game scenarios, even though their offline generation has been less researched than that of game worlds.

A. Game Worlds

Previous work in automatic content generation has traditionally relied on procedural methods and has often succeeded in creating visually convincing game environments. For the public eye, procedural generation has recently been successfully associated with games, due to *Spore* [48]. This game extensively uses procedural generation for player-created creatures animations and planet textures.

Regarding game worlds, many different procedures have been proposed to automatically create content such as terrain, trees, plants and urban environments. Procedural methods were

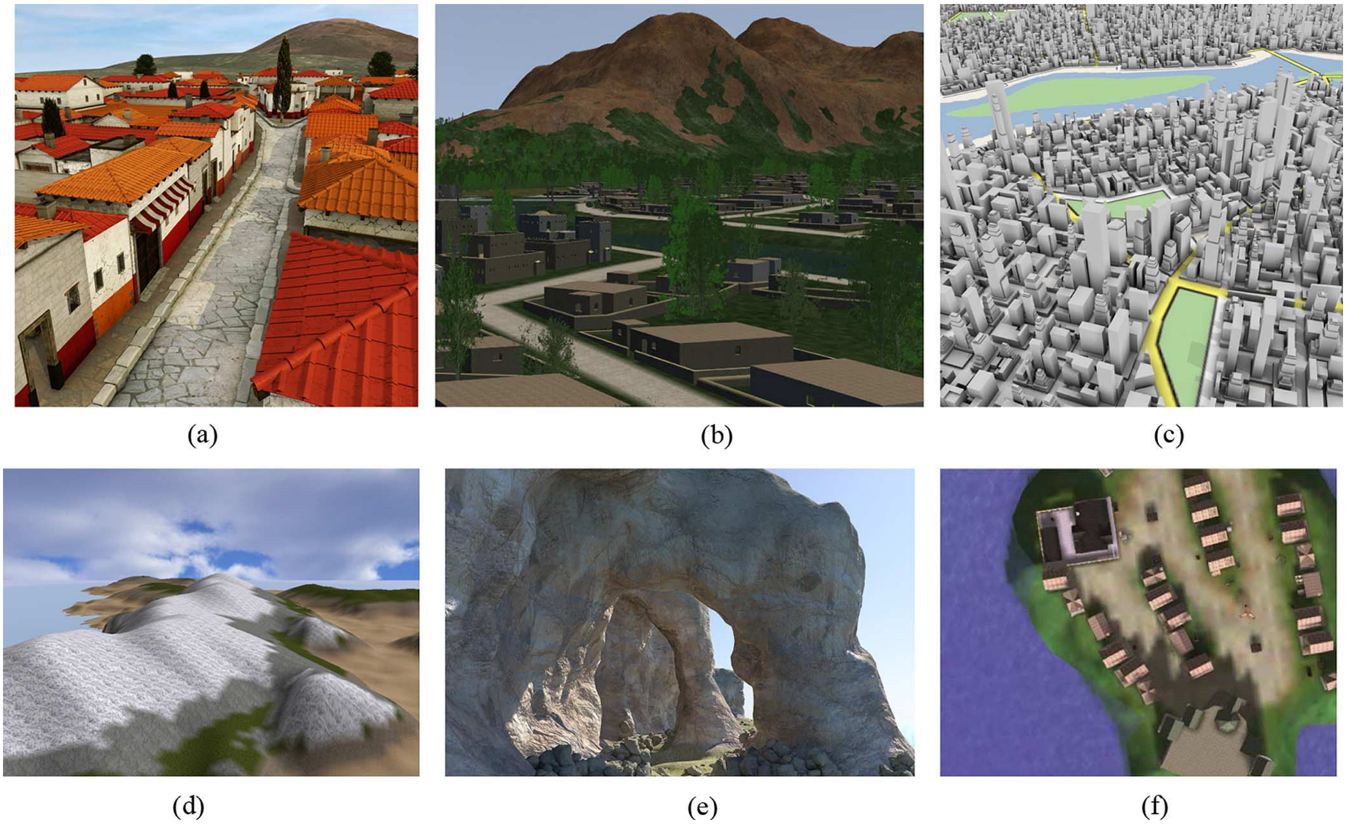

Fig. 4. (a) Urban environment generated by *CityEngine* [53]; (b) virtual world generated by *SketchaWorld* [68]; (c) road network and corresponding 3-D city geometry, generated by [56]; (d) height-map, generated by [54]; (e) complex terrain, with arches, created by [55]; (f) top view of town, generated by [60].

recently surveyed and discussed by Smelik *et al.* [49], who conclude that a common shortcoming in traditional methods is the lack of control over the generated output. Therefore, researchers are now aiming at more controllable procedural methods, seeking to allow designers to intuitively steer content generation.

In this direction, interesting work has been done in the generation of 2-D platform game levels. Compton and Mateas [50] use context-free grammars to generate platform levels, organized in patterns and branch structures. The generated level is controlled by a hill-climbing algorithm that adjusts patterns to suit a target controllable difficulty. Smith *et al.* [51] further developed these concepts, allowing designers to directly constrain properties in the generated platform levels (e.g., level path, jump rhythm and frequency, etc). Sorenson and Pasquier [52] propose another approach where genetic algorithms are used to evolve 2-D game levels towards satisfying designers constraints. An interesting result lies in how they evaluate generated levels: they are subjected to a fitness function that rewards levels based on how fun (in this case, challenging) they are. These results show that generation of 2-D level structures has been succeeding in considering important adaptivity concepts as difficulty, challenge or fun.

The generation of modern 3-D game worlds is facing other issues, more related with intuitive and interactive control. In this domain, Müller *et al.* [53] proposed the use of shape grammars to generate highly detailed cities. The grammar uses context sensitive rules to iteratively evolve building design, by creating more and more detail. Users can control the generation of a city using their *CityEngine* system, allowing them to create and edit grammar rules, in a similar way to using scripting languages. Fig. 4(a) shows a model for the ancient city Pompeii, as generated by *CityEngine*.

Recent research has been focused on creating new methods for designers to control game world generation, more intuitive than shape grammars. Doran and Parberry [54] propose an approach where terrain elevation heightmaps are generated by independent software agents, with different roles for coastlines, beaches, mountains, hills, and rivers. Designers are responsible for defining terrain features that constrain the amount of agents, their lifetime, and actions and, thus, the way the terrain is generated. Peytavie *et al.* [55] present a framework for generating complex terrains that include overhangs, arches, caves and different materials such as sand and rocks. Designers can control the terrain generation by sculpting bedrocks, modeling cracks, fractures, and tunnels, adding granular material, and controlling erosion tools. In Chen *et al.* [56], tensor fields are used to guide the generation of street networks. Users can control the generated street network by placing basis tensor fields, using tensor field patterns, smoothing fields to reduce its complexity, brushing the field to orient streets, or applying noise to make the road network less regular. Fig. 4(c)–4(e) shows, respectively, a road network created by Chen *et al.*, a height-map generated by Doran and Parberry, and a complex terrain modeled by Peytavie *et al.*

Even more interactive and user-centric methods have been proposed to control automatic content generation by: sketching the silhouette and bounds of a mountain in a 3-D interface [57], brushing and sculpting outdoor terrains [58] and sketching roads, which are automatically generated to fit with the surrounding environment [59].

Some recent research results have already shown that control over the generation process can extend beyond this type of interactive modeling of geometric world features. Bielikova *et al.* [60] propose a system for generating educational game content: quests, NPC, virtual worlds [see example in Fig. 4(f)] and narratives. In this case, domain experts, i.e., teachers, and not designers, control content generation. Teachers can select precreated game objects, add new learning content to them and create relationships between objects. Knowledge about objects and their relationships is the basis for solving and generating all the appropriate content. These results offer another valuable contribution: control on the generated content is applied at a higher level than geometric features, by using knowledge on objects and their relationships.

Nitsche *et al.* [46] introduce a case study for the procedural generation of game worlds based on the gaming style of its players. In *Charbitat*, players steer the generation of an infinite world through their in-game actions. The game world is split into individual tiles and each new tile is generated using noise functions and filters, where the underlying seed value is calculated based on player-dependent character data, i.e., his actions. Players are involved and conscious of this process: they can voluntarily influence the world generation in different directions as they please. Although this is an online method, this guided generation nature relates better with the methods and requirements for offline adaptivity.

Both approaches above show that automatic generation of game worlds can be controlled on a higher level (when compared to geometric features), and can be made dependent on player data. Both results seem successful advances towards customized content generation.

B. Game Scenarios and Quests

Offline automatic generation of game scenarios and quests has not been a subject of much research, specially when compared with online scenario adaptivity (Section V). The term game scenario, *i.e., the global progression within a game level, including its initial settings and the logical flow of events and actions that follow*, is mainly used in serious games and simulations. Its entertainment games equivalent, game missions or quests, also structures a sequence of events and actions, normally associated to a game task that must be completed.

Research on this field shows that there is a growing interest in creating player-centric quests that provide personalized gameplay. Sullivan *et al.*'s Grail framework [61] is aimed at providing customized quests, through online player-centered adjustments (analyzed in Section V), but it also includes an authoring tool for designers to control quest generation.

Although the following two methods are in essence also online based, their simple definition of quest avoids the usual design requirements of online methods (e.g., performance or consistency concerns). Therefore, they relate closely to possible offline techniques. Pita *et al.* [62] propose a system to dynamically generate quests in persistent massively multiplayer online role-playing games (MMORPG). Quest generation creates valid game goals, which are unique for each player and game. Quest uniqueness is ensured by three player-centric features that constrain the generation process to produce relevant quest paths: the memories (past quests) of the player, his relationship to the character assigning the quest, and player attributes (needed to complete quests). Ashmore and Nitsche [63] also investigate player-centric quest generation. They propose a new quest generator to include in the previously discussed *Charbitat* [46] system. Quests consist of key and lock puzzles (a key must be found to unlock an obstacle) and the generation process places within the game world, the obstacle, its key, and the challenges along the way to obtain it. Quest generation occurs during the generation of a new world tile: possible locations for keys and locks are scored by evaluators that are highly dependent on the procedurally generated tile. In the Charbitat case, quests become unique for each player because they are influenced by the game world which was itself generated in such a customized fashion.

These results evidence some of the potential in integrating and influencing game quest generation with its surrounding game worlds. As stated in Section III, integration with the game space is an important aspect to be considered in quest generation. In Pita's case, quests are generated in a game world that was manually designed, before-hand. In Ashmore's approach, the game world is first procedurally generated and is then evaluated for placement of quest elements. Though not adaptive in any way, Dormans work [64] is a good example of a different approach, a constructive integrated one. The generation of 2-D action–adventure game levels is broken down into two steps: a graph grammar generates mission structures that are used in an extended shape grammar, which grows a space that accommodates the generated game mission.

Offline generation of game scenarios, as defined earlier, is still far behind these concepts of customized quests or missions. Research in offline scenario generation is still more focused on the methods, i.e., on *how*, to generate and less on its purposes, i.e., on *what for*, e.g., steering them to be player-centric. As mentioned in Section II, Martin *et al.* [22] generate game scenarios for serious games. They use functional L-systems, a variant of formal grammars, to write generation rules which can expand training objectives into generated scenario elements, i.e., the initial settings and the progression of game events. Hullet and Mateas [65] also generate game scenarios from pedagogical goals, but using a planning system that decomposes pedagogical goals into tasks, subtasks, and methods, which encode knowledge to achieve that goal state.

Both approaches generate game scenarios from goals that capture which skills the players should apply during the game. However, in both methods, these declarable goals are simply a low-level and domain-dependent set of features that are implied by the higher level desired gameplay skills. For example, Hullet explicitly declares the goal "*a room should be blocked*" to implicitly capture the skill of breaching walls to rescue victims.

C. Discussion

In this section we surveyed research related to customized content generation, namely procedural generation of 2-D game levels, 3-D game worlds, quests, and game scenarios.

The surveyed methods show that the generation of 2-D levels is already capable of being controlled, or at least evaluated, by the same kind of criteria currently used to steer adaptivity: difficulty, challenge, and fun. These results highly encourage the further use of player data, i.e., their preferences or performance, for controlling the procedural generation of game levels. Even though level generation for the platform game genre is less complex than the generation of modern game worlds, the same conclusions could still hold for the latter.

However, research shows that this is still far away, since the generation of complex 3-D game worlds is facing other issues. The main challenge is to enable designers to control the generation process. Controllable content generation is enabling procedural methods to become more flexible and accurate. While maintaining its automatic nature, these methods are allowing game designers to steer automatic content generation by means of a better expression of their intent. Although these results are aimed at the design stage and through geometric features modeling, they seem promising steps towards customized content generation, as they allow procedural methods to be interacted with and controlled.

Moreover, Bielikova *et al.* and Nitsche *et al.* show that automatic generation of game worlds can be controlled from a higher level (when compared to geometric features), and can be made dependent on player data. The offline generation of game worlds, adapted to fit its players' purposes, could benefit from further research on these topics. The challenges ahead mainly relate to the integration and extension of these two topics by interpreting player data to reason about high level gameplay expectations (as discussed on Section II), and use this reasoning to steer content generation.

Regarding quest generation, results showed that customized quest generation is becoming more relevant, and it can be a successful mechanism to engage players in more enhanced, interactive and personal experiences. However, the methods surveyed are still somehow rudimentary, due to their *ad-hoc* nature. Quests are defined in an elementary manner, and generation is constrained to only one aspect of what a quest can include: goals to accomplish and locations for objects, in the cases analyzed. Open challenges remain for: 1) affecting all sorts of events and actions that can occur through the settings of a game world; and 2) seamlessly integrating quest and game world adjustments.

As stated in Section III, we think that fully integrating world and quest/scenario generation is a potentially important milestone. Multiple alternatives can be devised for this: creating worlds to better accommodate the scenario generation process, creating scenarios that fit the generated worlds or, in the best case, parallel or mixed approaches. In any case, the coherence and robustness of the gameplay would benefit from this integration, since generated events would take place in a space that is known and influenced by them, thus decreasing the risk for generation of related errors.

As for current research in offline scenario generation, and regardless of being based on declarable goals, these approaches show that the challenges ahead in researching customized scenario generation are the same as with game worlds: 1) considering higher level skills or goals (or the learning preferences discussed in Section II) in an explicit way; and 2) taking advantage of player-dependent data, as discussed with quest generation.

To adapt game worlds and scenarios, using controllable content generation, we identified the two challenges mentioned above. Overcoming these will make it possible to create customized content, in the sense that game worlds and scenarios will be generated before starting the game, according to knowledge about individual players. This knowledge should go beyond player preferences (e.g., favorite colors) and focus on what affects playing motivations or purposes. For example, consider a world domination strategy game and a player who aims to be an economic leader and use trade strategies. A matching virtual world and scenario could be generated so that, for example, the player homeland is located close to natural resources and trader routes in an intermediate position between not so aggressive opponent cities (all of which encourage the fast development of trade strategies).

Current research is already tackling some of the challenges identified above, and its methods could be valuable to future work in customized content creation. Semantic and declarative modeling techniques are already capable of controlling procedural methods by embedding and interpreting higher level knowledge in virtual objects. Tutenel *et al.* [66] define object semantics as all information, beyond its 3-D model, related to a particular object within the game world (e.g., functional information like how to interact with it, possible relationships with other objects, etc). With semantic modeling, object relationships, features and other semantic information can be used to guide the layout generation of a game world, whether designing it manually or creating it procedurally.

Bidarra *et al.* [67] introduce declarative modeling of virtual worlds, explaining how semantics can help designers to create virtual worlds by declaring *what* they want to create, instead of *how* to model it. Such declarative modeling enables designers to control and constrain virtual worlds, through semantic specifications that describe what the virtual world and its objects should be. Fig. 5 illustrates how this semantic level, presented to designers, is used to control the procedural level. This scheme differs from conventional procedurally based modeling, sporadically used by designers and technical artists, in that it incorporates a semantics layer, between the designer and the procedural techniques. This semantic level provides designers with a powerful front-end that generates and steers the underlying procedural level, while encapsulating the complexity of the latter.

Many of these methods have been integrated in *SketchaWorld* [68], a prototype system for declarative modeling of virtual worlds. In this declarative approach, designers state their intent by specifying which high-level features a virtual world should have, e.g., the layout of the landscape or the population size of a city. Designer's intent is used to generate a matching 3-D virtual world, where each specification is procedurally expanded to a visually convincing terrain feature. Within this declarative approach, interactions between terrain features are automatically solved using virtual world consistency maintenance, which consists of a combination of semantic definitions of the geometric and functional relationships between terrain features, and a set of generic resolution rules. A virtual world created with *SketchaWorld* is shown in Fig. 4(b).

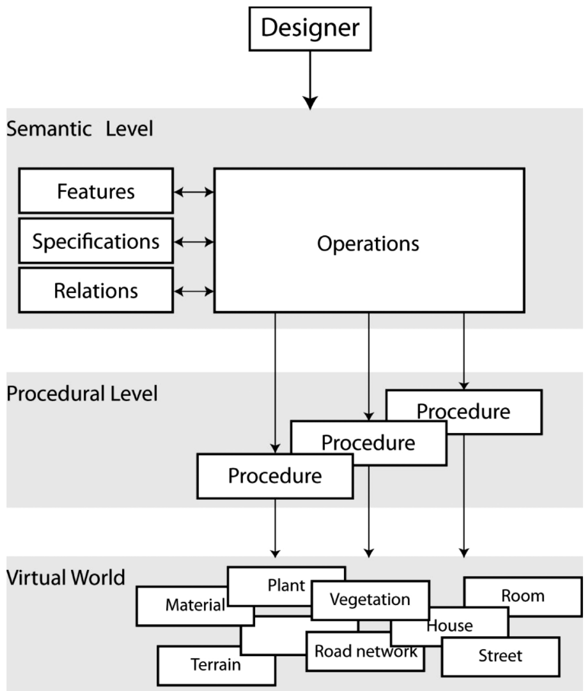

Fig. 5. Generic approach of declarative modeling of virtual worlds [67].

Semantic and declarative modeling can already help in tackling some of the challenges we identified throughout this section. The previously explained semantic layer deals with all high-level information relating to virtual world objects at the semantics level. This information helps convey the meaning and the role of an object in the virtual world, and consists of generic descriptions of classes of features, including attributes, properties, roles, relations, etc. This encourages the incorporation of further semantic information about player dependent gameplay purposes, and how these can be used to control object generation. For example, if a player needs obstacles in a race track, the semantic layer could indicate *what* and *how* obstacles can be used with that player. Finally, declarative modeling already includes semantics-based mechanisms to check and solve procedural conflicts in generated worlds. This shows that these techniques can be flexible enough to handle conflicting contexts, like those which would likely arise when integrating player-centric with designer-centric purposes, and virtual worlds with game scenarios.

The current state of research in semantic and declarative modeling, however, does not answer all of the issues discussed in this section and many challenges still remain open. Among them, supporting the generation of game scenarios in a similar fashion (enriching them with an analogous semantic scheme), integrating these with virtual world generation, and measuring player data into valid semantic knowledge, are some of the issues that need to be addressed to consider semantics as a possible relevant technique to customized content generation.

V. Online Adaptivity

As mentioned in Section III, offline customized generation is not the only *method* to support adaptivity through delayed authoring. Online, or in-game, adaptivity is the term many researchers use to describe the ability of a game to adjust to its players, in real time, as they play. Although this kind of adaptivity is still a recent research area, there are some significant results worth discussing here.

Most adaptivity research focuses on a low level, i.e., adapting specific game elements through nonintegrated approaches. However, Charles *et al.* [3] proposed a high-level framework to explain how online adaptivity should be supported in every domain, in an integrated manner. This framework captures the main abstract ideas and approaches that are currently adopted throughout this research area. A model of the player is used to capture the player habits and skills, and the player performance is monitored and compared with the model, while playing. Whenever an adaptation of the game is identified and performed, the framework measures its effectiveness, which can lead to either a new adaptation or an update of the player model.

Currently, online adaptivity mostly acts as a sandbox for researching new artificial intelligence concepts and methods. As such, most work in this field focuses on adjusting NPCs or other intelligent game agents to better suit players or even offer a more challenging game.

Peirce *et al.* [17] propose the ALIGN system as an approach for noninvasively adapting NPCs behavior to enable a personalized learning experience. ALIGN's architecture separates the logic of generic adaptation from game specificities, so that game logic and adaptation are independently authored and operated. Adaptive Elements (AEs) are the basic components that support possible adaptations. AEs are precreated and annotated with metadata describing both the game settings in which they can be used, and the abstract outcome of their use. Separate inference engines translate game events to AEs and create specific in-game interventions to match the selected AE. Each in-game intervention is influenced by a set of rules that examine a player model and determine the desired adaptation outcome.

Westra *et al.* [16] use agent organizations to adapt (in-game) the behavior of game elements in serious games. Uncentralized and independent (learning) agents choose the tasks to be performed by individual game elements, e.g., a burning fire or NPCs. Possible behavior variations for all agents are implemented *a priori*, using domain experts knowledge. During the game, each agent infers and proposes possible actions, based on its own in-game goals. The agent organization framework mediates this autonomy, by controlling which behavioral adaptation occurs for each agent. The agent organization framework coordinates individual adaptations into a combined one that adjusts the global behavior of game elements to fit the player skill level and a coherent storyline. For this coordination, Westra uses a player model that estimates the skill levels of each player. This model is continuously updated to accurately steer the agent organization framework.

Bakkes *et al.* [69] focus on adapting game AI, specifically an AI-controlled opponent in a particular real-time strategy game. In this case, online adaptivity takes place at the opponent AI, so that it can learn from its mistakes and act more effectively. The authors propose an approach where domain knowledge is gathered automatically by the game AI to form a case base (i.e., a compilation of solutions of similar past problems), which is exploited immediately to evoke effective behavior. The case base is compared with observations from previous games to allow improvement on past behaviors. Preliminary research [70] has been done to incorporate opponent modeling, in this case used for the AI to gain competitive advantage. Opponent models are established automatically, through clustering of strategic feature data in game observations. Past game observations are classified with such models, allowing a better matching with the case base.

On a similar direction, Hartley and Mehdi [71] also use a case-based approach that allows NPCs to learn, while playing, from the player actions, adapting the challenge level in the game. Game observations are gathered in cases that take the form of player state and action pairs. Matching these observations with previously registered cases, can be used to predict the next state-action pair and, therefore, enhance the NPC decision making. Results show that this approach succeeds in predicting player movement and actions in a FPS game. Therefore, despite its adaptivity focus on NPC (instead of player) goals, this case matching algorithm provides a method for adjusting the game according to player actions.

Dynamic scripting [72] is another technique proposed for adapting game AI, adopted for dynamic difficulty adjustment to the player skills. This learning technique is able to generate scripts (sets of behavioral rules), from rulebases associated with NPC classes, in order to control NPC behavior. Each rulebase comprises a set of manually designed rules and the probability that a rule is selected for a script is influenced by an attached weight value. Weights are updated according to their success rate in the game, which includes maintaining an even and challenging game for players.

Some promising work has already been done in directions other than AI and NPCs. Adaptive (simple) game worlds and levels are starting to be researched. Togelius *et al.* [73] propose an approach for generating, online, tracks for a racing game. Their goal is to augment player satisfaction, by creating a track that evolves with the player's characteristics. A player model is implemented by a neural network-based controller which infers and simulates the behavior of the real player. This player model is used to predict entertainment levels of specific players and decide how to evolve the track. Tracks are initialized as b-splines with 30 segments and they evolve through a mutation done by perturbing the positions of their control points. Although the focus is on 2-D racing tracks, this work shows promising results regarding the use of player modeling to generate adapted game content. Search-based procedural content generation is also used similarly for creating personalized *Super Mario Bros* platform levels [74], even though an exhaustive search approach is preferred over evolutionary algorithms.

Jennings-Teats *et al.* [75] also focus on dynamically constructing 2-D game environments that are adapted to players. In this case, the *Polymorph* algorithm generates 2-D platform levels, as you play, driven by the dynamic difficulty adjustment to the player skills. A statistical model of difficulty and a model of the player's current skill level are used, through mass data collection and machine learning techniques, to select the appropriate level segments to generate for each player.

Kazmi and Palmer [76] also direct their research to adapting game environments. They present a case study for a prototype of an adaptive FPS gaming environment. Player actions are recognized through a finite state machine approach, by which discrete actions reveal the skill level of players. Adaptation mechanisms try to make the game harder for players identified as experts and easier for beginners. In this research, finite state machines have also been used to implement all possible adaptations, i.e., adjustments on NPC behavior and movement, weapon mechanics, and game level geometry. Although this approach was mainly centered on adapting NPC behavior, the authors successfully explore other alternative ideas. They implemented a simple "modify geometry" mechanism that dynamically changes the game environment so that it becomes more difficult to navigate safely. They conclude that the "modify geometry" mechanism provides the most significant impact in terms of effectiveness.

In a different direction, some research has been done towards online adaptivity in quests and game scenarios. Magerko *et al.* [18] ISAT project uses an intelligent director agent for customizing simulation training scenarios to suit individual trainees. A skill model captures player proficiency levels in domain-specific skills, by monitoring and rating the trainee's actions. Scenarios are identified as sequences of plot points, i.e., actions, events and skills involved in them, which are selected, at run-time, for inclusion in the simulation. The director selects plot points by matching the list of tested skills with the current state of the skill model.

Sullivan *et al.* [61] also proposed a centralized approach, the Grail Game Manager, a run-time manager which dynamically generates quest structures using the player's history and current world state. This rule-based system is able to decompose quests (from a quest library) into separate entities (goals, actions, rewards, NPCs, dialog options) that can be dynamically recombined upon generation. This process filters possible quest entities through preconditions based on player history and current world state, thus creating a personalized experience.

A. Discussion

As surveyed above, current research results show that online adaptivity is mainly concerned with adjusting challenge levels of NPCs and game AI. In these approaches, online adaptivity is still characterized by a certain degree of predictability, since some of the analyzed methods require all possible variations to be created *a priori*. Performance and control over the game are the reasons why this balance, between static and dynamic techniques, is a recurring and important challenge in online adaptivity.

The current scope (AI and NPCs), purpose (challenge level balance) and techniques (combination of predefined content) in these approaches show that online adaptivity is in its first steps. Integrated approaches, embracing online adaptive game worlds and scenarios, are still far away. Confirmation of this is that player modeling and monitoring is still considered on an individual case, without sound and common theoretical foundations.

However, recent work has broadened the focus to adapting game environments and other game elements. Although, for example, Kazmi's "modify geometry" mechanism was simple and applied to only one type of situation, one can easily foresee the potential of adapting more than just the behavior of intelligent agents. Procedural content generation (as surveyed in Section IV) is becoming more powerful and can have a role to play in online adaptivity as well. An example of such potential is evidenced by Kenneth Stanley's Galactic Arms Race [77], a multiplayer game where players control a spaceship and collect weapons throughout a game world. Weapons are procedurally generated, at run-time, based on which weapons have been selected and used before by players.

Results both in 2-D game worlds and quest/scenarios confirm this observation: procedural content generation is becoming more and more online efficient and player-centric. This indicates a promising research direction on extending these procedural methods and online adaptivity to the integration of modern game worlds with the events and actions that comprise game scenarios.

In the previous section, we already discussed the benefits of embedding semantics, atop geometry, in a virtual world. A promising direction could be to integrate these methods in online adaptivity, and use such knowledge to adjust properties of events or objects to match the player performance. Such work could build up on Kessing *et al.* [78] who introduce the notion of services in game worlds. Services are defined as the capacity of an object to perform an action, possibly subject to some requirement. For example, an apple (or any food for that matter) "provides the service" of satisfying someone's hunger. These services are added to a semantic layer on an object level, which means that virtual objects *know* how they can affect other entities or how others can interact with them. Semantic services are specified as generic characteristics of classes of objects and, therefore, object instances can acquire their own independent behavior, instead of a rigidly predefined one. Including player data as a possible service requirement could act as a starting point for a new form of online adaptive worlds and scenarios. In this way, different services could act differently according to the player, therefore steering available events in the game world.

Finally, modifying landscapes and topography of virtual worlds has already been suggested as a valid direction for online adaptivity [3]. However, future methods for dynamically changing game environments, on a world scale, must tackle important challenges to be successful. Among them, maintaining coherence (e.g., mountains cannot magically change shape) and ensuring performance and scalability are important aspects, so that this type of adaptivity does not undermine the player experience.

VI. Conclusion

In this paper, we surveyed the present state of adaptivity in games and simulations. We focused on the *purposes*, *targets*, and *methods* that have been proposed so far to support adaptive game technology, from both academia and industry.

Our first conclusion is that adaptivity is already establishing itself as a rapidly maturing field regarding its *purposes*. Current advances, both in industry and academia, indicate good results in not only adapting towards an optimal challenge level (both in industry and academia), but also towards other affective states like fun, frustration, predictability, anxiety or boredom. With simulations, research is already successful in adapting to fit specific skill levels and incorporate learning styles. In this domain, assessment methods are becoming better at creating and using personal performance data.

Current advances also show that work is already being laid out to approach some still open issues. For both entertainment games and simulations, the main goal lies in materializing Magerko's vision of adapting to match the player's real motivations for playing [5]. To accomplish this, the challenge is to go further beyond current results, by capturing and incorporating higher resolution player data, including gameplay expectations, broad learning preferences and assessment data. There are many open and intriguing research questions in recognizing, capturing, and defining these steering purposes in a detailed and measurable way.

Concerning the *targets* of adaptivity, we have concluded that a large community both in industry and academia has already been focusing on game mechanics, AI, NPC, and narratives. Fewer research groups are already focusing, with success, on adaptive game scenarios or quests. On the other side, concerning adaptivity in modern and complex game worlds, many research questions are still unanswered.

We think that there is a lot of potential not only in adaptive game worlds, but particularly in their integration with adaptive scenarios/quests. The fulfillment of a game scenario within a virtual world is an essential aspect of gameplay, determining the sequence of events within an immersive environment, in which the player participates. Acting upon these, in an integrated manner, can create plenty of (yet unexplored) possibilities for improving gameplay.

Regarding the *methods* which can support adaptivity, some important advances have already been achieved with offline and online techniques. One of the most promising *methods* is the procedural generation of off-line (i.e., pregame) content that is customized to fit each player. Procedural content generation is becoming more controllable, although mainly through control over geometric features of that content. Some preliminary work has been done to: 1) incorporate player data; 2) control generation using high level information (e.g., object metadata, semantics); and 3) integrate game level and event generation. Further advances are being achieved in generating personalized basic quests, for entertainment games, and generating scenarios from declarable learning goals, for serious games. For offline adaptivity, customized content generation seems the most promising method in terms of future research. In this field, the main issues to address relate to the interpretation of player data to, for example, generate insight on gameplay expectations and learning preferences. Such "intelligence" could then control what and how to generate next, particularly in integrated game worlds and scenarios. This high level knowledge fits well with the concepts used in present semantic modeling research. We therefore conclude that virtual world and object semantics can be instrumental to customized content generation.

Online (i.e., in-game) adaptivity is also an essential *method* to consider. Current research is succeeding in using player

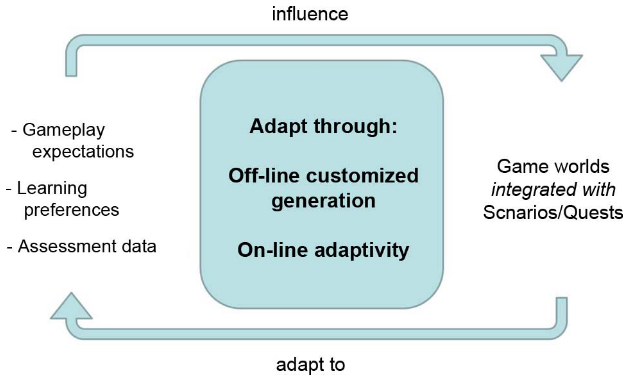

Fig. 6. Open challenges for adaptivity in games and simulations.

models to control the adaptation of NPC run-time behavior. Moreover, promising work is being done on dynamically constructing game scenarios, and even of game environments, that are adapted to the player in-game performance. For adaptivity's purposes, online adjustment of game worlds and scenarios is likely to achieve better results than with NPCs. However, new methods will need a paradigm shift, already encouraged by the recent advances in procedural content generation: from searching and selecting among predefined solutions to generating dynamic emerging ones.

Fig. 6 summarizes the challenges we discussed in this article and how they relate to each other. Gameplay expectations, learning preferences or assessment data can be used to guide both offline customized generation and online adaptivity of integrated game worlds and scenarios.

These challenges open up a variety of promising research directions. Pursuing them will likely lead to the development of new methods and techniques, which in turn will improve present adaptive game technology. As a result, games and simulations can become more flexible, agile, and complete in the way they adapt to the player. Ultimately, a better adaptivity will foster the potential to make games and simulations even more unpredictable, effective and fun.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] C. Aldrich, "A field guide to educational simulations," *Learn. Circuits*, 2002.

[2] S. Blackman, "Serious games … and less!," *ACM SIGGRAPH Comput. Graphics*, vol. 39, no. 1, pp. 12–16, 2005.

[3] D. Charles, A. Kerr, M. McNeill, M. McAlister, M. Black, J. Kücklich, A. Moore, and K. Stringer, "Player-centred game design: Player modelling and adaptive digital games," in *Proc. DiGRA Conf. Changing Views—Worlds in Play*, Vancouver, BC, Canada, 2005, pp. 285–298.

[4] K. M. Gilleade and A. Dix, "Using frustration in the design of adaptive videogames," in *Proc. ACM SIGCHI Int. Conf. Adv. Comput. Entertainment Technol.*, Singapore, 2004, pp. 228–232.

[5] B. Magerko, "Adaptation in digital games," *IEEE Comput. Mag.*, vol. 41, no. 6, pp. 87–89, Jun. 2008.

[6] R. Houlette, "Player modeling for adaptive games," in *AI Game Programming Wisdom II*. Boston, MA: Charles River Media, 2004, pp. 557–566.

[7] C. Beal, J. Beck, D. Westbrook, and P. Cohen, "Intelligent modeling of the user in interactive entertainment," in *Proc. AAAI Spring Symp. Artif. Intell. Interact. Entertain.*, Stanford, CA, 2002, pp. 8–12.

[8] D. Thue, V. Bulitko, M. Spetch, and E. Wasylishen, "Interactive storytelling: A player modelling approach," in *Proc. 3rd Artif. Intell. Interact. Dig. Entertain. Conf.*, Stanford, CA, Jun. 2007, pp. 43–48.

[9] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Trans. Affect. Comput.*, vol. 99, 2011.

[10] G. Yannakakis and J. Hallam, "Real-time game adaptation for optimizing player satisfaction," *IEEE Trans. Comput. Intell. AI Game.*, vol. 1, no. 2, pp. 121–133, Jun. 2009.

[11] C. Pedersen, J. Togelius, and G. N. Yannakakis, "Modeling player experience for content creation," *IEEE Trans. Comput. Intell. AI Game.*, vol. 2, no. 1, pp. 54–67, Mar. 2010.

[12] P. Rani, N. Sarkar, and C. Liu, "Maintaining optimal challenge in computer games through real-time physiological feedback," in *Proc. 11th Int. Conf. Human Comput. Interact.*, Las Vegas, NV, 2005, pp. 184–192.

[13] G. Chanel, C. Rebetez, M. Bétrancourt, and T. Pun, "Boredom, engagement and anxiety as indicators for adaptation to difficulty in games," in *Proc. 12th Int. Conf. Entertain. Media Ubiquitous Era*, Tampere, Finland, 2008, pp. 13–17.

[14] S. Tognetti, M. Garbarino, A. Bonarini, and M. Matteucci, "Modeling enjoyment preference from physiological responses in a car racing game," in *Proc. IEEE Conf. Comput. Intell. Games*, Copenhagen, Denmark, Aug. 2010, pp. 321–328.

[15] C. Harteveld, R. Guimarães, I. Mayer, and R. Bidarra, "Balancing play, meaning and reality: The design philosophy of LEVEE PATROLLER," *Simulation and Gaming*, vol. 3, no. 43, pp. 316–340, 2010.

[16] J. Westra, F. Dignum, and V. Dignum, "Keeping the trainee on track," in *IEEE Conf. Comput. Intell. Game.*, Copenhagen, Denmark, Aug. 2010, pp. 450–457.

[17] N. Peirce, O. Conlan, and V. Wade, "Adaptive educational games: Providing non-invasive personalised learning experiences," in *Proc. 2nd IEEE Int. Conf. Digital Game. Intell. Toys Based Edu.* , Banff, AB, Canada, Nov. 2008, pp. 28–35.

[18] B. Magerko, B. Stensrud, and L. Holt, "Bringing the schoolhouse inside the box—a tool for engaging, individualized training," in *Proc. 25th Army Sci. Conf.*, Orlando, FL, Nov. 2006.

[19] J. Niehaus and M. O. Riedl, "Scenario adaptation: An approach to customizing computer-based training games and simulations," in *Proc. AIED Workshop Intell. Edu. Game.*, Brighton, U.K., 2009, pp. 89–98.

[20] W. L. Johnson, C. Beal, A. Fowles-Winkler, U. Lauper, S. Marsella, S. S. Narayanan, D. Papachristou, and H. Vilhjalmsson, "Tactical language training system: An interim report," in *Proc. Conf. Intell. Tutoring Syst.*, Berlin, Germany, Jun. 2004, pp. 336–345.

[21] H. C. Lane, M. Core, D. Gomboc, A. Karnavat, and M. Rosenberg, "Intelligent tutoring for interpersonal and intercultural skills," in *Proc. Interservice/Industry Train., Simul., Edu. Conf.*, 2007.

[22] G. A. Martin, C. E. Hughes, S. Schatz, and D. Nicholson, "The use of functional L-systems for scenario generation in serious games," in *Proc. ACM Workshop Procedural Content Generation Games*, Monterey, CA, 2010, pp. 6:1–6:5.

[23] J. Rowe, B. Mott, S. McQuiggan, J. Robison, S. Lee, and J. Lester, "Crystal island: A narrative-centered learning environment for eighth grade microbiology," in *Proc. Workshop Intell. Edu. Game.*, Brighton, U.K., 2009, pp. 11–20.

[24] J. Rowe and J. Lester, "Modeling user knowledge with dynamic Bayesian networks in interactive narrative environments," in *Proc. 6th Annu. AI Interact. Digital Entertain. Conf.*, Stanford, CA, 2010, pp. 57–62.

[25] B. Magerko, C. Heeter, J. Fitzgerald, and B. Medler, "Intelligent adaptation of digital game-based learning," in *Proc. ACM Conf. Future Play*, Toronto, ON, Canada, 2008, pp. 200–203.

[26] S. Chen and D. Michael, "Proof of learning: Assessment in serious games," 2005 [Online]. Available: http://www.gamasutra.com/features/20051019/chen_01.shtml

[27] D. R. Lampton, J. Bliss, and G. A. Martin, "Performance measurement and training feedback in a military collaborative virtual environment," in *Proc. 1st Int. Conf. Virtual Reality*, Las Vegas, NV, Jul. 2005.

[28] E. M. Raybourn, "Applying simulation experience design methods to creating serious game-based adaptive training systems," *Interact. Comput.*, vol. 19, no. 2, pp. 206–214, 2007.

[29] M. G. Core, H. C. Lane, M. van Lent, D. Gomboc, S. Solomon, and M. Rosenberg, "Building explainable artificial intelligence systems," in *Proc. 18th Conf. Innovative Appli. Artif. Intell.*, Boston, MA, 2006, vol. 2, pp. 1766–1773.

[30] D. Gomboc, S. Solomon, M. G. Core, H. C. Lane, and M. van Lent, "Design recommendations to support automated explanation and tutoring," in *Proc. 14th Conf. Behav. Representation Modeling Simul.*, Universal City, CA, 2005.

[31] J. P. Rowe, L. R. Shores, B. W. Mott, and J. C. Lester, "Individual differences in gameplay and learning: a narrative-centered learning perspective," in *Proc. 5th Inter. Conf. Found. Digital Game.*, Monterey, CA, 2010, pp. 171–178.

[32] B. Brathwaite and I. Schreiber, *Challenges for Game Designers*. Boston, MA: Charles River Media, 2009.

[33] Remedy Entertainment, Max Payne, Gathering of Developers, 2001 [Online]. Available: http://www.rockstargames.com/maxpayne/splash.html

[34] Nintendo EAD, Mario Kart Wii, 2008 [Online]. Available: http://www.mariokart.com/wii/launch/

[35] Konami, Pro Evolution Soccer 2008, 2007 [Online]. Available: http://www.games.konami-europe.com/game.do?idGame=149

[36] Valve Corporation, Left 4 Dead, 2008 [Online]. Available: http://www.l4d.com/game.html

[37] Valve Corporation, Left 4 Dead 2 2009 [Online]. Available: http://www.l4d.com/game.html

[38] G. Newell, Gabe Newell Writes for Edge 2008 [Online]. Available: http://www.next-gen.biz/blogs/gabe-newell-writes-edge

[39] Quantic Dream, Heavy Rain, Sony Computer Entertainment, 2010 [Online]. Available: http://www.heavyrainps3.com/

[40] D. L. Roberts and C. L. Isbell, "A survey and qualitative analysis of recent advances in drama management," *Int. Trans. Syst. Sci. Appl.*, vol. 4, no. 2, pp. 61–75, 2008.

[41] H. Barber and D. Kudenko, "Dynamic generation of dilemma-based interactive narratives," in *Proc. 3rd Artif. Intell. Interact. Digital Entertain. Conf.*, Stanford, CA, 2007, pp. 2–7.

[42] B. W. Mott and J. C. Lester, "U-director: A decision-theoretic narrative planning architecture for storytelling environments," in *Proc. 5th Int. Joint Conf. Autonom. Agents Multiagent Syst.*, Hakodate, Japan, 2006, pp. 977–984.

[43] M. Sharma, S. Ontan, C. Strong, M. Mehta, and A. Ram, "Towards player preference modeling for drama management in interactive stories," in *Proc. 20th Int. FLAIRS Conf.*, Key West, FL, 2007, pp. 571–576.

[44] C. Fairclough, "Story games and the OPIATE system," Ph.D. dissertation, Dept. Comput. Sci., Univ. Dublin–Trinity College, Dublin, Ireland, 2006.

[45] Walker, J. for Rock, Paper, Shotgun Webzine, Left 4 Dead 2: Exclusive RPS Hands-On Preview, 2009 [Online]. Available: http://www.rockpapershotgun.com/2009/06/01/left-4-dead-2-exclusive-rps-preview

[46] M. Nitsche, C. Ashmore, W. Hankinson, R. Fitzpatrick, J. Kelly, and K. Margenau, "Designing procedural game spaces: A case study," in *Proc. FuturePlay*, London, ON, Canada, Oct. 2006.

[47] C. van Est and R. Bidarra, "High-level scenario editing for simulation games," in *Proc. 6th Int. Conf. Comput. Graphics Theory Appl.*, Algarve, Portugal, 2011.

[48] Maxis, Spore, Electronic Arts, 2008 [Online]. Available: http://www.spore.com

[49] R. Smelik, K. J. de Kraker, T. Tutenel, R. BidarraS, and G. Groenewegen, "A survey of procedural methods for terrain modelling," in *Proc. CASA Workshop 3D Adv. Media Gaming Simul.*, Amsterdam, The Netherlands, 2009, pp. 25–34.

[50] K. Compton and M. Mateas, "Procedural level design for platform games," in *Proc. 2nd Artif. Intell. Interact. Digital Entertain. Conf.*, Marina del Rey, CA, 2006, pp. 109–111.

[51] G. Smith, M. Treanor, J. Whitehead, and M. Mateas, "Rhythm-based level generation for 2D platformers," in *Proc. 4th Int. Conf. Found. Dig. Game.*, Orlando, FL, Apr. 2009, pp. 175–182.

[52] N. Sorenson and P. Pasquier, "Towards a generic framework for automated video game level creation," in *Applications of Evolutionary Computation, EvoApplicatons 2010, Proceedings*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, 2010, vol. 6024, pp. 131–140.

[53] P. Müller, P. Wonka, S. Haegler, and A. U. Van Gool, "Procedural modeling of buildings," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 614–623, 2006.

[54] J. Doran and I. Parberry, "Controlled procedural terrain generation using software agents," *IEEE Trans. Comput. Intell. AI Game.*, vol. 2, no. 2, pp. 111–119, Jun. 2010.

[55] A. Peytavie, E. Galin, S. Merillou, and J. Grosjean, "Arches: A Framework for modeling complex terrains," *Comput. Graphics Forum (Proc. Eurographics)*, vol. 28, no. 2, pp. 457–467, 2009.

[56] G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang, "Interactive procedural street modeling," *ACM Trans. Graphics*, vol. 27, no. 3, pp. 1–10, 2008.

[57] J. Gain, P. Marais, and W. Strasser, "Terrain sketching," in *Proc. Symp. Interact. 3D Graphics Game.*, Boston, MA, 2009, pp. 31–38.

[58] G. J. P. de Carpentier and R. Bidarra, "Interactive GPU-based procedural heightfield brushes," in *Proc. 4th Int. Conf. Found. Digital Game.*, Orlando, FL, 2009, pp. 55–62.

[59] J. McCrae and K. Singh, "Sketch-based path design," in *Proc. Graphics Interface Canadian Inform. Process. Soc.*, Toronto, ON, Canada, 2009, pp. 95–102.

[60] M. Bielikova, M. Diveky, P. Jurnecka, R. Kajan, and L. Omelina, "Automatic generation of adaptive, educational and multimedia computer games," *Signal Image Video Process.*, vol. 2, no. 4, pp. 371–384, Dec. 2008.

[61] A. Sullivan, M. Mateas, and N. Wardrip-Fruin, "Rules of engagement: Moving beyond combat-based quests," in *Proc. Intell. Narrative Technol. III Workshop*, Monterey, CA, 2010, pp. 11:1–11:8.

[62] J. Pita, B. Magerko, and S. Brodie, "True story: Dynamically generated, contextually linked quests in persistent systems," in *Proc. Conf. Future Play*, Toronto, ON, Canada, 2007, pp. 145–151.

[63] C. Ashmore and M. Nitsche, "The quest in a generated world," in *Proc. Digital Games Res. Assoc. (DiGRA) Conf.: Situated Play*, Tokyo, Japan, Sep. 2007, pp. 503–509.

[64] J. Dormans, "Adventures in level design: generating missions and spaces for action adventure games," in *Proc. Workshop Procedural Content Generation Game.*, Monterey, CA, 2010, pp. 1:1–1:8.

[65] K. Hullett and M. Mateas, "Scenario generation for emergency rescue training games," in *Proc. 4th Int. Conf. Found. Digital Games*, Orlando, FL, 2009, pp. 99–106.

[66] T. Tutenel, R. Bidarra, R. M. Smelik, and K. J. de Kraker, "The role of semantics in games and simulations," *Comput. Entertain.*, vol. 6, no. 4, pp. 1–35, 2008.

[67] R. Bidarra, K. J. de Kraker, R. Smelik, and T. Tutenel, "Integrating semantics and procedural generation: Key enabling factors for declarative modeling of virtual worlds," in *Proc. FOCUS K3D Conf. Semantic 3D Media Cont.*, Sophia Antipolis, France, Feb. 2010, pp. 51–55.

[68] R. Smelik, T. Tutenel, K. de Kraker, and R. Bidarra, "A declarative approach to procedural modeling of virtual worlds," *Compu. Graphics*, vol. 35, no. 2, pp. 352–363, 2011.

[69] S. Bakkes, P. Spronck, and J. van den Herik, "Rapid and reliable adaptation of video game AI," *IEEE Trans. Comput. Intell. and AI in Games*, vol. 1, no. 2, pp. 93–104, 2009.

[70] S. Bakkes, P. Spronck, and J. van den Herik, "Opponent modelling for case-based adaptive game AI," *Entertain. Comput.*, vol. 1, no. 1, pp. 27–37, 2009.

[71] T. Hartley and Q. Mehdi, "Online action adaptation in interactive computer games," *Comput. Entertain.*, vol. 7, no. 2, pp. 28:1–28:31, 2009.

[72] P. Spronck, M. Ponsen, I. Sprinkhuizen-Kuyper, and E. Postma, "Adaptive game AI with dynamic scripting," *Mach. Learn.*, vol. 63, pp. 217–248, Jun. 2006.

[73] J. Togelius, R. De Nardi, and S. Lucas, "Towards automatic personalised content creation for racing games," in *Proc. IEEE Symp. Comp. Intell. Game.*, Apr. 2007, pp. 252–259.

[74] N. Shaker, J. Togelius, and G. N. Yannakakis, "Towards automatic personalized content generation for platform games," in *Proc. 6th AAAI Conf. Artif. Intell. Interact. Digital Entertain.*, Stanford, CA, Oct. 2010, pp. 63–68.

[75] M. Jennings-Teats, G. Smith, and N. Wardrip-Fruin, "Polymorph: Dynamic difficulty adjustment through level generation," in *Proc. Workshop Procedural Content Generation Game.*, Monterey, CA, 2010, pp. 11:1–11:4.

[76] S. Kazmi and I. Palmer, "Action recognition for support of adaptive gameplay: A case study of a first person shooter," *Int. J. Comput. Game. Technol.*, 2010.

[77] E. Hastings, R. Guha, and K. Stanley, "Automatic content generation in the galactic arms race video game," *IEEE Trans. Comput. Intell. AI Game.*, vol. 1, no. 4, pp. 245–263, 2009.

[78] J. Kessing, T. Tutenel, and R. Bidarra, "Services in game worlds: A semantic approach to improve object interaction," in *Proc. 8th Int. Conf. Entertain. Comput.*, 2009, vol. 5709, pp. 276–281.

Ricardo Lopes received the B.Sc. and M.Sc. degrees in information systems and computer engineering from the Technical University of Lisbon, Lisbon, Portugal, in 2007 and 2009, respectively. His Ph.D. research subject is the "Generation of adaptive game worlds."

His current research interests include adaptivity in games, player modeling, interpretation mechanisms for in-game data, and (online) procedural generation techniques.

Rafael Bidarra graduated in 1987 in electronics engineering at the University of Coimbra, Portugal, and received the Ph.D. degree in computer science from Delft University of Technology, Delft, The Netherlands, in 1999.

He is currently an Associate Professor of Game Technology at the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology. He leads the research line on game technology at the Computer Graphics Group. His current research interests include procedural and semantic modeling techniques for the specification and generation of both virtual worlds and game play; semantics of navigation; serious gaming; game adaptivity and interpretation mechanisms for in-game data. He has published numerous papers in international journals, books, and conference proceedings. He integrates the editorial board of several journals, and has served in many conference program committees.