A review of interactive narrative systems and technologies: a training perspective

Linbo Luo[1], Wentong Cai[2], Suiping Zhou[3],Michael Lees[4], Haiyan Yin[2],
[1]School of Computer Science and Technology, Xidian University, China
[2]School of Computer Engineering, Nanyang Technological University, Singapore
[3]School of Science and Technology, Middlesex University, United Kingdom
[4]Section Computational Science, University of Amsterdam, Netherlands

Abstract—As an emerging form of digital entertainment, interactive narrative has attracted great attention of researchers over the past decade. Recently, there is an emerging trend to apply interactive narrative for training and simulation. An interactive narrative system allows players to proactively interact with simulated entities in a virtual world and have the ability to alter the progression of a storyline. In simulation-based training, the use of an interactive narrative system enables the possibility to offer engaging, diverse and personalized narratives or scenarios for different training purposes. This paper provides a review of interactive narrative systems and technologies from a training perspective. Specifically, we first propose a set of key requirements in developing interactive narrative systems for simulation-based training. Then we review nine representative existing systems with respect to their system architectures, features and related mechanisms. To examine their applicability to training, we investigate and compare the reviewed systems based on the functionalities and modules that support the proposed requirements. Furthermore, we discuss some open research issues on future development of interactive narrative technologies for training applications.

Index Terms—Interactive narrative, human-in-the-loop simulation, training, intelligent agents

1 Introduction

RECENT years have witnessed an increasing interest in developing simulation-based training systems for a wide spectrum of applications such as military operations [1], [2], [3], health care [4], [5], [6], [7], business management [8], [9], [10], [11], and education [12], [13], [14]. Simulation-based training is also referred to as human-in-the-loop simulation, where a trainee is situated in a synthetic environment and interacts with virtual entities for the purpose of knowledge and skill acquisition. Compared to traditional training methods, simulation provides an unique way for training that is more affordable and adaptive. For example, a soldier can practice military operation skills through simulation without being physically exposed to a real combat environment. It also has the advantage of offering an engaging, interactive and collaborative virtual experience, which can lead to better learning performance [14], [15].

Despite its increasing popularity, simulation-based training still faces many challenges in its development. One of the key challenges is the provision of scenarios. A scenario in terms of training refers to a sequence of events that a trainee experiences during the simulation. It is an essential component to drive the simulation towards achieving the desired training objectives. Through playing different scenarios, a trainee can exercise different skills or undertake certain missions for training. In practice, manual authoring of scenarios is a time-consuming and tedious process. Thus, the number of scenarios that can be played is a critical bottleneck to training [16], [17]. Moreover, during the simulation, the execution of a scenario has to be coupled with trainee's actions in an interactive manner. It requires that the training system has the capability to allow a trainee to influence the way in which a scenario progresses in the simulation. Such a capability is not commonly supported by traditional non-interactive simulations.

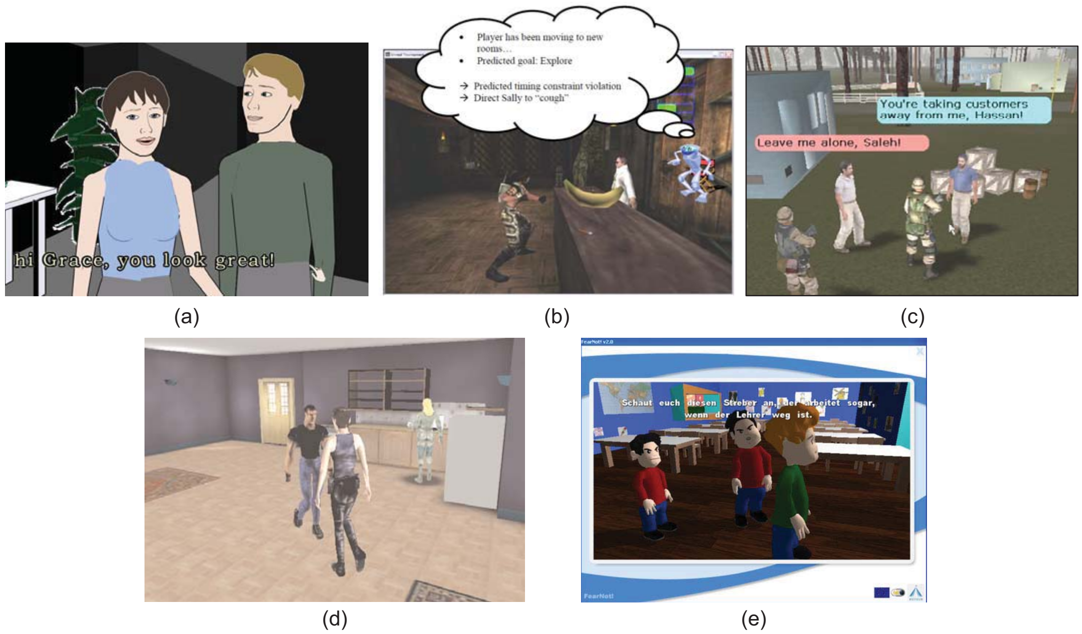

Fig. 1. Screenshots of prototype interactive narrative systems: (a) Façade [18], (b) IDA [19], (c) ISAT [3], (d) I-Storytelling [20], (e) FearNot! [21]

Over the past decade, interactive narrative (also known as interactive storytelling) has been rapidly developed as a new form of digital entertainment. More recently, there is an emerging trend to apply interactive narrative in training domain [3], [17], [22], [23]. In an interactive narrative system, a player is situated in a virtual world and she/he can take different actions in the progression of an unfolding story. The vision of interactive narrative is to create a dramatic "Holodeck"-like experience [24], in which a dramatically significant story can be dynamically generated via player's interaction in a highly-immersed virtual environment. To enable interactivity, interactive narrative system is designed to allow a player to influence the way in which the story unfolds. Most interactive narrative systems (although not all) also employ some automated means (e.g., AI planning [25], [26], [27]) to generate narratives, which helps to alleviate the burden of authoring. While the research on interactive narrative is still a young and emerging area, many interactive narrative systems and techniques [18], [19], [20], [25], [27], [28], [29], [30], [31], [32], [33], [34] have been proposed and steadily developed over the last twenty years. Fig. 1 shows the screenshots of some existing prototype systems developed for interactive narrative.

In this paper, we advocate the use of interactive narrative system as a viable tool to address the scenario provisioning challenge in simulation-based training. While a growing number of researchers have started to develop different interactive narrative systems for training, there is little work that has thoroughly examined the suitability of existing interactive narrative technologies in the application of training. To design an interactive narrative system for training, it requires the use of interactive narrative or scenario as a way to promote effective learning, rather than only to convey some dramatic meaning. Thus, the designer has to consider additional or even different requirements for such purposes. In this paper, we propose a set of key requirements in developing interactive narrative system for training. We use these requirements as the criteria to assess the existing interactive narrative systems. The paper aims to provide an in-depth review of the existing approaches and assess their strength and weakness from a training perspective. The assessment is intended to reveal potential improvements and identify open research problems for the future.

As interactive narrative is becoming an active area of research and practice, it is not surprising that literature review exists. In [35], Roberts and Isbell provided a comprehensive review and qualitative analysis on the existing interactive narrative systems. Specifically, they examined the type of systems that use the *drama manager* agent to direct the progression of narrative and they proposed a set of desiderata to analyze these systems. In a more recent review [36], Riedl and Bulitko suggested a taxonomy to classify existing approaches to interactive narrative along a three-dimensional space and they showed how a small sample of existing interactive narrative systems are placed within the space. The review provides a good overview on the landscape of interactive narrative research. However, it tends to focus on high-level approaches instead of diving into the specifics of any existing system.

The key difference of our review from the above mentioned reviews is that we examine the existing systems in a specific context of training applications, rather than in a general context. We identify a set of assessment criteria (i.e., the key requirements) that we believe are most relevant to training applications. Although some of the criteria are borrowed from the previous reviews, there are differences as we elaborate these criteria in the training context. Moreover, our paper also provides a more detailed review of the existing systems at a system design level. We investigate the key features and related mechanisms of the existing systems. We hope to provide researchers and developers with more concrete insights on the approaches and techniques in the literature that can be potentially applied in their own work.

The rest of the paper is organized as follows. Section 2 first proposes a set of key requirements of developing interactive narrative systems for training. Section 3 then presents a review of some representative interactive narrative systems, in terms of the system architectures, design features, and related mechanisms. Section 4 summarizes and compares the reviewed systems based on the proposed requirements. Section 5 rounds up the review and discusses some important issues for applying interactive narrative technologies to training and simulation.

2 Key Requirements in Developing Interactive Narrative Systems for Training

In this section, we propose a set of key requirements that need to be considered when developing an interactive narrative system for training. We aim to use these requirements as the criteria to assess the existing interactive narrative systems from a training perspective. The proposed requirements may also serve as a general framework for designers to assess their own interactive narrative systems for simulation-based training in future development. Generally speaking, the proposed requirements can be grouped into two categories. The first category is from a trainer's point of view and considers how an interactive narrative can satisfy trainer's need to achieve different training objectives. The requirements in this category include *controllability* and *robustness*. The second category is from a trainee's point of view and focuses on how an interactive narrative system can offer an engaging and individualized narrative experience to a trainee. The requirements in this category include *personalization* and *interaction*.

2.1 Controllability

In simulation-based training, a trainer has the role to direct the training towards certain training objectives. For example, a trainer may wish to let a trainee exercise a certain set of skills over other skills in a training session, depending on the trainee's existing abilities. Thus, it is necessary to allow the trainer to control how a narrative or scenario is generated, such that it can reflect the trainer's preferences over different training objectives. In this paper, we refer to the controllability as the capability of an interactive narrative system to allow a trainer to direct the trajectory of a narrative towards some desired ones.

To further clarify the controllability requirement, we first draw a distinction between the role of a trainer and an author in an interactive narrative system for training. While some existing works (e.g., [2]) assume the these two users have the same role, we consider a trainer as the user of the system and an author as the developer of the system. Specifically, an author is considered to be responsible for the creation of the narrative content (e.g., plot points) and/or narrative structure, which are used in interactive narrative system to generate variants of narrative (i.e., different narrative trajectories). In software engineering term, author can be referred to as library developer. A trainer, on the other hand, does not need to create the actual narrative content. Rather, an interactive narrative system should allow trainer to interface with the system in the way that she/he can control or influence the choice on which narrative trajectories are generated. In software engineering term, trainer can be referred to as library user.

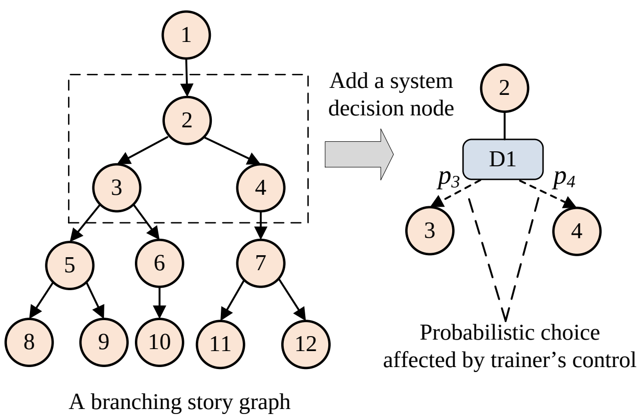

Fig. 2. An example of branching story graph and the way to ensure controllability. In the graph, nodes are authored narrative content (plot points), solid directed arcs are choices made by player (i.e., trainee), D1 is a decision node controlled by the system, and dashed directed arcs are choices made by the system

Based on the distinction between trainer and author, we emphasize that the controllability is from a trainer's point of view, rather than an author's. It differs from the authorial intent as discussed in [36], which refers to the extent to which an author's intent constrains the interactive narrative system. In [36], an example of high authorial intent is the systems that utilize a branching story graph (see Fig. 2), where a player can choose branches in the graph at her/his own will during game play. In such a system, as every narrative trajectory is manually created by an author, the author's intent is considered to be fully preserved. However, this does not necessarily suggest that the system has full controllability, as the trainer does not have the control on which narrative trajectory will be experienced by the trainee. One way to enable the controllability in a branching story graph is to embed some decision nodes (e.g., D1 in Fig. 2), which select branches based on the system's decisions instead of the player or trainee's decisions. By doing this, some mechanisms can be designed to allow a trainer to direct the system's decision and thus influence how a particular narrative trajectory is selected.

Depending on the way in which a trainer directs an interactive narrative system, we classify the controllability into two types: soft control and hard control. Soft control is the ability to influence the likelihood that certain narrative content will occur during game-play, while hard control is the ability to force certain narrative content to occur. In the example in Fig. 2, the effect of soft control could be to increase the probability of selecting node 3 (i.e., p_3) over the probability of selecting node 4 (i.e., p_4) or vice versa. The effect of hard control is to make one probability (either p_3 or p_4) to be one and the other to be zero.

2.2 Robustness

In an interactive narrative, a player is usually provided with a wide range of actions to perform during the game. In some cases, the player may perform some unexpected actions, which may cause the narrative to go outside of the story space (i.e., the space of possible narrative trajectories) constrained by the author. Brain Magerko [37] refers to this issue as the *boundary problem* in interactive narrative. In the context of training, such s story space may be further confined into a sub-space due to the trainer's control (as discussed in section 2.1). Thus, it is important to ensure the system is robust enough to recognize and respond to the potential deviation from this trainer-controlled story space. In this paper, we refer to the robustness as the capability of interactive narrative system to deal with the trainee's actions that can possibly lead the narrative to go beyond the bounds as confined by the trainer. This requirement is necessary not only to guarantee the coherence of narrative, but also to make sure that trainee can follow some narrative trajectories for some specific training objectives as defined by trainer.

To ensure the robustness of interactive narrative system, the approaches adopted by the existing systems can be generally classified into two categories: *prevention approach* and *mediation approach*. The prevention approach provides player only a limited set of actions to choose (usually at some prescribed times) during the game. In other words, the system disallows any actions that can potentially cause the boundary problem. One example is the systems using the branching story graph [38]. In these systems, a player can only perform the actions at some specific times with a set of alternatives to choose from. While the prevention approach is the most straightforward way to ensure the robustness, it limits the interactivity of the system to some extent. Thus, it is less desirable, especially in the context of training.

The mediation approach allows more freedom on the actions that a player can perform and it adopts some mediation strategies to solve the boundary problem caused by player's actions. Depending on how the boundary problem is handled, the mediation strategies can be further classified into two categories: *reactive approach* and *predictive approach*. The reactive approach handles the boundary problem only when player' actions cause the deviation from the desired story space. That is when the boundary problem actually occurs. One notable system that adopts such reactive strategies is the Mimesis system [34], where two specific mediation strategies, *accommodation* and *intervention*, are proposed to manage player interaction. In contrast, the predictive approach anticipates the possible boundary problem beforehand. The systems which adopt predictive strategies, such as IDA [33], usually make inference on the future behaviors of players and direct the narrative based on such inference.

2.3 Personalization

While the first two requirements focus on how to control the narrative trajectory (both at compile time and run-time) towards the ones preferred by the trainer, it is also necessary to consider how the system can generate narratives catered to different trainees. For instance, in challenge-focused training, the training process will be effective only when the scenario or narrative presented to the trainee is neither too difficult nor too easy. In this case, an interactive narrative system has to provide different narratives for different trainees depending on their existing skill levels. In this paper, we refer the personalization as the capability of interactive narrative system to adapt itself to different trainees.

To achieve personalization there is a growing trend [36] to incorporate player modeling for interactive narrative. A player model is used to learn and capture the differences of individual players (i.e., trainees). By using a player model, the interactive narrative system can infer how to select narratives that best fit the trainee's characteristics. For example, in [39], a player model is designed to automatically learn the player's preferred playing styles based on the player's in-game actions. The interactive narrative system then dynamically adapts the narrative content to match with the inferred playing style of the player.

Depending on how the player model is constructed, existing approaches to realize personalization for interactive narrative can be generally classified into two categories: *hypothetical approach* and *data-driven approach*. The hypothetical approach constructs the player model based on designer's experiences. The designer usually makes hypotheses on how to map the observed player actions to the player's characteristics (e.g., playing styles, preferences, and skill levels). For example, the player model in [39] uses the designer pre-defined rules for updating the player's playing style, given the actions taken by the player. The data-driven approach relies on the actual player game-play data and player feedback to build the player model. For example, in [40], past players' feedback and their game-place traces are collected.

A data-driven player model is built based on these data and it employs the case-based reasoning to predict the player preference by comparing the current player's behavior with the behavior of past players.

2.4 Interaction

The key difference of interactive narrative from traditional storytelling is that it allows player to interact with the system so as to influence the progression of a storyline. In this paper, we refer the interaction as the way how a trainee can interact with interactive narrative system. We consider the interaction as an important requirement in training, as the way in which a trainee can interact with the system largely affects the degree to which the individual difference of trainees can be exhibited during the game. For example, consider the choice of actions that an interactive narrative system provides to a trainee. If the system only allows a trainee to choose a set of actions at some prescribed times, it is more likely that different trainees will choose the same action at a given time because of the limited choices. In such cases, it is more difficult to recognize the difference of individual trainees in terms of their preferences, playing styles or abilities.

In the existing interactive narrative systems, the most common *mode* of player interaction is that a player plays the role of a first person character (e.g., protagonist). Acting as a first-person virtual character, a player can directly communicate with other virtual agents and manipulate the objects in the virtual world. Besides being a first-person participant, a player can also interact with story world "off-stage", as suggested by Cavazza et al. [41]. An example of the "off-stage" intervention is that as a spectator, a player may give advice to virtual agents via a communication link. Regarding the interaction *time*, some existing systems give the player the freedom to perform actions at anytime during the game, while other systems only allow players to perform actions at some prescribed times. In terms of the *choice* of actions, some systems allow players to perform all available actions, which are applicable to the current situation. In some other systems, players can only choose a limited set of actions at given times.

3 A Review of Nine Interactive Narrative Systems

In this section, we provide a review of existing work in terms of their key design features, system architectures, core modules and functionalities. Our aim in this section is not to give a complete account of all existing work. Instead, we will sample some representative work in interactive narrative research. We select some of the most influential work which we judge to have had significant impact on the later design of interactive narrative systems. We also select some more recent works which develop interactive narrative systems not only for entertainment purpose, but also for training and education purposes. We also tend to choose the work which provides a full-fledged system design of interactive narrative, rather than merely focuses on a particular issue, such as narrative generation. Thus, some earlier works on automatic story generation, such as Tale-Spin [42], Universe [43] and MINSTREL [44], are not included in this review. In the following sub-sections, the reviewed works are presented in chronological order to give the reader a general idea of how the technologies of interactive narrative have evolved over the years.

3.1 Mimesis

The Liquid Narrative Group is led by R. Michael Young at North Carolina State University. The Mimesis system [34], developed by the group, is one of the pioneering works on interactive narrative. The system uses a plan structure to drive the actions in the story world. One key feature of Mimesis is that it integrates plan-based behavior generation with interactive game environments. This feature allows Mimesis to automatically produce intelligent action sequences that are highly sensitive to run-time context. Another key feature of Mimesis is the proposed narrative mediation strategy to manage the user interactions that interfere with story structure. The mediation strategy helps to prevent the potential plan failure from the threats introduced by user's actions.

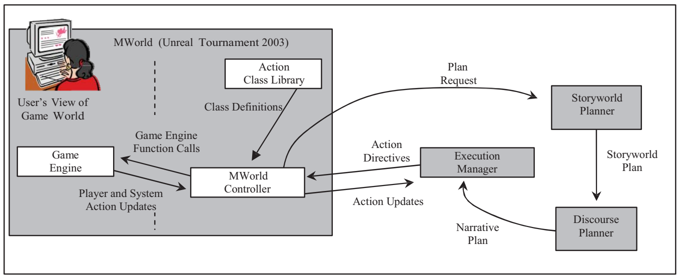

Fig. 3. The Mimesis system architecture (retrieved from [34])

The Mimesis system employs a component-based architecture to promote the modularity of the system. The architecture design of the Mimesis system is shown in Fig. 3. The execution of the system starts with *MWorld controller* sending a plan request to *storyworld planner*. The plan request contains the information on current and goal states of the game (used to form initial plan) and a set of pre-defined actions (operators) in the story world. The *storyworld planner* responds to the plan request by generating a storyworld plan that specifies the action sequences of characters/objects in the game. Based on the storyworld plan, the *discourse planner* creates another plan of action sequences carried out by the game engine's media resources (e.g., camera, narration and background music), and it integrates the two plans to form a final narrative plan. The *execution manager* receives the narrative plan and builds a directed acyclic graph (DAG) to represent actions in the plan (as nodes) and temporal orderings (as edges). The *execution manager* initiates an action's execution by selecting an action node from DAG and sending a corresponding action directive to *MWorld*. *MWorld* is a customized game environment built on top of the Unreal Tournament engine, and it consists of three elements: *game engine*, *action class library*, and *MWorld controller*. The *MWorld controller* serves as a central message router for other components in the system.

Mimesis provides an explicit support for user interaction management through the proposed mediation strategies [45]. User actions in the Mimesis system are classified into three categories: constituent, consistent and exceptional. Constituent actions are those prescribed as part of the narrative plan, whose effects can drive the story forward. Consistent actions are those, whose effects do not threaten any of the causal links for the remaining part of the plan. Exceptional actions are those, whose effects threaten at least one of the causal links in the plan. The mediation strategy of Mimesis always permits constituent and consistent actions to be performed. For exceptional actions, Mimesis handles the situation with one of two mediation strategies. The first strategy is to *accommodate* the user's action by re-planning the unexecuted portion of the plan. The re-planning process helps to reestablish any threatened causal links caused by user's exceptional action. The second strategy is to *intervene* with the user's action by replacing the action's effects with the ones consistent with the causal constraints. An example of intervention is a vending machine in the story world returns the money back to the user instead of delivering a can of beverage, as user's action (buying a drink) may threaten a pre-condition (possessing enough money) in the subsequent story plan. Given an initial storyworld plan, Mimesis predicates all possible exceptions caused by the user's actions, and it constructs a mediation table with a set of mediation policies to handle these exceptions based on either *accommodation* or *intervention*.

In terms of *controllability*, the Mimesis system does not provide explicit support to allow a trainer to direct the narrative trajectory. In Mimesis, the narrative plan request, which defines the goals and actions of the generated plan, is made by the game engine itself. No interface is provided to allow a trainer to influence how a narrative plan can be generated. However, as Mimesis adopts a partial-order plan structure for narrative representation, it is potentially possible to integrate some control mechanisms (e.g., through some hard or soft constraints to planning). The *robustness* of the Mimesis system is realized through the proposed *accommodation* and *intervention* strategies. Both strategies are considered as a *reactive approach* as we described in section 2.2. Mimesis does not address the *personalization* requirement, as the system does not adapt itself to cater for the player (i.e., trainee) differences.

However, as planning-based search is used for narrative generation, it is possible to use some heuristics to guide the searching process so that the generated narrative can be tailored to the individual difference. With regard to the *interaction*, Mimesis allows players to control a virtual character in a story world and issue commands to trigger specific user actions [46], [47]. The players can perform an action at anytime during the game and they can take any actions that are applicable to the current situation.

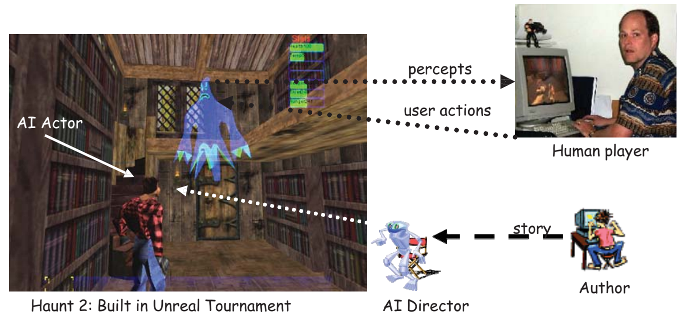

Fig. 4. The Interactive Drama Architecture (retrieved from [19])

3.2 IDA

The Interactive Drama Architecture (IDA) [33] proposed by Brain Magerko and John Laird at the University of Michigan is an interactive narrative system, which focuses on achieving the robustness of the system by introducing a *predictive approach* to deal with the player's actions in an interactive virtual environment. The proposed approach is intended to resolve the *boundary problem* [19], [37] that occurs, when a player takes some actions that lead to the consequences outside of the content (i.e., the story space) of pre-authored plot. To address this issue, the IDA system proposes a *director* agent to direct the behaviors of the virtual agents and influence the player's behaviors to confine with the pre-defined story space. The *director* in IDA does not wait to handle the player's actions until the *boundary problem* occurs. Rather, it employs a predictive model of player's behaviors and anticipates the possible *boundary problem* beforehand. The *director* will preemptively initiate some subtle strategies to direct the player away from the actions that may endanger the story plot. Compared to the reactive approach, IDA's predicative approach in handling user exceptions can help to improve the player's experience and believability of the system.

As shown in Fig. 4, the overall architecture of IDA comprises of an author, who writes the plot of story, the *director* agent, the virtual world populated with synthetic characters, and the human player who interacts with the virtual environment. The author, who could be a human writer or an autonomous authoring system, is responsible to produce the initial plot of a story, using a story representation language provided by IDA. The story plot in IDA is represented as a graph of *plot points*, which contains a set of preconditions and actions. It should be noted that even though IDA uses the plan-based narrative representation, no automated mean (e.g., planning algorithms) is employed to generate narrative plans. The narrative plan is pre-defined by the author and it is then used as used as the input to the *director* agent to drive the story in the virtual world. The functionality of the *director* agent is to monitor the behaviors of the player and direct the virtual character in the story world to ensure the executability of pre-defined narrative plan. The virtual characters in IDA are implemented as rule-based Soar agents [48].

The central role in the IDA system is the *director* agent, which is able to guide the player's behaviors, based on both the current actions and hypothesis of future actions of a player. The *director* agent employs a probabilistic model of player behaviors, as shown in Fig. 5. This player model represents a general hypothesis of how a player would behave in the virtual world. To predict the likelihood that the *boundary problem* might occur, the *director* agent first creates an internal simulation of the story world. Based on the simulated world, the *director* agent runs the player model and observes whether any plot elements may be affected by the anticipated player's actions. The probability of the player's actions fulfilling plot content, $P(F)$, is then computed. If $P(F)$ is less than an author-defined threshold, the *director* agent will take some preemptive strategies to prevent the *boundary problem*.

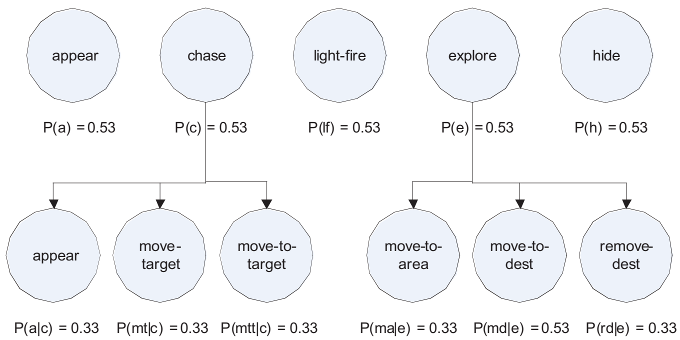

Fig. 5. An example of the probabilistic model of player behaviors in IDA (retrieved from [19])

The design of IDA helps to ensure that the executed story plot does not deviate from the one defined by an author. However, in terms of *controllability*, the system does not provide a way to allow a trainer to direct the story plot. The direction of story progression is fully controlled by the autonomous *director* agent. The *robustness* of the system is ensured through the use of the *director* agent, which uses a predicative approach to handle player's actions. It should be noted that even though a player model is used to predict the player behaviors in IDA, the objective is to check whether the predicted player behaviors will cause the deviation from the pre-defined plot. The player model is not utilized for the purpose of *personalization*. With regard to the *interaction*, IDA provides a proper management of player interactions. Thus, it does not enforce significant restrictions on the player's actions. A player in IDA plays a role of a ghost in the story world. She/he can freely move around the environment and interact with other virtual characters at any time during the game. The player's interaction with virtual characters will influence how the virtual characters will behave in the story.

3.3 I-Storytelling

Marc Cavazza and his colleagues in Intelligent Virtual Environments Lab at Teesside University have made extensive contributions towards character-based interactive narrative. The Interactive Storytelling (I-Storytelling) system [20], [49] developed by the group uses the structure of Hierarchical Task Networks (HTN) [50] to represent the roles and tasks of autonomous agents in an interactive storytelling environment. The system focuses on the modeling of virtual agents' behaviors. The story variants are expected to emerge from the dynamic interactions of the virtual agents. The HTN-based hierarchical planning adopted by I-Storytelling facilities the encoding of domain knowledge, and the interleaving of planning and execution.

In the I-Storytelling system, each main character's goal and tasks are represented using a HTN, which is formalized as an AND/OR graph. Fig. 6 shows a typical HTN representation of a main character Ross in a sitcom Friends^{TM}-like scenario. The character's main goal is defined at the top of the hierarchy. The top-level goal can be refined into the sub-goals (i.e., AND nodes) representing various steps to achieve the top-level goal in an implicit time ordering. For example, as Ross's main goal is to take character Rachel out, he may achieve this goal following the steps of acquiring information about Rachel, then gaining her friendship, and finally asking her out. For each sub-goal, there could be also a set of alternative solutions (i.e., OR nodes) to achieve it. For example, in order to acquire Rachel's information, Ross may either ask her friend, borrow her diary, or phone her mother. These alternative solutions are represented as sub-plans. At the lowest level of HTN, the leaf nodes are the primitive actions that can be directly executed by the virtual agents in the game environment.

For each virtual agent, a suitable plan from the HTN is generated using a heuristic search algorithm [20]. The algorithm uses some heuristic values to guide the searching process and these heuristic values reflect the personality and mood of the character. The virtual agents' behaviors are driven by the generated HTN plans. In I-Storytelling, the story variations are emerged from the dynamic interaction between virtual agents, and user intervention.

The dynamic interaction among agents may introduce some emergent situations that are not represented as part of virtual agent's HTN

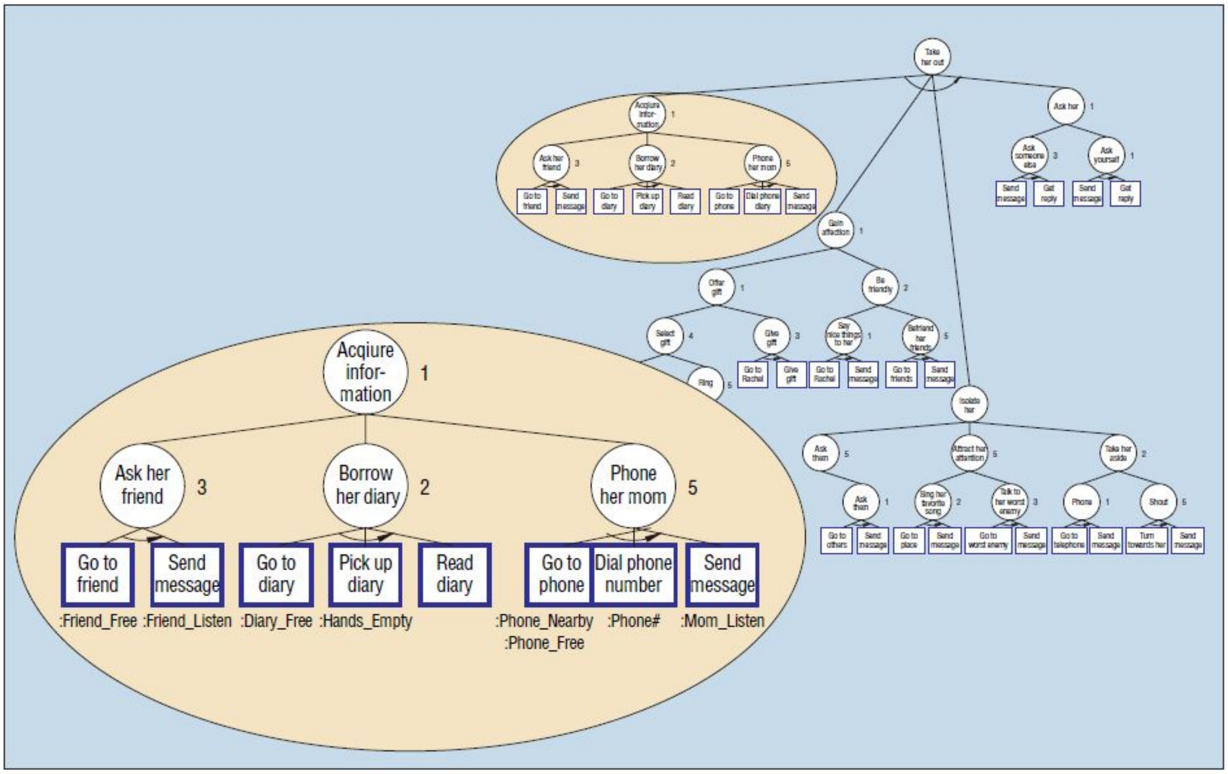

Fig. 6. An example of Hierarchical Task Network for character behavior (retrieved from [20])

plan. For example, Ross may meet Rachel accidentally at the early stage of Ross's plan. Apart from using re-planning, the I-Storytelling system has proposed two mechanisms, *situation reasoning* and *action repair* to handle emergent situations. The *situation reasoning* is used to avoid undesirable results. As for the case of accidental encounter with Rachel, one example of *situation reasoning* by Ross is to hide from her. By doing this, Ross can still resume his initial plan after Rachel passes. The *action repair* is used to restore the executability condition from action failure. One example is that Ross intends to perform the "read Rachel's diary" action, but Rachel is currently writing in the diary. In this case, the *action repair* to restore the executability condition (i.e., the diary is available for reading) could be simply to wait Rachel finishing writing. In the system, the *situation reasoning* and *action repair* are implemented as separate plans from the virtual agent's main plans.

In I-Storytelling, how the story will progress is largely driven by the character's interactions. The system has the advantage in maintaining character believability, as all the characters' behaviors are driven by their own goals and intentions (through their own HTN plans). However, it is difficult to enable *controllability* in such a system. This is because the system does not have overall control on how the story will progress over time. In terms of the *robustness*, I-Storytelling uses the reactive approach, which creates the contingency plans to handle the emergent situations. However, it should be noted that the proposed mechanisms are used to ensure the executability of the virtual agent's HTN plans. These mechanisms do not guarantee that the progression of the story will follow a certain direction during the execution. The *personalization* requirement is not considered in the I-Storytelling system. With regard to the *interaction*, I-Storytelling allows anytime player interaction. The player can choose to intervene at anytime by either manipulating the objects with narrative meaning or giving advice to virtual agents via natural language processing (NLP). It should be noted that the HTN-based story representation in I-Storytelling isn't capable of generating highly complex stories as it is bounded by its total-ordering and task independence assumption. Thus, the same group from Teesside University has proposed to use Heuristic Search Planning (HSP) [32] as an alternative solution.

3.4 Façade

Façade [18] developed by Michael Mateas and Andrew Stern is one of the most successful fully-realized interactive narrative systems. The design of the system explicitly addresses the issue of balancing character believability and plot coherence. In Façade, rather than solely relying on the behavior-based autonomous agents, an additional *drama manager* agent is introduced to direct the story direction. The drama manager operates on fine-grain plot elements called *beats*. A beat is defined as the smallest unit of story structure. Each beat contains a collection of characters' behaviors, tailored to a specific situation or context. One key feature of Façade is it aims to offer high agency [51] that allows the system to react to and incorporate user's interactions in a narratively meaningful way. The system is designed to respond to user's interactions both locally and globally in order to provide a dramatic-rich user experience.

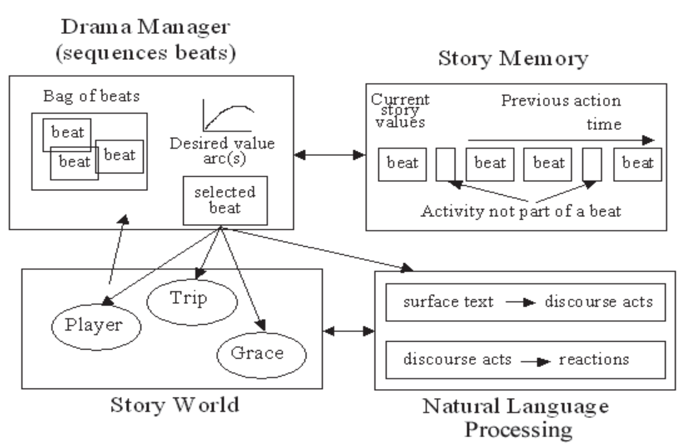

Fig. 7. Façade system architecture (retrieved from [18])

Fig. 7 shows the overall architecture of the Façade system. The *drama manager* is responsible for continuously monitoring the situation and selecting an appropriate beat when the current beat terminates or aborts. The beat is selected from all the available beats of a story (i.e., bag of beats). Each beat is annotated with preconditions, weight, priority and effects. This annotated knowledge is used by *drama manager* for selecting the next beat. For example, the effects of beat are used to compare with the current desired value in a user-defined dramatic arc, and the beats whose effects closely match with the desired value may be given high scores for selection. Once the *drama manager* has chosen a beat, it will update the *story memory* and send the selected beat to the *story world* for execution. In the *story world*, only one beat is active at a time. The beat contains the behaviors of virtual agents, as well as the reactions to the player in a given situation. The user interactions are primarily handled by *natural language processing*. The player's inputs to the system are interpreted into the *discourse acts*, which concisely represent the general meaning of the player's actions. Each beat is designed to respond to these *discourse acts*, based on a set of forward-chaining mapping rules.

In each Façade beat, the collection of behaviors are organized into a set of *beat goals*, which define different narrative goals within the context of a particular beat. A beat goal could be a transition-in goal that provides a transition to the current beat. Several main body goals could be defined to establish a main conflict or dramatic situation within the beat. There is also a transition-out goal to transit out of the beat. To handle the player interaction, a special type of beat goal called mix-in goal is defined to mix in a performance of the reaction to player's actions within the beat. When a mix-in goal needs to be executed, the system will abort the current beat goal and prioritize the mix-in goal to start immediately. For example, when a virtual agent is trying to offer a drink to a player, the player may start to discuss a topic about marriage. The virtual agent may mix-in a short beat goal discussing her feelings about marriage, then resume the previous beat goal by continuing offering a drink to the player. Each beat goal in a Façade beat is realized through a set of action/reaction pairs, where a virtual agent performs an action and other agents react accordingly. These pairs of joint behaviors support the coordination between the virtual agents, and they are implemented using a reactive behavior language called ABL [52].

The *controllability* of the Façade system is achieved by specifying the dramatic arc (as shown in Fig. 7). The system allows the user (e.g., trainer) to define the shape of the arc, which will influence how the beats will be selected in the beat sequencing process. It is a form of soft control, as it only increases the probability of certain type of beats (e.g., the beats with high story tension) to be selected. In terms of the *robustness*, Façade uses the mix-in goals to handle the player's behaviors. The mix-in goals are designed to reactively respond to off-topic behaviors initiated by the player. It should be noted that all the mix-in goals need to be defined at the authoring stage, which may cause authoring burden. The *personalization* is not considered in the Façade system. With regard to the *interaction*, a player in Façade is given the flexibility to speak to virtual agents via NLP, make gestures, and take actions in the virtual environment. It also does not limit the time and choice that a player can perform actions.

3.5 Thespian

Thespian [28], [53], [54], developed by the computational emotion group at University of Southern California, is a multi-agent framework for interactive narrative. The system is intended to be used for both entertainment (e.g., modeling the Little Red Riding Hood story) and pedagogical purposes (e.g., tactical language training). Thespian is built based on PsychSim [55], which models the agent's reasoning based on Partially Observable Markov Decision Problems (POMDPs) [56]. In Thespian, characters are controlled by goal-driven agents. A model of player [57] is explicitly introduced to represent different types of player and the model is utilized to predict the player's interactions with virtual agents. Based on the player model, a directional control mechanisms [53] is proposed to modify the virtual agents' goals and beliefs to prevent the violation of author's intentions.

Thespian adopts a two-layer approach, as shown in Fig 8, for authoring and simulating interactive narratives. At the bottom layer (i.e., character level), a multi-agent system is developed to support the simulation of various virtual agents. The interactions among virtual agents are realized in a turn-taking manner, with only one agent acts at each turn. The behaviors of an agent are motivated by the agent's goals being assigned by the author. The agent's personality is reflected by its various goals and the relative importance (weights) of these goals. To make sure the agents' behaviors are well-motivated and believable, a personality fitting process [28] is also proposed in Thespian for character authoring. The process requires the author to provide some desired story paths (i.e., linear sequence of agents' actions) as the inputs. The process can then automatically fit agents' personalities to their roles in the story by adjusting the goal weights of agents.

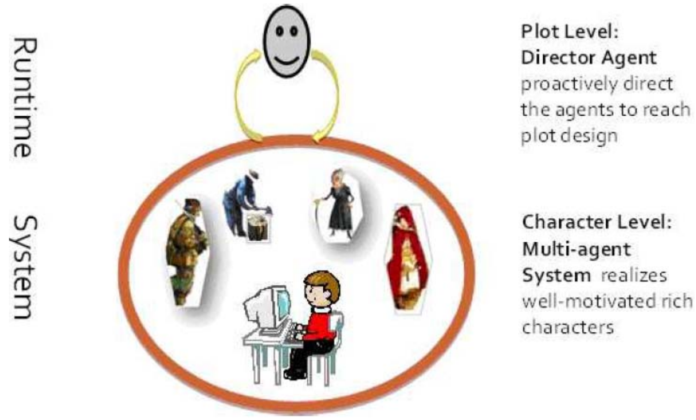

Fig. 8. Two-layer system architecture of Thespian(retrieved from [53])

At the upper layer (i.e., plot level), Thespian provides an omniscient agent, named the director agent, to proactively direct the agents' behaviors in accordance with the author's intentions. The director agent can access the beliefs of all virtual agents, reason about the future interactions between the agents and user, and check for the potential violation of the temporal and partial order constraints specified by the author. To prevent the potential violations, the director agent can either fit the agents' goals or modify the agents' beliefs to influence their behaviors. The director agent iteratively makes the adjustment to virtual agents and check for the violations in future steps, until a satisfactory solution is found or the maximum number of attempts is reached. To facilitate the authoring and evaluation of stories, a model of player is also incorporated in Thespian. The "simulated" user acts as a virtual agent in the story. By assigning different goals to the player agent, the player model can be used to represent different types of players (e.g., talkative and non-talkative). The player agent can choose actions at each turn of the player and the director agent is used to perform the directional control accordingly.

The *controllability* of the Thespian system is ensured by allowing the author or trainer to specify temporal and partial order constraints. It is a form of soft control, as the director agent only influences the virtual agents' behaviors by modifying their beliefs or goals. By incorporating a model of the player, Thespian adopts a predictive approach, similar to IDA, for achieving *robustness*. It simulates the future behaviors of agents and player, and anticipates the possible violations of the prescribed story requirements beforehand. Even though the player model is designed in Thespian, it is not utilized for the purpose of *personalization*. As for the *interaction*, a player acts in a first-person role in the environment and interacts with virtual agents individually via spoken language and gestures.

3.6 ISAT

The Interactive Storytelling Architecture for Training (ISAT) [2], [22] developed by Brain Magerko et al. is an interactive narrative system specifically targeted for training domains. The ISAT system distinguishes from the previously discussed systems in its ability to generate narratives that are tailored to individual players (i.e., trainees). ISAT introduces an intelligent *director* agent, which chooses the training narrative/scenario based on a trainee's skill level. A player's skill model is designed to track the skill proficiency of a trainee during the game-play. The key feature of ISAT is that it provides customized and individualized training experience through real-time narrative adaptation.

Fig. 9 shows the overall architecture of the ISAT system. In ISAT, the trainer also plays the author role. She/he can interface with a story authoring tool, named *Scribe* [58], to define both the story content and flow. The defined story/scenario is represented in the form of partially-ordered plot points. Each plot point is also referred to as a scene, which contains a set of events occurring within a short period of time in a specific location. The *director* agent is responsible for executing the trainer-specified story within the training environment. The director agent maintains a trainee skill model to dynamically track the trainee's proficiency in a set of domain-specific skills. Based on the state of the trainee skill model, the director agent can perform a set of actions in order to provide an individualized training experience to the trainee. These actions include spawning virtual characters and objects, modifying character's behaviors, and selecting appropriate plot points to execute.

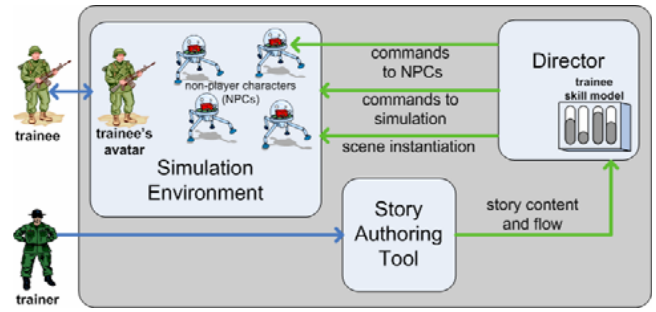

Fig. 9. ISAT system architecture (retrieved from [2])

During a training session, the director agent monitors the actions of the trainee and updates the trainee skill model accordingly. For instance, if a certain skill is performed incorrectly, the director agent will decrease the corresponding skill *score* in the trainee skill model. For the selection of plot points, each plot point is annotated with the set of skills that it tests. The director agent will select the next plot-point that best matches with the current state of the skill model. The selection process is illustrated in Fig. 10. As shown in Fig. 10, the current state of trainee skill model indicates the low proficiency in skill D and E and the current scene (i.e., the plot point) are used to test the skill A and B (the blue circles). Thus, given all the candidate scenes to be selected as the next

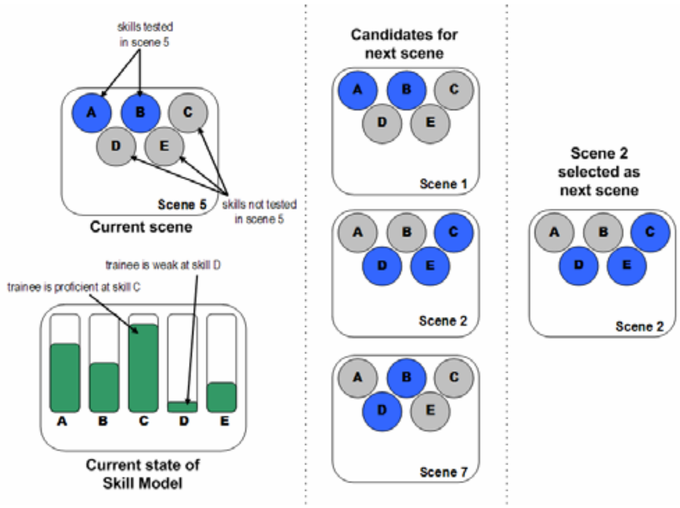

Fig. 10. Process of scene selection in ISAT (retrieved from [2])

scene, the scene that can practice both skill D and E is then selected.

In ISAT, the *controllability* is exercised by putting the trainer in the role of author. The trainer has the responsibility to define the story space with the assistance of an authoring tool. As discussed in section 2.1, the manual authoring of story content can ensure the author's intent (or trainer's in this case) is well preserved. Yet, it does not necessarily suggest full controllability. In ISAT, how the narrative content (e.g., plot points) that is selected and presented to the trainee is still managed by the director agent. For ensuring *robustness*, ISAT adopts a reactive approach, called *reactive direction* [2], to prevent the trainee from going into a situation outside the training space. The trainee skill model in ISAT is specifically designed to achieve the *personalization*. The skill model makes hypothetical mapping between trainee's in-game actions and their proficiency in various skills. Thus, it takes a *hypothetical approach* for player modeling. For the *interaction*, the trainee plays a first-person role to move around the environment and search for casualties in a combat medic training scenario. When a casualty is found, the trainee can choose all available medical treatment options from graphical menus and perform the medical procedures.

3.7 PaSSAGE

The PaSSAGE (Player-Specific Stories via Automatically Generated Events) system [39], developed by David Thue et al. at the University of Alberta, is an interactive narrative system that incorporates player modeling for generating narratives tailored to the player's preferred playing style. The purpose of player modeling in PaSSAGE is to improve the entertainment value of the narratives and thus make the player experience more enjoyable. Similar to the ISAT system, PaSSAGE steers the narrative in run-time based on the player's in-game actions.

The PaSSAGE uses the tree structure as shown in Fig. 11, similar to the branching story graph, to represent a story. The rounded boxes in Fig. 11 denote the in-game events (also known as encounters) and diamonds indicate different story endings. The square boxes (i.e., D_1 to D_3) are the system's decision nodes, which are utilized to make story adaptive to different players. At each decision node, the PaSSAGE system automatically selects the next encounter based on the current estimate of play style of the player. At the design stage, all the encounters are annotated with the information on which type of players it is suitable for. During the game-play, the system examines information about the encounter and its branches and it chooses the encounter whose branch best fits the current value in the player model.

The player model in PaSSAGE classifies different players into stereotype based on their play styles. Specifically, the model considers five types of play style: fighter who prefer combat, power gamer who prefer collecting items and gaining riches, tacticians who prefer think creatively, storytellers who prefer complex story plots, and method actors who prefer dramatic actions. The player model uses a weight vector to indicate the player's inclination to the five types of play styles and updates each weight based on player's actions. For example, if a player chooses to search for a valuable item, the weight for power gamer will be increased. At the design stage, all the possible actions taken by the player are pre-specified with the values that change the weights in the

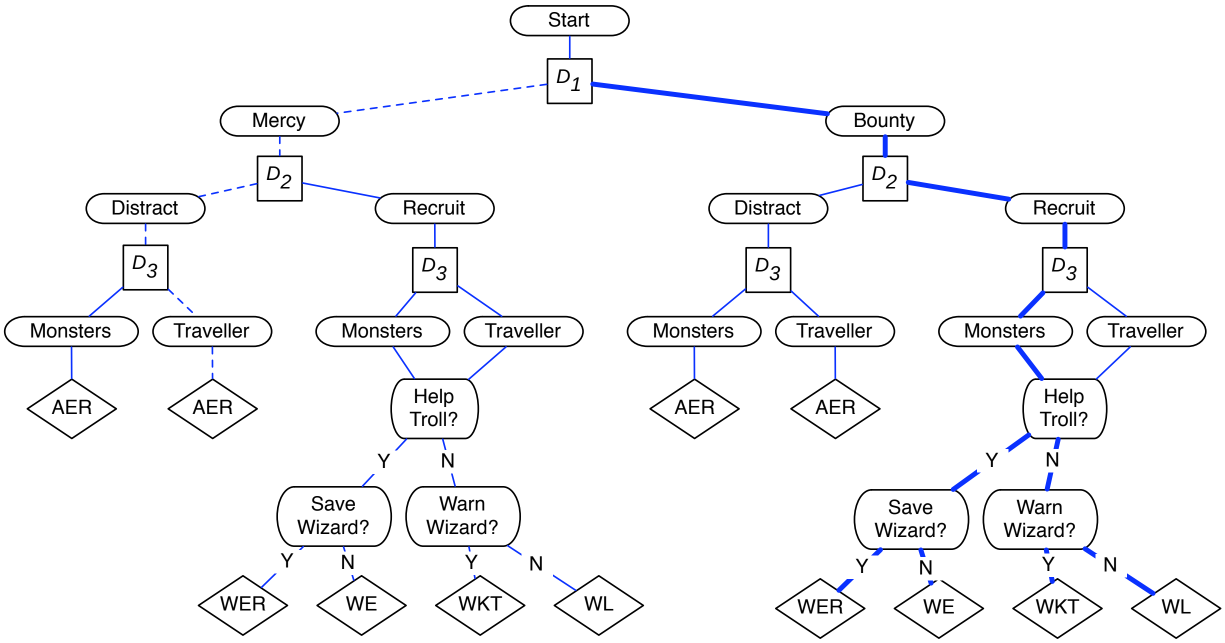

Fig. 11. Example of tree-based story representation in PaSSAGE(retrieved from [39])

player model.

The PaSSAGE system does not provide an explicit support for the *controllability* within its tree-based story structure. The encounters in the story tree are selected either by the player's own choices or the system's decision nodes that choose the encounter based on the state of player model. However, it is potentially possible to impose some trainer-specified constraints at the system's decision nodes so that the selection of encounters can be directed towards the trainer's desire. With regard to the *robustness*, PaSSAGE adopts the prevention approach, as it only allows a player to select a set of pre-defined choices of actions at some prescribed situations. The key strength of the PaSSAGE system is its ability to achieve the *personalization*. Similar to ISAT, the player model used in PaSSAGE also takes a *hypothetical approach*, as how the player's actions can reflect the play style and update the weight values based on the designer's hypotheses. For the *interaction*, even though a player in PaSSAGE can move around in the virtual environment, the key actions that a player can take are governed by the pre-defined story tree. The player can only choose from a pre-defined set of actions at certain stages as defined in the story tree.

3.8 Automated Story Director

The Automated Story Director (ASD) [3] developed by Mark Riedl et al. is an interactive narrative system that uses an *experience manager* to offer adaptive narratives in response to player's actions. The experience manager employs a generative approach, which can automatically generate variations of a narrative. It uses a planning-based narrative generator to produce alternative narratives (i.e., contingency plans) to deal with the inconsistencies caused by player's actions. The generative approach adopted by the ASD system helps to reduce the authoring burden, as it only requires an author to provide an exemplar narrative and a collection of plan operators for narrative authoring. The system has been applied to two prototype applications, one for entertainment purpose (i.e., an interactive Little Red Riding Hood story) and one for military training.

In ASD, the experience manager first starts to work on an *exemplar narrative*. This exemplar narrative is created by a human author and it is used as the narrative that the author or system expects the player will experience. To represent a narrative, a partially-ordered plan-based structure, similar to the one used in Mimesis [34], is used. Fig. 12 shows the

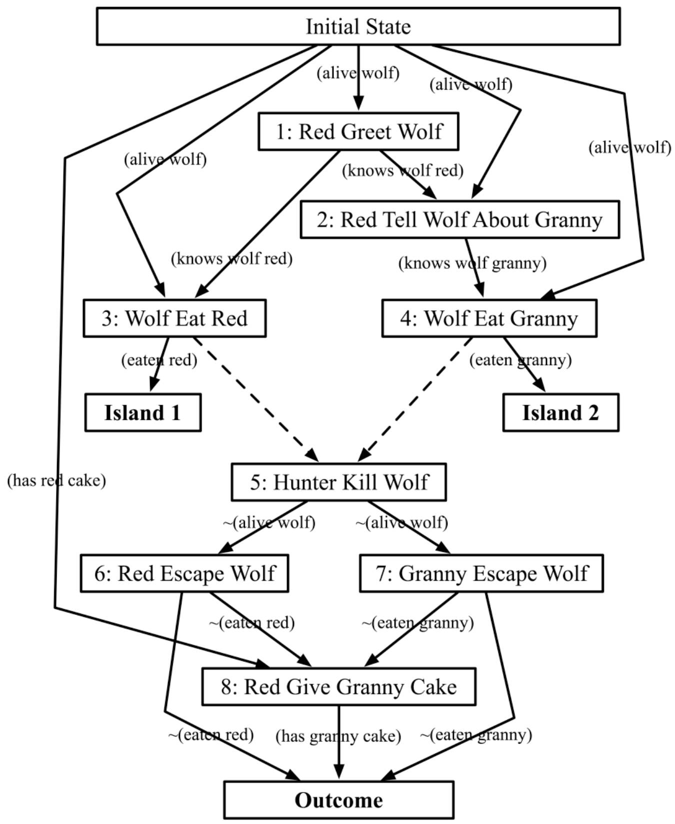

Fig. 12. Exemplar narrative plan in ASD(retrieved from [3])

plan structure of the exemplar narrative for the Little Red Riding Hood story. In such a plan structure, the boxes represent the plan steps, solid arrows are causal links and dashed arrows are temporal constraints. To ensure that the user (i.e., the author or trainer) can have the ability to control the possible narrative trajectories generated from such a partial-order plan, a special data structure called islands (as shown in Fig. 12), is used. The system allows user to inject these islands, which are the intermediate states that all narratives generated by a planner must pass. In the context of training, such islands can help to ensure that certain situations relevant to training (e.g., a tasking opportunity to practice a skill) will be presented to the trainee according to trainer's preferences.

The ASD system handles the player's actions by analyzing all the possible inconsistencies between narrative plan and the virtual world state that can occur due to the player's actions. Specifically, the system checks every inconsistency that can potentially threaten a causal link in the exemplar narrative plan. Once such a threatened causal link is detected, the system uses a tiered re-planning approach to search for the contingency plan to fix the broken link. The narrative re-planning process is conducted off-line to avoid the delays during the game-play. Given all the potentially threatened causal links, the re-planning will generate a tree of contingency plans, where each plan forms a branch of the initial exemplar narrative. During the game-play, the ASD system executes the narrative plan by generating directives to control the semi-automatous virtual characters in the story world.

The ASD system achieves the *controllability* through the use of islands in its partially-ordered plan narrative. The specification of islands allows a trainer to exercise hard control, as it ensures that the certain intermediate states (i.e., the specified islands) will be achieved during plan execution. The *robustness* of the system is realized by its re-planning mechanism to generate the contingency plans. It uses a reactive approach, as the contingency plans are invoked only when a player performs an action that threatens a causal link in the initial plan. Even though the system adapts its narrative (through invoking contingency plans) based on the player's actions, the adaptation is to preserve the desired properties as specified by the author or the trainer. The *personalization* of the system is not addressed, as the adaptation is not made to consider variation in the trainees. For the *interaction*, the player of the ASD system can control a virtual character and act as a role in the story world character. The player can perform a wide repertoire of actions at anytime during the game.

3.9 C-DraGer

The Case-based Drama manaGer (C-DraGer) developed by Manu Sharma et al. [40] is an interactive narrative system, which aims to generate narratives that can adapt to improve an individual player's experiences in an interactive fiction game. The system employs a case-based player model that leverages on the player feedback data to predict the interestingness of plot points in a story. Based on the built player model, a Drama Manager module is designed in C-DraGer to dynamically determine the next plot point that is best suited to a specific player. One salient feature of the C-DraGer system is that its player model is constructed based on real player's data rather than the empirically-derived models of player behaviors.

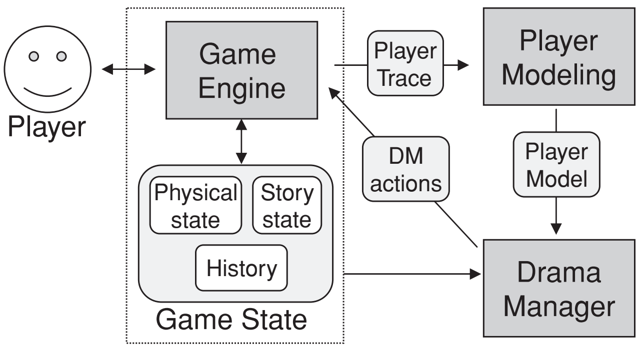

Fig. 13. Architecture of C-DraGer(retrieved from [40])

The architecture of the C-DraGer system is shown in Fig. 13. The architecture consists of three main modules, namely, the game engine, player modeling and drama manager. The game engine runs the narrative, executes player's actions and updates game states. In C-DraGer, a player interacts with the game engine through a text-based interface. The trace of the player's in-game actions are sent to the player modeling module (PMM). The PMM analyzes the player's trace to predict the interestingness of the candidate plot points. Based on the information obtained from PMM, the drama manager module (DMM) steers the narrative by sending DM actions to the game engine.

The player modeling module (PMM) in C-DraGer uses a case-based reasoning approach to construct the player model. The system first builds a case base by recording the past players' gameplay traces and their feedback on the interestingness of the plot points they experienced during the game. The feedback is obtained by asking the player to rank the experienced plot points and the whole story based on a 5-point Likert scale and give a confidence value on her/his ranking. Based on this recorded information, each case base is constructed and it contains the player trace, the interestingness values and confidence value of a past player. To predict the preference of the current player, the PMM compares the trace of current player with the ones of past players in the case base. The PMM selects the case with its player trace most similar to the current player's and uses the interestingness values in the matched case to predict the current player's interestingness values. The system assumes that the players with similar preferences (i.e., the story interestingness) tend to perform the similar actions in the game.

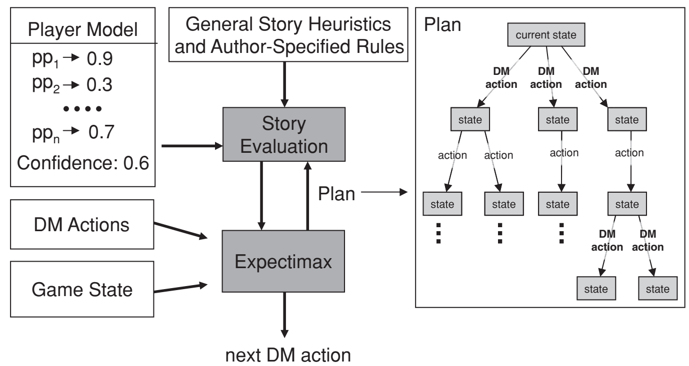

Fig. 14. Drama Manager Module in C-DraGer(retrieved from [40])

Based on the player model and the current game state, the drama manager module (DMM) in C-DraGer is designed to influence the story world by sending DM actions. In C-DraGer, a story is represented as a branching story graph as shown in Fig. 14 and DM actions can affect which plot points (i.e., states) will be selected given the current state of the game. To determine the next DM action to execute, the DMM uses an *expectimax* algorithm, which evaluates the story based on both the player interest and author-specified story guidelines. As shown in Fig. 14, the story evaluation takes the inputs of story heuristics and author-specified rules, which evaluate the story based on some desired properties (e.g., how much the story changes from one topic to another) as preferred by the author. The interestingness values obtained from the player model are then combined with the evaluation values based on author-specified heuristics and rules. The DM action, which gives the maximum combined value, will be selected for execution.

TABLE 1
Controllability in the nine interactive narrative systems

Systems	Mechanism for controllability
Mimesis	**Not supported.** Potentially applicable to use hard or soft constraints to direct the planning-based story generation.
IDA	**Not supported.** Story is pre-defined by author.
I-Storytelling	**Not supported.** The direction of story is only affected by dynamic interactions of virtual agents.
Façade	**Soft control.** A trainer can influence the story using dramatic arc.
Thespian	**Soft control.** A trainer can specify temporal and partial-order constraints.
ISAT	**Authoring.** Trainer has the role of author, which uses a story authoring tool to create the story space.
PaSSAGE	**Not supported.** Potentially applicable to impose trainer-specified constraints in the system decision nodes of tree-structure story
ASD	**Hard control.** Use islands to ensure certain trainer-specified plan states will be achieved in planning-based story generation.
C-DraGer	**Soft control.** Use author/trainer-specified heuristics and rules to influence the selection of plot points.

In terms of *controllability*, C-DraGer allows a trainer or author to explicitly specify the story heuristics and rules, which influence how the drama manager selects the plot points in the story. It is a type of soft control, as it affects the likelihood that certain types of plot points will appear in the generated story. The C-DraGer system uses a branching story graph representation, within which all the player actions are defined as the branches in the graph. Thus, a player can only choose the actions, which are pre-defined in the graph. Thus, the *robustness* of the C-DraGer system is achieved using the prevention approach. One salient feature of C-DraGer is the use of a data-driven player model to achieve the *personalization*. Compared to other hypothetical player models, C-DraGer does not need to make hypothesis on the mapping between player's actions and characteristics. It relies on the past players' play traces and ratings to infer the preferences of the current player. With regard to the *interaction*, C-DraGer has a text-based interface for player interaction, where a player is presented with a set of valid actions to choose at any given time in the game. The choice of player actions is constrained by the action choices as pre-defined in the branching story graph structure.

4 Functional Comparison

In the previous section, we have reviewed some influential existing work in interactive narrative. In this section, we will compare the nine reviewed systems based on the four proposed requirements that they support. Table 1 compares the controllability mechanisms of the nine reviewed systems. It can be observed that five systems provide an explicit support for controllability. Among these five systems, three systems (i.e., Façade, Thespian and C-DraGer) adopt the soft control approach, one system (i.e., ASD) adopts the hard control approach and one system (i.e., ISAT) provides an authoring tool to let trainer perform authoring task (strictly speaking, not controllability as described in section 2.1). In the rest of four systems that do not support controllability, two systems (i.e., Mimesis and PaSSAGE) are potentially applicable to enable controllability given their story representation and story generation mechanisms. Generally speaking, it is easier to enable controllability in the systems that use planning-based story generation. In such systems, a trainer can specify either hard or soft constraints to control the story planning process. For the systems (e.g., PaSSAGE and C-DraGer) that use a tree-based or branching story graph structure, a trainer can influence the progression of story if there exist some system decision nodes to determine the next

TABLE 2
Robustness in the nine interactive narrative systems

Systems	Mechanism for robustness
Mimesis	**Reactive approach.** Two strategies, namely accommodation and intervention, are used to react to the exceptional player actions.
IDA	**Predictive approach.** A director agent preemptively directs the story based on the predicted future player behaviors.
I-Storytelling	**Reactive approach.** Two mechanisms, namely situation reasoning and action repair, are used to ensure the executability of virtual agents' plans.
Façade	**Reactive approach.** Pre-defined mix-in goals in story beat are used to react player's behaviors.
Thespian	**Predictive approach.** A direct agent anticipates the possible violations of the prescribed story based on a model of player.
ISAT	**Reactive approach.** Reactive direction is used to prevent trainee going outside the training space.
PaSSAGE	**Prevention approach.** A player can only choose a set of pre-defined actions at some prescribed situations.
ASD	**Reactive approach.** An offline re-planning process is carried out to create a tree of contingency plans.
C-DraGer	**Prevention approach.** A player chooses the actions which are pre-defined in the branching story graph.

branch. It is relatively difficult to ensure controllability in the systems (e.g., I-Storytelling) in which the direction of story largely depends on the dynamic interactions of virtual agents.

Table 2 compares the robustness mechanisms of the nine reviewed systems. It can be found that all the systems provide some mechanisms for robustness, with the majority of them (i.e., five systems) adopting the reactive approaches. The popularity of the reactive approach may be due to the reason that it is an intuitive way to handle the undesired player behaviors and it can react to the changing situation in a prompt manner. However, it should be noted that most reactive approaches require the creation of contingency plans or strategies at the authoring stage. This may introduce additional authoring burden, especially when the story plot is complex. The prevention approach is usually used in the systems (e.g., PaSSAGE and C-DraGer) that employ a branching story structure and provide a pre-defined set of player actions at each branching point. Though the prevention approach is the most straightforward way to ensure robustness, it has the disadvantage of limiting the freedom of the player's interaction. Among all the reviewed systems, only IDA and Thespian adopt the predictive approach for robustness. One challenge in realizing the predictive approach is that it relies on an accurate estimation of the player's future behaviors.

Table 3 compares the personalization mechanisms of the nine reviewed systems. It can be observed that even though personalization is an important requirement for improving both entertainment and training value of interactive narrative, the work on personalization for interactive narrative is still limited. Among the nine reviewed systems, three systems (i.e., ISAT, PaSSAGE and C-DraGer) provide an explicit mechanism for personalization. Two of them (i.e., ISAT and PaSSAGE) adopt the hypothetical approach and one system (i.e., C-DraGer) adopts the data-driven approach. The hypothetical approach relies on the modeler's experiences to derive the mapping between player in-game actions and player's characteristics. Thus, it is relatively easy to construct the player model. In contrast, the data-driven approach requires a large amount of data of past players to construct the player model. Even though it may introduce some overheads, it is potentially more reliable as the model is constructed based on actual gameplay data. Note that even though a player model is introduced in the IDA and Thespian systems, it is not used for the purpose of generating personalized stories.

TABLE 3
Personalization in the nine interactive narrative systems

Systems	Mechanism for personalization
Mimesis	**Not supported.**
IDA	**Not supported.** Even though a player model is used to predict player behaviors, it is used to prevent the deviation from the pre-defined plot.
I-Storytelling	**Not supported.**
Façade	**Not supported.**
Thespian	**Not supported.**
ISAT	**Hypothetical approach.** A player skill model that estimates the proficiency of player's skills based on player's actions is used.
PaSSAGE	**Hypothetical approach.** A player model that infers the player's play style based on player's actions is used.
ASD	**Not supported.**
C-DraGer	**Data-driven approach.** A player preference model that is constructed from the past player's game traces and feedback is used.

TABLE 4
Interaction in the nine interactive narrative systems

Systems	Interaction Mode	Time	Choice
Mimesis	First-person. Control a virtual character	Anytime	All available actions
IDA	First-person. Play a role of ghost	Anytime	All available actions
I-Storytelling	Inspector. Manipulate objects and give advice to virtual agents	Anytime	All available actions
Façade	First-person. Speak to virtual agents via NLP	Anytime	Unlimited
Thespian	First-person. Speak to virtual agents and make gestures	Anytime	Unlimited
ISAT	First-person. Control a virtual character	Specific time. When a casualty event is presented	All available actions
PaSSAGE	First-person. Control a virtual character	Specific time. When a game event is encountered	A set of actions. Defined in branching story graph
ASD	First-person. Control a virtual character	Anytime	All available actions
C-DraGer	First-person. Interact using a text-based interface	Anytime	A set of actions. Defined in branching story graph

Table 4 summarizes the interaction mechanisms of the nine reviewed systems in terms of interaction mode, time and choice. It can be observed that most of the reviewed systems (i.e., eight systems) have a first-person interaction mode and allow a player to control a virtual character in the story. In the I-Storytelling system, the player is not involved in the actual story. But, she/he can influence the behaviors of virtual agents in an "off-stage" manner. All systems, except ISAT and PaSSAGE, also allow the player to interact with the system at anytime during the game, In ISAT and PaSSAGE, a player can issue an action only when certain story events are presented. In terms of the action choices, Façade and Thespian systems do not limit the actions a player can perform. A player can type any dialog and the system will handle the player's input using NLP. The rest of the systems provide a player a set of actions to select from. Among these systems, the PaSSAGE and C-DraGer only allows a player to select a pre-defined sub-set of actions as defined in the branching story graph. In other systems, a player can choose from the set of all available actions.

5 Conclusion

With the advancement of new media, AI and agent technologies, recent years have witnessed an increasing interest in interactive narrative from researchers, educators and entertainment industries, with many papers being published and many projects being conducted. Although some significant work has been done in designing interactive narrative systems, it is still a relatively young research area. In this paper, we review the existing interactive narrative systems from a training perspective. The objective is to gain understanding of the readiness of existing interactive narrative technologies for simulation-based training. To this end, we first identified some key requirements in developing interactive narrative systems for training. Then, we described some representative existing work and discussed the architecture design, major features and related mechanisms of these systems. Based on our review of the existing work, we made a comparison based on their functionalities and mechanisms to support the key requirements we identified. We hope that the review and assessment could provide readers with useful information and insights on the state of the art of the interactive narrative technologies as well as their applicability to training.

The primary application of interactive narrative system is oriented towards digital entertainment and games. Recently, there is an emerging trend to apply interactive narrative technologies for training [3], [17], [22], [23]. In fact, it has been observed by the stakeholders and government agencies that there is a need to provide a more intuitive and interactive way to train or educate the personnel with relevant skill sets. The interactive narrative systems and technologies have shown great potential to facilitate such interactive training and education. In our future work, we are particularly interested in the investigation of how to apply interactive narrative technologies to the scenario generation process in virtual training systems. Despite the applicability of interactive narrative technologies in training, there is still a gap between what existing systems can offer and meet the requirements imposed by the training applications. Based on our review in this paper, it can be observed that no single interactive narrative system has provided a full solution to satisfy all the requirement for training. To conclude this paper, we provide our observations on the key research issues for developing interactive narrative systems for training.

Trainer controllability. One essential requirement for a training system is that the trainer should be able to easily set up the training scenarios or narratives, based on the training objectives. This requires the interactive narrative system of training application to offer a high degree of controllability to let the trainer interact and direct the scenario set-up process. The trainer should be given the capability to direct the progression of the storyline, so that the generated scenario could reflect the training objectives the trainer aims to achieve. Currently, the issue of controllability is less explored in interactive narrative research. While some existing systems have provided some mechanisms for controllability, they tend to control the story for achieving some dramatic requirements (e.g., following the dramatic arc in Façade). Such control does not necessarily imply that it is a story complying to a desired training objective that a trainer aims to achieve. Therefore, when applying interactive narrative technologies to training applications, the research on trainer controllability still needs further investigation.

Trainee adaptation. Another key aspect for developing a training system is the adaptation to trainee (i.e., player). The training process will be effective only when the scenarios or narratives presented to a trainee are neither too difficult nor too easy, based on the trainee's existing skill levels. To this end, it is essential to have a player model to learn and capture the differences of individual trainees, so that the training system can adapt the scenarios accordingly. Among the reviewed systems, there are only three systems (i.e., ISAT, PaSSAGE and C-DraGer) that incorporate player models for the purpose of personalization and adaptation. Further research is needed to study how to construct the player model to accurately capture the trainee's characteristics and how to utilize the constructed player model for adapting the narratives in training.

Story variation and replayablility. The variation and replayablility of the scenario, or narrative, play an important role in building an effective training system. If a trainee practices with the repeated scenario multiple times, she/he tends to memorize the scenario, instead of learning the underlying concepts. Thus, an enriching learning experience for the trainee cannot be achieved if the number of available scenarios is limited. Thanks to the capability of generating story variation, the interactive narrative technology has a great potential to offer story variation and hence offer replayablility. However, the research on the assessment of the quality of the story variants still requires further investigation. To this end, we envisage two levels of story variations need to be achieved. The first type of variation is at a strategic level. This type of variation should reflect the different training objectives and intensities of these objectives to be trained. The second type is at an action level. This type of variation may exercise the same set of training objectives, but they aim to enhance the replayablility of the system. Example of such variation is to place an enemy in different, but strategically-similar locations.

References

[1] R. Hill, J. Gratch, S. Marsella, J. Rickel, W. Swartout, and D. Traum, "Virtual humans in the mission rehearsal exercise system," *Künstliche Intelligenz*, vol. 4, no. 03, pp. 5–10, 2003.

[2] B. Magerko, B. Stensrud, and L. Holt, "Bringing the schoolhouse inside the box-a tool for engaging, individualized training," in *Proceedings of the 25th Army Science Conference*, 2006.

[3] M. O. Riedl, A. Stern, D. Dini, and J. Alderman, "Dynamic experience management in virtual worlds for entertainment, education, and training," *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning*, vol. 4, no. 2, pp. 23–42, 2008.

[4] P. M. Kato, "Video games in health care: Closing the gap," *Review of General Psychology*, vol. 14, no. 2, p. 113, 2010.

[5] J. A. Sokolowski, C. M. Banks, and P. Hakim, "Simulation training to improve blood management: an approach to globalizing instruction in patient safety," *Simulation*, vol. 90, no. 2, pp. 133–142, 2012.

[6] J. Dequidt, H. Courtecuisse, O. Comas, J. Allard, C. Duriez, S. Cotin, É. Dumortier, O. Wavreille, and J.-F. Rouland, "Computer-based training system for cataract surgery," *Simulation*, vol. 89, no. 12, pp. 1421–1435, 2013.

[7] G. Bausch, J. Adermann, B. Andrack, M. Dengl, J. Handwerk, M. Müller, A. Seifert, H. Steinke, M. Sturm, W. Korb *et al.*, "Design and development process of a next-generation training system for spinal surgery," *Simulation*, vol. 89, no. 12, pp. 1436–1441, 2013.

[8] A. J. Faria, D. Hutchinson, W. J. Wellington, and S. Gold, "Developments in business gaming a review of the past 40 years," *Simulation & Gaming*, vol. 40, no. 4, pp. 464–487, 2009.

[9] D.-J. Van Der Zee, B. Holkenborg, and S. Robinson, "Conceptual modeling for simulation-based serious gaming," *Decision Support Systems*, vol. 54, no. 1, pp. 33–45, 2012.

[10] C. Cleophas, "Designing serious games for revenue management training and strategy development," in *Proceedings of the Winter Simulation Conference*, 2012, pp. 140:1–140:12.

[11] A. G. Bruzzone, M. Massei, A. O. Solis, S. Poggi, C. Bartolucci, and L. D. Capponi, "Serious games as enablers for training and education on operations on ships and off-shore platforms," in *Proceedings of the 2013 Summer Computer Simulation Conference*, 2013, pp. 36:1–36:8.

[12] W. Johnson, J. Rickel, and J. Lester, "Animated pedagogical agents: Face-to-face interaction in interactive learning environments," *International Journal of Artificial Intelligence in Education*, vol. 11, no. 1, pp. 47–78, 2000.

[13] M. Core, D. Traum, H. C. Lane, W. Swartout, J. Gratch, M. Van Lent, and S. Marsella, "Teaching negotiation skills through practice and reflection with virtual humans," *Simulation*, vol. 82, no. 11, pp. 685–701, 2006.

[14] A. Meluso, M. Zheng, H. Spires, and J. Lester, "Enhancing 5th graders' science content knowledge and self-efficacy through game-based learning," *Computers & Education*, vol. 59, no. 2, pp. 497–504, 2012.

[15] H. Lukosch, T. van Ruijven, and A. Verbraeck, "The participatory design of a simulation training game," in *Proceedings of the Winter Simulation Conference*, 2012, pp. 142:1–142:11.

[16] J. Caussanel, N. Giambiasi, and A. G. Bruzzone, "From abstract representation to formal modeling of tactical military operations," in *Proceedings of the 2007 Summer Computer Simulation Conference*, 2007, pp. 1094–1100.

[17] A. Zook, S. Lee-Urban, M. O. Riedl, H. K. Holden, R. A. Sottilare, and K. W. Brawner, "Automated scenario generation: Toward tailored and optimized military training in virtual environments," in *Proceedings of the International Conference on the Foundations of Digital Games*, 2012, pp. 164–171.

[18] M. Mateas and A. Stern, "Façade: An experiment in building a fully-realized interactive drama," in *Game Developers Conference*, 2003, pp. 4–8.

[19] B. Magerko, "Evaluating preemptive story direction in the interactive drama architecture," *Journal of Game Development*, vol. 2, no. 3, pp. 25–52, 2007.

[20] M. Cavazza, F. Charles, and S. J. Mead, "Character-based interactive storytelling," *IEEE Intelligent Systems*, vol. 17, pp. 17–24, 2002.

[21] N. Vannini, S. Enz, M. Sapouna, D. Wolke, S. Wat-

son, S. Woods, K. Dautenhahn, L. Hall, A. Paiva, E. André *et al.*, "FearNot!: a computer-based anti-bullying-programme designed to foster peer intervention," *European journal of psychology of education*, vol. 26, no. 1, pp. 21–44, 2011.

[22] B. Magerko, B. Wray, L. Holt, and B. Stensrud, "Improving interactive training through individualized content and increased engagement," in *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference*, 2005.

[23] M. Ponder, B. Herbelin, T. Molet, S. Schertenlieb, B. Ulicny, G. Papagiannakis, N. Magnenat-Thalmann, and D. Thalmann, "Immersive VR decision training: telling interactive stories featuring advanced virtual human simulation technologies," in *Proceedings of the workshop on Virtual environments*, 2003, pp. 97–106.

[24] J. H. Murray, *Hamlet on the Holodeck: The future of narrative in cyberspace*. MIT Press, 1998.

[25] M. O. Riedl, "An intent-driven planner for multi-agent story generation," in *Proceedings of the third international joint conference on Autonomous Agents and Multi-Agent Systems*, 2004, pp. 186–193.

[26] J. Porteous and M. Cavazza, "Controlling narrative generation with planning trajectories: the role of constraints," in *Proceedings of the 2nd Joint International Conference on Interactive Digital Storytelling: Interactive Storytelling*, 2009, pp. 234–245.

[27] R. M. Young, "Notes on the use of plan structures in the creation of interactive plot," in *AAAI Fall Symposium on Narrative Intelligence*, 1999, pp. 164–167.

[28] M. Si, S. C. Marsella, and D. V. Pynadath, "Thespian: Using multi-agent fitting to craft interactive drama," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 2005, pp. 21–28.

[29] Y. Cai, C. Miao, A.-H. Tan, and Z. Shen, "A hybrid of plot-based and character-based interactive storytelling," in *Proceedings of the 2nd international conference on Technologies for e-learning and digital entertainment*, 2007, pp. 260–273.

[30] A. B. Loyall, "Believable agents: Building interactive personalities," Ph.D. dissertation, Carnegie Mellon University, 1997.

[31] R. Aylett, J. Dias, and A. Paiva, "An affectively driven planner for synthetic characters," in *Proceedings of the 16th International Conference on Automated Planning and Scheduling*, 2006, pp. 2–10.

[32] F. Charles, M. Lozano, S. J. Mead, A. F. Bisquerra, and M. Cavazza, "Planning formalisms and authoring in interactive storytelling," in *Proceedings of the first International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 2003, pp. 216–225.

[33] B. Magerko and J. Laird, "Building an interactive drama architecture," in *Proceedings of First International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 2003, pp. 226–237.

[34] R. M. Young, M. O. Riedl, M. Branly, A. Jhala, R. J. Martin, and C. J. Saretto, "An architecture for integrating plan-based behavior generation with interactive game environments," *Journal of Game Development*, vol. 1, pp. 51–70, 2004.

[35] D. L. Roberts and C. L. Isbell, "A survey and qualitative analysis of recent advances in drama management," *International Transactions on Systems Science and Applications*, vol. 4, no. 2, pp. 61–75, 2008.

[36] M. O. Riedl and V. Bulitko, "Interactive narrative: An intelligent systems approach," *AI Magazine*, vol. 34, no. 1, pp. 67–77, 2013.

[37] B. Magerko, "Player modeling in the interactive drama architecture," Ph.D. dissertation, University of Michigan, 2006.

[38] M. O. Riedl and R. M. Young, "From linear story generation to branching story graphs," *Computer Graphics and Applications, IEEE*, vol. 26, no. 3, pp. 23–31, 2006.

[39] D. Thue, V. Bulitko, M. Spetch, and E. Wasylishen, "Interactive storytelling: A player modelling approach," in *Proceedings of the 3rd Artificial Intelligence and Interactive Digital Entertainment conference*, 2007, pp. 43–48.

[40] M. Sharma, S. Ontañón, M. Mehta, and A. Ram, "Drama management and player modeling for interactive fiction games," *Computational Intelligence*, vol. 26, no. 2, pp. 183–211, 2010.

[41] M. Cavazza, R. Aylett, K. Dautenhahn, C. Fencott, and F. Charles, "Interactive storytelling in virtual environments: Building the "Holodeck"," in *6th International Conference on Virtual Systems and Multi-media (VSMM 2000)*, 2000, pp. 678–687.

[42] J. R. Meehan, "The Metanovel: writing stories by computer." Ph.D. dissertation, Yale University, 1976.

[43] M. Lebowitz, "Creating characters in a story-telling universe," *Poetics*, vol. 13, no. 3, pp. 171–194, 1984.

[44] S. R. Turner, *The creative process: A computer model of storytelling and creativity*. Psychology Press, 1994.

[45] M. O. Riedl, C. J. Saretto, and R. M. Young, "Managing interaction between users and agents in a multi-agent storytelling environment," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 2003, pp. 741–748.

[46] R. M. Young, "An overview of the mimesis architecture: Integrating intelligent narrative control into an existing gaming envonment," in *Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, 2001, pp. 78–81.

[47] R. M. Young and M. O. Riedl, "Towards an architecture for intelligent control of narrative in interactive virtual worlds," in *Proceedings of the International Conference on Intelligent User Interfaces*, 2003, pp. 301–302.

[48] J. E. Laird, A. Newell, and P. S. Rosenbloom, "Soar: An architecture for general intelligence," *Artificial intelligence*, vol. 33, no. 1, pp. 1–64, 1987.

[49] M. Cavazza, F. Charles, and S. J. Mead, "Interacting with virtual characters in interactive storytelling," in *Proceedings of the first international joint conference on Autonomous Agents and Multiagent Systems*, 2002, pp. 318–325.

[50] D. S. Nau, S. J. J. Smith, and K. Erol, "Control strategies in htn planning: Theory versus practice," in *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, 1998, pp. 1127–1133.

[51] M. Meteas and A. Stern, "Structuring content in the façade interactive drama architecture," in *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference*, 2005, pp. 93–98.

[52] M. Mateas, "Interactive drama, art and artificial intelligence," Ph.D. dissertation, Carnegie Mellon University, 2002.

[53] M. Si, "Thespian: A decision-theoretic framework for interactive narratives," Ph.D. dissertation, University of Southern California, 2010.

[54] M. Si, S. C. Marsella, and D. V. Pynadath, "THESPIAN: An architecture for interactive pedagogical drama." in

Proceedings of the 12th International Conference on Artificial Intelligence in Education, 2005, pp. 595–602.
[55] D. V. Pynadath and S. C. Marsella, "PsychSim: Modeling theory of mind with decision-theoretic agents," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2005, pp. 1181–1186.
[56] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable markov processes over a finite horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, 1973.
[57] M. Si, S. C. Marsella, and D. V. Pynadath, "Proactive authoring for interactive drama: An authors assistant," in *Proceedings of the 7th international conference on Intelligent Virtual Agents*, 2007, pp. 225–237.
[58] B. Medler and B. Magerko, "Scribe: A tool for authoring event driven interactive drama," *Technologies for Interactive Digital Storytelling and Entertainment*, pp. 139–150, 2006.