Towards an Accessible Interface for Story World Building

Steven Poulakos* **Mubbasir Kapadia†** **Andrea Schüpfer‡**
Fabio Zünd‡ **Robert W. Sumner*‡** **Markus Gross*‡**

*Disney Research Zurich, Switzerland
†Rutgers University, New Jersey, USA
‡ETH Zurich, Switzerland

Abstract

In order to use computational intelligence for automated narrative synthesis, domain knowledge of the story world must be defined, a task which is currently confined to experts. This paper discusses the benefits and tradeoffs between *agent-centric* and *event-centric* approaches towards authoring the domain knowledge of story worlds. In an effort to democratize story world creation, we present an accessible graphical platform for content creators and even end users to create their own story worlds, populate it with smart characters and objects, and define narrative events that can be used by existing tools for automated narrative synthesis. We demonstrate the potential of our system by authoring a simple *bank robbery* story world, and integrate it with existing solutions for event-centric planning to synthesize example digital stories.

Introduction

A prerequisite to using computational intelligence for automated narrative synthesis (Riedl and Bulitko 2013) is defining the *domain knowledge* of the story world in which you want to author narratives. Domain knowledge includes annotated semantics that characterize the attributes and relationships of objects and characters in a scene (state), different ways in which they interact (affordances), and how these affordances manipulate their state. The domain knowledge definition entails both the state and action space within a story world. Current languages and interfaces for specifying domain knowledge are confined to experts and the overhead of domain specification is high, often comparable to authoring the story from scratch. In order to democratize the use of computational narrative intelligence, the research community must first provide an accessible interface for building story worlds, and this is the focus of this paper.

Current approaches to automated narrative synthesis use logical planners, such as STRIPS-like formalisms (Fikes and Nilsson 1971), for building the state and action space of individual characters in the story. The Planning Domain Definition Language (PDDL) (Mcdermott et al. 1998) provides a common formalism for describing planning domains with clear semantics. Although these tools are powerful, they are hard to use and restricted to experts. We want to democratize story world creation, enabling both expert and novice users to construct a space for compelling narrative content.

In this paper, we explore two different metaphors for story world building. Agent-centric authoring (Riedl and Bulitko 2013) defines the traits of each individual character or object in the story world. Authoring the characteristics and capabilities of individual characters is decoupled from the specifying the story itself, and the complexity of automated narrative synthesis is combinatorial in the number of characters and the different capabilities of each character. Event-centric authoring (Shoulson et al. 2013) encapsulates interactions that have narrative significance as logical constructs and provides an appropriate level of abstraction for authoring and reasoning about stories. This imposes an additional authoring overhead, as these events need to be specified, but mitigates the complexity of automation as it is now independent in number of actors and actor capabilities. Automated approaches explore the space of events to generate stories. We present the benefits and trade-offs between these two metaphors and motivate an event-centric approach for constructing story worlds. To this end, we introduce a graphical platform for event-centric authoring of story worlds and demonstrate its potential with preliminary results.

Related Work

The research community has addressed the problem of authoring interactive narratives in two main ways. Manual approaches provide domain specialists with complete control over creating rich narrative content, while automated approaches rely on computational techniques to generate emergent interactive experiences. We provide a brief review below, which builds on comprehensive surveys of narrative authoring (Riedl and Bulitko 2013; Kapadia et al. 2013).

Manual Authoring. Scripted approaches (Loyall 1997) describe behaviors as pre-defined sequences of actions. While providing fine-grained control, small changes often require far-reaching modifications of monolithic scripts. Improv (Perlin and Goldberg 1996) and LIVE (Menou 2001) describe actor behaviors as rules based on certain conditions. These systems produce pre-defined behaviors appropriate for specific situations. However, they are not designed to generate complicated agent interactions with narrative significance. Facade (Mateas and Stern 2003) utilizes authored beats to manage the intensity of the story in addition to

a generalized scripting language (Mateas and Stern 2002; 2004) to manually authoring character interactions based on preconditions for successful execution.

Story Graphs can represent branching story lines (Gordon et al. 2004) enabling user interaction as discrete choices at key points in the authored narrative. Behavior Trees (BT's) are applied in the computer gaming industry to design the artificial intelligence logic for non-player characters (Isla 2005). BT's enable the authoring of modular and extensible behaviors, which can be extended to control multiple interacting characters (Shoulson et al. 2014) and to provide a formalism for specifying narrative events.

Automated Narrative Synthesis. Domain-independent planners (Sacerdoti 1975) provide a promising direction for automated narrative synthesis, however, at the cost of requiring the specification of domain knowledge. The complexity of authoring is transferred from story specification to domain specification. For example, domain specification has been demonstrated to enable multi-actor interactions that conform to narrative constraints (Kapadia et al. 2011b; 2011a). However, they cannot be dynamically changed to accommodate user input. Narrative mediation systems (Riedl and Young 2006) automatically synthesize sub stories that consider the ramifications of possible user interaction. However, these systems produce story graphs with significant branching that are difficult to edit by humans. Virtual directors or drama managers (Magerko et al. 2004) may also accommodate user input while steering agents towards pre-determined narrative goals (Weyhrauch 1997). Thespian (Si, Marsella, and Pynadath 2005) uses social awareness to guide decision-theoretic agents. PaSSAGE (Thue et al. 2007) estimates a player's ideal experience to guide the player through predefined encounters.

Agent-Centric Domain Specification. Agent-centric approaches (Riedl and Bulitko 2013) build up each character as an individual and explore the space of all possible combinatorial character actions to synthesize stories. Authoring the characteristics and capabilities of individual characters is decoupled from specifying the story itself, and the complexity of automated narrative synthesis is combinatorial in the number of characters and the different capabilities of each character.

Event-Centric Domain Specification. Events are a layer of abstraction on top of agent-centric authoring which encapsulate complex multi-actor interactions that have narrative significance. Event-centric approaches (Shoulson et al. 2013; Shoulson, Kapadia, and Badler 2013) plan in the space of pre-authored narratively significant interactions, thus mitigating the combinatorial explosion of planning in the action space of individual character actions.

Automatic Domain Learning. Recent work (Li, Lee-Urban, and Riedl 2013) has shown the promise of using crowdsourcing and machine intelligence for automatically learning domains for automated story synthesis. These approaches greatly minimize the burden of domain specification, while sacrificing authoring precision.

Comparison to Prior Work. Our work complements ongoing research in computational narrative intelligence and advocates the need for providing accessible metaphors for building story worlds, in an effort to democratize story authoring for the masses. To this end, we discuss the benefits and limitations of agent-centric and event-centric authoring paradigms and present our ongoing work towards providing a graphical platform for end users to design their own story worlds, for authoring compelling digital stories.

Domain Specification for Automated Narrative Synethesis

In order to use computational intelligence for narrative synthesis, content creators and story writers need to specify the domain knowledge of the story world, which can be used by an intelligent system for reasoning, inference, and ultimately story synthesis. This includes annotating semantics that characterize the attributes and relationships of objects and characters in the scene (state), different ways in which they interact (affordances), and how these affordances manipulate their state. Many intelligent systems for automated synthesis are similar in this regard (Riedl and Bulitko 2013). However there exists a tradeoff between the complexity of authoring and the computational complexity of generating stories depending on type of domain representation used. The rest of this section first introduces some preliminary concepts for story world building. Using these building blocks, we describe two standard representations of domain knowledge and discuss their impact on automated narrative synthesis.

Preliminaries

We introduce smart objects and affordances as the building blocks for creating story worlds.

Smart Objects. The virtual world $\mathbf{W}$ consists of smart objects (Kallmann and Thalmann 1999) with embedded information about how an actor can use the object. We define a smart object $w = \langle \mathbf{F}, s \rangle$ with a set of advertised affordances $f \in \mathbf{F}$ and a state $s = \langle \theta, R \rangle$, which comprises a set of attribute mappings θ, and a collection of pairwise relationships R with all other smart objects in $\mathbf{W}$. An attribute $\theta(i, j)$ is a bit that denotes the value of the j^{th} attribute for w_i. Attributes are used to identify immutable properties of a smart object such as its role (e.g., a button or a bank robber) which never changes, or dynamic properties (e.g., `IsPressed` or `IsStanding`) which may change during the story. A specific relationship R_a is a sparse matrix of $|\mathbf{W}| \times |\mathbf{W}|$, where $R_a(i, j)$ is a bit that denotes the current value of the a^{th} relationship between w_i and w_j. For example, an `IsFriendOf` relationship indicates that w_i is a friend of w_j. Note that relationships may not be symmetric, $R_a(i, j) \neq R_a(j, i)\ \forall\ (i, j) \in |\mathbf{W}| \times |\mathbf{W}|$. The state of each smart object is stored as a bit vector encoding both attributes and relationships.

Affordances. An affordance $f = \langle w_o, \mathbf{w}_u, \mathbf{\Phi}, \Omega \rangle$ is an advertised capability offered by a smart object that takes the owner of that affordance w_o and one or more smart object users $\mathbf{w}_u$, and manipulates their states. For example, a smart object such as a ball can advertise a *Throw* affordance, allowing another smart object to throw it. A precondition $\mathbf{\Phi} : \mathbf{s_w} \leftarrow \{\texttt{TRUE}, \texttt{FALSE}\}$ is an expression in conjunctive normal form on the compound state $\mathbf{s_w}$ of $\mathbf{w} : \{w_o, \mathbf{w}_u\}$ that checks if f can be executed based on their current states. A precondition is fulfilled by $\mathbf{w}$ if $\mathbf{\Phi}_f(\mathbf{w}) = \texttt{TRUE}$. The postcondition $\Omega : \mathbf{s} \rightarrow \mathbf{s}'$ transforms the current state of all participants, $\mathbf{s}$ to $\mathbf{s}'$ by executing the effects of the affordance. When an affordance fails, $\mathbf{s}' = \mathbf{s}$.

Narrative Synthesis. The aim of narrative synthesis is to generate a narrative $\Pi(\mathbf{s}_s, \mathbf{s}_g)$, which satisfies an initial state $\mathbf{s}_s$ and through a series of state transitions results in the desired goal state $\mathbf{s}_g$.

Agent-centric Domain Knowledge

Following the definition of smart objects and affordances described above, we can define a domain $\Sigma = \langle \mathbb{S}, \mathbb{A} \rangle$ which includes the definition of the state space $\mathbb{S}$ of all characters and objects, and the action space $\mathbb{A}$, or the space of all permissible actions and interactions in this story world.

State Space. The overall state of the world $\mathbf{W}$ is defined as the compound state $\mathbf{s} = \{s_1, s_2 \cdots s_{|\mathbf{W}|}\}$ of all smart objects $w \in \mathbf{W}$, which is encoded as a matrix of bit vectors. $\mathbf{s_w}$ denotes the compound state of a set of of smart objects $\mathbf{w} \subseteq \mathbf{W}$. The state space $\mathbb{S}_a$ represents the set of all possible world states $\mathbf{s}$.

Action Space. The agent-centric action space $\mathbb{A}_a = \mathbf{F}_1 \times \mathbf{F}_2 \times \cdots \times \mathbf{F}_{|\mathbf{W}|}$ is the cross product of the set of all affordances of each smart object in the world.

Narrative Synthesis. Automated approaches explore the space of all permissible character actions to generate a narrative $\Pi(\mathbf{s}_s, \mathbf{s}_g) = \langle f_1, f_2 \dots f_n \rangle$ that is a sequence of actions and interactions between story characters that satisfies the desired story outcome, $\mathbf{s}_g$. Agent-centric domain representations have two main drawbacks. First, the computational complexity of automation scales exponentially in the number of characters and the set of affordances of each character. This limits story worlds to a small set of characters with limited degree of interactions. Second, the act of specifying the state and capabilities of characters is decoupled from the story itself, and it can be harder to maintain narrative coherence.

Event-centric Domain Knowledge

Event-centric domains introduce events as an additional layer of abstraction. Events are pre-defined context-specific interactions between any number of participating smart objects whose outcome is dictated by the current state of its participants. Events serve as the building blocks for authoring complex narratives. An event is formally defined as $e = \langle t, \mathbf{r}, \mathbf{\Phi}, \Omega \rangle$ where t is a logical representation of a coordinated interaction between multiple actors. t takes any number of participating smart objects as parameters where $\mathbf{r} = \{r_i\}$ define the desired roles for each participant. r_i is a logical formula specifying the desired value of the immutable attributes $\theta(\cdot, j)$ for w_j to be considered as a valid candidate for that particular role in the event. A precondition $\mathbf{\Phi} : \mathbf{s_w} \leftarrow \{\texttt{TRUE}, \texttt{FALSE}\}$ is a logical expression on the compound state $\mathbf{s_w}$ of a particular set of smart objects $\mathbf{w} : \{w_1, w_2, \dots w_{|\mathbf{r}|}\}$ that checks the validity of the states of each smart object. $\mathbf{\Phi}$ is represented as a conjunction of clauses $\phi \in \mathbf{\Phi}$ where each clause ϕ is a literal that specifies the desired attributes of smart objects, relationships, as well as rules between pairs of participants. A precondition is fulfilled by $\mathbf{w} \subseteq \mathbf{W}$ if $\mathbf{\Phi}_e(\mathbf{w}) = \texttt{TRUE}$. The event postcondition $\Omega : \mathbf{s} \rightarrow \mathbf{s}'$ transforms the current state of all event participants $\mathbf{s}$ to $\mathbf{s}'$ by executing the effects of the event. When an event fails, $\mathbf{s}' = \mathbf{s}$. An event instance $I = \langle e, \mathbf{w} \rangle$ is an event e populated with an ordered list of smart object participants $\mathbf{w}$.

State Space. The event-centric state space $\mathbb{S}_e$ is equivalent to $\mathbb{S}_a$.

Action Space. The event-centric action space $\mathbb{A}_e = \{e_1, e_2 \cdots e_m\}$ is defined as the set of all m events that may occur between any permutation of smart objects in the world $\mathbf{W}$.

Narrative Synthesis. A narrative $\Pi(\mathbf{s}_s, \mathbf{s}_g) = \langle e_1, e_2 \dots e_n \rangle$ is defined as a sequence of events that transform the state of the world from its initial state $\mathbf{s}_s$ to the desired goal state $\mathbf{s}_g$ that represents the desired outcome of the narrative. An event-centric representation of domain knowledge helps mitigate the combinatorial complexity of authoring individual characters in complex multi-character interactions and its variants have recently gained prominence in the games industry. However, this does impose the additional overhead of authoring events. For example, the work in (Shoulson et al. 2013) represents events using Parameterized Behavior Trees (Shoulson et al. 2011) and uses a total-order planner (Pearl 1984) to generate narratives as a sequence of events.

A Graphical Platform for Building Story Worlds

Our story world builder is designed to build up components of a full story world with required semantics to achieve automated narrative synthesis. Our graphical platform is built within the Unity3D game engine. Following the event-centric representation (Shoulson et al. 2013), events are defined as Parameterized Behavior Trees (Shoulson et al. 2011) which provide a graphical, hierarchical representation for specifying complex multi-actor interactions in a modular, extensible fashion. We use NodeCanvas for authoring Behavior Trees. To demonstrate the benefits of our system, we integrate the planner described in (Shoulson et al. 2013) to generate sample narratives that can be synthesized within the story worlds that are created using our platform. Our un-

derlying representation of the story world is general and can be easily used within other computational narrative synthesis systems.

Building a new story world entails three main steps, which are described below: (1) Story World Creation: defining all possible states and relationships in the world in addition to configuring the scene. (2) Smart Object Creation: defining a set of smart objects and characters, and instantiating them in the scene. (3) Event Creation: defining a lexicon of events for creating stories. Figure 1 illustrates the main steps of building a story world and Figure 2 illustrates an example story that was generated within a bank robbery scenario, that was authored using our platform.

Story World Creation

The first part of world creation is to instantiate a world with all of the basic functionality as well as the definition of all possible states and relationships available in the world. States include high level descriptions of objects. For example, a bank vault smart object is locked, or a smart character has a role of robber. Relationships can be defined. For example, two robbers may be allied to each other.

The second part of world creation involves setting up an environment instance (or scene) within the world. In our example, we define a bank within a city. Lights are added to the scene and navigation functionality is configured using NavMesh in Unity3D. Additional components are attached to the scene as required by an automated story planner.

Smart Object Creation

Although it is not necessary, we differentiate between smart objects and smart characters. The smart characters require additional components to enable additional functionality. For example, a component for inverse kinematics enables a character to do complex physical actions, such as pressing a button and grasping objects.

Figure 3 shows the interface for creating a smart object. The user chooses a model (of type FBX in Unity3D) and selects affordances to associate with that model. Initial states of the smart object are defined from those available in the world. For example, a bank robber is created in Figure 3 with the states `isStanding` and `HoldingWeapon` set to true. We also assign the role state of `RoleRobber` to true. We also add a relation `isAllied` with another character. The initial state of an instance of the robber will be standing, holding a weapon and allied with another character.

In a second step of smart object creation, the smart object can be instantiated multiple times. This part includes selecting a representative icon, setting the initial position and defining the instance of a relation to another smart object (or character). For example, two robbers are instantiated and allied with each other. Figure 4 shows the instantiation of a bank robber.

Event Creation

There are two steps for event creation: (1) desiging parameterized behavior trees that describe interactions between multiple smart objects and characters in the scene and, (2)

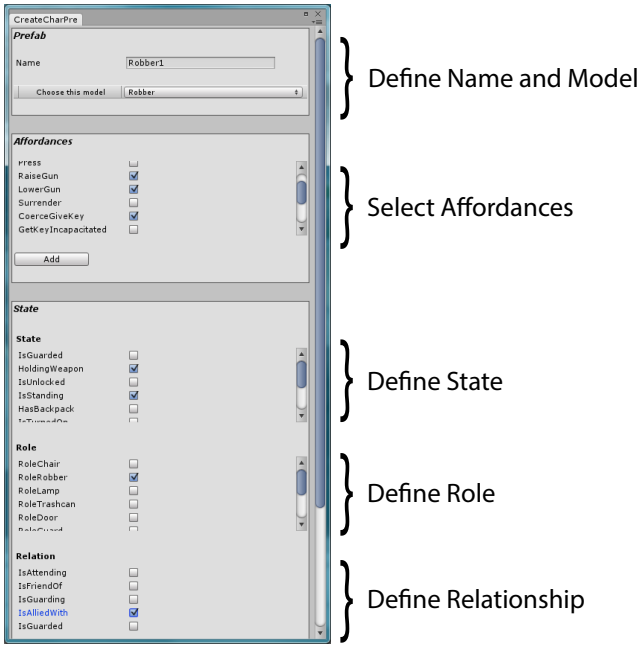


Figure 3: **Smart Object Creation.** Choose a model, add affordances and define a state to create a smart object.

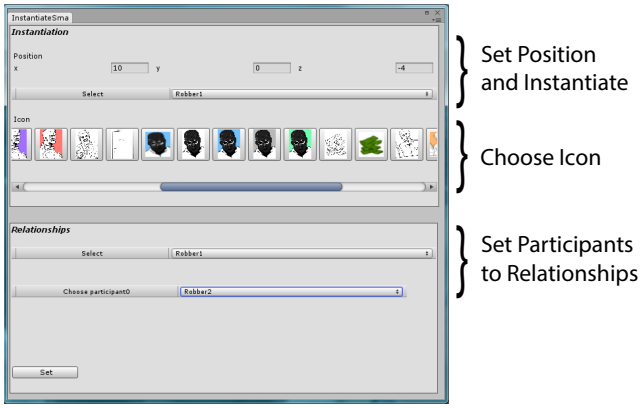


Figure 4: **Smart Object Instantiation.** Instantiate a smart object by defining position and a representative icon. Set the participants to the relationships.

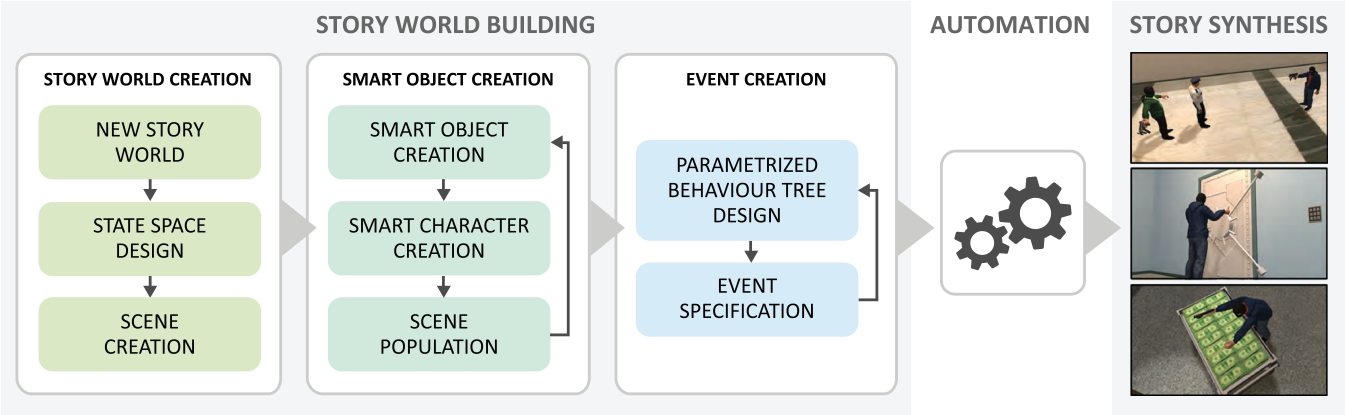


Figure 1: Overview of steps for building a story world and creating stories.

(a) (b) (c) (d)

Figure 2: An example narrative synthesized using an event-centric planner in the bank story world that was authored using our system. (a) Two robbers distract and incapacitate a guard. (b) One robber coerces the bank manager to press a button to access the vault. The robber unlocks the vault (c) and takes money from the vault (d) before escaping from the bank.

specifying the preconditions and postconditions of an event, and associating it with the corresponding PBT.

PBT's are inherently graphical in nature, and afford intuitive graphical user interfaces for design. We have integrated the popular NodeCanvas library for authoring behavior trees within our platform and extended it for PBT's, as utilized by our system. Figure 5(a) shows the process of creating a PBT by adding action tasks (leaf nodes) and control nodes to a PBT. In the example, an affordance task is selected. Figure 5(b) shows the selection of the affordance of the action task. Figure 5(c) visualizes a simple PBT that was designed using a sequence control node.

In the last step, an event instance is created. A name and representative icon are defined. A previously created PBT is designated for the event. Note that the same PBT may be used for multiple event definitions. For example, a logical conversation may have different ramifications depending on its event semantics which are defined at the event level. Pre- and post-conditions are defined as conjunctive normal form (CNF) expressions on the state and relations of the PBT participants. Figure 6 shows an example event called DistractAndIncapacitate. An icon to visually represent the event may be selected. The previously created PBT, also called DistractAndIncapacitate, is selected. The parameter, target, which represents the guard in the scene has initial state `RoleGuard` and `isStanding`, for example. As post condition, the state `isStanding` is negated.

The above three parts enable creation of the world with semantics required by a story planning tool, which is exe-

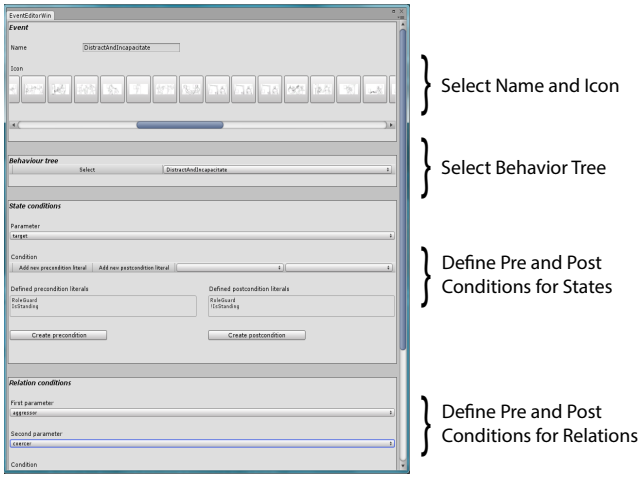
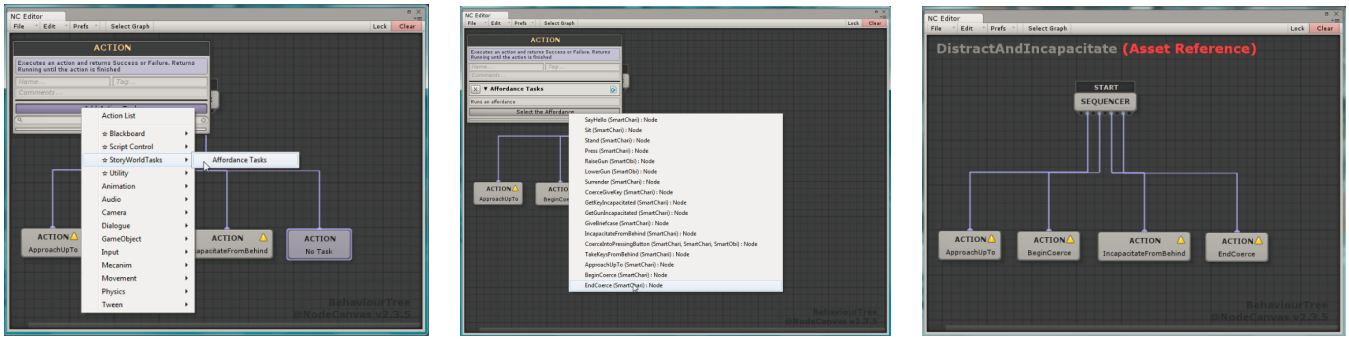


Figure 6: **Event Specification.** Create an event by selecting a representative icon and selecting a behavior tree. Define preconditions and postconditions on states and relations for the parameters of the behavior tree.



(a) (b) (c)

Figure 5: **Parameterized Behavior Tree Design**. (a) Select an affordance task to add to an action node. (b) Select an affordance to use in the behavior tree. (c) Graph of created behavior tree.

cuted to achieve the results visualized in Figure 2.

Conclusions

This paper motivates the importance of accessible metaphors for domain specification, as a precursor to computational narrative synthesis. We describe two representations of domain knowledge: agent-centric and event-centric. Agent-centric domain representations are used to author the traits and capabilities of unique characters at the cost of computational complexity and lack of scalability in number and complexity of characters. Event-centric domain representations mitigate the combinatorial complexity of authoring individual characters to more efficiently achieve desired narrative structure, at the cost of additional overhead of authoring events. Our aim of creating an accessible graphical interface for building story worlds can be applied to both domains.

We demonstrate a graphical platform for event-centric authoring of story worlds, which entails three main steps: Story World Creation, Smart Object Creation and Event Creation. These parts are designed to support the requirements of automated story planning systems to synthesize stories. We demonstrate the process of building a story world and creating stories in the context of a bank robbery scenario. Our tools are accessible to expert and novice users.

Limitations and Future Work. Development of our story building platform is ongoing. The interfaces and functionality continue to be tested and improved. Our story building platform is currently limited to creating events based on pre-defined affordances of smart objects, which constrains the narrative space. As future work, we intend to extend the platform to give more creative freedom to the author. Our tools may also be used to assist in the development of interactive narratives (Kapadia et al. 2015a; 2015b). We also plan to conduct user studies to improve the interface and identify the appropriate level of abstraction of domain representations. Hybrid domain representations (Riedl, Saretto, and Young 2003) must also be considered and integrated into the story building platform as well as techniques for automatic domain learning (Li, Lee-Urban, and Riedl 2013). The far-reaching goal of our research is to democratize story world building and digital story synthesis, by providing accessible metaphors for narrative content creation.

References

Fikes, R. E., and Nilsson, N. J. 1971. Strips: A new approach to the application of theorem proving to problem solving. Technical Report 43R, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025. SRI Project 8259.

Gordon, A.; van Lent, M.; Velsen, M. V.; Carpenter, P.; and Jhala, A. 2004. Branching Storylines in Virtual Reality Environments for Leadership Development. In *AAAI*, 844–851.

Isla, D. 2005. Handling Complexity in the Halo 2 AI. In *Game Developers Conference*.

Kallmann, M., and Thalmann, D. 1999. Direct 3d interaction with smart objects. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '99, 124–130. New York, NY, USA: ACM.

Kapadia, M.; Singh, S.; Reinman, G.; and Faloutsos, P. 2011a. A Behavior-Authoring Framework for Multiactor Simulations. *IEEE CGA* 31(6):45–55.

Kapadia, M.; Singh, S.; Reinman, G.; and Faloutsos, P. 2011b. Multi-actor planning for directable simulations. In *Digital Media and Digital Content Management (DMDCM)*, 111–116.

Kapadia, M.; Shoulson, A.; Durupinar, F.; and Badler, N. 2013. Authoring Multi-actor Behaviors in Crowds with Diverse Personalities. In *Modeling, Simulation and Visual Analysis of Crowds*, volume 11. 147–180.

Kapadia, M.; Falk, J.; Zünd, F.; Marti, M.; Sumner, R. W.; and Gross, M. 2015a. Computer-assisted authoring of interactive narratives. In *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*, i3D '15, 85–92. New York, NY, USA: ACM.

Kapadia, M.; Zünd, F.; Falk, J.; Marti, M.; Sumner, R. W.; and Gross, M. 2015b. Evaluating the authoring complexity of interactive narratives with interactive behaviour trees. In *Foundations of Digital Games*, FDG'15.

Li, B.; Lee-Urban, S.; and Riedl, M. 2013. Crowdsourcing interactive fiction games. In *International Conference on the Foundations of Digital Games, Chania, Crete, Greece, May 14-17, 2013*, 431–432.

Loyall, A. B. 1997. *Believable agents: building interactive personalities*. Ph.D. Dissertation, Pittsburgh, PA, USA.

Magerko, B.; Laird, J. E.; Assanie, M.; Kerfoot, A.; and Stokes, D. 2004. AI Characters and Directors for Interactive Computer Games. *Artificial Intelligence* 1001:877–883.

Mateas, M., and Stern, A. 2002. A behavior language for story-based believable agents. *IEEE Intelligent Systems* 17(4):39–47.

Mateas, M., and Stern, A. 2003. Integrating plot , character and natural language processing in the interactive drama facade. In *TIDSE*, volume 2.

Mateas, M., and Stern, A. 2004. A behavior language: Joint action and behavioral idioms. In *Life-Like Characters*. Springer. 135–161.

Mcdermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. Pddl - the planning domain definition language. Technical Report TR-98-003, Yale Center for Computational Vision and Control,.

Menou, E. 2001. Real-time character animation using multi-layered scripts and spacetime optimization. In *ICVS*, 135–144. London, UK: Springer-Verlag.

Pearl, J. 1984. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Perlin, K., and Goldberg, A. 1996. Improv: a system for scripting interactive actors in virtual worlds. In *Proceedings of ACM SIGGRAPH*, 205–216. New York, NY, USA: ACM.

Riedl, M. O., and Bulitko, V. 2013. Interactive narrative: An intelligent systems approach. *AI Magazine* 34(1):67–77.

Riedl, M. O., and Young, R. M. 2006. From linear story generation to branching story graphs. *IEEE CGA* 26(3):23–31.

Riedl, M.; Saretto, C. J.; and Young, R. M. 2003. 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*, AAMAS '03, 741–748. New York, NY, USA: ACM.

Sacerdoti, E. D. 1975. The nonlinear nature of plans. In *IJCAI*, 206–214.

Shoulson, A.; Garcia, F. M.; Jones, M.; Mead, R.; and Badler, N. I. 2011. Parameterizing behavior trees. In *MIG*, 144–155. Springer-Verlag.

Shoulson, A.; Gilbert, M. L.; Kapadia, M.; and Badler, N. I. 2013. An event-centric planning approach for dynamic real-time narrative. In *MIG*, 99:121–99:130.

Shoulson, A.; Marshak, N.; Kapadia, M.; and Badler, N. 2014. ADAPT: The Agent Development and Prototyping Testbed. *IEEE TVCG* 20(7):1035–1047.

Shoulson, A.; Kapadia, M.; and Badler, N. 2013. PAStE: A Platform for Adaptive Storytelling with Events. In *INT VI, AIIDE Workshop*.

Si, M.; Marsella, S. C.; and Pynadath, D. V. 2005. Thespian: An architecture for interactive pedagogical drama. In *Proceeding of the 2005 Conference on Artificial Intelligence in Education*. 595–602.

Thue, D.; Bulitko, V.; Spetch, M.; and Wasylishen, E. 2007. Interactive storytelling: A player modelling approach. In *AIIDE*.

Weyhrauch, P. W. 1997. *Guiding interactive drama*. Ph.D. Dissertation, Pittsburgh, PA, USA. AAI9802566.