Evaluating the Authoring Complexity of Interactive Narratives with Interactive Behaviour Trees

Mubbasir Kapadia*
Disney Research Zürich
Rutgers University

Fabio Zünd
ETH Zürich

Jessica Falk
ETH Zürich

Marcel Marti
ETH Zürich

Robert W. Sumner
Disney Research Zürich
ETH Zürich

Markus Gross
Disney Research Zürich
ETH Zürich

ABSTRACT

This paper evaluates the use of Behavior Trees (BT) for authoring compelling narrative experiences with free-form user interaction. We systematically study extensions to traditionally BT representations, which decouple the monitoring of user input, the narrative, and how the user may influence the story outcome – referred to as Interactive Behavior Trees (IBT's). By quantitatively evaluating the authoring complexity of BT formalisms with traditional story graph representations, we show that IBT's better scale with the number of story arcs, and the degree and granularity of user input. Our theoretical estimate of authoring complexity is corroborated with a qualitative user study, which confirms that subjects take lesser time with reduced effort to author narratives using IBT's. The subjective difficulty of IBT's is also lower than traditional story graphs.

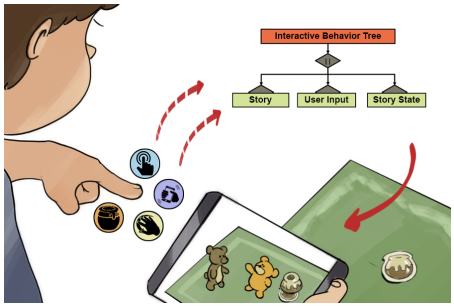


Figure 1: Augmented Reality application for Interactive Narratives authored using Interactive Behavior Trees.

Categories and Subject Descriptors

I.3.7 [**Computer Graphics**]: Virtual Reality; I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems—*Games*

Keywords

interactive narratives, behavior trees, augmented reality

1. INTRODUCTION

Interactive narratives place users in immersive virtual worlds where they become an integral part of an unfolding story. These users create or influence dramatic storylines through their actions. The growing maturity of Augmented Reality (AR) technologies opens up a new host of interaction possibilities and bridges the gap between the real world and virtual content to create immersive experiences. However, the ability to author interactive narrative content has not kept pace with the promise of AR technology. Modern and compelling interaction has seldom been combined with deep narrative experiences yet.

Computer games provide a prime example of interactive narrative: at predefined points in time, the player is asked to make a choice or take a certain action, which influences the course of the story. The quality of the narrative experience generally increases with the number of possible choices. However, the authoring complexity grows exponentially with the number of choices. Therefore, to keep the content authoring manageable in strong and compelling narratives the interaction is limited. This keeps the set of possible actions to alter the story minimal, and hence the player experiences only little user agency to influence the outcome of the story. Alternatively, a narrative with more interaction possibilities is created but at the cost of a simple and trivial narrative structure.

In this work we evaluate novel story authoring techniques, which overcome the problem of exponential growth of authoring complexity. Behaviour Trees (BT's) [5] allow authoring complex (non-interactive) stories containing various branching story arcs. They are created in a modular and thus extensible and easily maintainable fashion. As extension to BT's, Interactive Behavior Trees (IBT's) have been recently proposed and handle free-form user interaction, which is desirable in interactive narratives. IBT's are

*Email : mubbasir.kapadia@rutgers.edu

Proceedings of the 10th International Conference on the Foundations of Digital Games (FDG 2015), June 22-25, 2015, Pacific Grove, CA, USA.
ISBN 978-0-9913982-4-9.

split into three subtress: one for monitoring the user input, one defining the narrative, and one defining how the user may influence the story outcome. This empowers content creators to create compelling narratives involving free-form user interactions with minimal authoring complexity.

We study the authoring complexity of designing interactive narratives using BT formalisms and show that our IBT's increase the modularity, reusability, and maintainability of authored narratives, in comparison to traditional approaches. Our theoretical estimate of authoring complexity is corroborated with a user study, which confirms that subjects take lesser time with reduced number of errors to author narratives using IBT's. The subjective difficulty of BT's is also lower than traditional story graphs.

To demonstrate the potential of IBT's, we author an interactive narrative for an AR application where the user can freely interact with both real and virtual content to progress the narrative along a direction of their choosing. Fig. 1 depicts the interaction between the user and an example AR application. The user experiences an unfolding narrative through the lens of his or her portable device where the virtual characters are controlled using authored IBT's. Sensors on the device are used to register different interactions, which branch the narrative in different directions. This allows the user to freely engage with the characters and be an active participant in the story.

2. RELATED WORK

Interactive narratives have been studied from many different perspectives [21] and we provide a brief review below. Scripted approaches [11, 12] describe stories as pre-defined action sequences where small alterations often require far-reaching modifications of monolithic scripts. Improv [17] and LIVE [15] define rules, which govern how actors act based on certain conditions. These systems are not designed to generate complicated agent interactions that dramatically impact the story outcome. Facade [13] executes authored beats to manage the intensity of the story. A Behavior Language (ABL) [14] provides a generalized scripting language for single- or multi-character actions based on manually authored preconditions for successful action execution.

Dialogue Trees [18] and Story Graphs [4] accommodate user interaction as discrete choices at key points in the authored narrative. Behavior Trees (BT's) are gaining popularity in the computer gaming industry for designing the artificial intelligence logic for non-player characters [5, 6]. BT's offer graphical constructs for authoring modular, extensible behaviors which can be extended to control multiple interacting characters [24]. For communication between nodes, BT's rely on a blackboard [16], which is a centralized, flat repository of data that can be accessed by nodes in the tree. Animation systems [28, 27] can be used to visualize the execution of these behaviors on fully articulated virtual humans.

Computational tools have also been developed that automatically generate digital stories [10, 9]. Event-centric planning [25, 26] plans in the space of pre-authored behavior trees, thus mitigating the combinatorial explosion of planning in the action space of individual character actions. Recent work explores the use of partial-order planning to provide a computational tool for authoring interactive narratives [8]. PaSSAGE [29] and the Automated Story Director [22] monitor a user's experience through the story to choose between scenes and character behavior. The focus of this work is to revisit and evaluate the underlying representations of the stories themselves, in order to reduce the complexity of authoring interactive narratives.

User Interaction. Rapid advances in interaction modalities [30] have helped bridge the divide between real and virtual content where natural user interfaces are effectively invisible and heighten the degree of immersion. However, the ability to author interactive narrative content that can truly harness the power of these interaction experiences still remains an open challenge. Augmented Reality [31] helps bridge the gap between real and virtual content to create immersive experiences. Combining different input modalities such as touch, gestures, and voice with the inherent camera control in AR applications empowers content creators to create new forms of interactive experiences that were not possible before.

3. INTERACTIVE NARRATIVES

We identify three main requirements towards authoring free-form interactive narrative experiences:

1. **Modular Story Definition.** Complex interactive narratives have many interconnected story arcs that are triggered based on user input leading to widely divergent outcomes. The complexity of authoring narratives must scale linearly with number of story arcs, which can be defined in a modular and independent fashion.

2. **User Interactions.** User interaction should be free-form, and not limited to discrete choices at key stages of the story, with far-reaching ramifications on the outcome of the narrative. Monitoring user input and story logic should be decoupled to facilitate the modification of user interactions without requiring far-reaching changes to the story definition.

3. **Persistent Stories.** The actions and interactions between the user and characters over the entire course of the narrative must persist and influence story progression.

3.1 Behavior Trees for Narrative Authoring

Behavior Trees (BT's) enable graphical and modular authoring of complex narratives. Branching narratives can easily be created using BT control nodes that connect multiple subtress, which represent independent story arcs. Recent work [24] facilitate the authoring of complex multi-actor interactions in a parametrizable fashion, enabling the reuse of modular plot elements, and ensures that the complexity of the narrative scales independently of the number of characters. These properties of BT's make them ideally suited for authoring complex, branching narratives (Requirement 1). However, BT's are not suited to to handle free-form interactions and persistent state (requirements 2 and 3).

Monitoring User Interactions. Fig. 2 illustrates a naive approach to monitoring user interactions in a BT. This re-

quires the subtree that monitors user input to be inserted into the narrative tree at all points where user selects how the narrative proceeds, thus producing a tight coupling between the narrative definition and user interaction. Reducing the number of instances in the narrative where user input is monitored severely limits interactivity, while free-form interaction increases the complexity of the tree. Also, user input can be monitored only at the granularity of a single node in the tree, and not at every frame. These complications produce a tradeoff between complexity in the narrative and degree of interactivity.

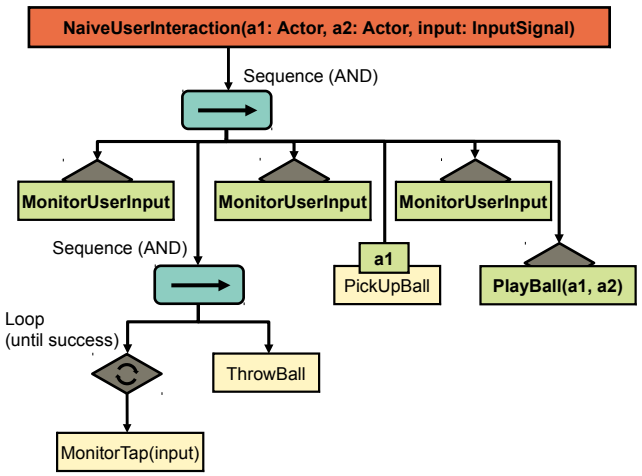


Figure 2: Handling user interaction in a traditional Behavior Tree. The MonitorUserInput subtree is required at all points in the narrative tree where the user selects how the narrative proceeds, resulting in unnecessary large trees.

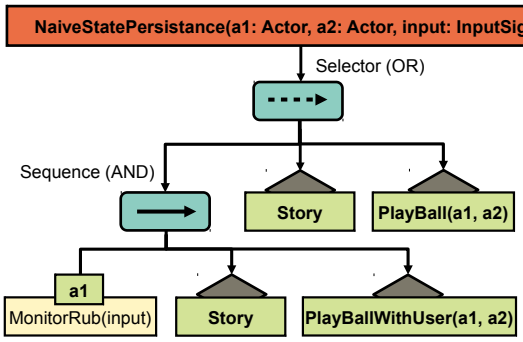


Figure 3: A naive approach to handle state persistence in Behavior Trees.

State Persistence. Behavior Trees traditionally don't have any means of explicitly storing the past state of the characters involved in the narrative, where the current state of the story is implicit in the current active node. Fig. 3 illustrates a simple narrative where the bears choose to include the user in the game of catch contingent on whether the user playfully interacted with the bear at the beginning of the story. A traditional BT requires a branch at the very beginning of the narrative where the presence or absence of the playful rub produces two largely redundant tree definitions with minor variations in its ending.

3.2 Interactive Behavior Trees

To address the challenges described above, Interactive Behavior Trees facilitate free-form user interaction and state persistence. We refer the readers to [8] for a detailed summary of IBT's and provided a brief overview below.

IBT's are divided into 3 independent sub-trees that are connected using a Parallel control node. An IBT

$$\mathbf{t}_{\text{IBT}} = \langle \mathbf{t}_{\text{ui}}, \mathbf{t}_{\text{state}}, \mathbf{t}_{\text{narr}} = \{\mathbf{t}_i^{\text{arc}} | \mathbf{t}_1^{\text{arc}} \dots \mathbf{t}_m^{\text{arc}}\}, \beta \rangle \quad (1)$$

comprises the following subtrees: (1) **Story Subtree.** $\mathbf{t}_{\text{narr}}$ is responsible for handling the narrative progression and is further subdivided into subtrees $\{a_i\}$ that represent a separate story arc. (2) **MonitorUserInput Subtree.** $\mathbf{t}_{\text{ui}}$ monitors the different interactions that are available to the user and can be easily changed depending on the application or device. Since $\mathbf{t}_{\text{ui}}$ is executed in parallel with the other subtrees, IBT's are able to immediately respond and register the interactions of the user and use it to influence the narrative outcome. (3) **MonitorStoryState Subtree.** $\mathbf{t}_{\text{state}}$ monitors the state of the story to determine if the current story arc needs to be changed.

The overall design of the IBT results in three subtrees that execute independently in parallel with one another. A blackboard β is used to store the states of the characters and the story and is used to communicate between the subtrees and also to maintain state persistence. $\mathbf{t}_{\text{ui}}$ updates β when any input signal is detected, which is queried by $\mathbf{t}_{\text{state}}$ to determine the current state of the story, and the active story arc. $\mathbf{t}_{\text{state}}$ contains separate subtrees for each story arc which checks if the precondition for the particular arc is satisfied. If so, β is updated to reflect the newly activated story arc which is used to switch the active story in $\mathbf{t}_{\text{narr}}$.

4. AUTHORING COMPLEXITY

4.1 Cyclomatc Complexity

Cyclomatic complexity [7] quantifies the number of linearly independent paths through the program by measuring the number of branches in the code, and serves as a standard criteria for optimizing the testability and maintainability of the code without the need for dynamic code analysis. Cyclomatic complexity estimates the number of decisions made by the source code where a complex, often poorly written program with many branches leads to a high value of cyclomatic complexity, indicating that the program is difficult to test, maintain and prone to errors. By analogy, narratives that account for the different ways a user may influence the story outcome require many decision points, and are complex to author. For this reason, we use cyclomatic complexity to quantify authoring complexity for interactive narratives.

Cyclomatic complexity, $c(\mathbf{g})$ is computed by first converting the program into its equivalent control flow graph (CFG) representation $\mathbf{g}$, where the nodes correspond to atomic commands, and the directed edges connect commands that execute in sequence. c can be calculated in a variety of ways, including:

$$c(\mathbf{g}) = p(\mathbf{g}) + 1 \tag{2}$$

where $p(\mathbf{g})$ is the number of decision points (e.g., if, while, ... etc.) in the program. However, Eq. 2 assumes that there is a single termination point in the program which is not the case with BT's where every node may terminate with success or failure. Additionally, there is a third "runnning" state that is returned at each frame while the node is still executing. To generalize the measure of $c(\mathbf{g})$ for multiple termination points, we use a modified equation shown below where $s(\mathbf{g})$ denotes the number of exit points in $\mathbf{g}$.

$$c(\mathbf{g}) = p(\mathbf{g}) - s(\mathbf{g}) + 2 \tag{3}$$

4.2 Computing Cyclomatic Complexity for Behavior Trees

Fig. 4 illustrates the equivalent control flow representations for the different control nodes used for defining behavior trees, which are used to compute its cyclomatic complexity. All subsequent calculations of $c(.)$ assuming that the behavior trees are converted to their equivalent control flow graphs.

Leaf Node. A leaf node $\mathbf{t}_{\text{leaf}}$ represents an atomic command in a BT which returns either success or failure. If it is in the "running" state, it continues executing itself until it succeeds or fails. Fig. 4(a) shows the CFG for a leaf node. Depending on the number of return states in the particular leaf node implementation, we have $p(\mathbf{t}_{\text{leaf}}) = \{0, 1, 2\}$ and $s(\mathbf{t}_{\text{leaf}}) = \{1, 2\}$. If the leaf node immediately returns success or failure without using the running state, we have $c(\mathbf{t}_{\text{leaf}}) = 1$. If the leaf node can enter the running state, the complexity is $c(\mathbf{t}_{\text{leaf}}) = 2$.

Sequence Node. A sequence node $\mathbf{t}_{\text{seq}}$ returns failure if any one of its child nodes fails, else it returns success. If a child returns "running", it simply continues executing this child until it has reached failure or success. Fig. 4(b) illustrates the CFG for $\mathbf{t}_{\text{seq}}$. To calculate $c(\mathbf{t}_{\text{seq}})$ of $\mathbf{t}_{\text{seq}}$ with a set of m child nodes $\{\mathbf{t}_i | \mathbf{t}_1, \mathbf{t}_2 \dots \mathbf{t}_m\}$, we need to consider that each child node may be its own subtree with multiple decision points.

$$\begin{aligned} p(\mathbf{t}_{\text{seq}}) &= \sum_{i=1}^{m} p(\mathbf{t}_i),\ s(\mathbf{t}_{\text{seq}}) = 2, \\ \therefore c(\mathbf{t}_{\text{seq}}) &= \sum_{i=1}^{m} p(\mathbf{t}_i) \end{aligned} \tag{4}$$

Selector Node. The selector node $\mathbf{t}_{\text{sel}}$ returns success as soon as its first child node returns success, and produces a similar control flow graph as compared to $\mathbf{t}_{\text{seq}}$ (Fig. 4(c)). Hence, $c(\mathbf{t}_{\text{sel}}) = c(\mathbf{t}_{\text{seq}})$.

Loop Node. The loop node $\mathbf{t}_{\text{loop}}$ is used to repeatedly execute its child node $\mathbf{t}_{\text{c}}$ until a certain condition is met. A loop node may only have a single decorator or leaf node as its child and does not define how to traverse through multiple children. The following termination conditions may be used: (1) loop until success, (2) loop until failure, (3) loop N times, (4) loop forever. Fig. 4(d) illustrates the loop node which terminates when its child node returns success. It has only one termination node, and an additional decision point is introduced for looping.

$$\begin{aligned} p(\mathbf{t}_{\text{loop}}) &= 1 + p(\mathbf{t}_{\text{c}}),\ s(\mathbf{t}_{\text{loop}}) = 1, \\ \therefore c(\mathbf{t}_{\text{loop}}) &= p(\mathbf{t}_{\text{c}}) + 2 \end{aligned} \tag{5}$$

The same calculations apply for a loop node that repeats until failure. For $\mathbf{t}_{\text{loop}}$ which repeats a fixed number of times.

$$\begin{aligned} p(\mathbf{t}_{\text{loop}}) &= 1 + p(\mathbf{t}),\ s(\mathbf{t}_{\text{loop}}) = 2, \\ \therefore c(\mathbf{t}_{\text{loop}}) &= p(\mathbf{t}) + 1 \end{aligned} \tag{6}$$

Fig. 4(e) illustrates a loop node that never returns. Since the node never terminates and has neither a decision point nor an exit point, we only need to consider the child of the loop node.

$$\begin{aligned} p(\mathbf{t}_{\text{loop}}) &= p(\mathbf{t}),\ s(\mathbf{t}_{\text{loop}}) = 0, \\ \therefore c(\mathbf{t}_{\text{loop}}) &= c(\mathbf{t}) \end{aligned} \tag{7}$$

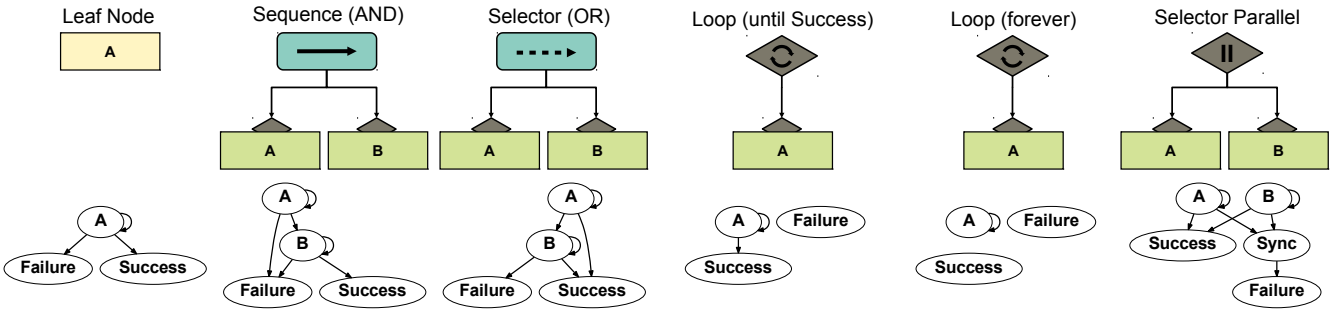
Parallel Node. The parallel node $\mathbf{t}_{\text{par}}$ executes its child nodes in parallel and has two types depending on the termination condition: (1) Selector Parallel: It executes until any child node returns success or all of them return failure. (2) Sequence Parallel: It executes until any child node returns failure or all of them succeed. Fig. 4(f) illustrates the control flow graph of a selector parallel node. An additional node "Sync" is used to symbolize the synchronization barrier between the child nodes and termination. The sequence parallel node exhibits similar behavior and both their complexity measures can be calculated as shown below.

$$\begin{aligned} p(\mathbf{t}_{\text{par}}) &= 2 \cdot \sum_{i=1}^{m} p(\mathbf{t}_i),\ s(\mathbf{t}_{\text{par}}) = 2, \\ \therefore c(\mathbf{t}_{\text{par}}) &= 2 \cdot \sum_{i=1}^{m} p(\mathbf{t}_i) \end{aligned} \tag{8}$$

4.3 Complexity Comparison

Using the complexity measure described above, we compare the authoring complexity of traditional story graphs, naive BT definitions for interactive narratives, and IBT's.

Story Graphs. A story graph $\mathbf{g}_{\text{s}}$ is a directed graph where the vertices correspond to story atoms during which the user has no outcome on the narrative, and the edges represent a discrete set of choices that the user has to influence how the story ends. A linear narrative represents a lower bound on $c(\mathbf{g}_{\text{s}}) = 1$ with no decision points. For interactive narratives



(a) Leaf (b) Sequence (c) Selector (d) Loop until success (e) Loop Forever (f) Selector Parallel

Figure 4: Control flow graphs for the different control nodes used in Behavior Trees.

however, an upper bound on the complexity represents a branching story graph where all possible user interactions are possible at each stage in the story, which is supported by IBT's.

For a story graph $\mathbf{g}_\text{s}$ with m story arcs $\{a_i|a_1 \dots a_m\}$, each with $|a_i|$ number of nodes, and d possible user interactions, there will be a maximum of $d+1$ outgoing edges per node in $\mathbf{g}_\text{s}$. Therefore, each node represents d decision points. Additionally, the number of exit points depends on how many story arcs can end the story $1 \leq s(\mathbf{g}_\text{s}) \leq m$. We calculate $c(\mathbf{g}_\text{s})$ as follows:

$$c(\mathbf{g}_\text{s}) = d \cdot \left(\sum_{i=1}^{m} |a_i| \right) - s(\mathbf{g}_\text{s}) + 2 \tag{9}$$

Behavior Trees (Naive Approach). Fig. 2 illustrates the naive approach to authoring interactive narratives using BT's. This BT $\mathbf{t}_\text{BT} = \langle \mathbf{t}_\text{mu}, \{a_i|a_1 \dots a_m\} \rangle$ comprises a subtree $\mathbf{t}_\text{mu}$ which monitors user input and story branching, and m story arc subtrees, where a_i represents the i^{th} arc with $|a_i|$ nodes. However, these subtrees are tightly coupled together as the user input must be monitored between each node of each story arc to ensure free-form user interaction. The number of decision points for each story arc a_i, $p(a_i) = |a_i| \cdot p(\mathbf{t}_\text{mu}) + p(a_i)$, and $\mathbf{t}_\text{BT}$ has two exit points. Hence, the cyclomatic complexity $c(\mathbf{t}_\text{BT})$ is calculated as follows:

$$c(\mathbf{t}_\text{BT}) = \sum_{i=1}^{m} (|a_i| \cdot p(\mathbf{t}_\text{mu}) + p(a_i)) \tag{10}$$

Interactive Behavior Trees. The benefit of the IBT formalism is that the 3 subtrees $\langle \mathbf{t}_\text{ui}, \mathbf{t}_\text{state}, \mathbf{t}_\text{narr} \rangle$ are all independent to each other and run in parallel using a Parallel Sequence node. Additionally, each subtree has a Loop forever node at its top, Hence, we can consider each subtree as an independent program and the total complexity $c(\mathbf{t}_\text{IBT})$ can be computed by adding the complexities of each independent subtree.

MonitorUserInput Subtree. $\mathbf{t}_\text{ui}$ monitors all possible user interactions which sets the appropriate state in the blackboard β. In our proposed solution, all the leaf nodes that monitor the respective user inputs were modified to only return success and thus have no decision points. $\mathbf{t}_\text{ui}$ simply checks the different interactions in a sequence without any branching. As a result, $c(\mathbf{t}_\text{ui}) = 1$.

MonitorStoryState Subtree. $\mathbf{t}_\text{state}$ checks if all the preconditions for each arc are satisfied and sets the current story arc if needed. For l_i nodes per story arc a_i, $(l_i - 1)$ of those node are assertion nodes that represent a decision point and the last one sets the current story arc. Hence, the complexity $c(\mathbf{t}_\text{state})$ is calculated as follows:

$$c(\mathbf{t}_\text{state}) = \sum_{i=1}^{m} (l_i - 1) \tag{11}$$

Story Subtree. $\mathbf{t}_\text{narr} = \{\mathbf{t}_i^\text{arc} | \mathbf{t}_1^\text{arc} \cdot \mathbf{t}_m^\text{arc}\}$ consists of a subtree for each story arc a_i which additionally checks if it is the current story arc before proceeding to execute the narrative. This ensures that story arcs can be switched seamlessly at any point in the narrative. This introduces 2 more decision points per arc, in addition to the arc complexity. The number of decision points $p(\mathbf{t}_i^\text{arc})$ for the i^{th} story arc, $p(\mathbf{t}_i^\text{arc}) = 2 \cdot (2 + p(a_i))$. The resulting complexity, $c(\mathbf{t}_\text{narr})$:

$$c(\mathbf{t}_\text{narr}) = 4 \cdot m + \sum_{i=1}^{m} (2 \cdot p(a_i)) \tag{12}$$

The overall complexity $c(\mathbf{t}_\text{IBT})$ is calculated as follows:

$$\begin{aligned} c(\mathbf{t}_\text{IBT}) &= c(\mathbf{t}_\text{ui}) + c(\mathbf{t}_\text{state}) + c(\mathbf{t}_\text{narr}) \\ &= 1 + 4 \cdot m + \sum_{i=1}^{m} (2 \cdot p(a_i) + (l_i - 1)) \end{aligned} \tag{13}$$

Conclusion. Eq. 9 shows that the number of user interactions and the number of nodes in the story arcs have a multiplicative effect on the authoring complexity for story graphs. This limits content creators to choose between freedom of interaction and the complexity of the narrative to prevent the authoring complexity from becoming prohibitive. In contrast, the number of decision points in the story arc have a linear effect on $c(\mathbf{t}_\text{IBT})$, while the number of ways a user can

interact has no impact on the authoring complexity. This allows content creators to create compelling interactive narrative experiences with free-form user interaction, without being burdened by limitations in the specification language.

5. APPLICATION

We developed an interactive narrative application in an AR setting, authored using IBT's, where the user can freely interact with the characters in the story (two bears) and his interactions influence the outcome of the narrative.

5.1 Augmented Reality Framework

Augmented Reality (AR) applications benefit from intuitive and versatile input mechanisms where the user can seemingly physically interact with the virtual world. For see-through AR applications, the physical movement of the mobile device serves as a direct means of exploring the digital content superimposed in a real environment, where, not only can the user interact with the virtual characters using the host of sensors available on the device, but the virtual characters can interact with the physical world as well.

Implementation. We used a natural image-based tracking approach [1], which registers the 2D marker image and estimates the camera location and pose in real-time for stable tracking. We used the implementation provided by Vuforia [19]. Additional image markers can also be tracked and used as triggers in the AR application, for example, to instantiate new objects in the world, which may branch the narrative in a different direction. The game application was implemented using the Unity3D game engine using a data-driven character animation system. The animation functionality is exposed to the author using a set of routines including **LookAt**(obj), **Reach**(target) etc., which can be invoked from BT's. For more details of the animation system, please refer to [28]. The narrative was authored using an extended version of the BT library described in [24]. The current version of the game was deployed and tested on multiple portable devices including Sony Xperia Tablet Z, Apple iPad 3rd generation, Apple iPad Mini Retina, and Apple iPad Air.

Interaction Vocabulary. Using the sensors available on the mobile device, the user can interact with the virtual characters in the following ways: (1) Moving the device to focus on different objects and characters. (2) Tapping on objects and characters to pick them up or interact with them. (3) Shaking the device. (4) Gestures to communicate user intent. (5) Using image markers to trigger objects (e.g., a honey pot sticker can be used to create a honey pot for the bears) in the world. The mode and effect of interactions is completely decoupled from the narrative and can be easily changed depending on the platform used without impacting the narrative definition.

5.2 Story Definition

We author an interactive narrative involving two bears. We provide a brief description below and refer the readers to the supplementary video for more details.

Playing Catch. The first bear B_1, enters the scene and looks up at the player with curiosity. The player can experiment with the bear by trying out different interaction possibilities including rubbing the bear, asking him to twirl using a circular gesture, tapping him to attract his attention, or zooming in to get a closer look. Soon, the second bear B_2 enters and asks B_1 for a beach ball so they can play catch. B_1 is unable to find a ball and turns to the player for help. The player may choose to give a soccer ball to the bears but they only want to play with the beach ball. Depending on where the player throws the beach ball, different branches of the story are triggered where giving the ball to B_1 enables him to help his friend.

State Persistence. The user's choices have ramifications later on in the story. For example, B_1 remembers if the player rubbed him at the beginning and includes him in the game by throwing him the ball. If the player chooses not to interact with the bear, the bears are less friendly and exclude him from the game of catch.

Additional Interactions. At any point, the player may use a honey pot sticker to trigger a honey pot in the world. The bears who are obsessed with honey leave aside whatever they are doing, even their beloved ball, and make a beeline towards the honeypot which represents one possible conclusion of the story. A more mischievous player may choose to trigger bees into the world which chase the bears and disrupt their game of catch. Only flowers may then be used to distract the bees and save the bears. These different story arcs are authored as modular, independent units in the IBT and can be triggered at any stage without the need for complex connections and state checks in the behavior tree definition.

Freedom of Interaction. The above narrative represents a very simple baseline to demonstrate the ability to author compelling interactive narrative experiences. Unlike traditional approaches where the user was limited to discrete choices at certain stages in the narrative, IBT's empower the player with complete freedom of interaction where he may choose to play ball with the bears, give them honey, or simply wreak havoc by releasing a swarm bees at any point of time. The interactions elicit instantaneous and plausible interactions from the characters while staying true to the narrative intent.

6. USER STUDY

We conducted a preliminary user study to assess the influence of the authoring methods on the user's authoring performance. Each subject was asked to implement a predefined, moderately complex narrative using two methods: (1) Story Graphs `SG` and (2) Interactive Behavior Trees `IBT`. We used a basic implementation of story graphs where nodes represent the execution of one or more affordances in sequence, and edges represent a condition on the status of the previous node, or a user interaction. Recent extensions [23] encapsulate state machines within individual graph nodes to create more complex narrative atoms. However, since our goal is to allow the user to interact at any point in the story, narrative atoms in story graphs are restricted to the set of available affordances.

6.1 Method

All subjects carried out the study on a similar PC with two monitors, where they used an extended version of the Unity game engine for authoring. Both methods were implemented

within the same GUI to mitigate the influence of the interface on the analysis. Each subject was first provided with a tutorial document to introduce each method and get accustomed to the application. For each tutorial, the subject could freely interact with the software and experiment with its functionality without a time limit. Following each tutorial, the subject was asked to author the narrative described below during which a variety of metrics were measured and logged.

Narrative. The subjects were presented with the following textual description of the narrative, which they had to author using each method. While relatively simple, the narrative exercises the use of multiple interconnected story arcs and freedom of user interaction. (1) *Start Arc:* Bear1 enters the scene. Bear1 is bored. After 2 seconds Bear1 realizes that the user is watching and waves at him. Bear2 enters the scene. (2) *Conversation Arc:* Bear2 greets Bear1. They start arguing. After 10 seconds Bear2 apologizes to Bear1. The story ends. (3) *Bee Arc:* The player may trigger the bees at any time. If the bees are in the scene, the bears run away from the bees and completely ignore any other story elements or user interactions. (4) *Flower Arc:* The users can spawn flowers at any time. If flowers are in the scene, the bees (if present) stop chasing the bears and fly to the flower. The bears are then free to continue with the previously active story arc, or respond to other user interactions.

6.2 Metrics

Independent Variables. The method of authoring and author proficiency are treated as independent variables (**IV**) in our experiment. Subjects were instructed to author the same narrative using the 2 methods: Story Graphs `SG` and Interactive Behavior Trees `IBT`. The order in which the user authored the story using each method was chosen at random. At the beginning of the study, the subject was asked to select his proficiency as an *expert* who has previous experience with BT's, or a *novice* with little or no experience.

Dependent Variables. We logged a variety of metrics to capture the authoring performance, which are treated as dependent variables (**DV**) in our experiment. (1) We measured the time in minutes t_a needed to author the narrative for each method. (2) We counted the number of mouse clicks n_c during each authoring session as an estimate of the effort to author a story. (3) For each authoring method, the user was asked to rate its subject difficulty d_s as a nominal value from 1 (very easy) to 5 (very hard).

6.3 User Study Results

We recorded 12 subjects at the age between 21 years and 35 years, 90 % male. They were Computer Science students (4 undergraduate, 8 graduate) and were proficient with using the Unity editor. Additionally, expert users had previous experience with Behavior Trees. All subjects (6 experts, 6 novice users) authored the same story using both methods. A MANOVA was conducted with **Method** and **Proficiency** as independent variables, with t_a, n_c, and d_s as dependent variables, and with the user's ID as covariate.

Findings revealed a statistically significant difference in user performance based on the **Method** prior (Roy's Largest Root = 6.282, $F(3, 17) = 35.587$, $p < 0.001$, partial $\eta^2 = 0.863$) as well as based on the **Proficiency** prior (Roy's Largest Root = 5.525, $F(3, 17) = 31.309$, $p < 0.001$, partial $\eta^2 = 0.847$). Furthermore, the multivariate model exhibits high R^2 values: t_a: $R^2 = 0.888$, n_c: $R^2 = 0.571$, d_s: $R^2 = 0.785$.

The univariate effects of **Method** and **Proficiency** are summarized in Table 1. The estimated means for both IVs are depicted in Table 2. Fig. 5 depicts the correlation matrix with Pearson's Correlation coefficients for all IVs and DVs. Red coefficients indicate a significant ($\alpha = 0.05$) correlation. The strongest significant correlation is observed between **Method** and d_s ($r = -0.87$).

Table 1: Univariate tests analysis for Method and Proficiency.

IV	DV	*df*	Error *df*	*F*	*p*
Method	t_a	1	19	30.320	**< 0.001**
	n_c	1	19	11.761	**0.003**
	d_s	1	19	67.308	**< 0.001**
Proficiency	t_a	1	19	103.665	**< 0.001**
	n_c	1	19	11.472	**0.003**
	d_s	1	19	0.731	0.403

Table 2: Estimated means for the multivariate model (Method and Proficiency interaction). Lower and upper bounds (LB, UB) are provided for 95% confidence intervals.

DV	Proficiency	Method	LB	Mean	UB
n_c	Expert	IBT	131.52	367.16	602.81
		SG	654.18	889.83	1125.48
	Novice	IBT	649.86	885.50	1121.15
		SG	899.19	1134.84	1370.48
t_a	Expert	IBT	12.87	25.52	38.16
		SG	46.04	58.68	71.32
	Novice	IBT	74.34	86.99	99.63
		SG	107.68	120.32	132.96
d_s	Expert	IBT	2.25	2.83	3.40
		SG	4.42	4.99	5.57
	Novice	IBT	1.93	2.51	3.08
		SG	4.27	4.84	5.42

Interpretation of Results. The results clearly show that independent of the user's proficiency with IBT's, there is a significant decrease in authoring time, number of clicks, and subjective authoring difficulty when comparing `SG` and `IBT`. The novice users consistently performed worse than the expert users, which confirms that our performance metrics were appropriate. Finally, while the method indicates a strong effect on the subjective difficulty, no significant effects of the subject's proficiency with IBT's on the subjective difficulty could be measured.

The study yielded positive results. However, it must be noted that the scope of the study was limited due to the significant amount of time it took to author even the moderately complex narrative described above. The average time to complete the study for the *novice* group was over 2 hours. More complex narratives and a larger user base are needed to create more substantial results. Furthermore, the quality of the tutorial documents explaining the methods have certainly a strong influence on how well the subjects perform with that method. Yet it is inherently difficult to teach all methods to exactly the same extent. In a fu-

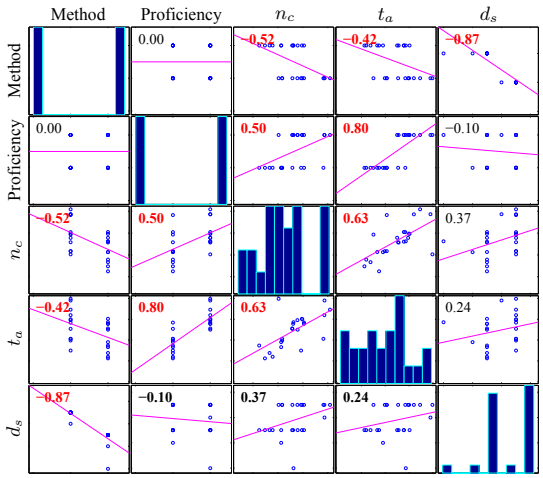


Figure 5: Correlation matrix for the IVs and DVs depicting Pearson's Correlation Coefficients. A red coefficient indicates a statistically significant correlation. The plots on the diagonal depict a histogram for each distribution.

ture study, maybe carefully created video tutorials for each method might mitigate the effect of training on the study outcome. Our user study confirms the theoretical measures of authoring complexity that were described earlier.

7. DISCUSSION AND FUTURE WORK

In this paper, we demonstrate the benefits of using Interactive Behavior Trees for authoring free-form interactive narratives. The hierarchical, graphical nature of BT's makes them suitable for authoring complex narratives with branching story arcs in a modular, extensible fashion. By decoupling the monitoring of user input and the narrative definition, users who experience the narratives can freely interact with the characters in the story, limited only by their creativity and interaction device. This enables content creators to author compelling interactive narrative experiences with free-form user interaction.

We calculate the authoring complexity of authoring narratives by mapping different narrative specifications to their corresponding control flow representations, which can be used to compute their cyclomatic complexity. Comparing the cyclomatic complexity of IBT's vs. traditional story graphs reveals that IBT's scale better with the number of story arcs, and degree and granularity of interactions. This makes them suitable for authoring complex narratives with free-form user input. Our user study confirms the premise that users, be it expert users, who are familiar with IBT's, or novice users, spend in average less time, require less effort, and rate it subjectively easier to author a predefined narrative employing IBT's compared to traditional story graphs in our Unity authoring editor.

Limitations and Future Work. IBT's enable content creators to easily author complex interactive narratives without being burdened by limitations in the specification language. However, all the story arcs and responses to different user inputs need to be authored beforehand and there are no emergent responses to unforeseen situations which may often arise in interactive applications. There is a growing trend to use automated narrative tools [21] to produce emergent interactive experiences. Recent work [8] explores the use of automated narrative tools to assist content creators to handle all possible user interactions during the authoring process. For future work, we would like to study the potential benefits and tradeoffs of automation for authoring interactive narratives.

The cyclomatic complexity provides an estimate of the number of decision points in the authored story, but does not account for the number of decisions needed while authoring the narrative. Additional measures [2] that account for the high-level structure of a story may also be used to complement this analysis. The use of story graphs provides a suitable baseline for comparing the benefits and tradeoffs of IBT's for interactive narratives. For future work, we would also like to compare IBT's with many additional alternatives including hierarchical task networks [3], narrative mediation [20], and other planning based approaches.

8. REFERENCES

[1] F.-e. Ababsa and M. Mallem. Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems. In *ACM SIGGRAPH VRCAI*, pages 431–435, 2004.

[2] S. Chen, M. J. Nelson, and M. Mateas. Evaluating the authorial leverage of drama management. In *Proceedings of the Fifth Artificial Intelligence and Interactive Digital Entertainment Conference, AIIDE 2009, October 14-16, 2009, Stanford, California, USA*, 2009.

[3] K. Erol, J. Hendler, and D. S. Nau. Htn planning: Complexity and expressivity. In *Proceedings of AAAI*, pages 1123–1128. AAAI Press, 1994.

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

[5] C. Hecker, L. McHugh, M. Argenton, and M. Dyckho. Three approaches to halo-style behavior tree ai. In *Game Developers Conference*, 2007.

[6] D. Isla. Halo 3: Building a better battle. In *Game Developers Conference*, 2008.

[7] G. Jay, J. E. Hale, R. K. Smith, D. P. Hale, N. A. Kraft, and C. Ward. Cyclomatic complexity and lines of code: Empirical evidence of a stable linear relationship. *JSEA*, 2(3):137–143, 2009.

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

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

[10] M. Kapadia, S. Singh, G. Reinman, and P. Faloutsos.

A behavior-authoring framework for multiactor simulations. *Computer Graphics and Applications, IEEE*, 31(6):45 –55, nov.-dec. 2011.

[11] A. B. Loyall. *Believable agents: building interactive personalities*. PhD thesis, Pittsburgh, PA, USA, 1997.

[12] M. Mateas. *Interactive drama, art and artificial intelligence*. PhD thesis, Pittsburgh, PA, USA, 2002.

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

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

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

[16] I. Millington and J. Funge. *Artificial Intelligence for Games, Second Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2009.

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

[18] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics*, 7:25–75, 1997.

[19] Qualcomm. Vuforia Developer SDK, 2010.

[20] M. 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*, AAMAS '03, pages 741–748, New York, NY, USA, 2003. ACM.

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

[22] 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*, 4(2):23–42, 2008.

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

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

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

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

[27] A. Shoulson, N. Marshak, M. Kapadia, and N. Badler. ADAPT: The Agent Developmentand Prototyping Testbed. *IEEE TVCG*, 20(7):1035–1047, July 2014.

[28] A. Shoulson, N. Marshak, M. Kapadia, and N. I. Badler. Adapt: the agent development and prototyping testbed. In *ACM SIGGRAPH I3D*, pages 9–18, 2013.

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

[30] D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2011.

[31] F. Zhou, H.-L. Duh, and M. Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *Mixed and Augmented Reality. ISMAR 2008*, pages 193–202, Sept 2008.