Evaluation of an Automatically-Constructed Graph-Based Representation for Interactive Narrative

Nathan Partlan  
nkp@ccs.neu.edu  
Northeastern University

Elin Carstensdottir  
elin@ccs.neu.edu  
Northeastern University

Erica Kleinman  
erica@ccs.neu.edu  
Northeastern University

Sam Snodgrass  
s.snodgrass@northeastern.edu  
Northeastern University

Casper Harteveld  
c.harteveld@northeastern.edu  
Northeastern University

Gillian Smith  
gsmith@wpi.edu  
Worcester Polytechnic Institute

Camillia Matuk  
cmatuk@nyu.edu  
New York University

Steven C. Sutherland  
sutherland@uhcl.edu  
University of Houston-Clear Lake

Magy Seif El-Nasr  
m.seifel-nasr@northeastern.edu  
Northeastern University

ABSTRACT

Interactivity and player experience are inextricably entwined with the creation of compelling narratives for interactive digital media. Narrative shapes and buttresses many such experiences, and therefore designers must construct compelling narrative arcs while carefully considering the effects of interaction on both the story and the player. As the narrative becomes more structurally complex, due to choice-based branching and other player actions, designers need to employ commensurately capable models and visualizations to keep track of that growing complexity. However, previous models of interactive narrative have failed to fully capture interactive elements with automated, operationalized visualizations. In this paper, we describe an algorithm for automated construction of a framework-driven, graph-based representation of interactive narrative. This representation more fully and transparently models structural and interactive features of the narrative than did prior approaches. We present an initial evaluation of this representation, based on modified cognitive walkthroughs performed by interactive narrative design and research experts from our research team, and we describe the takeaways for future improvement on interactive narrative modeling and analysis.

CCS CONCEPTS

- Computing methodologies → Knowledge representation and reasoning; Simulation evaluation; Applied computing → Computer games; Software and its engineering → Design languages.

KEYWORDS

Games, Interactive Narrative, AI, Computational Co-Creativity, Graphical Models, Design Representation, Structural Models, Design, Computational Media, Player Agency, Knowledge Representation

ACM Reference Format:


1 INTRODUCTION

When building any interactive experience, such as a game, it is critical for the designer to deeply consider and shape its interactive elements. They must form models of the player’s likely experiences and interactions, and consider how the player’s actions can impact their artifact and inform its reactions. This dialogue between the designer’s vision and the players’ perceptions shapes and defines the artistic and experiential results of the work.

With the increasing complexity of modern games, these experiences may be many and varied; they may depend deeply on the player’s choices, and the designer may have difficulty theorizing about player experience in a vacuum. Narrative-heavy games, such as roleplaying games and visual novels, can encompass a broad spectrum of designs. They can range from linear, movie-like experiences to highly open and branching ones. As a result, designers must make careful choices about how to structure their games to achieve their desired player experiences.

Interactive narrative often requires complex and intensive authoring to track and maintain the narrative through many potential choices and interactions. Authoring tools can assist designers in building a coherent, satisfying narrative by representing and visualizing the design appropriately. A good tool can organize important information, reducing the need for designers to keep separate, complex mental or physical models. Branches, choices, and other player interactions are sources of complexity and may benefit from helpful, clear representation. Authoring tools that can automatically assist
the designer in representing and reflecting on their design could prove valuable in aiding their analysis of these experiential aspects of their work.

Interaction, in this context, encompasses the flow and structure of the narrative, the opportunities for player input, and the responses to that input – including story progression, user interface, feedback, animation, and more [8]. When representing interactive narrative, it is important to capture all of these, and to present them in an approachable and comprehensible manner. This representation can be understood as an analogue to the “knowledge representation” component of an agent-based AI. It provides the knowledge base about the “world” of the game that can be used by other AI systems for creativity support and automated playtesting.

In recent work, we proposed such a representation, composed of multiple graphs to capture the scene flow, layout, scripts, and interactions [34]. However, this prior work did not fully detail the methods for computationally constructing such a representation, nor did it deeply explore the effectiveness of the representation in illuminating the design. It also neglected to deeply interrogate the representation’s flaws and limitations. Here, we describe the computational process for building such a representation, as well as an initial expert evaluation that we performed in order to evaluate and find opportunities for improvement in that representation.

The representation is implemented in the StudyCrafter platform for interactive narrative scenario creation. The tool enables novice designers to develop playable scenarios, often for the purpose of performing social science experiments [18]. StudyCrafter has been employed in several classrooms, resulting in a small body of student-built experiments that have been found to represent a variety of styles, strengths, and weaknesses [34]. We use two of those experiments, “An Unusual Situation” and “The Research Riddle,” as the subjects of our evaluation.

Our evaluation consists of in-person, in-depth interviews with members of our research team. Each researcher critically examined the representations of individual interactive scenarios using a modified cognitive walkthrough method [25]. This work’s primary contribution is to describe and critically evaluate our computational representation of interactive narrative scenarios, and outline requirements for developing a representation that fully encompasses and exposes a scenario’s interaction patterns and potential experiences. This understanding enables improvements to computational narrative analysis and future work on AI support for interactive narrative design, with a focus on modeling player interactions in a way that is interpretable and usable for designers.

2 RELATED WORK

Academic analysis of interactive narrative has a long history, dating back at least to the early 1980s with Laurel’s work based on Neo-Aristotelian theory [22], if not earlier [31, 37]. In these analyses, however, comprehensively representing player interaction has not usually been the primary focus. Most work has explored the potential and best-practices for authorship [27, 32, 40] and defined interactive narrative structure from the designer’s perspective, as in the works of Bernstein, Lindley, Ashwell, Short, and others [2, 4, 24, 30]. While these structures are useful for tracking the potential branches of a story, they do not explicitly represent the details of the player’s interaction with that structure.

In its totality, interaction includes such elements as the structural and choice affordances, narrative progression, UI, and other forms of feedback for player action. These elements were explored and outlined in recent work by Carstensdottir et. al. [8]. More recently, this research has expanded into a proposal for “interaction maps,” designed to capture some of these elements of interaction [9]. Here, we measure that framework and explore the ways in which it successfully captures elements of interaction, and, more importantly, the opportunities for improving it.

Player experience in interactive narrative has also been studied separately from structure, as in research on agency [7, 11, 15, 45, 46, 48], and on engagement and immersion [5, 6, 10, 13, 14, 29, 39]. This work, however, has usually operated at the level of holistic measurement of player experience, rather than at the level of detailed representation of each interaction.

In the IRIS project and following work, researchers have begun an effort to more directly extract and separate metrics of player experience in interactive narrative [35, 36, 47]. These metrics are promising, but they do not yet propose a fully computational, operationalized model of interactive narrative that would enable their proposed measurements and support designers. Recently, Partlan et. al. [34] developed such a representation to measure and expand on the “objective metrics” proposed by Szilas and Ilea [44] as part of that project, and here we evaluate and suggest improvements to that representation.

Outside of interactive narrative, researchers have performed formal modeling and analysis of game design. One of the most comprehensive such efforts is the development of Operational Logics, a theory that categorizes many families of game design elements [28]. To operationalize these logics, Osborn [33] has since proposed computational models of specific logics. The algorithm below can be seen as formally representing certain logics from several families, such as character-state, control, and linking logics.

Computational analyses of game design have employed graphs, such as Petri nets [1, 23] and Bayes nets [17]. Petri nets, however, quickly become too complex for manual analysis. Our layout graph, described below, is inspired by the D-nodes in Guzdial and Reil [17], but they focus on physics-based interactions extracted from gameplay videos, for machine learning and game generation, which does not necessarily support manual analysis nor interactive narrative. Bakkes and Dormans [3] proposed modeling mission design and physical level layout as two related graphs, but their work applied specifically to physical level layouts and was primarily generative. They did not deeply explore models of player interaction, design metrics, nor how to interpret existing designs into graphs.

Other researchers have developed programming-language-based models of game design, such as those proposed by Martens and Hammer [26], Gemini by Summerville et. al. [43] and the VGD by Schau [38]. These models are still in the early stages, however, and it is unclear how interpretable they will be for game designers. Our work focuses on modeling interactivity, narrative, and gameplay in transparent and visible form, to enable tractable analysis.

Finally, some researchers are exploring automated playtesting, in which AI agents play through games either in their original form or
We use these to build the edges of the scene flow map. Each scene’s layout is represented as a single node in the scene flow map, with properties that determine what scene will be loaded next. For each scene, we create a node in the scene flow map.

The scene flow map represents how the scenes connect to each other. In StudyCrafter, "end" points in the script for each scene contain a property that determines what scene will be loaded next. We use these to build the edges of the scene flow map. Each scene’s out-edges lead to the next scenes that may be shown, depending on the player’s actions.

For each scene, we build layout and script graphs, and we post-process the script graph into the interaction map. The scene flow map ties the representation together by keeping references to these other graphs associated with each scene.

### 3.2 The Layout Graph

There is a layout graph for each scene, representing the physical placement of objects in the visual environment. The scene may contain animation, but the layout graph contains information only about the initial placement of the objects. To construct the layout graph, we copy the properties of each object from the saved scene data, including its position, name, type, image, and so forth. We then build edges between each object and each other object, and we annotate the edges with the distances and directions between the objects’ centers.

### 3.3 The Script Graph

The script graph for each scene contains the code to operate the scene’s gameplay logic. StudyCrafter already contains a visual scripting language, which is displayed with a partially-connected graph-like view in the StudyCrafter editor. Our representation rebuilds this as a true directed graph, ensuring that we can use standard graph algorithms to operate on it.

We keep a data-driven mapping of StudyCrafter script node types to our representation, with specifications of the properties they may contain, including the types of data those properties represent.

As we encounter each node in the saved script data, we look up this mapping and load each of its properties into an appropriately typed object. Each script node, in addition to containing a set of properties, has a unique ID that we use to connect it with other script nodes. Specifically, we analyze the potential transitions from each script node, based on its type, and connect it to all nodes to which it may transition next. This process differs for different types of script nodes; for instance, an animation node has a single "next" node, whereas a branch node contains one or more condition statements, each of which may transition to a different node.

### 3.4 The Interaction Map

We post-process the script graph using static graph analysis to build the interaction map. This is possible because we have already built the graph-based representation of the script, enabling us to search through it and create nodes and edges based on its structure.

To build the interaction map, we begin by performing a depth-first search over the script graph. After building the start node, at each edge between nodes in the script, we construct the proper node type for the end of the edge, if applicable, by inspecting the script node. If the script node has a visible or interactive impact, such as displaying dialogue, animating a character, or modifying the UI, we build a new node of undetermined type; it might be a feedback or event node, depending on as-yet unknown context (i.e., whether it is related to prior choices by the player). If it has no direct impact on the player’s perception of the scenario (e.g., a hidden branch or changing a variable value), we build a temporary, “non-interaction” node just for tracking its edges, to be removed in post-processing. If it is a choice or event-based interaction, we build an interaction point vertex. We connect each edge as in the script.

There are a few special cases for the construction process. When we come across script nodes that are not related to player input, such as special events based on the passage of time or based on random branching, we mark them as event nodes, since they are
The fourth had experience with game AI and player modeling revolving narrative elements and performing player experience research. In one of the former evaluations) evaluated the representation for “An Unusual Situation.” In the third evaluation, a single researcher (who also was involved in one of the former evaluations) evaluated the representation for “The Research Riddle.”

The evaluation employed a modified cognitive walkthrough method. Cognitive walkthroughs are forms of usability analysis in which the facilitator steps the evaluators through the process of completing a task, asking questions at each step. The original questions are directly tied to usability (i.e. “Will the user notice that the correct action is available?”) [25]. We therefore replace the original questions with our own, designed to apply directly to the evaluation of a representation of interactive narrative scenarios. In our method, we perform two walkthroughs: one on the designer’s original view of the script of the scenario within the StudyCrafter editor itself, and one on our current graph-based representation.

In the first step in the protocol, experts play through the scenario to be studied using the StudyCrafter scenario player, see Figure 2 (a), as many times as they feel they need to understand and experience it in its entirety. After the experts are comfortable enough with the scenario to understand the script, the first modified cognitive walkthrough begins. StudyCrafter’s built-in visual script editor, shown in Figure 2 (b), serves as our point of reference for the existing visual representations for interactive narrative design tools. As experts examine the script for each individual scene in the scenario, we ask the following questions:

1. Would the representation, assuming it has been modified to capture anything already noted as missing earlier in the walkthrough, give you all the information you need about player interaction and narrative for this scene?
2. (If the answer is “no” to question 1) What information should be captured that currently is not?
3. What narrative structural features does this scene have that you would want to capture in a representation?
4. What interaction features does this scene have that you would want to see in the representation?
5. What other experiential features of this scene do you want to capture in the representation?

These questions are designed to target the narrative structure, the scenario’s interactions with the player, and any other aspects of the scenario’s design that affect the player’s experience. After recording the observations about each of these questions, we move on to the next scene in the scenario.

After completing the walkthrough of the script, Figure 2 (b), we progress to the second modified cognitive walkthrough. This second walkthrough uses the visualizations of the script graphs and interaction maps as described by Partlan et. al. [34] and seen in Figure 2 (c). As the experts examine these representations, we ask the following questions:

1. Would the representation, assuming it has been modified to capture anything already noted as missing earlier in the walkthrough, give you all the information you need about player interaction and narrative for this scene?
2. (If the answer is “no” to question 1) What information should be captured that currently is not?

There are several post-processing steps to complete the creation of the interaction map. We destroy non-interaction nodes and connect their predecessors and successors to each other, simplifying the map. We then need to determine which nodes are related to which interaction units. To do this, we perform a search through the graph starting at each interaction point, where other interaction points are treated as impassible (infinite cost). All reachable nodes, those that have a non-infinite cost, are marked with the interaction unit from which we began the search. We can now mark all nodes of unknown type that are related to an interaction unit as feedback, and all others as events (since they are unrelated to player input).

Finally, we must decide what to do with interaction points that can be triggered by the player at multiple times, such as button-press-based events. To let automated playthroughs traverse these events rather than getting stuck, we need to ensure that they have entry edges. In theory, there could be edges to these events from many nodes in the scene, as they can be triggered at multiple times. However, this would clutter the graphs. Therefore, we currently add indirect edges to events only from nodes that would otherwise be dead ends or cycles.

Currently, we build one interaction map for each scene. In theory, we could also build a map of the entire scenario, by combining the scene-based maps. This, however, we leave to future work.

4 EVALUATION METHOD

One researcher led three evaluations with different subsets of our research team serving as experts. Each expert was academically trained in computer science and game design, with three of the four experts having experience studying and analyzing games that involve narrative elements and performing player experience research. The fourth had experience with game AI and player modeling research. The experts also had prior experience with StudyCrafter and the representation. In two of the evaluations, separate pairs of researchers evaluated the representation for “An Unusual Situation.” In the third evaluation, a single researcher (who also was involved in one of the former evaluations) evaluated the representation for “The Research Riddle.”

The evaluation employed a modified cognitive walkthrough method. Cognitive walkthroughs are forms of usability analysis clearly not feedback. When we find functions, which can be called from multiple points in the script, we connect them as if they directly follow from each of their call points. Finally, when we find end nodes, we mark them with a special end node type.

In the third evaluation, a single researcher (who also was involved in one of the former evaluations) evaluated the representation for “An Unusual Situation.” In two of the evaluations, separate pairs of experts having experience studying and analyzing games that involve narrative elements and performing player experience research. The experts also had prior experience with StudyCrafter and the representation. In two of the evaluations, separate pairs of researchers evaluated the representation for “An Unusual Situation.” In the third evaluation, a single researcher (who also was involved in one of the former evaluations) evaluated the representation for “The Research Riddle.”

The experts also had prior experience with StudyCrafter and the representation. In two of the evaluations, separate pairs of researchers evaluated the representation for “An Unusual Situation.” In the third evaluation, a single researcher (who also was involved in one of the former evaluations) evaluated the representation for “The Research Riddle.”

Currently, we build one interaction map for each scene. In theory, we could also build a map of the entire scenario, by combining the scene-based maps. This, however, we leave to future work.

There are several post-processing steps to complete the creation of the interaction map. We destroy non-interaction nodes and connect their predecessors and successors to each other, simplifying the map. We then need to determine which nodes are related to which interaction units. To do this, we perform a search through the graph starting at each interaction point, where other interaction points are treated as impassible (infinite cost). All reachable nodes, those that have a non-infinite cost, are marked with the interaction unit from which we began the search. We can now mark all nodes of unknown type that are related to an interaction unit as feedback, and all others as events (since they are unrelated to player input).

Finally, we must decide what to do with interaction points that can be triggered by the player at multiple times, such as button-press-based events. To let automated playthroughs traverse these events rather than getting stuck, we need to ensure that they have entry edges. In theory, there could be edges to these events from many nodes in the scene, as they can be triggered at multiple times. However, this would clutter the graphs. Therefore, we currently add indirect edges to events only from nodes that would otherwise be dead ends or cycles.

Currently, we build one interaction map for each scene. In theory, we could also build a map of the entire scenario, by combining the scene-based maps. This, however, we leave to future work.

4 EVALUATION METHOD

One researcher led three evaluations with different subsets of our research team serving as experts. Each expert was academically trained in computer science and game design, with three of the four experts having experience studying and analyzing games that involve narrative elements and performing player experience research. The fourth had experience with game AI and player modeling research. The experts also had prior experience with StudyCrafter and the representation. In two of the evaluations, separate pairs of researchers evaluated the representation for “An Unusual Situation.” In the third evaluation, a single researcher (who also was involved in one of the former evaluations) evaluated the representation for “The Research Riddle.”

The evaluation employed a modified cognitive walkthrough method. Cognitive walkthroughs are forms of usability analysis in which the facilitator steps the evaluators through the process of completing a task, asking questions at each step. The original questions are directly tied to usability (i.e. “Will the user notice that the correct action is available?”) [25]. We therefore replace the original questions with our own, designed to apply directly to the evaluation of a representation of interactive narrative scenarios. In our method, we perform two walkthroughs: one on the designer’s original view of the script of the scenario within the StudyCrafter editor itself, and one on our current graph-based representation.

In the first step in the protocol, experts play through the scenario to be studied using the StudyCrafter scenario player, see Figure 2 (a), as many times as they feel they need to understand and experience it in its entirety. After the experts are comfortable enough with the scenario to understand the script, the first modified cognitive walkthrough begins. StudyCrafter’s built-in visual script editor, shown in Figure 2 (b), serves as our point of reference for the existing visual representations for interactive narrative design tools. As experts examine the script for each individual scene in the scenario, we ask the following questions:

1. What narrative structural features does this scene have that you would want to capture in a representation?
2. What interaction features does this scene have that you would want to see in the representation?
3. What other experiential features of this scene do you want to capture in the representation?

These questions are designed to target the narrative structure, the scenario’s interactions with the player, and any other aspects of the scenario’s design that affect the player’s experience. After recording the observations about each of these questions, we move on to the next scene in the scenario.

After completing the walkthrough of the script, Figure 2 (b), we progress to the second modified cognitive walkthrough. This second walkthrough uses the visualizations of the script graphs and interaction maps as described by Partlan et. al. [34] and seen in Figure 2 (c). As the experts examine these representations, we ask the following questions:

1. Would the representation, assuming it has been modified to capture anything already noted as missing earlier in the walkthrough, give you all the information you need about player interaction and narrative for this scene?
2. (If the answer is “no” to question 1) What information should be captured that currently is not?
We expect answers to these questions to be related to the desires for the representation that the experts formulate based on the first walkthrough. We also expect unforeseen problems and new ideas to emerge as the experts examine and react to the existing representation. During this process, for both walkthroughs, new ideas are noted as soon as they are clearly described, after which the team moves on. This is to prevent defensiveness or long design discussions, an approach recommended by Spencer [42].

### 4.1 Evaluation Scenarios

The two scenarios employed in our evaluation process were created by game design students in a graduate-level research methods course. The students were prompted to design scenarios that would serve as social science or psychology experiments.

*“An Unusual Situation”* begins with a framing narrative and tutorial. Then, it presents a narrative in which the player’s character is, depending on an initial randomized condition, either publicly praised or insulted by an old friend in front of a large gathering. Next, the player is given a choice to punch or not punch someone who looks like that friend. Finally, the scenario presents a debrief about the research goals and theories involved.

In *“The Research Riddle,”* the player is either given a specific character or asked to choose between several characters. Then, they are presented with a situation in which a researcher needs their help in finding the answer to a (made up) academic question. To find the answer, the player must navigate a looping set of user interface menus, choices, and text entry fields, simulating a directed search through journals and computer software.

We selected these two scenarios based on their use of interactive narrative elements. They were chosen due to their employment of a variety of interactive affordances, including character selection, UI interactions, dialogue choices, and text entry. Compared to other available scenarios, their narrative framing, characterization, and feedback for player choices presented more opportunities for analysis and discussion.

### 4.2 Data Analysis

We analyzed the results of the cognitive walkthrough-based evaluation by extracting all comments that were relevant to the representation or user experience, and categorizing them into emergent groups based on their themes. We separated comments first by their topic: the experiences of playing through the scenarios, insights available from the original StudyCrafter scripts, experts’ expectations for the computational representation, and comments about the actual representation. In each topic, we further defined themes: reflections on narrative design and structure, player interaction, content and characters, and experiment design and style.

### 5 RESULTS

The analysis produced four distinct themes, and further sub-themes within each, derived from the expert comments. Table 1 lists these themes for the playthroughs and script walkthroughs, and Table 2 does so for the representation expectation discussions and walkthrough-based reviews. In this section, we will elaborate on particularly important subsets of the themes. We combine our discussions of the playthrough and script walkthrough sections of Table 1 for brevity, as both sets of comments relate to the original StudyCrafter representation of the scenarios.

#### 5.1 Narrative Design and Structure

Comments related to this theme tended to discuss how and to what extent the narrative changed in response to player choices. They also related to the use of particular narrative patterns, such as framing narrative or exposition. For a more complete discussion of structures of interactive narrative, see [8], as well as [2, 4, 24, 41].

**Playthrough and Script:** During the playthroughs and script walkthroughs, the experts most frequently commented on the linearity of the scenarios. They often focused on structural themes, such as the lack of structural impact of choices and the looping and hidden structures of puzzles.

**Representation Expectations:** The experts expected the representation to clarify what conditions led to specific branches. If variables were used to create long-term dependencies or branches on prior choices, they wanted the representation to capture that. They also wanted to be able to see randomized variable assignments.

**Representation Review:** In general, the experts felt that the representation successfully captured linear and branching structures. They felt, however, that it needed more details, especially in terms of dependencies.

#### 5.2 Player Interaction

Player interaction encapsulates comments about the types and frequencies of player-driven choices and actions, the feedback for and results of those actions, or the lack thereof.

**Playthrough and Script:** The experts repeatedly discussed agency. For example, in *“The Research Riddle,”* character selection had too little impact on the story to lead to feelings of agency. They also noted that the provided options were sometimes unexpected or did not reflect all actions the player might want to take.

**Representation Expectations:** The experts hoped to clearly see how and when the scenario provides feedback for actions, and whether branching is visible or invisible.

**Representation Review:** The experts agreed that the feedback was clearly marked, and that branching was visible. In general, they had mostly positive comments about this theme. However, they noted a need to account for differing perspectives between players.

#### 5.3 Content and Characters

Though the representation focuses on structure and interactivity, the experts often commented on the aesthetic and characterization elements of the narrative.

**Playthrough and Script:** The experts discussed the relationships between, perspectives of, and setting behind the characters, as well as the player’s and narrator’s relationships to those elements. In *“An Unusual Situation,”* for example, they pointed out the scenario’s use of an unreliable narrator.

**Representation Expectations:** The experts expressed a desire to see analysis and context for dialogue and narration, including character relationships and perspectives.
### Representation Review
The representation did not fulfill most of these expectations, since it is heavily focused on structure. One easily addressable comment was that the “caption” presented alongside certain dialogue choices should be treated separately, as either an event or feedback.

### Representation Expectations
The experts sought clarity on the types and presentation styles of choices, interactions, and feedback. They also wanted information about the context of the scenario and its relationship with the player.

### 5.4 Experiment Design and Style
This theme reflects the experts’ comments on how the scenario design and the designer’s individual stylistic choices led to particular experiences and affected the experimental validity.

#### Playthrough and Script
“An Unusual Situation” presents an example of this interplay between stylistic choices and player experience. Experts felt that the choices and unreliable narrator were artificial and manipulative. As a result, they argued that it was unsuccessful in testing its hypothesis.

#### Representation Review
The experts found the representation confusing in some cases, due to designers’ stylistic preferences causing unusual results. They also recommended collapsing and combining nodes to reduce complexity.

### 6 DISCUSSION
How do we make sense of these detailed and complex insights gleaned from the cognitive walkthrough process? We begin by determining which aspects of the representation were successful, according to our experts’ goals, and then continue by analyzing the desires, concerns, and comments about the representation to find actionable, operationalizable feedback.
### 6.1 Representational Successes

The algorithm described in 3.4 is a new model for static analysis of interactive narrative capable of transforming narrative scenarios into a visual and structured representation. It was used in previous work to build a set of metrics [34] for StudyCrafter scenarios, however it is generalizable to other contexts. For instance, we have manually analyzed other interactive narratives using the Interaction Map framework that underlies parts of this representation. With modifications to incorporate particular interaction affordances of other game design tools, a similar algorithm could automatically derive representations for scenarios built in those tools.

The evaluation walkthroughs highlight several ways in which the representation reveals and clarifies aspects of the scenario designs that our experts consider important. The graph-based structure of the representation enabled our experts to visually comprehend several relevant features of the scenarios: their linearity and branching structure, the relative lengths of branches, and the possibility spaces of interaction. Whereas the script editor built into StudyCrafter felt disorganized and overly complex for this analysis, the representation alleviated the confusion.

The experts found the clarity of the interaction maps’ annotations of which nodes represented events versus feedback for player actions to be relevant and helpful. This visual depiction of the flow of particular actions and dialogue could enable designers to track whether, and to what extent, players would receive specific feedback for their choices. Had the designers who built these two scenarios visualized their linearity, their limited player interaction, and the lack of meaningful feedback for some of those interactions, they might have been able to use those insights to ameliorate some of the design choices that led to negative comments from our experts.

### 6.2 Elaborating and Clarifying the Representation

Though the representation fulfilled many of the experts’ desires and clarified several aspects of the scenario designs, the experts also described several potential improvements, and they commented on several points of confusion. These included a lack of detail about branches and variable usage, the need for further clarification of perceived events versus feedback, a desire for additional annotations of specific sub-types and perceptual attributes of nodes, and a request for collapsing or merging of nodes to reduce clutter.

In the interaction map, we chose to depict only visible interactions with the scenario, theorizing that the designer would want to see only the parts of the scenario that were directly perceivable by the player. Experts, however, found the resulting connections between nodes, as shown in Figure 2 (c), confusing and difficult to interpret. Because we did not include branches as separate nodes,
it looked like players could transition back to previous nodes when, in fact, the scenario was simply waiting for them to continue.

This particular interaction also appeared to allow any branch to be reached from any button. Only one transition would actually occur, depending on the button pressed, but we did not analyze the variable usage to show this. Visualizing variables and conditions, along with a separate branch node, would resolve this issue and enable designers to understand the actual flow of their scenario.

This ties into the more general need the experts expressed to better understand dependencies in the scenario. Many choice-based narrative games, and narrative-heavy games more broadly, include puzzles or gating on prior actions to unlock future interaction options. Thus, we see a general need for dependency visualization.

The experts also requested additional annotations in other cases. In the interaction maps, we visualized the overall types of nodes (interaction points, events, feedback), but we did not fully explain the ways in which those nodes would be presented to the player. Was an interaction point a dialogue choice or a button press? Was a feedback node an animation, a sound, or a dialogue? The experts would need to refer to the script graph to find out, and the connections between the graphs were not always easily determined. We could annotate these categories on the interaction map.

The experts also felt that some nodes, such as repetitive visibility-change actions that had little effect on player experience, were redundant and unimportant. We could combine or collapse some of the nodes that are merely linear segments of events or feedback, replacing them with compact annotations. Alternatively, we could build a dynamic viewer for the representation that allows expanding and collapsing sections as designers work at various levels of granularity. This might also help us to build graphs of the entire scenario, combining all scenes, as requested by our experts.

A more dynamic set of views could also be beneficial in visualizing differing player perspectives. We could annotate certain nodes as feedback in some such perspectives, based on the player’s knowledge of the consequences of choices gained by repeated playthroughs. In a first-playthrough perspective, however, the experts could be marked as events due to lack of knowledge of the narrative’s scope. This is feasible because the representation is already designed to support structural analysis of interaction and its effects.

Finally, the experts suggested several improvements that are significantly out of scope for our current representation. These are primarily related to deep analysis of the content and meaning of a narrative, such as analysis of the perspectives and viewpoints of players, the relationships between various characters, and between a player and those characters, the use of biased language and stereotypes, and other questions of culturally-informed meaning.

7 CONCLUSION

We have described our procedure for constructing a theory-driven representation to model narrative-driven games, as operationalized for the StudyCrafter platform. We have presented an initial evaluation of this representation, in which experts on our research team used a cognitive walkthrough method to perform a detailed examination of two interactive narrative projects. The evaluation indicated that our representation was successful in illuminating the structures and interactive elements of these scenarios. On the other hand, it also revealed significant shortcomings and points of confusion. The evaluation itself was limited in scope, in that it only included members of our research team.

The insights from this expert evaluation can be useful to other game designers, researchers, or critics. The cognitive walkthroughs evinced many critiques of the scenarios’ narrative structures: the overuse of framing narrative, the linearity and lack of interaction opportunities, and the feelings of constrained agency that resulted from these choices. These areas of focus and critique, and the representations and procedures that enabled them, may be relevant to many other games and interactive experiences.

By visualizing the narrative structure in graph form, we empowered experts to develop critiques with reference to specific points in the scenario design. They interrogated particular design decisions, such as the use of complex branching structures to simulate simple dialogue choices. Combining the representation’s visual structure with the cognitive walkthrough method ensured that each moment of the scenario could be fully explored and described with clarity, common reference points. A melding of theory-driven visualization and careful design review can thus be a valuable combination for work on interactive experiences.

In limiting the evaluation to members of the research team, however, this evaluation is necessarily constrained by our own perspectives, biases, and goals. Several members of the research team who acted as experts were also involved in the development of the representation, which influenced their comments. Future studies should incorporate designers from varied backgrounds, with various goals, to evaluate the representation in varied contexts.

Additionally, the scenarios in this study were built by designers who were not involved as experts. It would be useful to study designers’ impressions of the representation when analyzing their own work, and how those impressions differ from those of external experts who did not create the scenario.

We plan to iterate on and improve the representation in response to the experts’ feedback. First, we will update the Interaction Map framework to incorporate additional node types, annotations, and levels of detail to respond to the experts’ feedback. Then, we will build a new computational implementation and visualization that incorporates this new theory and better supports designers. We plan to test this improved version of the representation in the context of designers’ real workflows.

These representation efforts are building towards an AI-assisted creativity support tool that incorporates the representation as a sort of “knowledge base,” much as other game AI agents use a perception and world representation system. We believe that these insights about how experts use our representation to evaluate interactive narrative scenarios will be essential in building automated tools that prompt reflection, provide feedback, or generate suggestions for narrative-focused games.

ACKNOWLEDGMENTS

This work was made possible by a grant from the NSF (IIS-1736185). Other work on StudyCrafter has also been supported by DARPA and Northeastern University. Thanks to the entire StudyCrafter team for making this work possible, and especially to Dylan Schuten for leading important organizational efforts.