Analysis of ReGEN as a Graph Rewriting System for Quest Generation

Ben Kybartas and Clark Verbrugge

***Abstract*—Using procedural narrative generation in video games provides a flexible way to extend gameplay and provide more depth to the game world at low cost to the developers. Current examples of narrative generation in commercial games, however, tend to be simplistic, resulting in repetitive and uninteresting stories. In this paper, we develop a system for narrative generation using a context-aware graph rewriting framework. We use a graph representation of the game world to create narratives which reflect and modify the current world state. Using a novel set of metrics to evaluate narrative quality, we validate our approach by comparing our generated narratives to other procedurally generated stories, as well as to authored narratives from commercially successful and critically praised games. The results show that our narratives compare favourably to the authored narratives. Our metrics provide a new approach to narrative analysis, and our system provides a unique and practical approach to story generation.**

***Index Terms*—computer games, narrative, procedural content generation, graph rewriting**

I. Introduction

SIDEQUESTS, minor stories or tasks tangential to a game narrative are often used as additional content in narrative-oriented computer games. Such stories enhance gameplay by augmenting the core game goals with a variety of interesting, usually optional possibilities for the player to experience. An abundance of sidequests extends gameplay, enriching the virtual environment and giving players motivation to fully explore a game world, increasing the sense of depth in the virtual world and thus deepening the immersive experience. For larger games, however, producing large numbers of sidequests can become a significant development cost, and various games have explored automatic generation as a potential solution. Beyond the complexities of generating appropriate accompanying text, naively generated story/quest structure tends to be easily recognized as repetitive by players [1], [2], with a consequent reduction in player interest and motivation. This problem is exacerbated by the need in real games for automatically generated quests to always be completable, and to avoid perturbing the more important main quests that drive the storyline, and thus often end up being overly simple and extremely generic in structure.

Our approach to story generation focuses on developing a feasible technique that avoids the more obvious shortcomings of traditional quest generation, and in particular we aim to produce quantifiably good story structure, appropriate for use in minor sidequests. Our design is based on a *graph rewriting* approach, applied to a simple but formalized representation of the game state or context. Use of graph rewriting allows us to grow a story to arbitrary proportions and with arbitrary complexity through the repeated application of relatively simple narrative structure patterns. Building the system with awareness of the underlying context ensures the quests match the game state, and can have an observable and interesting impact on the game world, without overly interfering with the rest of the game design. Use of graph-rewriting builds on the advantages of grammar-based designs over goal-oriented designs in guaranteeing story correctness/validity throughout an incremental development process, with the significantly greater flexibility of graph-based rule specification and design. This allows us to generate story structures that have both non-trivial complexity and actual game relevance in combination with the incorporation of contextual requirements, and which are not reduced to simple, repetitive tasks.

Generated narrative quality is justified and further guided in our system by the design of several metrics that attempt to measure story quality from different perspectives. We define a number of properties based on analysis of the story graph structure, and in relation to the game state, that can be used to determine whether a narrative structure is of an appropriate "scale" for a sidequest in terms of size and complexity, the degree of repetition, as well as more abstract properties such as narrative depth or richness. We validate these metrics by quantifying the narratives found in two popular, large and modern role-playing games (a genre well known for the importance of narrative in game play), and comparing the results found for our generated narratives. Quantifying story quality with respect to player experience is of course a difficult problem that depends to a large extent on the elegance of story description and other artistic properties, but our approach shows that many aspects of basic structure are also important parts of the narrative design, and moreover are properties that can be captured and used to guide narrative generation.

Specific contributions of our work include:

- We describe a context-sensitive graph-rewriting approach to automatic story generation. This technique has been much less explored than search-based, goal-oriented approaches and combines awareness of game context with a flexible strategy for incrementally growing narratives to arbitrary complexity.
- To quantifiably validate our narratives we define a number of game narrative metrics. These (mainly) graph-based measurements allow for numerical comparison of narrative quality, measuring both basic properties as well as abstract properties such as narrative "richness."

School of Computer Science, McGill University, Montréal, QC, H3A OE9 Canada (e-mail: ben.kybartas@mail.mcgill.ca, clump@cs.mcgill.ca).

- Using our metric design, we show our narrative generation framework can generate narrative structures of better quality than other systems, comparable to the quality of manually designed modern games.

In the next section we discuss related work in the field of narrative generation as well as narrative analysis. Section III describes our overall design for a narrative generation system using graph rewriting. In Section IV we present the metrics used for our narrative analysis, followed in Section V by experimental analysis computing and comparing metrics for our system with another, successful narrative generation approach as well as with modern commercial games. Lastly, in Section VI we conclude and discuss directions for future work.

II. Related Work

An early example of a formal story structure for describing Russian folk-tales was proposed by Propp in 1928 [3]. This structure consisted of thirty-one "functions" through which the hero of the story progresses in a linear fashion. A "function" in essence refers to an event in the story, such as the hero's departure from home, or the defeat of the main villain. Additionally categorized were the main characters prevalent in each of the folk-tales. This concept of defining a narrative in a formal way according to story structure and events, called *formalism*, is the approach we take to defining narratives in our work. A critique of this method argues that simply understanding the structure of a story is ignorant of symbolism of cultural significance [4]. This critique is valid and represents a continual limitation of many narrative generation techniques, but our system is aimed at game narratives, which in most modern games remain structured in a formal way. Elements such as symbolism may be prevalent in a game narrative but are not reflected in narrative structure.

Grammar-based Narrative Generation Systems

Colby presents a story grammar for analyzing Inuit folk-tales which follows a similar form pattern to Propp's [5]. The main story is a string generated from the three main categories: *motivation*, *engagement* and *resolution*. Each of these categories can contain a sequence of many possible narrative events called *eidons*. The final narrative is therefore represented as a string of *eidons*, similar to the narrative structure in modern games. Rumelhart created a more generalized grammar aimed at being able to generate any form of narrative instead of just folk-tales [6]. The stories resulting from this grammar consisted of a linear progression of narrative events that could either *cause* or *allow* different narrative events or evoke reactions to events by elements within the narrative. The sequence of these events and their results would constitute the final story.

An argument against the lack of formal proof and experimentation led Black and Wilensky to perform a rigorous evaluation of story grammars [7]. It was argued that there were certain story formats such as non-linear narratives and embedded narratives which could not be generated by many of the existing story grammars. Also, certain non-stories could be generated such as instructional manuals. The inability for story grammars to account for meaning and symbolism was a critique of the argument that story grammars could be used to represent *any* story [8]. Our system is geared specifically towards game narratives and even more specifically sidequests within role-playing games, allowing us to create a specific grammar. This avoids the pitfalls of trying to create a general grammar.

Goal-oriented Narrative Generation Systems

One of the earliest narrative generation programs, TALE-SPIN, did not use a grammar approach, but rather used a *character goal-oriented* approach to creating stories [9]. TALE-SPIN contains a *map* of the world wherein the narrative takes place. This map contains information about the locations, items and characters within this narrative world. To create a narrative the program must be given a goal. It then biases the world in such a way that facilitates achieving the desired goal. The system then simulates this virtual world, and the linear sequence of events which occur therein produce the output for the story.

TALE-SPIN presented a planning approach to generating narrative, centered around achieving the goals of the characters in the world, and not the goals of an author. Other planning based systems focus on the *authorial goals* in which the planning methods still attempt to account for the desires of an author. Porteous *et al.* created a narrative generation system focusing on achieving specific authorial goals, which they referred to as *constraints*, and placed this system in a form of narrative game called an *interactive story* [10]. The game's setting used Shakespeare's *The Merchant of Venice* as the base story, using a goal-oriented planning approach, but where re-planning was used to account for the changes made in the game world environment by the user. At the same time, the system planned in such a way that certain *constraints* imposed by the author were met, expressed through the use of important scenes in the original play. Riedl and Young 's IPOCL system attempted to balance both the authorial and character goals by planning events in such a way that achieving the character goals also achieves the authorial goals [11]. Our system, while using graph-rewriting over planning, likewise aims to balance both authorial and character goals by ensuring that each segment of the narrative generated would correspond to the desires of the characters involved within the narrative.

Emergent Narrative Generation Systems

The world in which the story takes place is considered paramount to many of the developed narrative generation systems. *Virtual Storyteller* is one such system, where the main focus was to make a dynamic story world and simply recount events in order to construct a narrative [12]. The dynamic world was achieved by giving each character a goal and a personality as well as relations to different characters. A later version of the same tool modified the system to pick only specific events from the game world which corresponded to the viewpoint of a specific protagonist, since the original output was frequently found to simply be a disinteresting and disconnected set of events [13]. Chang and Soo provide an additional system of this form wherein they create a social world resembling the world of William Shakespeare's *Othello*, and give their characters beliefs and motivations to allow for actions such as deception and misjudgement [14]. They propose their system as a means for making NPCs more socially reactive, and therefore their form of narrative generation is not aimed at creating narratives within a game in which an unpredictable game player is in the role of lead protagonist. MEXICA has a similar focus, generating stories by picking the next event in the narrative considering the past actions of the characters in the world, as well as how these events modify interpersonal relationships [15]. Our system likewise considers interpersonal relationships as a means of generating narratives which are *believable*, given the world in which they occur. A recreation of the classic story *Madame Bovary*, focused on giving the user control over the social relations between characters in the game world, and relied on narratives *emerging* from how these changes affect the NPCs and their actions [16]. This was achieved by having the user physically read the lines of game dialogue into a microphone. The game then detects the user's emotion and updates the game characters according to the perceived emotional state of the user. *Façade* allows the user to talk to two characters within a game, and what they say modifies the relationship between the characters, and the relationship between the characters and the user [17]. These dynamic alterations produce noticeable changes in what the characters say, and in their interactions between each other, creating a unique narrative for each playthrough provided the user behaves differently each time. *Prom Week* places the user within a social environment mimicking that of a group of high-school characters several days before the prom. The user must then complete certain goals, such as becoming the prom king/queen, by manipulating the relations between characters to gain favour or disapproval [18]. The above three games are often referred to as *social games*. Their importance to narrative generation is that there is no predefined narrative; rather, the games are structured so that the actions and interactions of the NPCs create the narrative, and the player's interaction in this environment is to manipulate these actions and interactions. This type of narrative generation is often referred to as *emergent narrative* since the narrative is expected to emerge through these interactions, but there is no formal narrative defined within the game. As with the systems above, our narrative generation system attempts to model the game world and social environment. By this, we mean that our game world consists of a set of characters and the relations between characters, but also the different locations and objects within the game world as well. Our system is different, however, in that we are generating a defined narrative, rather than relying on an emergent method, and the purpose of the game world is to make reasoned decisions about how this narrative is to be structured.

Graph-Grammar and Quest Generation Systems

For our system we will be using a graph grammar to define our stories, similar to the string grammar approach used notably in the folk-tale generation system by Colby [5]. Our system, however, uses graph grammars to overcome some of the limitations of a string grammar approach. An example of this is that in representing the story as a graph we are able to provide branching stories. This is a relatively unexplored field as most generation systems tend to use goal-oriented or emergent techniques. A notable rewriting approach is found in SQUEGE, which represents narratives as a graph and uses rewrite rules to create a unique narrative based on a set of game objects [19]. The system was later expanded to have the quests converted into scripts for CD Projekt RED's *The Witcher* [20]. SQUEGE takes into account the objects within the game world, but is restricted in that it never considers the *relationships* between these objects. Graph rewriting has also been shown to be effective in generating the shape and contents of a game "dungeon" [21]. A skeleton of the dungeon is created in the form of a graph, stating where important events should occur. Situations such as puzzles, fights and boss-battles are then added in. A *shape grammar* is then used to define the shape of the final dungeon. The concept of quest-specific generation can also be compared to the military training tool by Zook. *et. al* [22]. The authors used a planning techniques to generate small, self-contained scenarios to be carried out by the user training with the system, and even liken this structure to that of quests in role-playing games. As opposed to evaluating the quality of their narratives using metrics, as we have done, they instead evaluate each narrative against the user's skill level, and generate quests that assist the user in increasing their skill level.

Narrative Quality Metrics

One of the especially interesting aspects of this work is in attempting to find metrics that in some respect represent the quality of a game narrative. Previous examinations of narrative quality are usually carried out by means of performing a human survey. One experiment carried out by Peinado and Gervás aimed to have the participants manually rate a story on the *linguistic quality*, *coherence*, *interest* and *originality* [23]. The GADIN tool aimed to create soap-opera style narratives, and the narratives were evaluated using a "Turing test" where participants were asked to pick, out of two stories, which one was generated by the system and which story was a soap-opera plot [24]. Other work has been focused on analyzing certain features of a narrative, such as presenting metrics to measure the quality of conflict [25]. These metrics were defined as *balance*, *directness*, *intensity* and *resolution*. A set of metrics to judge the *novelty* of a generated narrative compared to the previous narratives generated by the system was presented [26]. The similarity of characters and actions between stories was examined to compare the similarity between these narratives. This approach is perhaps similar in motivation to our design for a *uniqueness* metric but aimed between narratives, rather than within a narrative. Tomaszewski chose to evaluate the quality of his *Marlinspike* engine according to the amount of 'reincorporation;' *i.e.,* the number of player actions that were referenced by or which influenced later events in the narrative [27]. This measure is similar to our measure for *narrative richness*, although we evaluate our metric according to the sidequests which were influenced by player actions in previous sidequests. Throughout this paper, we will use the term *inter-narrative* metrics to define metrics between narratives and *intra-narrative* metrics to define metrics within a single narrative.

III. Narrative Generation

In our approach to narrative generation we combined the concept of using a formal representation of the game world in conjunction with *graph rewriting*. We call this system *ReGEN*, which stands for REwriting Graphs for Enhanced Narratives. In our case we are interested in generating *side-quests* for *role-playing games*, and thus within this paper we define a narrative as being an instance of a side-quest. Developing a system for side-quests has strongly influenced the decisions we made in designing our generation system. Our criteria for a reasonable system are:

- Being powerful in its ability to generative narratives but also easy to implement and simple for potential users to understand.
- Generating narratives which are reactive to the game world environment in which they occur.
- Providing uniqueness between narratives, and variety to the individual actions occurring within the narrative itself.

As opposed to many previous generation tools, we will not be working with emergent narrative although we do employ the concept of using a dynamic social environment to create engaging narratives. We do not call our system emergent, because we use this environment to create the narrative in full, rather than relying on stories emerging based on the interaction between the player and characters. This avoids some of the pitfalls of emergent narratives, such as the possibility that the stories will merely feel like disconnected narratives, a problem reported in the *Virtual Storyteller* system [13]. Our tool also allows for arbitrary authorial control, which is a short-coming noted in many emergent systems [17], [14]. The heightened level of control we provide with the ReGEN system allows the author to directly affect the quality and structure of the narratives produced.

ReGEN is likewise not goal-oriented, although we follow the same concept of generating a complete narrative and we attempt to balance character and authorial goals. Rather than use planning techniques we start with a basic story and progressively build it up using graph-rewriting techniques. We maintain the authorial goals by providing the author complete control over the set of rewrite rules used by the system. Although we do not explicitly model character goals, we still believe our system maintains character goals in that each narrative is centered around the player attempting to achieve the goals of a character in the game world. By avoiding a direct goal-oriented approach, we avoid the concerns involved in goal-oriented planning. For example, any choices made by the game player could result in the planner having to re-plan a path to achieve the desired goal, as with the planner used for *The Merchant of Venice* game [10]. Computational cost associated with replanning can be quite high, and imposing limitations on cost result in incomplete narratives. Also of importance is that we aim for a consistent game world that we do not necessarily modify to suit our planning approach. Such an approach was used in, for example, the TALE-SPIN system which begins by biasing the game world in such a way that makes the desired goal possible to achieve [9], but has disadvantages in terms of generality and ensuring (side)quests do not interfere with the main, manual narrative design.

Our design starts by generating a minimal base narrative, one with a definite beginning and end. We then apply rewrite rules to to flesh out the story and make it interesting. The system is designed so that the rewrite rules never modify the beginning or end state of the story, meaning that we can guarantee complete narratives. Additionally, the author has complete control over the narrative and any changes in the narrative, meaning that the author can impose additional consistency or other constraints as desired, while still ensuring a complete and coherent narrative.

We will begin by presenting the two main components of our ReGEN system, and how both components affect each other and the narrative. Following this, we will discuss the stages of the generation process and the application of rewrite rules.

Components of ReGEN

There are two main components to our graph-rewriting system, the *game world* and the *narrative*, both of which are represented by graphs. The *narrative* is generated by looking for potential stories which can arise out of the objects and relations in the *game world*. Conversely, the actions taken in the *narrative* modify the objects and relations in the *game world*.

The *game world* is a directed labelled multigraph where the nodes in the graph indicate objects and the edges indicate relations. An example of an object could be a *non-player character* (NPC), item or location. An object can contain any number of *attributes*. All objects, for instance, will have a *type* attribute which states if they are an NPC, or a location or any of the other possible object types. One or multiple *relations* may exist between any two objects and these can take many forms. In our system we define a relationship as having an *identifier* and an optional *reason*.

Fig. 1. A basic example of a relationship between two NPC objects

In Figure 1 we see a simple example of a relationship. The identifier is *Hates*, which implies that King Arthur *hates* Morgan Le Fey. The second term, *evil*, describes the **reason** for the identifier; *i.e.,* King Arthur *hates* Morgan Le Fey **because** Morgan Le Fey is *evil*.

Note that all relations are directed and not bi-directional, King Arthur may hate Morgan Le Fey, but Morgan Le Fey does not hate King Arthur in return. Furthermore, we do not place any limits on the number of possible relations between objects. King Arthur may hate Morgan Le Fey for one reason, but love Morgan Le Fey for another reason. This allows for a variety of complex interrelations. We find this system to be simple way to represent the game world and is also similar to the way data is stored in many RPG games.

The *narrative* is represented as a directed acyclic graph with labelled nodes. In the narrative, the nodes represent *events* and the edges indicate the *links* between events. Each event represents, as its name implies, a basic event in the narrative. An event may also have multiple attributes, but requires a *target*. The target refers to the object in the game world which is the focus of that narrative event. For example, an event where the player walks to a specific location will have a location object as its target. We call the set of all the target objects for our narrative the *cast*. Links in this case simply give the ordering of events in the narrative. We impose the restriction that a story graph may not contain cycles. This means that, following the story sequentially, it is impossible to visit the same event more than once.

Fig. 2. A basic sample of a narrative relating two narrative events

A very basic set of two events and their link is shown in Figure 2, with the two events connected by a single link showing that the retrieve object event occurs before the return event. Note that from a gaming perspective, each of these events relate to one action that must be taken by the user. In Figure 2, the player would be required to acquire the Holy Grail, and then would have to return to Camelot. If the player returns to Camelot before acquiring the Holy Grail, that will not allow the character to skip the previous event. This is the structure used in most commercial RPGs, where the player is free to roam around the world, but in order to progress through the actual quest, they must perform an ordered series of actions.

One attribute, which is important for narrative rewriting and is later used for our metric analysis, is the event *type*. We would, using Figure 2 as our example, call the first event a *retrieve* event and the second a *return* event. Using this allows us to determine the variety of actions performed by the user in a given narrative which we will later use to create a metric describing the number of *unique* actions taken by the player in a given narrative.

By using a graph, we can likewise allow for *branching narratives*. Since our research is centered within a gaming environment, a branching narrative indicates an event in the narrative after which the player is given a choice about which action they wish to perform next, resulting in two possible narrative events. The narrative the player experiences is the path the player takes from the starting node to the ending node. Since the player is not restricted in taking any path, whenever a given event is linked to two or more subsequent events, this indicates a branch in the narrative.

Generation Process

The generation process in ReGEN is based on graph rewriting. Graph rewriting is a generalization of the string rewriting strategy commonly associated with computer language grammars. In graph rewriting one defines rules that search for patterns in graphs (as opposed to strings), rewriting the resulting matched areas to produce a new graph. This allows us to dynamically change any aspect of the graph arbitrarily. A graph rewriting system thus consists of an initial graph along with rewrite *rules* which, much like a grammar rules, consist of both a left-hand side and a right-hand side. The left-hand side shows the pattern being searched for in the main graph, and the right-hand side shows the way that resulting pattern will be rewritten if found in the input graph. While the process of searching for a pattern within a graph is NP-complete in general, we find that the cost of this search is minimal on the size of graphs used within our system. Having labelled edges and nodes further reduces this search time. For example if we are looking for two NPCs who love each other, we search only through NPC nodes and only through "love" relationships.

The first step in our narrative generation process is to examine our game world environment for a *potential story*. To do this we define what could be known as a set of *initial rewrite rules*, which we will refer to as IRR from here on. It is using these rules that we create our initial narrative. The IRR has a left-hand side which searches for a condition within our game world, which we will henceforth call the *game world condition* and the right-hand side generates a narrative. The game world condition is also a directed labelled multigraph and the system checks if this graph exists as a subgraph within our main game world graph. A potential condition for say, a murder story, would be to locate two NPCs within our game world, where one NPC has a hate relation to another NPC. If this condition exists in the game world, then the resulting right-hand narrative is generated. The right-hand narrative is user-defined and consists of a complete narrative as defined above, by using a directed acyclic graph. We impose the condition that the narrative defined must have a beginning and end and consists of at least a single event. The length and detail of this narrative is completely customizable. We find that this helps alleviate the disconnect created between author and narrative found, for example, in emergent narrative systems. The cast in the resulting narrative is derived from the results of the condition we searched for in the game world graph. In our murder story, for example, we would define an event where the player must talk to the first NPC to receive the murder request. This would be followed by a murder event with the target being the second NPC.

At this point in time our generation process resembles the *Radiant Quest* system from *Skyrim* in that we have essentially created a skeleton narrative which is filled in by targets in the game world [2]. The main difference is that the *Radiant Quest* system looks primarily at types, for instance a role can be filled by a "bartender' and this could be any "bartender" within the *Skyrim* universe. Our system uses a conditional graph representing all roles and their interrelations, and we use sub-graph isomorphism to determine whether this graph exists in our game world. This means we can search for much more complex conditions, such as love-triangles or adulterous characters. We use the object attributes and relation identifiers/reasons to tune our search. This allows us to search for very specific patterns within our game environment, or very vague ones. Figure 3 shows an example of such a condition.

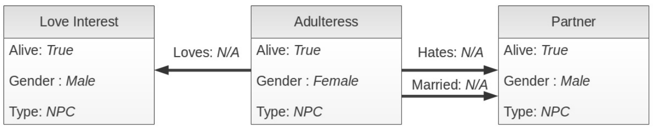


Fig. 3. A sample game world condition showing an "adulterous" character

In the example given in Figure 3 we define an adulterous condition as being a situation where one character *hates* and is *married* to another character, but *loves* a third character. We use *N/A* as the *reason* for these relationships because, for this condition, we are not interested in the reason these relationships exist, simply that they exist. We could have provided the reason if we wished to exert tighter authorial control. To show an example of tight authorial control, in the example figure we search for the specific situation where it is a female NPC in the position of the adulteress. We could remove the gender attributes from our above graph, and this would result in searching for any three characters of any gender mix who fill this pattern. Also important to note, the names *love interest*, *adulteress* and *partner* are not actually used in the search, since each node has a unique name. These names will, however, be of use as they will correspond to the *cast labels* used in the resulting narrative. Using labels becomes important in the subsequent stages of our narrative generation process.

The next step of the generative process involves rewriting the starting narrative to make it more "interesting". To do this we define a new set of rules, which we call *secondary rewrite rules* (SRR). The SRR differ from the IRR in that they can contain both a game world condition *and* a narrative condition with the results being a narrative rewrite. For a narrative to be a potential candidate for a given SRR, it must first satisfy the narrative condition. The narrative condition is represented as a directed acyclic graph and must be present as a sub-graph within the actual narrative in order for the condition to be met. If the narrative condition is met, then it is verified that the game world condition is also met. The game world condition is defined in the same manner as the game world condition in the IRR. The game world condition for these rules can also reference objects in the cast of the narrative. This is done by having labels for the cast. For example, the murder victim may fall under the "Victim" label. Thus, they can be referred to as victim in the game world condition and the system will automatically fill this in with the corresponding object used as the victim when checking the condition.

If both conditions are met then the sub-graph of the narrative representing the narrative condition is rewritten to be the narrative result. The narrative result or *narrative rewrite* is a *directed acyclic graph*. An example SRR is given in Figure 4. The first two conditions represent the *left-hand side* of our graph rewrite rule and the *right-hand side* shows the rewrite. The SRR in the figure states that if there is a story event of type *murder*, and if there exists an alive NPC who has a *love* relationship to the *Victim* of the murder event, then we can rewrite the murder event to consist of three events, the first being the murder event and the second being an ambush event, where the player is forced to kill the lover of the victim. The third event, in which the player spares the victim instead of killing them, is unique since it is not connected to either of the other two events. When the narrative is rewritten with this rule, this will generate a branch in the narrative. In order to apply this example SRR, the system will replace the murder event with the new graph shown in the narrative rewrite. It will then take any previous incoming edges to the murder event and make them incoming edges to any event in the new graph which does not have any incoming edges, ie. the Murder event and the Spare event. Likewise, it will attach any of the previous outgoing edges from the murder event to any of the events which do not have outgoing edges (ie. the Ambush event and the Spare event). In short, we have rewritten the quest to now have two paths. The first is that the player kills the victim and then must also fight their lover. The second path has the player spare their victim instead of killing them; they will then no longer have to fight the lover. Note that since the author designs the rules within the system, they may write rules which specify any number of branches.

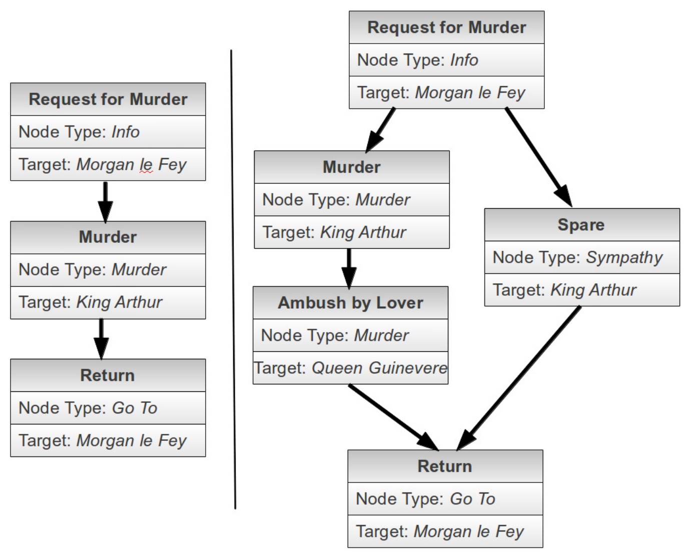


Fig. 5. The initial murder story generated *(left)* and a rewritten version using the secondary rewrite rule provided in Figure 4 *(right)*

As an example, imagine we have a game world with three NPCs, *Morgan le Fey*, *King Arthur* and *Queen Guinevere*. In this world, *Morgan le Fey* hates *King Arthur*, and *King Arthur* and *Queen Guinevere* both love each other. Imagine we have defined an IRR which states that if one NPC hates another, then a potential quest structure would have the player murder the enemy of that NPC. With this game world, a plausible narrative would be one in which the player aims to kill *King Arthur*, as requested by *Morgan le Fey*. The resulting narrative structure could be represented as three events, where the player receives the murder request, murders the victim, and then returns for a reward. This structure is shown on the

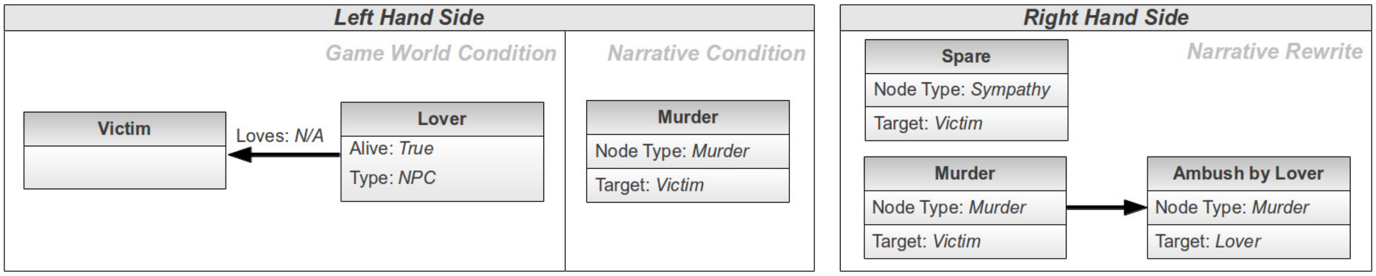


Fig. 4. An example of an SRR

left side of Figure 5. In the next phase, the system will see if there is a valid way to rewrite this narrative using the SRR in Figure 4. It will note that there is an event with *Murder* as the *type*, meaning that the narrative condition has been met. At the next step, the system will replace the *Victim* object in the game world condition with the target *Victim* in the murder event. The system will then check if this updated game world condition is valid. In this example, *King Arthur* is the victim, so the system will check if there are any alive NPCs that have a *loves* relation towards *King Arthur*. This condition is satisfied since *Queen Guinevere* loves *King Arthur* in our game world graph. Since both conditions have been met, we may rewrite this event using our *narrative rewrite*. We start by replacing the *Lover* and *Victim* targets with the actual targets, *Queen Guinevere* and *King Arthur* respectively. Next we replace the original murder event with the new graph. We attach the events following the logic explained above, where any events without incoming edges, will be linked by an incoming edge to any events which were previously linked to the murder event by an outgoing edge. In this example this corresponds to the *Request for Murder* event being linked to the *Murder* and *Spare* events by outgoing edges. Likewise any events without an outgoing edge are linked with an outgoing edge to any events which were previously linked to the murder event by an incoming edge. In this example this corresponds to the *Return* event being linked to the *Spare* and *Ambush* events by incoming edges. The resulting narrative is shown on the right side of Figure 5.

The last step in our process is to simulate each event in the narrative and determine its *effect* on the game world. These effects are declared when the author defines an event in the story. The system proceeds through the narrative in a linear fashion and whenever an event has an effect, the system applies it to the game world. The final result is a modified game environment. Note that in the case of branching narratives, a different final game environment is created for each branch. In our murder story example the act of murdering the victim may make all of the victim's friends hate the player, as well as the NPC who instigated the murder. This modified environment serves as the starting point for the next narrative, again going through the potential IRR rules and picking one at random. Within a game setting, the path taken by the player would be the path used to update the game world in real-time. By performing this step, our generation tool is able to generate narratives "on the fly". This means that we are able to generate narratives one after the other, and that each new narrative respects the changes to the game world of all the previous narratives.

The ReGEN system takes much of the philosophy behind the socially themed narrative generators, but provides an alternative to emergent and goal-oriented planners. It provides much authorial control to the one using the system, as the user is responsible for defining the initial story structures as well as any additional narrative rewrites. One of the downsides of not providing an emergent narrative system is that we restrict the freedom of player choice. We can however provide branching narratives which allow for player choice and give some measure of freedom. This design trades potential emergence for narrative completeness and control, a trade-off that seems appropriate for sidequests in commercial games.

An advantage of the use of Initial Rewrite Rules and Secondary Rewrite Rules is that we begin from and ensure a successful narrative at all points, but can incorporate relatively arbitrary constraints. The metrics we describe in the next section, for instance, could be evaluated at each stage to ensure we only pick the "best" rewrite rule at each step in the generation process. The implementation of this metric optimization remains as future work for the system.

IV. METRICS

A central difficulty in narrative generation is in terms of evaluation. Player opinion and enjoyment is of course paramount but requires procedurally complex and necessarily noisy human evaluation, which is typically done in the form of questionnaire evaluations [23], [24]. Our approach here is to develop a novel set of calculable metrics that intuitively relate to *narrative quality*, at least in terms of overall narrative structure. Our metrics take into account basic narrative features such as length and story branching, as well as more complex inter-story relations that commonly associated with narrative depth/complexity. We have defined an *inter-narrative* metric as being any metric which serves as an evaluation between different narratives, while an *intra-narrative* metric refers to an analysis on a single story, which does not relate to any other of the stories generated. Note that most metrics used in this paper are intra-narrative metrics, while defining more inter-narrative metrics remains a future goal of this research.

Below we define nine metrics that are used to give formal and quantifiable insights into the quality of the narrative being evaluated. The metrics are *narrative content*, *longest/shortest path*, *number of branches*, *cost*, *highest/lowest cost*, *encounters*, *uniqueness*, *narrative richness* and *weight of choices*. We will explain what each metric defines and provide justification in choosing these metrics in the following section.

Narrative Content

Our first metric relates simply to narrative length, and the total number of story events. While it is impossible to directly relate story length to how good a story is (such as determining if short stories are better than novels or vice versa), there are certainly extremes on both sides that could negatively impact story quality. For example, a side-quest with the following description: *Please go outside and collect one flower* may be disappointingly short, while an alternative quest stating *Please go outside and collect one hundred flowers* may feel much too long. The "One Hundred Flowers" quest could also appear lengthy due to the fact that the player is repeating the same actions for the entire quest, which we will discuss below in the "uniqueness" metric. One measure of narrative length is by simply looking at how many narrative events there are overall, regardless of whether the player will be able to experience all events in a single play-through of the quest. We will dub this basic metric *Narrative Content* as this is not necessarily a measure of narrative length, but rather of how many potential narrative events could occur within the quest.

Path Length

We cannot directly measure the length of a narrative when games allow for branching stories as each branch could be any arbitrary length. This means that the best measure for narrative length would involve looking at both the longest, shortest and average path through a narrative, and the amount of narrative seen by the player when taking either of the extreme paths. As with most metrics, it is controversial to state these directly relate to the narrative quality, but we can say that they are one of many metrics which are important to understanding narrative quality. For example, a large difference between the amount of narrative seen when taking the longest path as opposed to the shortest path may show that the player is missing out on much of the potential *narrative content* if they take the shorter path. Conversely, if both the longest and shortest path are very short compared to the total overall *narrative content*, we may be in a position where there are many short branching paths, which can be viewed as either a good or bad feature.

Number of Branches

We take the number of branches as being the number of times within a narrative that the player may experience an event that leads to two or more possible events. As previously explained, the selection of which event to experience is made by the player, and they may select only one of the possible events. We call this selection process *player choice*. Note that we are not analyzing the total number of paths, rather, we are trying to analyze the number of times within the narrative that the player will be given a choice, depending on the path they take through the given narrative. Thus we are aiming to determine, the most, fewest and average number of choices a player could be given within the quest. This is one metric which would not normally be a feature of more traditional narrative analysis, but it is specific to an interactive narrative context. We can view the number of branches in a narrative as a measurement of player freedom and/or narrative complexity.

Cost

Cost is a metric closely tied with narrative generation systems. In our system, actions such as murdering a person or destroying an item make irreversible changes to the game world. After a point these actions could lead to states where it is no longer possible to generate any narratives since most of the NPCs in the world had been murdered by a player in previous quests. We therefore suggest that certain actions should have a *cost* associated with them. This lead us to assign any event in the narrative that resulted in an object being effectively removed from the game world as having a cost of one.

In *Skyrim*'s *Radiant Quest* system, this potential cost is by-passed by having most of the targets be procedurally generated [2]. For example, in a murder quest, the game will procedurally generate an arbitrary NPC to be the player's target. Thus, murdering them does not make any changes to the game world. While this alleviates the concerns of cost, this takes away a feeling of importance from the quest. It is interesting to have quests which make definite and consequential changes, even if at a high level these pose a threat to the size or state of the game world. Cost thus represents an important property of narrative quality when narratives have a meaningful interaction with the game world.

Highest/Lowest/Average Cost

For our analysis of cost, we take a similar approach to the narrative length in that we are interested in the highest, lowest and average possible cost. Again these values let us determine if we are entering any extremes, as well as exploring the variety of possible outcomes based off of player choice. We do not see any purpose in including a metric of *Cost Content*, similar to *Narrative Content* as we are not interested in the overall presence of cost events, we are simply interested in how much cost is guaranteed to occur (*lowest cost*) and how much cost is possible to occur (*highest cost*).

Most/Fewest/Average Encounters

Encounters is a metric which can be considered an alternative to the cost metric. An encounter relates to an exciting event such as a fight with a monster that does not make any irreversible changes to the game world. Essentially, a fight encounter may be the same as a cost action, but whether or not it is an encounter or cost action depends on the target of the event. If the target is *renewable*, meaning the game may generate an infinite number of said objects, then the action is an encounter action. If the game cannot generate more of a given object, then the target is *non-renewable* and the corresponding action is a cost action. This metric has the disadvantage that in many of the examined narratives, the game world contained many random monsters placed in the world environment, but fighting them was incidental and not directly represented as part of the quest structure. We retained this metric since there were some quests where certain types of encounters were deemed important to quest progression, such as quests wherein the player must kill X number of creatures to proceed. Once again, we analyze both the most and fewest possible encounters, as well as the average number of encounters.

Highest/Lowest/Average Uniqueness

As mentioned in the preceding section, for metric analysis we assigned a key action to each event in our story. This we use to help define our *uniqueness metric*. For example if a node involves murdering someone, then the node is a *Murder* node. If the node involves travelling to a location, then this is a *Go To* node, and so on. Using this we can determine how many *unique node types* there are in a story. A story with two murder nodes, for example, would qualify as having only one unique node type. If we divide the number of unique node types by the *narrative content*, we get a measure of what percent of our total story nodes are unique, which we choose to call the story's *uniqueness*. This is shown in Equation 1

$$uniqueness = \frac{unique\ node\ types}{narrative\ content} \tag{1}$$

This means that our measure of uniqueness can be considered as an *intra-narrative* measurement of uniqueness. Referring back to the *One Hundred Flowers* quest, we can break the quest down into one hundred events, where each event is a *COLLECT* event. This would yield a narrative content of 100, but a uniqueness of only one percent. Conversely, the *One Flower* quest would have narrative content of one, but a uniqueness of 100 percent. An example of this in the commercial game *Skyrim* would be a number of quests within the *Civil War* chain of side-quests. These quests, usually prefixed with *The Battle of* followed by the location name, consist entirely of killing a certain number of enemies at different locations. This results in a repetitive experience for the user which regardless of initial interest inevitably becomes dull. We do believe that uniqueness has a direct effect on narrative quality, as it helps to determine whether a narrative faces issues of *repetitiveness*. Again, while there have been many examples of games that use repetitive actions, these games are often not story based. Future work of this research is to create the inter-narrative equivalent of this metric, allowing us to compare narratives to see how similar one generated narrative is to another. This prevents a case where a narrative might in itself be unique, but many of the same narratives may be occurring in a row. Returning to the *Skyrim* example, the *Battle of* side-quests are additionally repetitive in that there are several of them and in no case is there any change in the actions which need to be taken by the player. Once again, since the uniqueness of a quest may change depending on the path the player takes, we look at the highest, lowest and average uniqueness score for each narrative.

Narrative Richness

In an intuitive sense, narrative "richness" or "depth" closely relates to the how surprising or interesting narrative events appear to the player. Plot twists and realization of sub-plots add to perceived complexity and interest, but require the player to experience and even influence narrative events in a way that does not directly relate to the current goal. We interpret this as a metric in terms of the *unintentional consequences* of a narrative, since it attempts to measure how much of a given narrative has been influenced from past narratives, without being a direct goal of adjoining steps. The richness of a narrative may be influenced by multiple features, but we are currently only evaluating a narrative for richness in terms of these consequences. Future work in this area would aim to formalize more features of narrative richness, and validate it against the opinion of human players, in order to further this particular metric.

In our system, we keep a store of all the changes made to the game world by each narrative created. We call these the *postconditions* of our narrative. We also view the game world conditions for the IRRs and SRRs as potential *preconditions*. Preconditions can either be satisfied by conditions in the game world which were not the result of previous narratives, and conditions which were. For example, if in a previous narrative the player made two characters hate each other, then if a narrative is generated with the precondition that those two characters hated each other, the hate precondition was only satisfied because of the actions unknowingly taken by the player in the previous narrative. This then allows us to define narrative richness in terms of the percentage of the preconditions for our narrative which were satisfied by the postconditions of any previous narratives, as shown in Equation 2. This metric could be viewed as an inter-narrative metric. CD Projekt RED's *The Witcher* is a game known for using this feature, where actions taken by the user in previous quests result in them experiencing different events in later quests. An example of this occurs in an early quest entitled *Of Monsters and Men*, where the player is given the choice to defend a character accused of witchcraft, or leave her to be killed. In a much later quest, *Frozen Reflections*, the player encounters the witch. If the player saved the witch, then she is alive and provides the player with potions. If the player left the witch to die, then she is instead a vengeful spirit who attacks the player. Our narrative richness metric aims to capture the concept that the narrative experienced by the player has changed due to seemingly arbitrary choices made by the player in an earlier quest. Note that this metric cannot be evaluated based on a user's path, as with the above metrics, since richness depends on the impact of a choice on all possible futures—we need to look at how the whole quest itself is the result of previous actions, and not just for an individual user path.

$$narrative\ richness = \frac{|all\ postconditions \cap preconditions|}{|preconditions|} \tag{2}$$

Weight of Choices

Our final metric is to examine the effect that choices have on the number of final possible game worlds. As mentioned in the previous section, whenever our stories have a branching event we split our simulation into two parts to represent the two new possible game worlds. We then continue generating narratives from these two new game worlds. After some predetermined number of iterations, we then compare each final game world to each other final game world. This involves comparing each object's attributes and relations by dividing the number of attributes and relations that are the same between both worlds by the total number of attributes and relations in each game world. This gives the similarity between each game world, as shown in Equation 3. We believe this metric is important since it highlights the importance of the choices made by the player in the game in a quantifiable way.

$$similarity_{1,2} = \frac{|attributes_1 \cap attributes_2| + |relations_1 \cap relations_2|}{|attributes_1 \cup attributes_2| + |relations_1 \cup relations_2|} \quad (3)$$

V. Experiments

We designed and performed five tests to quantitatively evaluate the narratives generated by our system and compare them to narratives produced by two other narrative generation systems, as well as two narratives considered "good". Our definition of a "good" narrative is one written manually by an author and is part of a commercially successful RPG game which is known for strong narrative content.

We picked *Skyrim*'s *Radiant Quest* generation system as well as the SQUEGE Sub-quest generator for our narrative generation tools. We picked the *Radiant Quest* system since it is a fairly basic example of narrative generation that produces very short and basic quests [2]. SQUEGE was picked as it proposed similar goals to our quest generator, but did not utilize a game world based approach to narrative generation [19].

For our "good" narratives we examined the main quests from *Skyrim* as well as the main quests from *The Witcher*. Both are commercially successful games known their strong narrative content [28]. The main quests should be representative of the best-written of all the game quests as they are the primary narrative for the game. We additionally picked *The Witcher* as an example of a game which received significant critical acclaim for an especially interesting game narrative where player actions have non-trivial and interesting consequences [28]. In order to faithfully convert each quest in the *Skyrim* and *The Witcher* into a form that may be analyzed by our system, we gathered the quest data from the highly detailed and exact Wikis [29], [30]. These provide analyses for each narrative that in many cases are based upon the actual source files from the game. In the case of *Skyrim*, the quest descriptions are based upon the source files available through the editor provided by the developers. The *Witcher* wiki is fan-based, but also has detailed information extracted from an associated game modding tool. Each narrative is presented in the form of the actions which are needed by the user in order to progress in the narrative. Since this matches our definition of a narrative, it is simple to convert these narratives into our story graph format with each action being converted into an event. This event is then linked to the next action(s) that will need to be taken by the player. We then use keyword parsing to determine the node-type of each individual event (for example, we search for "Murder" to label a node a "murder node") intervening only when a specific action does not have an associated keyword. Such decisions can introduce potential bias, but were mostly trivial, and we are confident that our versions of the commercial narratives primarily represent a purely mechanical translation.

For our experiment we created a basic implementation of our system using Python. We created a sparse game world consisting of 25 objects which include NPCs, locations, items and enemies, which have a sparse set of relations. Examples of these include *loves*, *hates*, *owns* and *lives* relations. We had five rules in our Initial Rewrite Rules creating basic quest structures for stealing, fighting monsters, overthrowing tyrants, murdering hated individuals and surviving assassination attempts. We had five rules in our Secondary Rewrite Rules which allowed for ambushes by loved ones of the player's victims, sparing the individuals they are meant to murder, looting corpses for rewards, using stealth as a means of assassination and getting caught by the owner of items you are intending to steal. Again, while these rules are preliminary, they still represent a variety of unique and interesting narratives.

Comparatively, the evaluated version of SQUEGE uses rewrite rules to produce two main types of quests: item retrieval quests and assassination quests. The *Radiant Quest* system provides 24 basic quest structures, but these are largely similar. Quest types include assassination quests, fighting monster/NPC quests, item retrieval quests, thieving quests, rescue missions.

Note that our metric analysis is meant as a tool to be used by authors. Our metrics measure structural properties, and are meant to approximate narrative quality, but are not of course a complete replacement for aesthetic judgement. We expect an author would use metric analysis to study the types of stories produced by the *ReGEN* system. This gives insight into whether the author should add or modify specific rules in order to better control the system's output. An example of this would be an author who values player choice and therefore desires a large amount of branching in the generated narratives. If they are unhappy with the *number of branches* metric, then it may be an indication that they need to provide more rewrite rules which result in a branching narrative. For our experiment, we define the following criteria for a "good" narrative:

- A longer shortest/longest/average path is better as it increases the lifespan of the quest. Additionally, we desire shortest/longest/average paths that are roughly equal, as this indicates that regardless of player choice, they will still experience a similar number of events.
- A larger number of branches allows for more player control within a given quest, so we value a narrative with a higher number of branches. Having roughly equal scores for all three measures, again indicates that a player is guaranteed a consistent experience which is

independent of the path taken.

- A relatively low cost is preferable for all measures of cost, as it increases the lifespan of the system. Having a minimal difference between highest and lowest cost will further show that the cost effect is consistent, regardless of the choices made by the player.
- A higher number of encounters are good across all measures, as encounters provide exciting scenarios that do not reduce the size of the game world graph.
- A higher uniqueness is preferable as it indicates the presence of multiple unique actions in a narrative. Once again we search for roughly equal measures to ensure that there are no largely repetitive paths a player could take within a quest.
- A higher narrative richness shows that player actions significantly affect the narratives they experience, and is therefore viewed as a positive.
- We value a higher result for weight of choices as it indicates that the players can significantly affect the game world they are playing in.

Validity of these criteria are based on our own perceptions of what constitutes a good narrative, and further verified in our experimental results by comparing with narratives that are more well known in the gaming community. We expect that our system will be able to outperform the basic narrative generation of *Skyrim*'s *Radiant Quest* tool. We additionally wish to check for our short-comings and advantages over SQUEGE, since both our systems are aiming to produce interesting side-quests for RPG games. Lastly, we anticipate that our system will be of similar quality to that of the main narratives from *The Witcher* and *Skyrim*. It is important to note that we are not saying that our system can compete with the main quests of either game, since we cannot compare writing, voice-acting or any of the other features provided by both games. Rather, we are trying to show that the structure of our narrative can compare to that of those narratives. Below we present a series of tests examining and comparing the narratives in terms of our previously defined metrics.

Experimental Results

For our tests we parsed all four of the other narratives into the format of our narratives. This was relatively straightforward as all narratives follow a similar narrative structure that can easily be represented in graph form. As explained above, we used highly detailed wikis for to get the structure for *Skyrim* and *The Witcher* whereas the SQUEGE output is already presented in the form of a directed acyclic graph. For evaluating uniqueness, this involved giving each event in the narrative a *type,* assuming each event generally revolved around one action, such as fighting, gathering info or travelling. We could verify this once again by using the wikis, which define each action that must be taken in order to proceed in the quest. We then used our metrics to analyze each of the stories in terms of *narrative content*, *longest/shortest/average path*, *most/fewest/average branches*, *highest/lowest/average uniqueness*, *most/fewest/average encounters*, and *highest/lowest/average cost*. The results are presented in Table I. Note that for analyzing our system, we generated one hundred quests and averaged the results of each individual quest's metrics.

Narrative Content: The results shown in Table I show our system's metrics being comparable to both the *Skyrim* main quests, as well as the SQUEGE output in some respects, with our system scoring 5.32, compared to 5.08 in SQUEGE and 5.12 in *Skyrim*. The results of the analysis for *The Witcher*'s main quest-line appear to be, in general, very different than our previously examined systems. As expected, our system, the "good" quests, and the SQUEGE output show an improvement over *Skyrim*'s *Radiant Quest* results given that we defined a larger narrative content as being indicative of a better narrative. In terms of narrative content, our system, SQUEGE and the *Skyrim* main quests exhibit on average five distinct narrative events. The *Skyrim Radiant Quests* average only two events which, by our definition above, we consider to be a poor result. *The Witcher* exhibits a much a larger set of narrative events on average, providing ten more events on average than our system, *Skyrim*, or the SQUEGE quests.

Longest/Shortest/Average Path: Our system's longest and shortest path were 4.8 and 4.4 respectively with the average path being 4.6, meaning on average a player will see between *83-89%* of the narrative's content on a single playthrough. Comparatively, a player will see around *59-80%* of narrative content for a given SQUEGE narrative and *72-79%* for *The Witcher* Main Quests. For the *Skyrim* main and *Radiant Quests* we see a much higher percentage of narrative content, 100 and 96 percent respectively, which is due to the complete lack of branching paths in *Skyrim*'s main quests and the minimal use of branching paths in the *Radiant Quests*. If we follow our assumption that the "good" quests present superior quest design, and assume that a well design quest will allow the user to see between 70-100% of narrative content independent of the path followed, then we can state that our quests fall within this boundary. Alternatively, we could state that a good branching story presents between 70-80% of narrative content, comparable to the narrative content results from *The Witcher*, and somewhat above the amount of choice (59%) present in SQUEGE output. In this case, our results indicate that we should have more of our narrative contained within branching paths on average. Using our criteria for a good narrative, we find that our system outperforms the *Radiant* and SQUEGE quests. We approach the values of path length found within the *Skyrim* main quest, scoring on average 4.6 events comparing to the 5.1 events contained on average in a *Skyrim* main quest. Some difference between the longest and shortest path are to be expected since the lack of branching in the *Skyrim* quests mean that users will always see all possible narrative content.

Looking at the difference between the amount of narrative seen when following the longest and shortest path, based on the percentages we previously gave, we see that there is a difference of *6%*. This means that if the player follows the shortest path in an average narrative they see six percent less of the total content than if they took the longest path. We see that this difference is similar to the *7%* difference in *The Witcher*, and much less than the *21%* of the SQUEGE narratives. A lower difference implies that regardless of which

TABLE I
THE MEAN AND STANDARD DEVIATION EVALUATED FOR VARIOUS TYPES OF GENERATED AND HAND-AUTHORED QUESTS

Metric	ReGEN	SQUEGE	Radiant Quests	Skyrim Main Quests	Witcher Main Quests
Narrative Content	5.32 ± 0.90	5.08 ± 3.25	2.33 ± 0.85	5.12 ± 2.37	15.41 ± 9.52
Longest Path	4.75 ± 0.54	4.13 ± 1.64	2.17 ± 0.80	5.12 ± 2.37	12.09 ± 6.41
Shortest Path	4.43 ± 0.59	3.04 ± 0.79	2.17 ± 0.80	5.12 ± 2.37	11.18 ± 6.58
Average Path	4.59 ± 0.52	3.78 ± 1.29	2.17 ± 0.80	5.12 ± 2.37	11.61 ± 6.46
Most Branches	0.57 ± 0.50	1.63 ± 1.07	0.17 ± 0.37	0	1.26 ± 1.09
Fewest Branches	0.57 ± 0.50	1.17 ± 0.37	0.17 ± 0.37	0	1.26 ± 1.09
Average Branches	0.57 ± 0.50	1.48 ± 0.88	0.17 ± 0.37	0	1.26 ± 1.09
Highest Cost	0.84 ± 0.55	1.42 ± 0.91	0	0.29 ± 0.46	0.21 ± 0.40
Lowest Cost	0.43 ± 0.50	1.29 ± 0.93	0	0.29 ± 0.46	0.12 ± 0.32
Average Cost	0.63 ± 0.42	1.38 ± 0.89	0	0.29 ± 0.46	0.17 ± 0.34
Most Encounters	0.25 ± 0.43	0	0.42 ± 0.57	0	1.44 ± 1.90
Fewest Encounters	0.0 ± 0.0	0	0.42 ± 0.57	0	1.24 ± 1.88
Average Encounters	0.12 ± 0.22	0	0.42 ± 0.57	0	1.33 ± 1.89
Highest Uniqueness	0.95 ± 0.09	0.68 ± 0.14	0.95 ± 0.12	0.72 ± 0.20	0.59 ± 0.25
Lowest Uniqueness	0.94 ± 0.09	0.58 ± 0.10	0.95 ± 0.12	0.72 ± 0.20	0.53 ± 0.24
Average Uniqueness	0.94 ± 0.09	0.62 ± 0.09	0.95 ± 0.12	0.72 ± 0.20	0.56 ± 0.24
Narrative Richness	0.03 ± 0.12	0	0	0	0.03 ± 0.16

path the player takes through the narrative they are still able to see much of its content, and that choices impact specific narrative results rather than representing fundamental, highly disjoint story branching. These results correspond nicely with our criterion that having a smaller difference between a longest and shortest path is indicative of a good story.

One important result, which is not explicitly stated in our criteria but is still worth noting, is that the standard deviation on our narrative content is lower than that of the other systems and is much closer to that of the *Radiant Quest* system. It is difficult to state if this is indicative of a good/bad story, but it does indicate that our generation system frequently generates narratives with the same amount of narrative content, resulting in a more consistent experience. If desired, we expect greater variance could be achieved by adding more and more complex rewrite rules in the system.

Most/Fewest/Average Branches: The number of branches show that on average, one in every two of our generated stories will contain at least one branching path. Following our criteria, this falls well below the results of both the SQUEGE narratives and *The Witcher* narratives, where *The Witcher* has on average one branching path per narrative and the SQUEGE narratives have closer to two branching paths. The *Skyrim* main quests are always linear and therefore do not have any branches, whereas the *Skyrim Radiant Quests* very infrequently contain branching paths. Since our criteria state that a large number of branches indicate better narratives, then the SQUEGE and *Witcher* narratives are superior in this area to our generated metrics. Since we have made the assumption that *The Witcher* quests are representations of good narrative structure, we should therefore add more rewrite rules which create branching paths in the narrative in order to compete with its results. This is again one of the strong benefits of defining narrative metrics and comparing our generated narratives to others, since it gives insight into how we can restructure our narrative generation tool in such a way that it produces measurably better narratives.

Highest/Lowest/Average Cost: Comparing costs, we see that our stories contain, on average, at least one potentially irreversible action per narrative. The results for SQUEGE are even higher, with at least one cost unit per narrative regardless of path chosen. The *Radiant Quests* always have no cost because all destroyed objects are procedurally generated for each quest. While this is an interesting means of having narrative generation with no potential cost to game environment, this takes away a sense of purpose from the side quest, since as a result the quest makes no noticeable difference in the game world. In spite of this, our costs are still high compared to the *Skyrim* and *Witcher* main quests, which have costs between 0.1 and 0.3. Following the criteria, aiming to reduce the number of rules which include an irreversible action would improve not necessarily narrative quality, but rather the lifespan of the quest generation process itself.

Most/Fewest/Average Encounters: In examining encounters, we reiterate that in most RPGs, encounters are implied but not explicitly stated in quest structure. For example, in *Skyrim*, a player encounters many monsters travelling through the game world, but these are random encounters and not stated in the quest description. SQUEGE, makes no such explicit definitions of encounter either. However, encounters are explicitly defined in our system, *The Witcher* main quests and the *Radiant Quests*. We define having more encounters as being a positive, and in this instance our system generates encounter events much less frequently than either the *Radiant Quests* or *The Witcher*. Again, this is a feature which can be tweaked with the creation of and/or modification to, the sets of rewrite rules in our system.

Highest/Lowest/Average Uniqueness: One of the more interesting metrics analyzed is that of uniqueness. As mentioned before, this metric is not perfect since we take only the primary action of each narrative event to be a description of that event. This ignores, for example, the implied random encounters in the *Skyrim* world, and does not account for player preference with regards to narrative content. We still feel however, that provides strong insight into a narrative's uniqueness as it is the primary event that is usually of most interest and importance to the user. Given that we regard repetitive events as detrimental to narrative quality, this is one metric of which a higher value directly implies an increase in narrative quality.

What is noted in these results is that the main quests of *Skyrim* and *The Witcher* have much lower uniqueness scores compared to our system and the *Skyrim Radiant Quests*. The uniqueness for *Skyrim* is on average 0.72, whereas the average uniqueness for *The Witcher* is only 0.56. Conversely, ReGEN's average uniqueness is 0.94 and the *Radiant Quest*'s average uniqueness is 0.95. One reason for this may be that our system and the *Radiant Quest* system are both focused on creating *side quests*, which are separate from the actions in the main quest. For example, a side quest may involve stealing an item, and a player who is not interested in being a thief may not perform this quest. However, in the main quest the player must perform all of the events in order to complete the game, so the main quest may aim to make stories which are of interest to all possible players. This is a relatively small subset of all the possible narrative events, and would result, therefore, in a lower uniqueness score. Conversely, the size of the narrative content may have an effect on the uniqueness. Since, as more content is added, the more likely it is that an action in the set of all possible actions will appear more than once. Thus, the uniqueness score of 0.5 for *The Witcher* could be a consequence of it having a much higher narrative content, that of fifteen narrative events on average per story. We found that the SQUEGE generator performed poorly in this metric, having an average uniqueness of only 0.6 despite the smaller narrative size.

Narrative Richness: We then examined both our narratives and the main quest narrative of *The Witcher* in terms of our *narrative richness* metric. The *narrative richness* metric could really only be evaluated for these two sets of quests since it is only in these two systems that actions taken by the player can have unforeseen consequences in later narratives. Although the *Skyrim* quests have some indirect effects on the *Skyrim* world, these changes do not effect the main quest, whereas in *The Witcher* indirect impact is an important part of player (and narrative) choice [28]. In this case ReGEN results are shown to be approximately the same as the results for *The Witcher*. What is interesting is that for both systems, the narrative richness is very small. While we have fallen within the metrics given by the "good" narrative, it would be interesting to further explore ways of making generated narratives more dependant on the results of previously generated narratives.

Radiant Quest Experiment: As an additional experiment, we performed a test to analyze the effect of using our system with the narratives given in *Skyrim*'s *Radiant Quest* system as the base narratives in our IRR. In this way, we could then observe the changes in metrics when our SRRs are applied to the resulting narratives. This experiment shows the flexibility of our graph rewriting system, and how it can adapted to suit the system in *Skyrim*. We also show that by applying our SRR rules, we can create measurably better narratives, which is an argument again in favour of our system as a means for narrative generation. The results for this experiment are shown in Table II, with the first column showing the results for the *Radiant Quest*, the second column showing the results of simply using the base narratives from the *Skyrim Radiant Quest* system and the third showing the results after our SRR rule layer is applied.

The first thing to note is that, due inherently to the way we designed our system, murder actions now have cost. This is in contrast to the *Radiant Quests* in *Skyrim* where all murder victims are procedurally generated. We left in this cost simply to show how it changes once we apply our SRR rules. The first column effectively matches the metrics given in the results for the *Radiant Quests* in Table I. Differences can be attributed to the fact that our system picks the next narrative at random from all potential narratives, meaning some variation is likely to occur simply due to the randomness of the selection process. After modifying the system to use our SRR graph rewrite rules, we see an noticeable improvement in the metrics based around our original definition of narrative quality, such as with the content and number of branches. There is a raise in the highest cost, which we define as detrimental to narrative quality, but interestingly the lowest cost has been reduced to zero (one of the SRR rules allows the player to spare the victim they are supposed to murder, which likely resulted in this change). The number of encounters remained the same—as we currently do not have any rewrite rules that create more encounters, adding such a rule would most like improve our encounter metric given in Table I. The uniqueness score declined slightly, showing a loss in narrative quality in this area. This supports the argument that an increase in narrative content will invariably lead to a decrease in uniqueness since the set of possible narrative events in a game is finite.

Weight of Choices: Note that for the *weight of choices* metric we do not have comparable data from any of the other four systems. In the case of *Skyrim*, this is due to the fact that *Skyrim*'s main quest do not branch, and therefore the state of the world does not change due to player choice. Likewise the *Radiant Quests* use only procedural content which makes no modification to the game environment. Note however, that due to this inherent design we can state that the *Radiant Quest* system creates a 0% result for the *weight of choices* metric. The SQUEGE quests represent the game world only as objects, not as objects and relations. Lastly, we do not have enough data on *The Witcher*'s game environment to represent the game world in our system.

After generating one hundred stories, we are left with 77 possible final social states. The metric shows that the attributes and relations of the objects within these states are about 94% similar to the attributes and relations of all other possible social states. This means that the choices made in our narrative can cause around a 6% difference in the social state. This is not an enormous difference, but it must be balanced against the need to enforce some uniformity of experience, and it still shows that the player can exert some influence over the game world. Since our criteria states that a higher weight of choices

TABLE II
THE RESULTS OF RUNNING OUR SYSTEM AS A *Radiant Quest* SYSTEM

Metric	Original Radiant Quests	Radiant Quests (ReGEN)	Radiant Quests + SRR (ReGEN)
Narrative Content	2.33 ± 0.85	2.51 ± 1.25	3.68 ± 1.95
Longest Path	2.17 ± 0.80	2.30 ± 1.12	2.74 ± 1.11
Shortest Path	2.17 ± 0.80	2.30 ± 1.12	2.36 ± 1.04
Average Path	2.17 ± 0.80	2.30 ± 1.12	2.57 ± 1.07
Most Branches	0.17 ± 0.37	0.21 ± 0.41	0.37 ± 0.48
Fewest Branches	0.17 ± 0.37	0.21 ± 0.41	0.37 ± 0.48
Average Branches	0.17 ± 0.37	0.21 ± 0.41	0.37 ± 0.48
Highest Cost	0	0.32 ± 0.47	0.66 ± 0.89
Lowest Cost	0	0.19 ± 0.40	0.0 ± 0.0
Average Cost	0	0.26 ± 0.40	0.29 ± 0.40
Most Encounters	0.42 ± 0.57	0.26 ± 0.44	0.28 ± 0.45
Fewest Encounters	0.42 ± 0.57	0.14 ± 0.34	0.0 ± 0.0
Average Encounters	0.42 ± 0.57	0.20 ± 0.36	0.12 ± 0.20
Highest Uniqueness	0.95 ± 0.12	1.0 ± 0.0	1.0 ± 0.0
Lowest Uniqueness	0.95 ± 0.12	1.0 ± 0.0	0.91 ± 0.14
Average Uniqueness	0.95 ± 0.12	1.0 ± 0.0	0.96 ± 0.07

indicates a more interesting narrative experience for the player this is an improvement over the *Radiant Quest* system, which as mentioned above has a *weight of choices* value of 0%.

Analysis of Skyrim as One Long Quest: Our final experiment was to metrically analyze *Skyrim* as one long main quest, as opposed to several shorter quests making up the main quest. Since there are seventeen quests in the *Skyrim* main quest, we tested the result against linking seventeen of our quests together into one long quest. Here, we were trying to examine the decline of uniqueness in narratives as dependent on content. The results show that the *Skyrim* main quest consists of 87 events, but its overall uniqueness is only 0.13. Likewise, when linking seventeen of our own quests together, we ended up with 100 events, but a uniqueness score of only 0.11. Additionally, linking 100 of our side quests into one quest, we end up with 269 events but a uniqueness score of only 0.04.

These results show the decline in uniqueness for both the *Skyrim* main quests, and of our own system over time. Interestingly, since uniqueness is in terms of how many narrative events are unique, multiplying narrative content by uniqueness gives us the total number of unique narrative events for each of the narratives tested. For both our system, and *Skyrim*'s main quest, this gives a total of eleven unique narrative events possible, meaning we again have measurable comparable metrics in terms of quality to those of *Skyrim*.

Understandably, many of these metrics are the result of the ways our rules are defined, and tweaking the rules can improve/worsen many of the results. However, this is one of the important reasons why we chose to design metrics. By performing these analyses we are able to determine which story structures are considered "good" and can then tweak our generation tool to make improved narratives.

VI. CONCLUSION

In this paper we presented a system for generating game narratives using graph rewriting. We developed a set of novel metrics which we could use to formally define various features of game narrative relating to narrative quality. Lastly, we used these metrics to analyze the narratives generated by our system and compare them against the existing game narratives, and narratives generated by SQUEGE, which is another narrative generator similar to our own. Our results show that, even at an early stage, our narrative generation system creates narratives which can quantitatively compare to those of *Skyrim*. The system additionally provides branching stories and allows for stories to be generated based off changes in the game world from previous narratives. As shown by the metrics, these features provide player choice as well as narrative depth. We demonstrated the flexibility of our system by modifying it to behave as the *Skyrim Radiant Quest* system, and then showed how our graph rewriting methods could create a measurable improvement in the quality of narratives generated. These results show the potential of using a formal analysis of narrative quality to examine both the narratives themselves and the narrative generation system itself.

We believe that further development of our own system will lead to better results comparable to commercial games when analyzed using our set of metrics. In particular, further development of the rewrite rules may lead to a highly dynamic and robust procedural narrative system. We are interested in validating our metrics through human studies, and also in further validating the narratives themselves, using formal methods to assert that each quest is possible for the user to complete. Finally, we wish to integrate our system into a gaming environment, to show proof-of-concept. This was found to be non-trivial in current game editors without direct source access, but remains a future goal of this research.

VII. Acknowledgements

The authors would like to thank Bentley James Oakes for valuable feedback and naming our ReGEN system. This research was supported by the Natural Sciences and Engineering Research Council of Canada.

References

[1] J. Horsey, "The Elder Scrolls V: Skyrim Radiant Quest system, will customise your game as you play," On-line, Geeky Gadgets, November 2011.

[2] H. Lenhardt, "Bethesda's Nesmith reflects on the difficult birth of Skyrims 'Radiant Story' system," On-line, Venture Beat, January 2012.

[3] V. Propp, *Morphology of the Folktale*. University of Texas Press, 1968.

[4] C. Levi-Strauss, "Structure and form: Reflections on a work by Vladimir Propp," in *Structural Anthropology, Volume 2*. University of Chicago Press, 1983.

[5] B. N. Colby, "A partial grammar of Eskimo folktales," *American Anthropologist*, vol. 75, no. 3, pp. 645–662, 1973. [Online]. Available: http://dx.doi.org/10.1525/aa.1973.75.3.02a00010

[6] D. E. Rumelhart, *Notes on a schema for stories*. Academic Press, Inc, 1975, pp. 211–236.

[7] J. B. Black and R. Wilensky, "An evaluation of story grammars," *Cognitive Science*, vol. 3, no. 3, pp. 213–229, 1979. [Online]. Available: http://dx.doi.org/10.1207/s15516709cog0303_2

[8] A. Garnham, "What's wrong with story grammars," *Cognition*, vol. 15, no. 13, pp. 145 – 154, 1983. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0010027783900379

[9] J. R. Meehan, "TALE-SPIN, an interactive program that writes stories," in *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, ser. IJCAI'77. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1977, pp. 91–98. [Online]. Available: http://dl.acm.org/citation.cfm?id=1624435.1624452

[10] J. Porteous, M. Cavazza, and F. Charles, "Applying planning to interactive storytelling: Narrative control using state constraints," *ACM Trans. Intell. Syst. Technol.*, vol. 1, no. 2, pp. 10:1–10:21, Dec. 2010. [Online]. Available: http://doi.acm.org/10.1145/1869397.1869399

[11] M. O. Riedl and R. M. Young, "Narrative planning: balancing plot and character," *J. Artif. Int. Res.*, vol. 39, no. 1, pp. 217–268, Sep. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1946417.1946422

[12] M. Theune, S. Rensen, R. op den Akker, D. Heylen, and A. Nijholt, "Emotional characters for automatic plot creation," in *Technologies for Interactive Digital Storytelling and Entertainment. Second Intern. Conf., TIDSE 2004*. Springer-Verlag, 2004, pp. 95–100.

[13] I. Swartjes and M. Theune, "A fabula model for emergent narrative," in *Proceedings of the Third International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, ser. TIDSE'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 49–60. [Online]. Available: http://dx.doi.org/10.1007/11944577_5

[14] H.-M. Chang and V.-W. Soo, "Planning-based narrative generation in simulated game universes," *IEEE Trans. Comput. Intellig. and AI in Games*, vol. 1, no. 3, pp. 200–213, 2009.

[15] R. P. ý Pérez and M. Sharples, "MEXICA: A computer model of a cognitive account of creative writing," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 13, no. 2, pp. 119–139, 2001.

[16] M. Cavazza, D. Pizzi, F. Charles, T. Vogt, and E. André, "Emotional input for character-based interactive storytelling," in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, ser. AAMAS '09. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 313–320. [Online]. Available: http://dl.acm.org/citation.cfm?id=1558013.1558056

[17] M. Mateas and A. Stern, "Procedural authorship: A case-study of the interactive drama facade," in *In Digital Arts and Culture (DAC*, 2005.

[18] J. McCoy, M. Treanor, B. Samuel, B. Tearse, M. Mateas, and N. Wardrip-Fruin, "Authoring game-based interactive narrative using social games and *Comme il Faut*," in *Proceedings of the 4th International Conference & Festival of the Electronic Literature Organization: Archive & Innovate (ELO 2010)*, Providence, Rhode Island, USA, June 2010.

[19] C. Onuczko, D. Szafron, and J. Schaeffer, "Stop getting side-tracked by side-quests," in *AI Game Programming Wisdom 4*. Course Technology, 2008.

[20] C. Onuczko, D. Szafron, J. Schaeffer, M. Cutumisu, J. Siegel, K. Waugh, and A. Schumacher, "A demonstration of SQUEGE: A CRPG sub-quest generator," in *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference*, 2007, pp. 110–111.

[21] J. Dormans, "Level design as model transformation: a strategy for automated content generation," in *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, ser. PCGames '11. New York, NY, USA: ACM, 2011, pp. 2:1–2:8. [Online]. Available: http://doi.acm.org/10.1145/2000919.2000921

[22] A. Zook, S. Lee-Urban, M. O. Riedl, H. K. Holden, R. A. Sottilare, and K. W. Brawner, "Automated scenario generation: toward tailored and optimized military training in virtual environments," in *Proceedings of the International Conference on the Foundations of Digital Games*, ser. FDG '12. New York, NY, USA: ACM, 2012, pp. 164–171. [Online]. Available: http://doi.acm.org/10.1145/2282338.2282371

[23] F. Peinado and P. Gervás, "Evaluation of automatic generation of basic stories," *New Generation Computing*, vol. 24, pp. 289–302, 2006, 10.1007/BF03037336. [Online]. Available: http://dx.doi.org/10.1007/BF03037336

[24] H. Barber and D. Kudenko, "Generation of dilemma-based interactive narratives with a changeable story goal," in *Proceedings of the 2nd international conference on INtelligent TEchnologies for interactive enterTAINment*, ser. INTETAIN '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 6:1–6:10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1363200.1363208

[25] S. G. Ware, R. M. Young, B. Harrison, and D. L. Roberts, "Four quantitative metrics describing narrative conflict," in *Proceedings of the 5th International Conference on Interactive Storytelling*, ser. ICIDS'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 18–29. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-34851-8_2

[26] R. P. y Pérez, O. Ortiz, W. Luna, S. Negrete, V. Castellanos, E. Peñalosa, and R. Ávila, "A system for evaluating novelty in computer generated narratives," in *Proceedings of the Second International Conference on Computational Creativity*, 2011, pp. 63–68. [Online]. Available: http://iccc11.cua.uam.mx/proceedings/the_narrative/perezyperez_iccc11.pdf

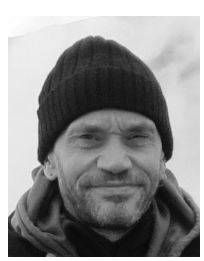
[27] Z. Tomaszewski, "On the use of reincorporation in interactive drama." in *Intelligent Narrative Technologies*, ser. AAAI Workshops, vol. WS-11-18. AAAI, 2011. [Online]. Available: http://dblp.uni-trier.de/db/conf/aiide/narrative2011.html#Tomaszewski11

[28] RPS, "Game logic vs choice & consequence," http://www.rockpapershotgun.com/2011/11/21/game-logic-vs-choice-consequence/, Rock, Paper, Shotgun, November 2011.

[29] The Unofficial Elder Scrolls Pages. (2013) Skyrim: Main quest. http://www.uesp.net/wiki/Skyrim:Main_Quest.

[30] The Witcher Wiki. (2013) The witcher primary quests. http://witcher.gamepedia.com/The_Witcher_primary_quests.

Ben Kybartas is currently a PhD student at the Delft University of Technology in Delft, Netherlands. He completed his Master's in the *GR@M* group at McGill University, Montréal, Canada. His research interests include procedural content generation and artificial intelligence in games.



Clark Verbrugge is an associate professor at the School of Computer Science of McGill University in Montréal, Canada. He is a member of both the *Sable* and *GR@M* groups at McGill, and does research in game formalization and analysis as well as compiler and runtime optimization. He holds a Ph.D. in Computer Science from McGill University and a B.A.(Hon.) in Film Studies from Queen's University.