Procedural Generation of Narrative Worlds

J Timothy Balint[1] and Rafael Bidarra[2], *Senior Member, IEEE*

***Abstract*—A Narrative World typically consists of several interrelated locations that, all together, fully support enacting a given story. For this, each location in a narrative world features all the objects as required there by the narrative, as well as a variety of other objects that plausibly describe or decorate the location. Procedural generation of narrative worlds poses many challenges, including that (i) it cannot lean only on domain knowledge (e.g. patterns of objects commonly found in typical locations), and (ii) it involves a temporal dimension, which introduces dynamic fluctuations of objects between locations. We present a novel approach for the procedural generation of narrative worlds, following two stages: first, a *Narrative World Mold* is generated (only once) for a given story; second, the narrative world mold is used to create one (or more) possible narrative worlds for that story. For each story, its narrative world mold integrates spatio-temporal descriptions of its locations with the object semantics and the domain knowledge previously acquired on typical locations. We describe how a narrative world mold can be generated, as well as how it can be fed to existing procedural generation methods, to create a variety of narrative worlds that fit that narrative. We evaluate our own implementation of this approach, performing a number of experiments that illustrate both the expressive power of narrative world molds and their ability to steer the generation of narrative worlds.**

***Index Terms*—Procedural Content Generation, Computational Narratives, Narrative Worlds, Narrative World Mold**

I. Introduction

A *NARRATIVE WORLD* (NW) [1] is an environment consisting of several interrelated locations, or *spaces*, which are intrinsically associated to a given *story*, in such a way that this story can be enacted in that NW. In a way, theaters likely provide one of the oldest and most common NW examples, and they serve to illustrate an essential property of NWs: that, despite all possible constraints put forward by the story, there is a huge variety of potential NWs, all of which support that very same story. In this article, we are specifically concerned with a variety of generative (or procedural [2]) methods able to (semi-)automate the generation of such NWs, and particularly, of virtual NWs. Achieving this is a rather challenging endeavor because the resulting world needs to:

- provide full support for each and *every action* in the narrative (e.g. include each action's location);
- integrate and layout in a sensible manner all the *essential objects* required by those actions, at the appropriate locations (e.g. have a chair in a location where a character sits down);
- complete each location with a variety of other objects that, despite being *non-essential items*, are typically found in it (e.g. have paintings or lamps hanging in a living room);
- take into account the spatio-temporal interrelationships among locations of the narrative (e.g. what is moved when, by which characters, between which locations).

Many techniques have been investigated for computational story generation, but the majority of that work has been dedicated to automating plot generation rather than space generation [3]. Furthermore, many of these techniques require an initial, high-level declaration and placement of essential and decorative objects, known as the full initial state. In this article we focus on the question: *given an input story lacking a full initial state, what does it take to automate the generation of a NW for it?* To preserve the focus on world (rather than story) generation, we will assume that this input is a *valid and coherent story*, typically consisting of a sequence of actions on objects, performed by characters in a given location, as used previously by many other authors [4]–[7]. By starting from a *coherent* story we are simply assuming that it has been validated, and deemed free of logical or spatio-temporal contradictions among the plot points. That excludes, for example, that a character picks up the apple from the table, hours after (i) that character has been killed or (ii) someone else has eaten that apple. In other words, we assume the story 'does not prescribe' any impossibility for the world.

At times, researchers have used a variety of specific narrative formats, including e.g. PDDL, which provide many features to process natural language stories into interactive adventures [8], to visualize story states [9], [10], as well as to generate action operators from natural language descriptions [11]. The reverse side to those approaches, however, is that the resulting research results become dependent on the presence of such specific features and formats. We therefore opted for a simple, agnostic input narrative format, assuming no specific features nor advanced structures.

In order to properly build each location for such a NW, both its objects (*content*) and the relationships between them (*structure*) need to be determined. These can be either explicitly described by a (world) designer (as is e.g. done in WordsEye [12] and SceneSeer [13]), or implicitly derived, e.g. through domain knowledge and/or rules about locations. This knowledge on what objects are, how they behave and relate to others, etc. is usually designated as their *semantics*. Semantics plays an important role, both among concrete object instances (as in *this couch is against the left wall*), and among generic classes (as in *Chairs are often placed around flat furniture*). Semantics involving object classes typically provides useful generalizations about both the content and structure of locations. Various scene synthesis systems that build a single, static 3D environment from data operate on both the content and the structure levels [14].

1 J Timothy Balint performed this work while at Delft University of Technology, The Netherlands and Stonehill College, MA, USA.

2 Rafael Bidarra is with Delft University of Technology, The Netherlands. (e-mail: R.Bidarra@tudelft.nl)

We posit that taking only into account spatial relationships between objects is insufficient for a scene synthesis system to automatically generate a narrative world as defined here, because it overlooks the temporal constraints that narrative actions impose on many objects. To the best of our knowledge, so far no scene synthesis system has taken into account the generation of dynamic locations, aimed at undergoing changes according to a narrative time-line. Likewise, very few systems have attempted using a narrative, and particularly its temporal dimension, to control scene synthesis for the entire story's world.

We propose a novel approach for the procedural generation of narrative worlds that overcomes these limitations, based on the notion of *Narrative World Mold* (NWM). A NWM integrates, for each story, the spatio-temporal descriptions of its essential narrative elements (objects, characters, locations, and their interrelationships), with object semantics and domain knowledge.

Once a NWM has been derived for a given story, it can be used to generate a variety of alternative narrative worlds, each of which is guaranteed to fulfill the requirements posed by that story. This NW generation phase can then be steered in a variety of ways, yielding e.g. a minimalistic, a futuristic, or an overly decorated NW, according to the control possibilities offered by the particular procedural generation techniques used. By using a NWM that guarantees spatio-temporal coherence of a NW, the aforementioned challenges to procedurally generating virtual narrative worlds can be overcome.

II. Related Work

Arguably the most studied area around narrative world generation consists of generating and laying out the locations, otherwise known as *(virtual) world generation*. World generation methods can be classified into *single-location generation* and *multi-location generation*, as well as *data-driven* and *mixed-initiative*. Data-driven methods are (almost) solely automated processes, and mixed-initiative methods combine data, automation and human input in order to generate the world.

The most common approach to generate a virtual world is to autonomously create a *single location* from data-driven domain knowledge, on which topic a recent survey has been published [15]. Knowledge is most often developed through learning the common objects typically found in a location, as well as the relationships normally seen between those objects. Relationship representation methods can be categorized into controlled rules [16]–[20], object-object constraints [21]–[24], and human-centric object relationships [25]–[27]. Each of these methods provides ways to create a variety of object configurations for a location, generally through some optimization over the known relationships for that location. This favors location diversity, allowing for the creation of distinct location variants from a learned location pattern.

Unlike data-driven methods, mixed-initiative single location methods rely on human insight to steer the generation of a space. Earlier work such as the WordsEye system [12], requires explicit specification of the objects in the scene and of their relative placement. Ma [14] later extended this idea for a single animation based on a single sentence. More recent work allows authorial control for creating a procedurally generated scene. SceneSeer [13] and Ma et al. [28] place objects in a room based on natural language text. Unlike WordsEye, SceneSeer uses data-driven domain knowledge, allowing it to add implicit objects that are related to the explicit objects mentioned in the text. Fu et al. [29] complete a room pattern based on the objects an author places in the environment. All of these methods provide a scene author with some control (or almost total control, in WordsEye's case) in order to generate a representation closer to what the author has in mind.

Some mixed-initiative methods have been developed to generate an entire world at once, with *multiple locations* [30]. These methods mostly focus on indoor scenes, consisting of interconnected locations, such as the various rooms in a house or dungeon. Camozzato et al. [31] use a sketch to layout rooms in an apartment. This sketch does not contain furniture but rather the walls within which the entire apartment exists. Similarly, Wu et al. [32] generate layouts through hierarchical decomposition and quadratic programming over a set of constraints representing high level specifications from a user. Finally, in one of their case studies, Karavolos et al. [33] use a sketch of a mission graph (where each node represents an action) in order to generate several interconnected locations in a dungeon.

There has been less work on mixed-initiative generation of complete worlds based on a provided narrative. Both GameForge [34] and Valls-Vargas et al. [35] are concerned with generating locations from a sequence of *plot points*, understood as consecutive actions performed by characters on objects. Valls-Vargas et al. only create and connect locations from the narrative, while GameForge places decorations in their environment based on authored or automatic (Gaussian) distributions.

The methods discussed above aim at creating locations from a time-invariant perspective, i.e they all focus on building the initial set-up of locations. However, many virtual environments, and specifically narrative worlds, evolve over time. When using a narrative to steer their generation, there are time-dependent fluctuations of the objects in the various locations, which must be taken into account. To the best of our knowledge, this dynamic aspect has not been handled in previous research work.

III. Overview of Narrative World Generation

A narrative can be enacted in a large variety of possible narrative worlds, as long as each of them meets two requirements: (i) the world must contain the content required in each location throughout the narrative, and (ii) the world must be set-up and laid-out taking into account the fluctuation of content in and out of its locations, due to the narrative actions occurring over time. These two requirements set the bottom line for any NW, enforcing that it respects and supports the essence of its narrative, above all other secondary details and contingencies. We call this 'common ground' of all those NWs the *Narrative World Mold* (NWM) for that narrative. Given a narrative, its

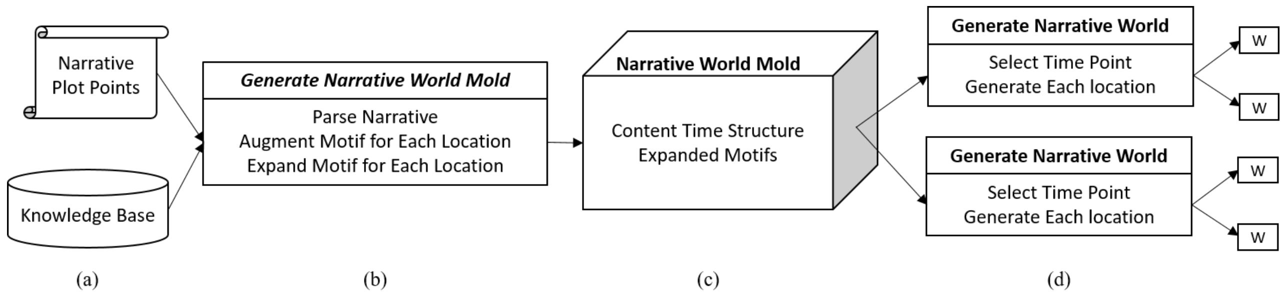

Fig. 1: The two main phases of the narrative world generation pipeline: (a) a Narrative and Knowledge Base are used to (b) generate the components of a Narrative World Mold; (c) this Narrative World Mold can then be used to (d) generate one or more narrative worlds, all of which, by definition, support enacting the input narrative.

NWM is a semantic model integrating (i) the spatio-temporal descriptions of its essential narrative elements (objects, characters, actions, locations, and their interrelationships), with (ii) object semantics and domain knowledge on location structure.

In this section, we summarize the proposed approach, from taking a narrative and generating its NWM up to creating one or more narrative worlds for it, as depicted in the pipeline of Figure 1.

The first input required for the NWM generation is the narrative itself. Here, we will consider a narrative to be essentially comprised of plot points, each plot point consisting of an action, any subjects and objects involved in the action, as well as the location that action takes place in. For simplicity, we will consider that these plot points impose a strict ordering of events; see Appendices A and B for examples.

Implicitly, plot points define the minimal set of locations needed in the narrative world, as well as the requirement of specific objects, per location, at specific points in time; we call these *narrative objects*. However, in order to actually generate each complete location, one needs more than just a minimal set of objects, indispensable for a few actions. For example, imagine at some point in the narrative a discussion takes place between two characters siting in a living room. In order to look plausible, this location needs not only, say, the couch where they are sitting, but also a table, chairs, bookshelves, pictures on the wall, some decoration, etc., even if they are not mentioned at all in any plot point. Therefore, we also need knowledge about how ‘typical’ locations, as those needed in the narrative, may ‘look like’. This includes, for example, which types/quantities of which objects they usually contain, relations among them, etc. This information, referred to as domain knowledge, can be represented and kept in a knowledge base. In order to support reasoning about the objects in a location and their inter-relationships, this knowledge base should contain, at least:

- a taxonomy of *actions*, needed to understand and process the plot points;
- an ontology of *objects* (e.g. organized into some taxonomy), needed to understand the content; and,
- a taxonomy of *motifs*, defined as pattern-based probabilistic descriptions of typical locations, needed to generate plausible location layouts.

A NWM integrates, for each location, the information in the respective motif with the object fluctuations that occur in it throughout the narrative. The purpose of this integration is to capture the narrative-specific requirements in terms of the role of individual objects over time and space, and thus help to shape the generic descriptions contained in motifs.

In a way, we can say that motifs bring in the large-scale composition and the plausibility to location descriptions, while the narrative fine-tunes these, assuring they fit the particular story taking place there, over time. This temporal dimension captured by a NWM gives it the ability to overcome potential mismatches between what is *usually found in a location* and what *this narrative needs there at a given moment*.

The following two sections describe, respectively, how the NWM can be derived (Section IV) and how it can be subsequently used to generate a variety of narrative worlds (Section V).

IV. NARRATIVE WORLD MOLD GENERATION

Generation of the Narrative World Mold, as depicted in Figure 1, requires two inputs: the narrative and the knowledge base. Basically, generating the NWM accounts for extracting the essence of the narrative into a compact and usable form. This can be done in three stages: classifying the set of locations that are required by the narrative, determining spatio-temporal fluctuations of objects throughout the narrative, and integrating those fluctuations with the motifs of the various locations.

Much of the essence of a narrative is inherent to the actions in its plot points, which relate characters to objects at some location(s). In the knowledge base, on the other hand, actions are described with their semantics (including e.g. their possibly different uses, meanings, contexts, frames, etc.). To make this explicit, we first identify for each plot point the corresponding action in the knowledge base, and associate them so that this inherent semantics gets captures in what we call the *compiled narrative*.

The actions in the knowledge base include *predicates*, with information e.g. on what is needed in the world for the action to commence, as well as on how the action changes the world state. In particular, so-called *inventory changing predicates* are instrumental to capture when content is added, used, transferred or removed. For example, a *pickup* action has an inventory changing predicate, transferring an object from a location to the inventory of the character performing the action.

By analyzing the compiled narrative, the objects in its locations can be determined and reasoned about. By discovering and capturing how the world changes over time throughout the narrative, all essential objects required by the narrative are identified, and combined with a variety of other objects that typically belong in such locations, considering all their spatio-temporal interrelationships. The three stages of NWM generation are now discussed in more detail.

A. Classification of locations

The set of all locations in a narrative can be easily extracted by walking through all its actions. However, for NW generation purposes, it is convenient to distinguish two types of locations: *motif-only locations* (i.e. those locations without particular narrative requirements, other than those objects in the respective motif) and *narrative locations* (i.e. those in which the narrative requires specific objects and/or layouts).

An initial set of motif-only locations comprises those locations where the only actions taking place neither use nor require any (local) objects; for example, a simple *Talk* action only requires one or more characters, but no local objects.

By analyzing the compiled narrative, we can expand this set of motif-only locations to also include those locations where the only actions taking place leave the location inventory unaffected. For this, it suffices to perform a simple simulation that sequentially scans the compiled narrative, and only processes the inventory changing predicates of each action, in order to track how objects move among characters' and locations' inventories over time. This allows the identification of actions by which e.g. objects are exchanged between characters' inventories, or consumed from a character's inventory. All such actions have no net effect on the location inventory. For example, if a character gives *poison* to another character (as mentioned in plot point 14 of Appendix A), then this action does not add nor remove poison to/from the location: since the character had the poison, it was in their inventory, not in that of the location; and this remains unaffected, might either character leave the location; hence the re-classification of the Guest Room, depicted in Figure 2. Naturally, this narrative simulation assumes that all its actions do take place, which is a reasonable assumption, consequence of the fact that the narrative has the authorial control, thus representing the author's vision of the NW.

B. Dynamic inventory determination

Once the set of locations is classified into motif-only and narrative locations, we can analyze how the narrative locations change over time. This temporal understanding of the narrative will provide the temporal basis to the NWM. As mentioned above, the narrative simulation, in essence, determines the way the content of each location changes over time. This results in a table of *Spatio-Temporal Inventory Fluctuations* (STIFs) indicating, per location, *when* and *which* content is added or removed from its inventory.

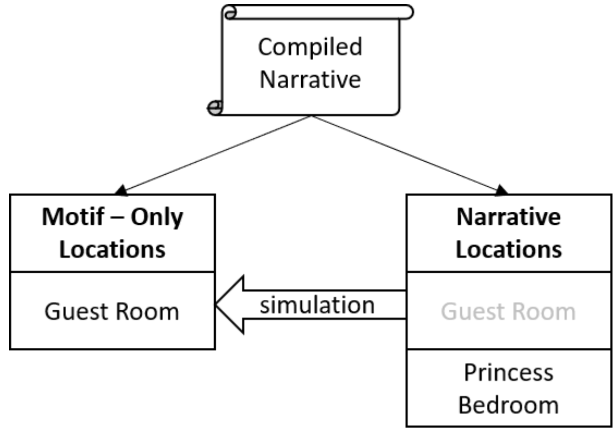

Fig. 2: Example of classification of locations for Appendix A: the *Guest Room*, initially classified as narrative location (from plot point 14), is finally identified as a motif-only location, by simulating the story.

However, this dynamic nature of STIFs, generated from a sequential narrative simulation, is not yet enough to fully determine the complete inventory of every narrative location at any given moment. The reason is that a STIF entry can tell us, at most, when (and where) an object gets mentioned in the narrative for the first time (e.g. 'the Princess picks up a bottle'); but that does not tell us where (nor since when) that object has been before that moment (e.g. whether the bottle was there from the very beginning, or was brought in by someone). To solve this, we have to analyze the inventory changing predicates, deriving whether that object had always been in that location (in case the action is e.g. destroying, moving or picking it up), or whether it is only then joining its inventory (in case the action is e.g. creating or putting down). In this way, we can determine the initial inventory of every narrative location. Moreover, combining that initial state with the STIFs, one can also determine what the inventory of that location will look like at any later stage of the narrative.

C. Integrating narrative inventories with motifs

The location inventories computed above express only the indispensable objects required there by the narrative. In order to make that location appear as a location of a specific type (e.g. a kitchen or a restaurant), a variety of *decoration objects* will have to be added to it, following the typical patterns captured by the motifs, in the knowledge base. Therefore, for each location, we need to integrate its narrative inventory with the respective motif, merging their objects and, possibly re-arranging, their interrelationships in a plausible and meaningful manner.

1) Motifs: Motifs were introduced in Section III as probabilistic descriptions of a given location type, defining the typical patterns found in locations of that type. Essentially, a motif consists of a set of objects and a set of relationships between them, as well as the probability of that content or relationship to occur. For example, a bedroom motif could contain beds, lamps and treadmills: the bed would have a very high probability in the motif, meaning that it commonly occurs in a bedroom; conversely, a treadmill might have a low probability, and be thus seldom found in a bedroom.

For each object in a motif, its semantics can be retrieved from the object taxonomy in the knowledge base [36]. For this reason, it is often possible to trace specific narrative content to some generalization found in the motif, thus increasing the likelihood of that concrete object to assume relationships in the motif. Imagine, for example, one of the objects in the narrative is a *rocking chair*, and the motif has a *chair* object. Once *chair* is a parent of *rocking chair* in the taxonomy, then, for many (generative) purposes, *rocking chair* can be treated the same as the chair object in the motif.

In short, motifs provide a very rich description for creating locations in a NW: for motif-only locations, the motif provides all information needed to procedurally generate them; for narrative locations, that information has to be combined with the STIFs for that location to accurately determine its actual inventory, as explained next.

2) Expanding motifs with the narrative: It may happen that some of the objects the narrative requires at a location are already found in its respective motif. In such cases, we just have to modify the occurrence probability of such narrative objects to 1, in the motif. All other objects required by the narrative at a location have to be also appended to its motif, as well as tagged with an occurrence probability of 1.

Appending each of these new objects, say *x*, to the motif, can be done by connecting it through one or more relationships to existing objects. However, this motif expansion should not be improvised, but rather done sensibly, so that *x* 'plausibly fits into' the motif. A reasonable way to do this is to lookup, in the knowledge base, for other motifs in which *x* occurs, in order to 'find inspiration' for frequent and plausible objects *x* relates to. A chain of relationships from another motif may provide the necessary structure that connects each object *x* to content already in the motif. Typically, multiple such chains can be found; using different criteria for selecting one of them will guarantee diversity in the resulting expanded motif. One such criterion for choosing the best relationship chain is to find the most succinct, or shortest, chain. This criterion neither clutters the base motif with too many new objects, nor obfuscates the learned pattern of the location. An example of motif expansion is given in Figure 3.

All together, the STIFs, the expanded motifs for narrative locations, and the motifs for the motif-only locations, constitute the Narrative World Mold. The NWM allows for plenty of diversity and variability in the world generation process, given the stochastic nature of its motifs. Moreover, because a NWM explicitly includes all content required throughout the whole narrative, any world correctly generated from such a NWM is guaranteed to support that narrative.

V. Preparing Narrative World Generation

As shown in the previous section, the inclusion of the Spatio-Temporal Inventory Fluctuations in the Narrative World Mold can help us determine the precise inventory of a location at any given point in time. Therefore, the first step in preparing

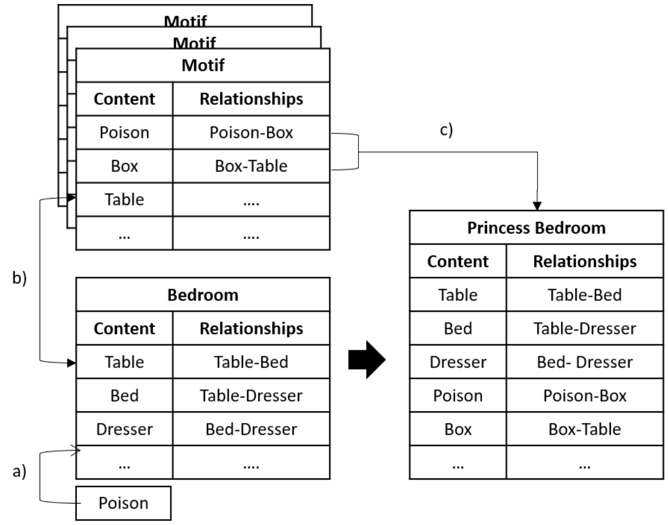

Fig. 3: Expanding the *Princess Bedroom* motif to include content and relationships from the STIF: (a) the *Poison* object is not normally found in a *Bedroom*; (b) other motifs in the knowledge base are searched until a connection between *Poison* and an object in the bedroom is found (*Table*); (c) the new content (*Poison*) and its relationship (to *Table*) are appended to the expanded *Princess Bedroom* motif.

the NWM for generation consists of (i) choosing a time point t_i for the desired NW (for example, the start of the narrative), and (ii) reducing the spatio-temporal dependencies in the NWM, by grounding the motif of each location onto that time point t_i. As a result, each grounded motif will not feature content which will only appear in that location later in the story; but it will contain all content which was brought in to that location (and not taken away) up to time point t_i in the narrative. In other words, solving the spatio-temporal dependencies in the NWM for time point t_i, should guarantee that the grounded motif for each location contains all objects that are relevant for (and from) that moment.

These grounded motifs can typically serve as the input to a location procedural generator such as the ones mentioned in Section II. Still, care must be taken when grounding a motif onto a time point t_i, so that the resulting structure still supports the subsequent narrative. Firstly, as each motif in the NWM contains the content and relationships for all times in a location, one has to guarantee that the pruning of 'future' objects/relationships from the motif does not leave it disconnected, with dangling (sets of) objects lacking a relationship to that location. Secondly, when pruning 'future' objects/relationships from a motif, one should also prevent removing *Implied Necessary Content* (INC), defined as other content that is going to be required in that location at a later stage (though not at stage t_i). INC is instrumental to guarantee that those 'future' objects can find some relationship to the environment. An example of this is given in Fig. 4. The objects connected by any relationships to the removed objects are then marked as INC, and their probabilities re-weighted in the motif, so that they stay in the motif to be taken in the generation process.

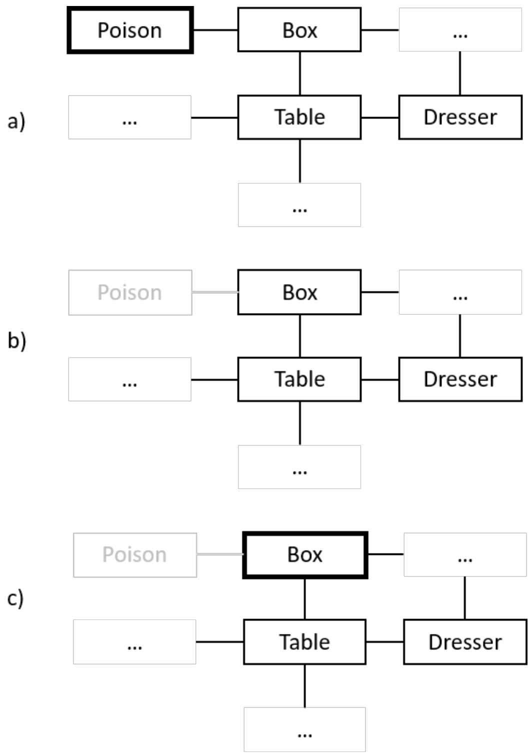

Fig. 4: An example of INC in a location: (a) At some moment t_i in the story, *poison* (in bold) will be brought in and hid within a *box*; (b) to generate that location at some time point $t_j < t_i$, the *poison* is removed from the motif (grayed out); (c) however, the *box* is marked as INC and kept for the generation at time point t_j, to allow for that 'future' relationship with the *poison*.

INC is a way to safeguard against creating an environment ill-equipped for the narrative. Naïvely generating a room without consideration for past and future narrative events might mean that necessary content may be unable to form a relationship with the generated world. For example, if a cup is brought into a room and left there, common sense implicitly states that it would be left on some flat surface. In the mold, this manifests itself as a relationship between a cup object and other objects in the location, e.g. a table. Grounding at time point t_i ensures the cup will not be generated prematurely in the room. However, the table should always be generated in the room to allow the location to accommodate the later addition of the cup. By imposing that those implied relationships (and therefore, the content they involve) are still necessary, the narrative will remain properly supported.

After determining all Implied Necessary Content for the chosen time point t_i, the NWM can be translated into a suitable format, and input to any of the data-driven procedural generation methods discussed in Section II. However, as the content in the motifs of the NWM is probabilistic, there is plenty of room to steer the quantity and variety in that input. Examples of this can include adjusting the probability threshold when selecting which objects to include in the generation, or using specific designer preferences, possibly filtering the selection and instantiation of decorative objects and relations to be actually requested from the generator.

VI. Implementation Aspects

Narrative world generation is dependent on the setup of the knowledge base. We now briefly describe how we structured and populated our object taxonomy, which resembles the knowledge graphs of Liang et al. [37]; see an example taxonomy in Figure 5. In this taxonomy, each object has a single parent and one set of related semantics (including a geometric model and an axis-aligned bounding box). The taxonomy is derived from WordNet [38], [39]. For each object, its semantics contains information used to instantiate and reason about it, similar to the attributes of Entika [40]. In addition, we also store a reference to possible 3D geometric representations, including their approximate bounding box dimensions, used during the final step of world generation to lay out the objects in the world.

Following previous work [41], we store motifs for each individual location in the knowledge base as a factor graph. We generate each graph using the method of Kermani et al. [24]; see an example in Figure 6. Nodes in the graph represent particular content, and include its frequency of occurrence, i.e. the probability of that object occurring in that location. Edges between nodes are pair-wise relationships. The same two nodes can be connected by multiple edges, provided these are semantically different (e.g. one is a spatial distance relationship and the other is an orientation relationship). Concrete pairwise spatial distances between objects are stored separately and are determined only during the procedural generation phase. Relationships also feature a probability, representing the frequency with which that relationship was detected in the data-set used (in our case, the SUN RGB-D data-set [42]).

As we analyze all locations in the Sun-RGBD dataset, and not just locations with a large number of scenes, we also perform some additional manual pre- and post-processing of the rule-set, e.g. using a cut-off threshold to prune objects/relationships with very low occurrence probability. Furthermore, motifs that do not contain a wall or floor object are merged with parents that do, higher up in the taxonomy. Finally, we use Dijkstra's algorithm to keep necessary con-

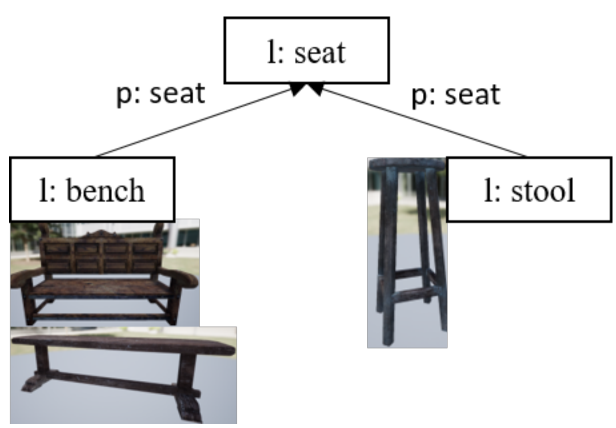

Fig. 5: A portion of an example object taxonomy containing three objects, with their relationships and labels.

tent connected when expanding the motifs (as discussed in Section IV-C).

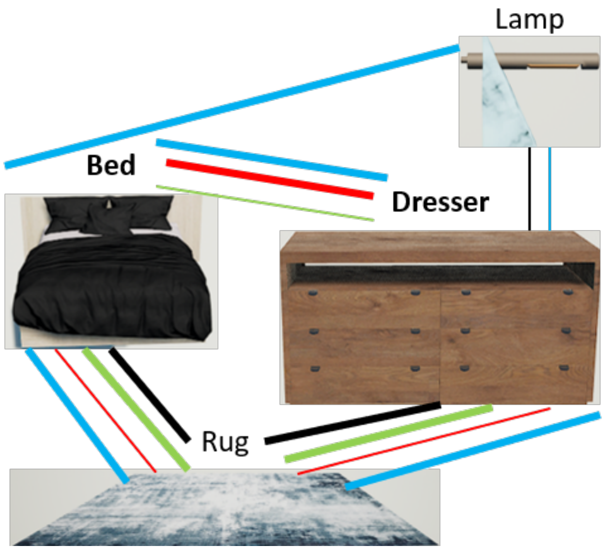

Fig. 6: An example factor graph from a *Bedroom* motif. Object probabilities are shown as high (bold) or low (plain) probabilities. The relationships represented are *support* (black), *proximity* (blue), and two *orientations* (green and red). Relationships between objects are shown with high probability (thick line) or low probability (thin line) of existing in a given scene.

VII. Evaluation

The Narrative World generation approach described so far aims at procedurally generating a large variety of environments, all of which adhere to the requirements set by a given narrative. One can devise at least two ways in which this generative process can fail: (i) the story over-constrains the generator to such a degree that it cannot generate *varied environments*, or (ii) the generator creates *locations that do not fully support the story*. In order to evaluate the NWM approach presented so far, and show that it does not fail in neither way, we formulate the following three hypotheses:

- H1: Calculating the Spatio-Temporal Inventory Fluctuations (STIFs) of a narrative world, one can always (i) distinguish motif-only locations from narrative locations, and (ii) determine the location to place necessary content used in the narrative.
- H2: Expanding a motif is a necessary step to generate a narrative world containing relational information to objects that do not usually exist in its locations.
- H3: An Implied Necessary Content (INC) analysis over the entire narrative is necessary to generate narrative worlds.

The examination of H1 (Section VII-A) will determine the suitability of plot point analysis to deduce the extent of the narrative domain. It will also show the usefulness of applying STIFs to determine the initial locations of objects.

Examination of H2 and H3 will be combined in a quality-similarity study (Section VII-B). For the former, we will show that motif learned data alone cannot always guarantee narrative enactment. For the latter, we will show the ability of the STIFs to capture that intent. Together, ascertaining these three hypotheses will ground the suitability and usefulness of Narrative World Molds for creating time-varying worlds, a distinguishing note of NWs, thus improving upon current time-invariant approaches.

TABLE I: Number of motif-only locations compared to the total number of locations, as determined by the *Narrative* (i.e. all locations in its domain that contain narrative objects), by the *Compiled Narrative*, and by the *STIF* (i.e. the full derivation).

Story	Narrative	Compiled Narrative	STIF
Heist	4/9	2/9	4/9
Medieval	4/7	3/7	4/7

To validate these hypotheses, we use two narratives that were initially generated using GLAIVE [43]. By testing with a narrative that contains an over-specified initial state, we can better determine the accuracy of the information inferred by the NWM. GLAIVE provides several example domains[1], of which we use the medieval and western domains. For completeness, the generated narratives can be found in Appendices A and B.

The domain of both stories was simulated using the Parameterized Action Representation [44], which treats all characters and locations as props, allowing them to be tracked similarly to objects[2]. It also contains an action field that transforms the story state before the action is executed, which is used to ensure that each plot point is feasible. For locations in the story that did not have a motif in the knowledge base (e.g. the *Princess Bedroom* in the medieval story), we manually connected a generic motif to the name of that location (e.g. the *Bedroom* motif). Worlds are generated using both an optimization generator [24] and a planning-based generator [45][3].

A. Temporal Evaluation

To test hypothesis H1, we examine the ability to determine motif-only locations from the compiled narrative. One advantage to using generated stories is that objects are given an initial location in the narrative domain, which can be used as ground truth to compare against narrative objects found by the NWM. Furthermore, the domain specifies all objects that *can* be in the narrative, but are not always used in that particular story. Unused specified objects are decorations, allowing us to test the NWM's ability to fill in locations even when it is not explicitly told that those elements exist (as they are not used in the narrative).

Table I presents how many locations can be identified as motif-only locations from the original narrative and compiled

[1] https://nil.cs.uno.edu/projects/glaive/

[2] Source code can be found at https://github.com/j-timothy-balint/Simplified-Simulation

[3] Source code that performs the entire method as well as the dataset of generated worlds can be found at https://github.com/j-timothy-balint/Narrative-World-Molds

TABLE II: Number of objects used in the narrative, as determined by the *Domain*, by the *Narrative* objects, by the narrative objects initially assigned to a *Location*, by the STIF, and by the full Narrative World Mold.

1. Story	2. Domain	3. In Narrative	4. Narrative In Location	5. STIF	6. Narrative World Mold
Heist	13	11	6	6	6
Medieval	11	4	4	4	9

narrative. Moreover, Table I shows that calculating the fluctuation of objects among locations can detect several more motif-only locations than simply using the compiled narrative. In most cases, this is because an object enters a location with a character, but is never used within it. Such objects, therefore, play no narrative role in that location, which confirms the importance of calculating inventory fluctuations.

We test out the ability of a NWM to anticipate the objects required in each location by determining if they appear in either the STIF or NWM, as shown by the results in Table II. NWMs are aimed at populating the locations with narrative and decorative objects. However, objects may be specified in the world *domain* (column 2) but not necessarily used in the story. These latter are therefore unknown to the Narrative World Mold; all other objects are specified in the plot points of the narrative (column 3). Some of these objects may be only transferred between characters. All other objects are either explicitly or implicitly tied to a location by some plot point (column 4).

In Table II, the computed STIF (column 5) captures exactly the same location-bound objects of column 4, demonstrating their efficacy. Computing the STIF can capture any fluctuations based only on the narrative, without needing any world domain. Finally, the motif data in the NWM is able to bring in many story-unmentioned objects. For example, in the Medieval narrative the domain (column 2) includes *chairs* around a table; however, these objects occur nowhere in the narrative. The NWM (column 6) is able to include such objects via the motif data.

In conclusion, a Narrative World Mold can not only capture and place all narrative content, but the remaining (decorative) content gets mostly captured in its motif. This is done without modifying any location generators, as all information they need is already captured in the NWM.

B. Quality-Similarity Experimentation

While STIFs in NWMs may capture the intent of an author by correctly identifying the initial location of objects, we still must show that the generated NW can anticipate and accommodate the needs of the narrative (i.e. it is a correct NW), while ensuring that the NWM does not suppress the generators ability to create diverse locations. We therefore scrutinize hypotheses H2 and H3 by examining the difference in quality and similarity that occurs at each step of the NW generation process. Using a leave-one(-step)-out approach, we determine its necessity for the full NW generation, outlined in Fig 1. Worlds are generated using both an optimization generator [24] and a planning-based generator [45].

For the *quality analysis*, we request each of the two generators to produce 1000 worlds for each step left out, and also for each of two time points, the beginning and end of the story. We then check each resulting world to determine if it correctly meets the NW definition. Therefore, for a given time point, both the narrative objects that should exist in the room should be in the room, and objects that can form a relationship with narrative objects, which either were previously in the room or will exist later in the story, should also be in the generated world. H2 and H3 are satisfied as long as leaving one step out leads to the method failing to always guarantee a correct NW output (within a margin of error). The percentage of correctly generated worlds is shown in Table III. As the difference in narrative worlds is skewed based on the number of rooms in that world that require additional objects, we present both the mean and median percentage of correct worlds.

TABLE III: Median and mean percentage of correct narrative worlds for removing one step of the generation pipeline.

Story	Full NWM	No Motif Expansion	No INC
Heist	$100/92$	$1/20$	$93/71$
Medieval	$100/96$	$1/1$	$97/78$

Table III shows the necessity of including both narrative objects and the objects that support them when generating a NW. The second column shows that oftentimes narrative objects were not learned as part of the motif, and therefore, how to fit them into the motif must be accounted for. This provides strong evidence for H2. When INC is not included (third column), the generator simply omits preparing the world for some event(s) in the narrative, typically failing to yield a correct NW.

While Motif Expansion and INC inclusion are required to ensure creation of a correct NW, their inclusion in the NWM obviously also constrains the world generators, making it harder to converge upon a solution. At times, a generator will simply fail to find a world that satisfies all constraints, as can be seen in the first column of Table III. However, this is a rare occurrence compared to excluding one of the steps of NW generation, confirming that both motif expansion (H2) and INC inclusion (H3) are necessary steps to generate a correct NW.

Since a narrative may sometimes over-constrain a world generator, we must assess not only the quality of the narrative worlds it generates, but also their similarity. For this analysis, we probe the ability of the generator to still create a varied environment by measuring the similarity among the inventory set of rooms generated for the NW. Using a leave-one(-step)-out approach, we generate, for each of the two generators, 50 worlds for both the Heist and Medieval stories. For this experiment we are concerned with generation diversity rather than quality. Therefore, unlike for the previous experiment, we consider all generated worlds, even if some of them may have rooms that are missing some narrative object(s). For the same reason, we leave motif-only rooms out from this analysis. We determine every object that appears in a location for a condition, and then calculate its average appearance in the same room for the compared condition. This is then averaged over all objects to create the average similarity between conditions, see Table IV. H2 and H3 are satisfied if the Full NWM method has a similar average scoring to all other methods.

TABLE IV: Comparison of the average similarity scores (in percentage) for removing one step in the generation pipeline. Associated error is shown as the 95% confidence interval.

-	**Full**	**No Motif Expansion**	**No INC**
Full	55 ± 8	39 ± 7	50 ± 9
No Motif Expansion	44 ± 8	50 ± 6	42 ± 9
No INC	53 ± 8	41 ± 7	50 ± 8

Table IV shows that each condition has a similar impact on the similarity of generated rooms, regardless of which step is left out. Scores on the *No INC* row of Table IV, show that not including INC will often create a similar set of worlds as from the full NWM. However, as we saw from the results on Table III, that similarity is no guarantee that they are narrative worlds. Altogether, the two experiments show that the full NWM is required to generate a NW, and that it does not adversely effect the diversity of the generated worlds, thus proving H2 and H3.

C. Evaluation conclusion

The temporal evaluation in SectionVII-A shows that, for NW generation, it is necessary to determine how the narrative changes the surrounding environment over time. The quality-similarity evaluation in Section VII-B further shows that motif expansion and INC are two narrative-specific steps of a NWM, which are indispensable for the NW generation process, and have little effect on NW diversity. Altogether, this evaluation has shown that the various NWM features analyzed must be considered to guarantee that a generator yields a real NW, fully supporting the entire narrative from begin to end.

VIII. Conclusion

Narrative worlds (NWs), consisting of various interrelated locations that contain many required and plausible objects, are complex and difficult to create. Their procedural generation therefore poses huge research challenges, including taking into account the spatio-temporal interrelationships between narrative locations, characters and objects. To approach and solve these challenges, we proposed an intermediate representation, the Narrative World Mold (NWM), which integrates, for each narrative, the spatio-temporal descriptions of its essential narrative elements with object semantics and domain knowledge.

We have shown that, by first building a NWM for a story, various challenges of procedural NW generation can be overcome, without detriment of the variety in the generated worlds. In particular, using a full-fledged NWM as proposed here is indispensable whenever the narrative involves content that is either not normally found in the motifs, or fluctuating between locations over time. Furthermore, we have shown that the constraints imposed by a NWM have a minor impact on the ability of a procedural world generator to create a varied environment. As long as the NWM captures all necessary semantic information on narrative objects, a variety of narrative worlds can be generated, all of them supporting the narrative.

In addition to the planner output narratives used throughout this article (Apendices A and B), a variety of other narrative content could serve as input to the NW generation approach described here, including gameplay traces (which typically include many alternative variants of a narrative), annotated natural language stories (as e.g. those created by TaleMaker [46]) and even simple theatrical scripts or other plot-point-based stories (as e.g. those often told by children [47]). Naturally, for all these narrative alternatives to work, it is instrumental that the knowledge base contains appropriate motifs for the respective locations. NWMs are highly dependent on the semantics in the knowledge base. Therefore, a limitation of generating NWMs using data-driven knowledge is that a mold can very well semantically agree with the knowledge base, but be nonsensical to the narrative author. For overcoming this, it is crucial to improve scene recognition and mining systems, an area of active research [48]–[50]; this will capture richer semantics in the motifs in the knowledge base, and thus enable the generation of better NWMs and, ultimately, of even even better NWs.

Procedural generation of narrative worlds has the potential to open up many new possibilities for game development [51]. Among the numerous open research challenges ahead, we see the convenience of (i) exploring action-centered data-driven knowledge to complement the contribution of location motifs to NW generation, and (ii) investigating the role that global narrative context features (such as theme, epoch, genre, etc...) can play in determining and refining narrative worlds. We believe that Narrative World Molds, as proposed in this article, provide a solid and sound foundation for successfully approaching such challenges.

Acknowledgments

This work was partly supported by the Netherlands Organisation for Scientific Research under Project 314-99-104.

Appendix A
Medieval Story

1) Prince talk-multiple princess king *in* throne-room
2) Princess fall-in-love prince *in* throne-room
3) Prince fall-in-love princess *in* throne-room
4) Maiden hides *in* guestroom
5) Prince move *in* throne-room *to* guestroom
6) Princess talk king *in* throne-room
7) Princess move *in* throne-room *to* corridor
8) Princess move *in* corridor *to* guestroom
9) Princess talk prince *in* guestroom
10) Princess move *in* guestroom *to* bedroom_princess
11) Princess pickup poison *in* bedroom_princess
12) Princess move *in* bedroom_princess *to* guestroom
13) Prince talk princess *in* guestroom

14) Princess give prince poison *in* guestroom
15) Prince move *in* guestroom *to* dining-room
16) Princess move *in* guestroom *to* dining-room
17) King move *in* throne-room *to* dining-room
18) Maiden move guestroom *to* kitchen
19) Maiden hides *in* kitchen
20) Prince talk-multiple princess king *in* dining-room
21) Prince move *in* dining-room *to* kitchen
22) Prince poisons cup_king poison *in* kitchen
23) Prince move *in* kitchen *to* dining-room
24) Maiden swaps cup_king cup_prince *in* kitchen
25) Maiden pickup cup_prince *in* kitchen
26) Maiden move *in* kitchen *to* dining-room
27) Maiden place-on cup_prince table *in* dining-room
28) Prince drink cup_prince *in* dining-room
29) Prince dies *in* dining-room
30) Princess talk king *in* dining-room
31) Princess move *in* dining-room prison
32) King marry maiden *in* dining-room

APPENDIX B HEIST STORY

1) Robbie hatch six-shooter horse_brown bank mother-lode
2) Jill open *in* general-store
3) Sally move *in* main-street *to* bank
4) Sally withdraw money_dress *in* bank
5) Sally move *in* bank *to* main-street
6) Sally sell Anne tomatoes money_tomato *in* main-street
7) Robbie move *in* main-street *to* alley_dark
8) Robbie hide *in* alley_dark
9) Barney move *in* barneys_room *to* saloon
10) Robbie pick-pocket sally money_dress *in* main-street *from* alley_dark
11) Robbie move *in* alley_dark *to* main-street
12) Robbie move *in* main-street *to* saloon
13) Robbie buy-drinks barney money_dress *in* saloon
14) Robbie escort barney *in* saloon *to* main-street
15) Robbie escort barney *in* main-street *to* alley_dark
16) Robbie lay barney *in* alley_dark bank
17) Robbie take barney six-shooter *in* alley_dark
18) Robbie move *in* alley_dark *to* main-street
19) Robbie move *in* main-street *to* saloon
20) Robbie cheat game_poker money_dress money_poker *in* saloon
21) Robbie move *in* saloon *to* main-street
22) Robbie pawn seller_broker locket money_locket *in* main-street
23) Robbie buy seller_horse horse_brown money_locket *in* main-street
24) Robbie ride horse_brown *in* main-street *to* bank
25) Robbie hold-up six-shooter tom *in* bank
26) Robbie collect mother-lode *in* bank
27) Robbie getaway mother-lode horse_brown *in* bank *to* out-of-town
28) Tom pick-up handcuff *in* sheriffs-office
29) Tom move *in* sheriffs-office *to* barber-shop
30) Tom ride horse_white *in* barber-shop *to* out-of-town
31) Robbie arrested-by tom handcuffs mother-lode *in* out-of-town

REFERENCES

[1] B. Kybartas, R. Bidarra, and E. Eisemann, "Integrating semantics and narrative world generation," in *Proceedings of FDG 2014*, Apr. 2014.

[2] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, 1st ed., ser. Computational Synthesis and Creative Systems. Berlin, Germany: Springer, 2016.

[3] B. Kybartas and R. Bidarra, "A survey on story generation techniques for authoring computational narratives," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 3, pp. 239–253, 2017.

[4] M. O. Riedl and R. M. Young, "Narrative planning: Balancing plot and character," *Journal of Artificial Intelligence Research*, vol. 39, no. 1, pp. 217–268, Sep. 2010.

[5] B. Kartal, J. Koenig, and S. J. Guy, "User-driven narrative variation in large story domains using monte carlo tree search," in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 69–76.

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

[7] R. Farrell and S. Ware, "Fast and diverse narrative planning through novelty pruning," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 12, no. 1, Sep. 2016.

[8] P. Ammanabrolu, W. Cheung, D. Tu, W. Broniec, and M. Riedl, "Bringing stories alive: Generating interactive fiction worlds," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, no. 1, pp. 3–9, Oct 2020. [Online]. Available: https://ojs.aaai.org/index.php/AIIDE/article/view/7400

[9] J. Robertson and R. M. Young, "Perceptual experience management," *IEEE Transactions on Games*, vol. 11, no. 1, pp. 15–24, 2019.

[10] M. Diamanti and D. Thue, "Automatic abstraction and refinement for simulations with adaptive level of detail," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 15, no. 1, pp. 17–23, Oct 2019.

[11] T. Hayton, J. Porteous, J. Ferreira, and A. Lindsay, "Narrative planning model acquisition from text summaries and descriptions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 02, pp. 1709–1716, Apr 2020.

[12] B. Coyne and R. Sproat, "WordsEye: An automatic text-to-scene conversion system," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: Association for Computing Machinery, 2001, p. 487–496.

[13] A. X. Chang, M. Eric, M. Savva, and C. D. Manning, "SceneSeer: 3D scene design with natural language," *arXiv preprint arXiv:1703.00050*, 2017.

[14] M. Ma, "Automatic conversion of natural language to 3D animation," Ph.D. dissertation, University of Ulster, 2006.

[15] S.-H. Zhang, S.-K. Zhang, Y. Liang, and P. Hall, "A survey of 3D indoor scene synthesis," *Journal of Computer Science and Technology*, vol. 34, no. 3, pp. 594–608, 2019.

[16] T. Tutenel, R. Bidarra, R. M. Smelik, and K. J. De Kraker, "Rule-based layout solving and its application to procedural interior generation," in *Proceedings of the CASA Workshop on 3D Advanced Media In Gaming And Simulation*, 2009, p. 10.

[17] T. Yamakawa, Y. Dobashi, and T. Yamamoto, "Efficient simulation of furniture layout taking into account lighting environment," in *Proceedings of the 29th International Conference on Computer Animation and Social Agents*. New York, NY, USA: ACM, 2016, pp. 173–179.

[18] P. Henderson and V. Ferrari, "A generative model of 3D object layouts in apartments," *arXiv preprint arXiv:1711.10939*, 2017.

[19] T. Weiss, M. Nakada, and D. Terzopoulos, "Automated layout synthesis and visualization from images of interior or exterior spaces," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 41–47.

[20] P. Kán and H. Kaufmann, "Automated interior design using a genetic algorithm," in *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '17. Gothenburg, Sweden: Association for Computing Machinery, 2017.

[21] L.-F. Yu, S.-K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher, "Make it home: Automatic optimization of furniture arrangement," *ACM Transactions on Graphics*, vol. 30, no. 4, pp. 86:1–86:12, Jul. 2011.

[22] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan, "Example-based synthesis of 3D object arrangements," *ACM Transactions on Graphics*, vol. 31, no. 6, p. 135, 2012.

[23] M. Yan, X. Chen, and J. Zhou, "An interactive system for efficient 3D furniture arrangement," in *Proceedings of the Computer Graphics International Conference*. New York, NY, USA: ACM, 2017, pp. 29:1–29:6.

[24] Z. S. Kermani, Z. Liao, P. Tan, and H. Zhang, "Learning 3D scene synthesis from annotated RGB-D images," *Computer Graphics Forum*, vol. 35, no. 5, pp. 197–206, 2016.

[25] M. Fisher, M. Savva, Y. Li, P. Hanrahan, and M. Nießner, "Activity-centric scene synthesis for functional 3D scene modeling," *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 179:1–179:13, Oct. 2015.

[26] Y. Jiang, H. S. Koppula, and A. Saxena, "Modeling 3D environments through hidden human context," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2040–2053, Oct. 2016.

[27] R. Ma, H. Li, C. Zou, Z. Liao, X. Tong, and H. Zhang, "Action-driven 3D Indoor Scene Evolution," *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 173:1–173:13, Nov. 2016.

[28] R. Ma, A. G. Patil, M. Fisher, M. Li, S. Pirk, B.-S. Hua, S.-K. Yeung, X. Tong, L. Guibas, and H. Zhang, "Language-driven synthesis of 3D scenes from scene databases," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 212:1–212:16, Dec. 2018.

[29] Q. Fu, X. Chen, X. Wang, S. Wen, B. Zhou, and F. Hongbo, "Adaptive synthesis of indoor scenes via activity-associated object relation graphs," *ACM Transactions on Graphics*, vol. 36, no. 6, nov 2017.

[30] R. v. d. Linden, R. Lopes, and R. Bidarra, "Designing procedurally generated levels," in *Proceedings of the second AIIDE workshop on Artificial Intelligence in the Game Design Process*, Nov. 2013.

[31] D. Camozzato, L. Dihl, I. Silveira, F. Marson, and S. R. Musse, "Procedural floor plan generation from building sketches," *The Visual Computer*, vol. 31, no. 6, pp. 753–763, Jun. 2015.

[32] W. Wu, L. Fan, L. Liu, and P. Wonka, "MIQP-based layout design for building interiors," *Computer Graphics Forum*, vol. 37, no. 2, pp. 511–521, May 2018.

[33] D. Karavolos, A. Bouwer, and R. Bidarra, "Mixed-initiative design of game levels: integrating mission and space into level generation," in *Proceedings of FDG 2015*, jun 2015.

[34] K. Hartsook, A. Zook, S. Das, and M. O. Riedl, "Toward supporting stories with procedurally generated game worlds," in *2011 IEEE Conference on Computational Intelligence and Games*. IEEE, 2011, pp. 297–304.

[35] J. Valls-Vargas, S. Ontanon, and J. Zhu, "Towards story-based content generation: From plot-points to maps," in *2013 IEEE Conference on Computational Intelligence and Games*. IEEE, 2013, pp. 1–8.

[36] J. T. Balint and R. Bidarra, "A generalized semantic representation for procedural generation of rooms," in *Proceedings of FDG 2019*. San Luis Obispo, California, USA: Association for Computing Machinery, 2019.

[37] Y. Liang, F. Xu, S.-H. Zhang, Y.-K. Lai, and T. Mu, "Knowledge graph construction with structure and parameter learning for indoor scene design," *Computational Visual Media*, vol. 4, no. 2, pp. 123–137, 2018.

[38] Princeton University, "About WordNet," Princeton University, Tech. Rep., 2010.

[39] C. Pelkey and J. M. Allbeck, "Populating virtual semantic environments," *Computer Animation and Virtual Worlds*, vol. 24, no. 3, pp. 405–414, May 2014.

[40] J. Kessing, T. Tutenel, and R. Bidarra, "Designing semantic game worlds," in *Proceedings of The Third Workshop on Procedural Content Generation in Games*. Raleigh, NC, USA: Association for Computing Machinery, 2012, p. 1–9.

[41] Y.-T. Yeh, L. Yang, M. Watson, N. D. Goodman, and P. Hanrahan, "Synthesizing open worlds with constraints using locally annealed reversible jump MCMC," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 56:1–56:11, Jul. 2012.

[42] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[43] S. Ware and R. M. Young, "Glaive: A state-space narrative planner supporting intentionality and conflict," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 10, no. 1, Sep. 2014.

[44] R. Bindiganavale, W. Schuler, J. M. Allbeck, N. I. Badler, A. K. Joshi, and M. Palmer, "Dynamically altering agent behaviors using natural language instructions," in *Proceedings of the Fourth International Conference on Autonomous Agents*. New York, NY, USA: ACM, 2000, pp. 293–300.

[45] M. Savva, A. X. Chang, and M. Agrawala, "Scenesuggest: Context-driven 3d scene design," *arXiv preprint arXiv:1703.00061*, 2017.

[46] M. Bueno and R. Bidarra, "Mixed-initiative story co-creation with TaleMaker," in *Proceedings of 17th International Conference on the Foundations of Digital Games (FDG)*, 2022, pp. 1–12.

[47] M. Kegeleers and R. Bidarra, "Story authoring in augmented reality," in *Proceedings of 16th International Conference on Computer Graphics Theory and Applications (GRAPP)*, 2021.

[48] S. Satkin, M. Rashid, J. Lin, and M. Hebert, "3DNN: 3D nearest neighbor," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 69–97, Jan. 2015.

[49] Z. Liu, Y. Zhang, W. Wu, K. Liu, and Z. Sun, "Model-driven indoor scenes modeling from a single image," in *Proceedings of the 41st Graphics Interface Conference*. Halifax, Nova Scotia, Canada: Canadian Information Processing Society, 2015, p. 25–32.

[50] X. Song, C. Chen, and S. Jiang, "RGB-D scene recognition with object-to-object relation," in *Proceedings of the 25th ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2017, pp. 600–608.

[51] A. Liapis, G. N. Yannakakis, M. J. Nelson, M. Preuss, and R. Bidarra, "Orchestrating game generation," *IEEE Transactions on Games*, vol. 11, no. 1, pp. 48–68, 2019.

J. Timothy Balint received his bachelors degree in Physics from Roanoke college in 2011. He received his M.S in computer science in 2014 and his Ph.D. in computer science in 2017 from George Mason University. He then was a post-doctoral researcher at Delft University of Technology, where he worked on this research project until 2019, before becoming an assistant professor at Stonehill College, Mass. There he led the Data Science Major and the *Generative Worlds of Autonomous, Interactive Non-Player Characters* (GWAIN-PC) lab, focusing on virtual characters, procedural content generation, and narrative development. He is currently a Data Scientist at Battelle Memorial Institute.

Rafael Bidarra graduated in 1987 in electronics engineering at the University of Coimbra, Portugal, and received his Ph.D. in 1999 in computer science from Delft University of Technology, The Netherlands.

He is associate professor Game Technology at the Faculty of Electrical Engineering, Mathematics and Computer Science of Delft University of Technology. He leads the research lab on game technology at the Computer Graphics and Visualization Group. His current research interests include procedural and semantic modeling techniques for the specification and generation of both virtual worlds and gameplay; serious gaming; game adaptivity and interpretation mechanisms for in-game data. Rafael has numerous publications in international journals and conference proceedings, integrates the editorial board of several journals, and has served in many conference program committees.