Making sense of emergent narratives: An architecture supporting player-triggered narrative processes

Simon Chauvin, Guillaume Levieux, Jean-Yves Donnart, Stéphane Natkin

Making Sense of Emergent Narratives: An Architecture Supporting Player-Triggered Narrative Processes

Simon Chauvin
C.N.A.M / C.E.D.R.I.C
Paris, France
simon.chauvin@cnam.fr

Guillaume Levieux
C.N.A.M / C.E.D.R.I.C
Paris, France
guillaume.levieux@cnam.fr

Jean-Yves Donnart
Thales Training and Simulation
Osny, France
jean-yves.donnart@thalesgroup.com

Stphane Natkin
C.N.A.M / C.E.D.R.I.C
Paris, France
stephane.natkin@cnam.fr

***Abstract*—Emergent games have the particularity to allow more possible situations to emerge than progression games do. Coupled with procedural content generation techniques they also tend to increase the number of possible situations that players can encounter.**

However, in case the player is not creative or lucky enough these many emergent situations can have a low narrative value. This article addresses this problem through an architecture that gives players more responsibilities towards the story by allowing them to trigger Narrative Processes. A Narrative Process is a script capable of making meaningful modifications to the story in real time. Our proposed architecture relies on an Interpretation Engine whose role is to make sense of the emergent world as it is changing and inform the Narrative Processes with high level story concepts such as actors and places.

We first cover the basics of emergent games and interactive narratives and then present the architecture behind the Narrative Processes as well as the Interpretation Engine. We conclude by a discussion of the potential impact of our architecture on the fundamental characteristics of emergent games.

Keywords*—*Interactive Storytelling, Emergent Narrative, Emergent Game, Procedural Content Generation, Minecraft

I. Introduction

The number of emergent games based on procedural content generation techniques has been growing these past few years. Many of them are among the most successful video games released each year [1], [2], [3], [4]. While this might be a temporary trend, the specific approach to game design taken by emergent games has produced new play schemes and new game narrative possibilities [5], [6]. It laid the ground for a more thorough investigation of their narrative potential and improvement.

In the following sections we introduce our approach to emergent game narratives as well as our plan for improving them through player-triggered Narrative Processes.

A. Emergent Games

Emergent games are created through an approach to game design that emphasizes the creation of rule-based systems rather than pre-scripted paths [7]. On the one hand, emergent games provide consistent rules that can be combined in multiple ways to create many new situations that are unique and that were not predefined. On the other hand, scripted games use contextual rules that lead to accurately predefined situations. As a result, authors of emergent games favor combination since they have less control on the sequence of events. Authors of scripted games instead favor control over combination.

Even though the author cannot precisely design the specific story that players will experience within an emergent game, many emergent games can still provide a strong narrative experience [5]. Players take part in the game's story creation and express themselves through their choices so that each play session becomes a player-driven narrative [8]. We consider that this kind of narrative is a major aspect of emergent games and has greatly contributed to their recent success. Moreover, the more a game relies on procedural content techniques, the more it tends to be systemic, allowing for more emergent gameplay and more player-driven narratives.

To further our understanding of emergent games we first defined their fundamental properties in a previous publication [9]:

- Coherence: the coherence of the game world enabled by the use of consistent rules
- Agency: the ability offered to players to have an impact in deeply interactive and immersive worlds
- Possibility Space: a large space of possibility that allows for many possible situations to emerge
- Uncertainty: the ability of the game world to generate many surprising situations
- Codirection: the critical role of players in experiencing the narrative but also participating in its creation

A major part of the most recent and successful emergent games make extensive use of procedural content generation. It allows them to increase their space of possibility as well as their potential for uncertainty. Procedural content generation directly sustains the fundamental characteristics of emergent games. For instance, always playing Minecraft in the same world with the same characters would still be interesting but it would set aside the great pleasure of exploration [1]. In the case of Dwarf Fortress for instance it would undermine the very premise of the game, which is to manage a colony of dwarves that the players is not supposed to know at first [10].

The most common generated parts of emergent games directly concern the structure of the world: how the different parts come together to form a coherent and interesting world [11]. This goes hand in hand with the generation of the elements populating these worlds such as the characters, wildlife and items. Procedural content generation allows the designer to make sure that no players will be able to know the world before hand. This ensure that the exploration is a mandatory and central aspect of gameplay.

In spite of the very innovative narrative approach shown by emergent games, they are still limited to situations that could be considered too simple. For instance, among the very successful emergent games cited in this article most of them rely on combat and survival and lack more complex, and more human, narratives [1], [2], [4], [10]. Relying too much on players to build stories is a risky operation. Few players can actually make use of the system to create rich and meaningful stories. While codirection is an important property of emergent games, it could still be possible to provide players with more meaningful stories, partially authored, without taking off too much of their ability to codirect the story.

The field of interactive narrative has worked at length on ways to enrich games with meaningful interactive stories. They have developed models to help build interactive stories and made many theoretical advancements regarding interactive storytelling [12], [8], [13].

B. Interactive Storytelling

Mixing narrative content and interactivity is a strong dilemma [14]. To have the player experience specific feelings, like for instance, the pain of losing a child, we need to virtually put her in such a situation [15]. However, crafting this moment is only possible by unfolding a specific course of events. The player must play the role of a parent, while there must be a child that she is attached to, and then this child may eventually die. Of course, in an interactive setting, the player may strive to save the children but to fulfill our narrative goals, she may never be able to save him or her.

The most simple way to create such a narrative experience is to use pre-scripted events, like most of the story based games do. The player will play the role of a parent, but at a specific moment, the children will die and the player will not be able to do anything about it. But while the use of scripted events will make sure that the narrative is unfolding, it also strongly limits the interactive part of the experience. Did we really need to take control of the simulation and force this event to happen to create a strong narrative experience, or is it possible to have a much more dynamic system that maintains the narrative value of the experiment while giving much more freedom to the player? The interactive storytelling research field looks for answers to this question.

Since the creation of the pioneer interactive narrative game Facade, drama managers have been one of the most promising ways to add a dynamic narrative dimension to an interactive experience [16], [17]. The concept of drama management can be used to adapt a narrative to a player, making sure that the story stays consistent and interesting [18], [19], [20], [21], [22]. Indeed, in an interactive narrative, the player can interact with the game world at anytime and the consequences of her actions modify the current story. The drama manager is responsible for taking these actions into account while still reaching predefined narrative goals, e.g. making specific narrative events happen or respect a certain aesthetic evaluation of the current story [23], [24], [19], [25], [20], [26].

Research in interactive narratives offers two general approaches for introducing more narrative content while taking into account the players' actions, the top-down and the bottom-up strategies [27]:

1) Bottom-Up Approach: The bottom-up approach implies that the story is emergent and results from the different interactions between the autonomous agents and the player [28], [29], [30]. No external narrative logic comes between the player and the story. In this approach the narrative is driven by players and can generate a great number of possible stories while ensuring that players have high agency regarding the sequence of events. However, since the authors have limited control, the potential for unsatisfying story is important. It is worth noting that one of the experiments in the bottom-up approach, FearNot!, was later improved with a double appraisal feature similar to a decentralized drama manager [31].

2) Top-Down Approach: In the top-down approach to interactive narratives, a *drama manager* is used to ensure that the stories created respect certain authored conditions [23], [24], [19], [25]. The drama manager is thus responsible for altering the environment and the characters in real time. Decentralized versions of the drama manager have also been developed and allow each character to make decisions regarding the unfolding of the story [25]. Therefore, these characters are not only characters of the story but also directors. The top-down approach tends to favor control from the authors over the sequence of events. However, it limits the number of possible stories and reduces player agency.

Both approaches have the potential to enrich the narrative capabilities of emergent games if used accordingly. The bottom-up approach, by providing more complex characters would be beneficial for developing stronger narratives and more believable game worlds. The top-down approach would allow to pursue more specific narrative goals, but at the expense of the player's agency.

However, as we explain in the next section, emergent games have properties that prevent us from directly using state of the art interactive storytelling systems.

C. Interactive Storytelling and Emergent Games

In the specific case of emergent games, and especially with procedurally generated content and user generated content like Minecraft or Dwarf Fortress, bottom-up or top-down approaches may not be directly usable, and this for two main reasons [1], [4], [10].

1) Abstract Game Worlds: First, using a drama manager requires an heavily formalized game world. To write a story, and moreover a story that can be processed and manipulated by some kind of planning engine, we need to define the world's objects, characters, places and all the actions that the planner may or may not do on this world. For instance, Cavazza et al's HTN based system has been demonstrated with characters from the Friends series or from the Madame

Bovary's novel [28], [32]. Facade system tells the story of Trip and Grace relationship's issues, in a flat where every object has a specific meaning and can change the story if the player chooses to interact with it [12]. Mimesis is demonstrated with a Bank Robbery story in a world with Fred, that can take a gun from an armory, to maybe kill Barney and rob the bank [33].

However, when dealing with emergent games based on procedurally generated content or user generated content, the endless amount of possibilities comes at the expense of the accuracy of the world's description. For instance, in Minecraft, almost everything that we know is that the player must eat, can craft objects, gather resources, and move blocks. The only available NPCs have no complex goals: some of them may kill the player, some others may be killed to provide resources and that is it.

Of course, worlds such as Minecraft's are not that empty or players would have stopped playing it a long time ago. The game's narrative content is just not predefined. First and foremost, Minecraft is a game where players create or adopt places and give them the meaning they decide. The first thing that the player is pushed to do by the system is to find a way to survive, as dying means starting all over again. To do so, the easiest way is to create a home, that is, a place where the player may store food and hide behind walls. As soon as her *home* is totally lit with crafted torches, no dangerous NPC may spawn inside and she may steer clear of dying every night. Thus, in many Minecraft games, the player has some kind of home, and a drama manager may use this information to craft a story: send a new NPC to create a neighbor? Place a dangerous NPC just behind the walls to increase the dramatic tension? Anyway, the drama manager must know that the player is in a safe place where she spends a lot of time, and use this information to define new interesting narrative events. However, just allowing the drama manager to read the game's state will not provide it with enough information: just the position and type of a collection of cubes and mindless NPCs.

Emergent games featuring procedural content generation and user generated content represent extreme cases of worlds that lack predefined content. But this lack of information can be found in more regular games. For instance, in a game such as Skyrim, the player might use certain places as a home even though it was never intended for[34].

The interpretation engine that we present in the remaining of this article represents the layer of interpretation between the raw information provided by Minecraft and the drama manager that will transform the world and the story in real time. This layer takes for input the numerous data retrieved from Minecraft and outputs structured and interpreted information for our drama manager to use directly. It serves both as an analyzer and filter, making the world of Minecraft more than a matrix of cubes: a world with places and characters.

2) Emergent Games' Aesthetics: Using drama management techniques always comes at the expense of the player's freedom and agency. As soon as we shape a part of the game's content to pursue an aesthetic goal, we make a choice that the player will not be able to change anymore. If we decide to kill a NPC, then the player will not be able to interact with it anymore.

As we detail in a previous work, emergent games have specific fundamental properties, most of them centered around the player's freedom and agency. Using a drama manager implies a trade-off between the amount of narrative content and these fundamental properties. We will discuss this trade-off at the end of the paper, and propose a way to alleviate this problem, especially in the kind of games that we are interested in.

In emergent games featuring procedurally generated content, the player does not only suspend her disbelief, but creates a big part of the story by herself. We named this property of emergent games *codirection*. We think that as the player is already assuming both the role of player and a part of the role of narrator, she may directly interact with the drama manager to customize the game's narrative.

Indeed, a drama manager can also be considered as a means for players to change the story and the world at run time. The various modifications would still be described by the inner mechanisms of the drama manager but it would not act in a totally autonomous fashion anymore.

We believe that a deconstructed form of the drama manager could be used by players to alter the story in a structured way in real time. Indeed, it appears to us that players are fully capable of assuming bigger responsibilities regarding the game and the story being created in real time, if given the right tools. A perfect example would be the various narratives that children develop while playing [35]. By using existing materials to create worlds and characters, children improvise narratives as they play.

Enriching emergent games with more complex narrative content can thus take advantage of players' ability to create the story as they play. This led us to take the core concepts of drama management while giving more control to the player. We thus developed a system of *Narrative Processes* that can be used by players, in real time, to modify the story in meaningful ways.

In the remainder of this article, we present the architecture we developed to enrich emergent games based on procedurally generated worlds with interactive storytelling techniques. Then we discuss the advantages and drawbacks of our system in regards to the fundamental properties of emergent games.

II. Expanding Emergent Stories Through Narrative Processes

In this section we describe the architecture that aims to enrich the narrative possibilities of emergent games. It was built using a popular emergent game based on procedural content generation: Minecraft. While the architecture was built and thought to be usable with any emergent games, Minecraft was an ideal choice for its popularity and modding capabilities.

A. Processes

As we explained in the previous section, any kind of automated system that modifies the game world in real time will have a negative impact on codirection. Since we consider that codirection is a fundamental aspect of emergent games, we need to limit the use of automation. Thus, we designed our system so that it gives players more responsibilities toward

the story. To this end we developed the concept of narrative processes, that we define as follows:

> ***Narrative Process****: a narrative process is a set of instructions that modifies the game world, and thus the story, in real time. When a process is activated, the engine will execute its instructions according to an event based system. A process must be triggered by the player in order to be considered by the engine, and will keep running until the player decides it should stop.*

By providing players with many different narrative processes we allow them to modify the story in many different ways. Moreover, triggering a narrative process becomes a conscious act that players make to modify the current story. This choice is a very high level choice and may lead to some unpredictable results. However, we postulate that making this choice will give the player a sufficient amount of control to maintain the codirection aspect of emergent games while still providing her with more narrative possibilities.

A narrative process is a script written in Lua that runs in the background as soon as the player selects it. Anyone can write new processes and each script has access to our engine's API, allowing it to 1) register new functions for specific events 2) modify the current state of the world 3) query the world's story, that is, both the world's current state and the story log.

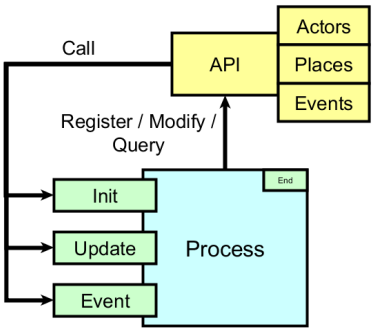

Fig. 1. Narrative Process

1) Registering custom events: Processes can register custom functions to be called when certain events happen. Since the API records many events that happen in the game world, functions can be associated with them. For instance, it is possible to register a function that will be called automatically when the player enters a new place. The default and most basic events redefined by processes are the *Init* function, called when the process is launched, and the *Update* function, called at each main loop of the game engine.

2) Modification of the current state of the world: We consider that the act of modifying the story should only be undertook by one or more characters in order to strengthen the believability of the world. As a result, processes can only modify the game world using characters' actions. Hypothetically, it would be absolutely justified to allow modification of the story through a character that has no presence in the world, such as a god but it would still be a character in the story anyway.

a) Character selection: The characters responsible for carrying the modifications can be chosen among the ones that already exist in the story or be created from scratch. Selecting an existing character can be a good way to keep the game world and the story as coherent as possible. Indeed, rather than creating new characters the system can find the best fitted one. The selection is based on a list of requirements defined by the creator of the process. A requirement is a condition that the character to be found has to comply with.

A valid condition for selecting a character is one, or a combination, of the elements of the list of information regarding characters that are tracked and recorded by the engine:

- A character already went to a certain place
- A character already used a certain item
- A character already met another character

For instance, a requirement could be that the character to be chosen cannot have met the player already or that she should possess a certain item. In the case no character that fits the requirements of the selected process can be found, a character can be created entirely instead. The more the story advances the more it will contain characters. New characters created by processes will become unusual.

b) Modifications: In order to bring more complex narrative situations, processes can use the various characters they created or selected to alter the environment and to reconfigure the potential interactions between characters. Similar to the character-based interactive storytelling experiments of Cavazza et al, the simple displacement of objects and characters alone can generate typical narrative situations [28]. By creating new characters with specific goals, processes can prevent other characters, or even the player, from achieving their own goals. They can also modify the environment in concrete ways by ordering characters to build, dig or destroy anything. Similarly, characters can be commended to interact with items and other characters.

The following list shows to what extent a character can interact with its environment. A character can:

- Add a block in the world, which removes this particular block from her inventory
- Remove a block from the world, which adds this particular block to her inventory
- Pick up items and blocks laying around the ground, which adds them to her inventory
- Put an item down, which removes this particular item from her inventory
- Put an item in a chest or in another character, which removes this particular item from her inventory
- Attack another character with or without an item
- Push and move another character

3) Story queries: In order to create meaningful narratives, processes can take advantage of the information about the story provided by the API. Indeed, a process needs to have access to the current state of the game world but also to what happened in the past, that is, the game world's story. Processes can thus have access to a log of the meaningful events that happened in the game world.

The story log is organized as follows: it is a chronological list of events, each event defining a relationship between actors, game objects and places. Using this set of events, a process can adapt to the current context of the story by acting not only according to the current state of the world but also to what happened in the past.

At first glance, it seems obvious that writing interesting narrative processes using the raw log of events might be complicated, this log being a very low level description of what happened in the game world. In the next section, we introduce the *Interpretation Engine*, that allows a higher level of interaction between processes and the story log.

III. Interpretation Engine

Processes aim to enrich the narratives of emergent games with new content and various modifications of the story. Enriching the current story implies that the process needs to know what happened in the story before it was triggered. Indeed, if the process was written to help or bother the player then it needs to be able to guess what the player likes or dislikes, where she spends most of her time and so on. Being able to know the current story also allows to act more coherently with the current context of the game world. A process that knows the various characters the player has met and interacted with might use them to act so as to reinforce the impression of a living and coherent world.

As we discussed in a previous section, emergent games and generated content make for unpredictable worlds that cannot be directly translated into a meaningful story: the events are often low level and very generic. For instance, Minecraft only provides us with information about collisions and other kinds of basic interactions between game objects. Therefore, we need to interpret the raw data coming from the game so that it can be used by processes to modify the story.

Our system records a series of basic events happening in the game. We call this raw log of events the Objective Story. Then, we call our interpretation of this event log the Subjective Story: a possible version of the story that bears more insights.

The overall architecture revolves around this interpretation engine whose role is to retrieve low level information from the game and transform it into a form that bears more meaning. The processes can then have access to this high level information and use it to plan their next action.

A. Objective Story

The objective story basically represents the state of the Minecraft world at different point in time. A collection of low level information about collisions, positions and actions undertaken by the various characters of the world. Nothing is interpreted, hidden or manipulated. This data is directly used by the interpretation engine to try and get a bigger picture of the story.

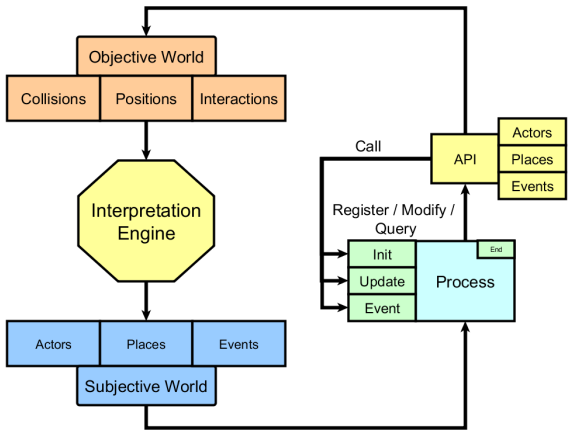

Fig. 2. The Interpretation Engine

The first step towards a more formalized world is to structure this information for allowing interpretation. This can provide a common language for describing events of the story and to search for new events, aggregate these events and draw statistics from them. It can help us, for instance, to retrieve the time and position at which a character used a specific object.

Therefore, each recorded event is made of one or two characters doing an action at a specific time and position that possibly involves an item.

Event structure:

character1 - [character2] - action - [item] - position - time stamp

This simple formalism allows the representation of every event of the story in a chronological order. The time stamp ensures that the time of the events is recorded along and that events are situated in time. Each event is recorded in a database so that it is possible to keep track of everything that happens in the game world.

The events recorded by the objective story are not those of the whole world but only of the world discovered by the player. When a player meets a new character or picks up a new item this will be recorded and the new character and item will also be tracked from this point forward. Through the actions of the player the story world gets bigger and contains more and more game objects. Every time a tracked object is modified, the modification is then recorded.

The objective story takes the form of a table in a database whose records represent a chronological list of alterations of the game world:

- A character picks up an item
- A character puts an item down
- A character sees another character
- A character pushes another character
- A character trades with another character
- The player position (every two seconds)

B. Subjective Story

The subjective story represents the world as it was interpreted by the engine. It is one version of many possible others that depends on the criteria used for interpretation. This is the material with which processes will create new events and modify the current story.

Unlike the objective story, the subjective story is made of high level information regarding the world such as the characters and their relations, the places in which events take place and so on. On the one hand, the objective story records events as fact. But on the other hand, the subjective story uses these recordings to label the events and to aggregate them together to provide a better understanding of the story. For instance, if a character attacks another the system should interpret this data as a bad relationship between the characters. Meaningful information like this allows processes to build upon the current story. They can be written to make decisions depending on the information coming from the interpretation engine, such as the presence of a character in a certain place for instance.

C. Interpretation

While recording the changes as they happen is pretty straightforward, the retrieval of a story from often unrelated events requires a deep understanding of the data. A few minutes of play can generate hundreds of new recordings, especially if the story has already started for some time and has multiple characters, places and objects involved. This is where interpretation steps in to translate the objective story into a possible subjective story.

Interpretation can happen inside the engine as a built-in way of labeling events but also through processes since they can make their own interpretation. When a process uses statistics or low level information about a character for instance, it actually makes hypothesis that might sometimes turn out to be true or false.

The main goal of the Interpretation Engine is to provide as much built-in interpretation as possible while also allowing parameterization of the interpretation. For instance, a process creator can use the interpretation engine to know what happened to the player two to three hours ago rather than from the beginning of the story. It could also ask for the monitoring of only a subset of the characters of the story.

Different processes require different levels of details about the story. Some processes might require very precise information such as the time, place and characters involved in a specific event or even the state of the relationship between two characters. Though, most story processes will only need partial information about the story regarding, for instance, the number of characters that went in a particular place.

For sense to emerge from context-poor recordings of in-game events, we need to introduce various data mining methods to create a meaningful abstraction of this data. These can help form the shape of the story made of the places in which it takes place as well as the emerging character relationships.

In the next section we describe a first step towards this goal: a method that allows us to retrieve the possible places of the story through the analysis of the player behavior.

The Places of the Story: The various places involved in the story are not predefined since the world is entirely generated and the players are free to do what they desire. As such, the places emerge from the behavior of players during play and continuously change over long periods of play. This concerns not just Minecraft but any emergent games that use procedural content generation or user generated content.

As players build their house, work on other buildings, go hunting or fetching resources they demarcate new places. Players quickly rationalize the space into different places with different purposes. Some are for storing resources, sleeping or crafting and others are for leisure or hunting.

Enriching an ongoing story without being able to identify the places in which it happens would be nearly impossible. This is why we decided to feed a clustering algorithm with all the positions of the player taken at regular intervals. By grouping the closest positions together, the algorithm tends toward an optimum size and number of clusters for the entire area represented by all the positions of the player.

By using clustering it becomes possible to identify the main places of the story and help the processes act in more meaningful ways. Indeed, knowing the layout of the world is important but even more so for allowing variations. Giving an area rather than a position for moving characters for instance is more coherent with our emergent approach.

As shown in Fig 3 the set of positions recorded over a period of time can be grouped together in an optimal fashion through the use of the expectation maximization clustering algorithm provided by the Java Machine Learning library [36]. On Fig 3, we in fact see the specific areas in which the player spent an important amount of time. For instance, the green area is the home of the player and the purple one is an area for fetching stones.

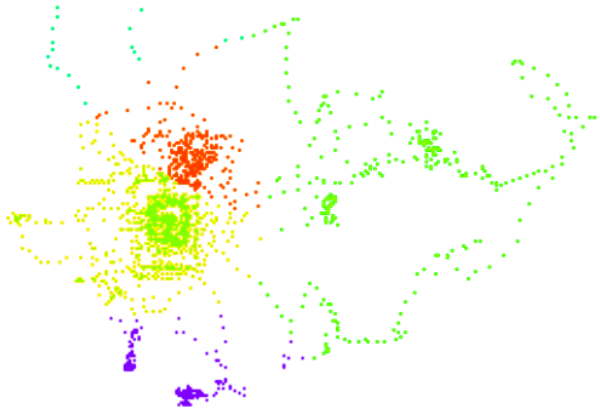

Fig. 3. Top view of the clusters of the recorded positions of the player. EM algorithm from Java Machine Learning library (min σ: 10, 100 iterations)

IV. Impact of the Engine in Regards to the Emergent Games' Aesthetics

We consider that using processes to modify the story in real time can enrich the narratives of emergent games with more authored content and increase the possibilities offered to players for modifying the story. However, our system may

also have negative side effects on the fundamental properties of emergent game narratives.

In the following section we consider the changes that our system bears on the fundamental characteristics of emergent games.

A. Coherence

Emergent games are simulations of alternate realities. They allow many situations to emerge without imposing a specific story. Each entity in the world behaves according to consistent rules which tend to allow emergent game worlds to stay coherent and to appear more believable.

The subjective story that processes will use to plan their next actions is the result of an interpretation process. This process is not perfect and relies on a certain number of assumptions, and thus may not correspond to what the player is really doing. For instance, we consider that players will spend time into specific places. But what if the player always goes straight toward the rising sun, as the experiment of Kurt J. Mac shows [6]? In this case, any process relying on places may not act in a fashion that is coherent to the player's behavior. The characters' behaviors may seem foolish and not related to the current story.

Our approach does not cover every possible player behavior and this is mainly due to our extensive use of the concept of place. This is especially true in the case of nomadic players since the places in which the story happens are always changing. Thus, we can see that our architecture aims to enrich the possible narratives of players that have a tendency to settle.

B. Agency

Agency represents the experience of a player having an impact in a dynamic, responsive world [37]. Emergent games often allow players to have a great amount of impact on the world and the sequence of events. Indeed, strongly coherent worlds allow players to fully understand the inner mechanisms of the virtual worlds. The more they play the easier it becomes for the players to anticipate and plan their actions.

Our system aims to provide players with more ways to affect the sequence of events through the use of narrative processes. But rather than directly increasing agency it instead provides more content that enriches the possible stories. Thus increasing the size of the possibility space as well as the number of potential situations of player agency.

C. Possibility Space

A large space of possibilities implies that the game can be replayed many times while still providing new worthwhile situations and choices, therefore preventing players from finding themselves in frustrating positions. They can always explore the game for more and more distinct experiences. Emergent games and more specifically user generated content based games actually provide a great number of play states that can be attained through play. This is made possible by the various interrelated subsystems that simulate complex and persistent worlds in which many situations can emerge but also by the new content that players are capable of bringing to the game.

The processes provided by our system have the potential to create many new ramifications from the main possibility space. By adding new content processes generate new possibility spaces around the main one and increase the pleasure of discovery and surprise that emergent games can offer.

Another important aspect of a large possibility space is that it also allows for dramatic actions that greatly impact the course of the game [9]. The ability of processes to create new situations that can deeply modify the story also allows for dramatic events to happen. For instance, a process could damage the house of the player but another could also allow the player to rebuild her house quickly. This could lead the player towards more risky situations that might turn out to be great stories.

D. Uncertainty

Situations and events emerge from the complexity of the various systems present in emergent games. They are often hard to predict since these situations were never predefined. As a result a critical element of surprise can be created and can increase the experience and enjoyment of emergent games.

Narrative Processes are systems that adds to the complexity of emergent games. From a high level point of view they guide the overall story and reduces uncertainty. But, each process reacts differently to the context of the story, they adapt, leading players back into uncertainty. Thus, narrative processes tend to reduce uncertainty regarding the main story arc while increasing uncertainty regarding the details of the story.

For instance, a process responsible of an opponent to the player might act differently whether the player has many friendly characters around her home or not. It could turn one of them against the player without her being wary of anything. The abilities of processes to act differently depending on the context of the story and the behavior of the player ensures that uncertainty is respected.

Since multiple processes can run in parallel, unpredictable situations can also emerge from the interactions between them. A process responsible for opposing the player running at the same time of another helping the player could actually cancel each other out or creates a tension that would be more difficult to obtain through the use of only one process.

E. Codirection

Emergent games allow players to be more responsible for the story and for the direction it takes. Rather than imposing mandatory events, authors build constrained spaces in which player can create new situations and events that were not predefined. Both authors and players are responsible for the final experience, they create a unique story in an asynchronous way.

By providing players with various narrative processes, authors give players more responsibilities toward the story since they can act on it in an out of character fashion. Each process represents a new way for players to intervene on the story through the tools written by the authors. Where a drama manager would take full responsibility of the story and use the player's behavior to make decisions, our system instead assumes that players can directly use the capabilities of a

deconstructed drama manager to take themselves the decision to lead the story in certain directions.

V. Conclusion and Future Works

In this article we presented our approach that aims to improve the narratives of emergent games through the use of interactive storytelling techniques. To do so, we propose an architecture that relies on Narrative Processes and an Interpretation Engine. The Narrative Processes empower players with the ability to modify the story in real time. Our goal is that these processes extend the narrative possibilities of emergent games without breaking the core fundamental properties that define them. As we detail in the previous section the processes require a thorough understanding of the story being played to act in meaningful ways. This is where the Interpretation Engine intervene to make sense of an emergent story.

In order to assess the true possibilities uncovered by our architecture we need to evaluate each property through a panel of players. This will allow us to get measurable effect of the processes on the enjoyment of emergent games.

Before setting up the experiment we believe that the Interpretation Engine can be improved through the use of a clustering algorithm for other dimensions than only the positions of the player. We could, indeed, use it to gain insights on other areas than the places of the story. Using clustering on player-character interactions for instance we could discover more about the player and her interactions with the other characters.

References

[1] Mojang, "Minecraft," 2009.

[2] Re-Logic, "Terraria," 2011.

[3] Maxis, "Spore," 2008.

[4] K. Entertainment, "Don't starve," 2013.

[5] B. Keogh, "When game over means game over: using permanent death to craft living stories in minecraft," in *9th Australasian Conference on Interactive Entertainment: Matters of Life and Death*. Melbourne, Australia: ACM, 2013, pp. 20:1—-20:6.

[6] S. Parkin, "A journey to the end of the world (of minecraft)," 2014.

[7] P. Sweetser and J. Wiles, "Scripting versus emergence: Issues for game developers and players in game environment design," *International Journal of Intelligent Games and Simulation*, vol. 4, no. 1, pp. 1—-9, 2005.

[8] S. Louchart and R. Aylett, "The emergent narrative theoretical investigation," in *Narrative and Interactive Learning Environments*, 2004.

[9] S. Chauvin, G. Levieux, J.-Y. Donnart, and S. Natkin, "An out-of-character approach to emergent game narratives," in *Foundations of Digital Games 2014*, 2014, p. 4.

[10] T. Adams, "Dwarf fortress," 2006.

[11] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, "Procedural content generation for games: A survey," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. February, pp. 1:1—-1:22, 2013.

[12] M. Mateas, "A neo-aristotelian theory of interactive drama," *Working notes of the AI and Interactive Entertainment*, 2000.

[13] E. Aarseth, "A narrative theory of games," in *Proceedings of the International Conference on the Foundations of Digital Games*, no. c. New York, New York, USA: ACM, 2012, pp. 129–133. [Online]. Available: http://doi.acm.org/10.1145/2282338.2282365

[14] G. Frasca, "Ludology meets narratology: Similitude and differences between (video) games and narrative," *Parnasso*, pp. 365–71, 1999.

[15] Q. Dream, "Heavy rain," 2010.

[16] M. Arinbjarnar and H. Barber, "A critical review of interactive drama systems," *AIS'B09 Symposium: AI & Games*, 2009.

[17] M. Mateas and A. Stern, "Build it to understand it: Ludology meets narratology in game design space," in *DiGRA Conference, Vancouver, BC*, vol. 2, 2005.

[18] M. T. Kelso, P. Weyhrauch, and J. Bates, "Dramatic presence," 1993.

[19] N. Szilas, J. Barles, and M. Kavakli, "An implementation of real-time 3d interactive drama," *Computers in Entertainment*, vol. 5, no. 1, pp. 1–18, 2007.

[20] M. Mateas and A. Stern, "Integrating plot , character and natural language processing in the interactive drama façade," in *In Proceeding of the Technologies for Interactive Digital Storytelling and Entertainment*, vol. 2, 2003.

[21] R. M. Young, "An overview of the mimesis architecture: Integrating intelligent narrative control into an existing gaming environment," in *The Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, 2001, pp. 78–81.

[22] G. Delmas, R. Champagnat, and M. Augeraud, "From tabletop rpg to interactive storytelling: definition of a story manager for videogames," in *Proceedings of the 2nd Joint International Conference on Interactive Digital Storytelling: Interactive Storytelling*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 121–126.

[23] B. Magerko and L. John, "Building an interactive drama architecture," in *International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, Darmstadt, Germany, 2003.

[24] M. Mateas and A. Stern, "Structuring content in the facade interactive drama architecture," in *American Association for Artificial Intelligence*, vol. 3, 2005.

[25] R. Paul, D. Charles, and M. McNeill, "Mist: an interactive storytelling system with variable character behavior," *ICIDS*, pp. 4–15, 2010.

[26] B. Mallon, "Towards a taxonomy of perceived agency in narrative gameplay," *Computers in Entertainment*, vol. 5, no. 4, p. 1, Mar. 2008.

[27] S. Louchart, R. Aylett, M. Kriegel, J. Dias, and R. Figueiredo, "Authoring emergent narrative-based games," *Journal of Game Development*, vol. 3, no. 1, 2007.

[28] M. Cavazza, F. Charles, and S. J. Mead, "Emergent situations in interactive storytelling," in *ACM symposium on Applied computing*. ACM, 2002, pp. 1080–1085.

[29] R. Aylett, S. Louchart, J. Dias, A. Paiva, and M. Vala, "Fearnot - an experiment in emergent narrative," in *Intelligent Virtual Agents*. Springer, 2005, pp. 305–316.

[30] 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," *4th International Conference & Festival of the Electronic Literature Organization: Archive & Innovate*, 2010.

[31] R. Aylett, S. Louchart, A. Tychsen, M. Hitchens, R. Figueiredo, and C. D. Mata, "Managing emergent character-based narrative," in *2nd international conference on INtelligent TEchnologies for interactive enterTAINment*, 2008.

[32] M. Cavazza, J.-L. Lugrin, D. Pizzi, and F. Charles, in *Proceedings of the 15th International Conference on Multimedia*. ACM, 2007, pp. 651—-660.

[33] M. Riedl, C. J. Saretto, and R. M. Young, "Managing interaction between users and agents in a multi-agent storytelling environment," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*. New York, New York, USA: ACM, 2003, pp. 741–748.

[34] B. G. Studios, "Skyrim," 2011.

[35] R. K. Sawyer, "Improvisation and narrative," *Narrative Inquiry*, vol. 12, no. 2, pp. 319–349, 2002.

[36] T. Abeel, Y. V. D. Peer, and Y. Saeys, "Java-ml : A machine learning library," *Journal of Machine Learning Research*, vol. 10, pp. 931–934, 2009.

[37] J. H. Murray, *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. New York, NY, USA: The Free Press, 1997.