A System for Simulation-based, Emergent, Interactive Narrative Generation

INTERIM REPORT
(16/11/2018)

Oliver Mitchell
University of Nottingham
psyom1@nottingham.ac.uk

Peter Blanchfield
University of Nottingham
peter.blanchfield@nottingham.ac.uk

ABSTRACT

Recent research in the field of interactive storytelling (IS) has focused on the advantages of "emergent narratives" (ENs). With this approach, the storyline 'emerges' at run-time from the interactions between author-programmed autonomous agents and the player. The benefit is a non-linear narrative, allowing for greater player interactivity. However, ENs are unpredictable at authoring time, removing control over the plot's structure. The struggle to balance the author's control over the narrative, whilst simulatenously allowing the player to influence the story's direction, has been dubbed the "Narrative Paradox". We discuss the design and implementation of a new hybrid emergent storytelling system for creating unscripted narratives which aims to integrate the successful components of existing approaches, including an appraisal-driven agent architecture extended to model autobiographical memory, and a "virtual director" which maintains storyworld knowledge and uses a novel metaheuristic genetic algorithm to generate suitable global events for agents to appraise and respond to. An evaluation of the system's effectiveness (based on comparative user-testing) is discussed with a particular regard to player 'immersion' and the believability of the generated narrative.

Keywords

interactive storytelling; games; autonomous virtual agents; emergent narratives

1. BACKGROUND

1.1 Defining Interactive Storytelling

Interactive storytelling is a unique form of narrative design seen in digital entertainment mediums where no predetermined storyline exists. Instead, the user can change the story and influence the characters themselves. Bostan and Marsh define interactive storytelling as an approach to computer game narrative design where "real-time feedback collected from the players is used by the game engine to continuously modify the content as it is being delivered." [1]. The key aspect of interactive storytelling is that the player is an active participant and their in-game actions have a direct effect on the narrative that is told. This differs from traditional computer game narratives which typically consist of an unchanging linear story with a pre-defined outcome.

1.2 Problems with the Integration of Interactivity into Game Narratives

In most computer games, attempts at complex, dynamic storylines are usually "confined to simple branching tree structures" [3]. Furthermore, as the scope of games increases and development effort is focused on assets such as textures, 3D-modelling and other gameplay content, a greater number of narrative possibilities arise. However, the "static nature of these [tree] structures affects the range of options covered [and] reduces the framework for player interaction". Louchart et al. suggest "the integration of interactivity into game narratives requires a rethinking of the design process" [3]. This is because story trees have to expand exponentially with the inclusion of more narrative elements (such as additional characters added to the game) or risk remaining so simple and rigid that the player's immersion in the game world is threatened.

In short, with a greater amount of gameplay content comes infeasible exponential growth in the number of branches of a narrative tree structure. When there are too many branches, developers are forced to simplify the story, resulting in less player interactivity.

1.3 The Narrative Paradox

It is common for narrative design in computer games to follow a plot-based approach. Louchart and Aylett recognised that the "authoring of interactive narrative presents the paradox that ... the author requires control over the unfolding narrative whilst ... the user expects freedom over their decisions." [3][4]. These two goals clearly conflict. The greater the level of interactivity provided to the player, the less control the author has over the story's events. Equally, the more control the author has over the story, the less room there is for the player to interact with characters and change the course of the narrative. This concept has been dubbed the "Narrative Paradox" [5]. Louchart and Aylett suggest that a solution to this paradox should divert from the branching plot-based approaches implemented in current computer games.

© 2018 ACM. ISBN xxx-xxxx-xx-xxx/xx/xx.
DOI: xx.xxx/xxx_x

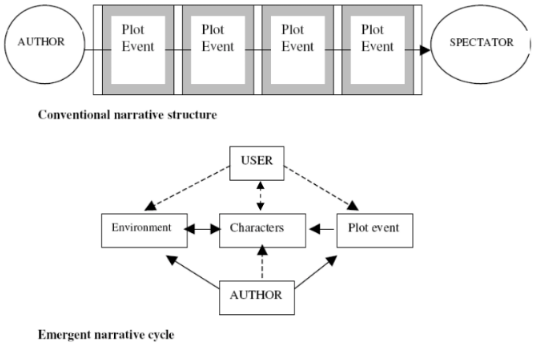


Figure 1: Diagram showing the difference in structure between convential plot-based narratives and interactive emergent narratives. The former is static and sequential, whereas the latter is simulation-based and reacts to player interaction. [3]

1.4 Emergent Narrative

Aylett proposed Emergent Narratives (EN) as an alternative. ENs are an approach to interactive story design where, instead of a static authored plot, the "narrative weight of the application is shared by authors and players." [3][5][6]. This is typically achieved by simulating a virtual world, including characters with different behaviours, personalities and goals created by the author. The storyline then 'emerges' at run-time from the interactions between these autonomous agents and the player [2]. The advantage of an EN approach is a non-linear narrative that allows for a greater level of player interactivity **(Figure 1)**. Despite this, ENs are unpredictable at authoring time. Kriegel and Aylett explain how authors are forced to "let go of specific story lines altogether and ... focus on creating the elements from which the story will emerge." [7]. ENs work as a sort of 'sandbox' where the author defines the limits of the story world and creates the characters and behaviours that fill it, then leaves the simulation to create its own story, providing no further input during run-time. As a result, authors of ENs will find it very difficult to ensure the story includes specific plot points.

ENs benefit from having no rigid structure to their story because there is no branching tree restricting the range of storylines. This solves the problem of infeasible exponential growth because there are no branches to manage and the storyline emerges on its own. This allows for a potentially inumerable amount of emergent storylines (should the simulation be complex enough).

Despite this, a disadvantage of emergent narratives are their reliance on well implemented virtual agents and their control logic. If the AI controlling each agent is too simple, then the story is unlikely to be complex enough to satisfy the player. Or, if the AI has too limited a set of possible responses to stimuli in its environment, the player may see the same storylines again and again, threatening the narrative's believability.

1.5 Motivation

The motivation for this project is to develop a new approach to interactive storytelling which utilises and combines components of successful existing emergent narrative systems in order to generate compelling stories and give players a greater sense of agency. It is also crucial to provide adequate control of the narrative to authors. This will be achieved through a suite of story creation tools which allow authors to define characters personalites, goals and actions in detail. This paper focuses on simulation-based approaches which exploit the usefulness of emergent narratives and omit the restrictions of plot-based designs.

2. RELATED WORK

2.1 History of Digital Storytelling

2.1.1 Early approaches

Since the 1970s, there have been many digital storytelling systems developed both in research and for commercial games. Early examples of story generation include *Novel Writer*, a system that generated 2000+ word murder mystery stories that emerged from the actions simulated by each character [8], and *TaleSpin*, a complex system that modelled rational behaviour for each character, including the production of personal goals that each agent could attempt to achieve autonomously through the creation of its own plan; a series of actions it may take [9]. *Virtual Storyteller* was another system that, despite a lack of interaction, created stories via the actions and cooperation of intelligent agents. In this system plots are not pre-defined but created by the actions of the agents alongside a "virtual director"; a seperate agent that maintains general story-world knowledge about the plot structure. It uses this to judge whether a character's intended actions fit into the narrative in order to produce a "good" plot; a believable and generally compelling story. Consequently, unlike the full-autonomy displayed in *TaleSpin*, the agents in *Virtual Storyteller* can only be considered semi-autonomous because they are consistently guided in their actions to create a well-structured plot [10]. Note that other systems explored in this paper utilise a similar "virtual director" type agent. However, those systems use such a device to manage and facilitate the creation of new storylines, **not** to perform global action control for individual agents, as seen in *Virtual Storyteller*.

Despite producing compelling emergent narratives, none of these systems can be considered examples of interactive storytelling because they do not allow a user to change the path of the story at run-time (*Novel Writer* and *TaleSpin* only accepted user input for the initial storyworld state and *Virtual Storyteller* allows for no user input whatsoever, relying solely on its encapsulated virtual director to direct the plot).

2.1.2 Towards Emergent Narratives

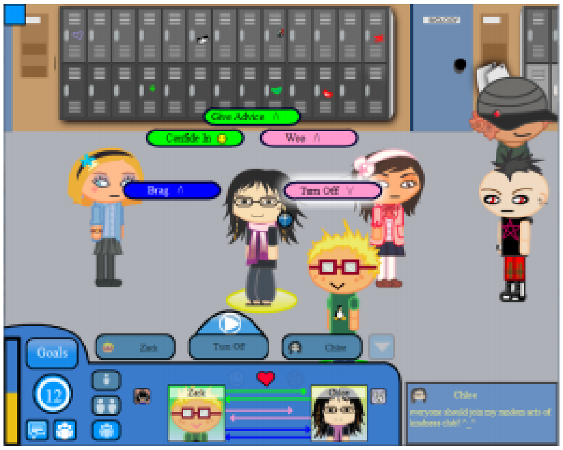
Following these initial approaches came a greater focus on emergent storytelling and a move towards non-branching, character-focused systems. Aylett first coined the term "emergent narrative" in 1999 when discussing the narrative issues arising from placing children's TV characters in a virtual environment where they and the user had a joint spacial existence [6]. The original television narrative was wholly pre-scripted and viewed passively by the audience from camera positions determined by the creators. This strategy is unsuitable for use in a virtual environment where the user is transformed from a spectator to a potential participant. Aylett argues that a pre-determined narrative means the role of the user must also be pre-determined. Therefore, relaxing this constraint allows the user to be more freely involved [6].

The solution is to build the narrative from the bottom-up through the interactions of the characters. Many groups have looked at producing emergent narratives by treating characters as virtual actors [23][24]. In such a scenario, the focus of the user is moved from a participant to a director, where they're able to construct narratives involving virtual actors and themselves. This has been compared to the playtime of young children where they switch repeatedly between particpant and director in order to control the direction of their roleplay [6].

2.2 Existing Emergent Narrative Systems

2.2.1 Automated Planning with HTNs

Cavazza et al. produced a prototype character-based system which focused on defining the possible actions of individual characters in hierarchical task networks (HTN) in order to create automated plans and achieve character desired goals [11]. Characters act autonomously, executing available actions in their HTN in order to reach their goal, similar to the *TaleSpin* system's character plans [9]. Each action is categorised and used to determine other characters' reactions (e.g. characters will react negatively to "rude" actions). The system also includes "mood values" for each character which can affect other characters' plans. Because the system allows for emergent behaviour, unexpected situations may arise that are unaccounted for in a character's plans. This creates the need for "situated reasoning" and "action repair". Cavazza et al. explain that situated reasoning "originates from the discrepancy between an agent's expectations and action preconditions" and is focused on the character aiming to achieve a specific result in a given situation or avoiding an undesirable result. Typically, this is seen when a character's current plan is interrupted and they must deal with a specific unanticipated situation. Action repair is focused on recovering from action failure, when external factors threaten the satisfaction of executability conditions. This system is primarily plan focused and authoring constitutes describing the sub-tasks in each character's hierarchical task network, and the character's reactions to "various generic situations, mostly arising from the conjunction of actions from the characters' respective plans" [11].



everyone should join my random acts of kindness club! ^_^

Figure 2: A screenshot of *Prom Week* showing the social interaction options available to the player. The HUD also shows the relationship between the player and the target player. [12]

2.2.2 Modelling Social Interaction: Prom Week

McCoy et al. produced *Prom Week* [12], a game described as a "social physics" simulator which utilises a bespoke AI system: *Comme il Faut* (CiF) [13]. The goal of the project was to design a game that could provide "satisfying stories that reflect player choices in a wide possibility space" [12]. As the title suggests, *Prom Week* revolves around the social lives of a group of teenagers the week before their prom night. The player is given a set of goals which they must complete within the week, for example: finding a date to attend the prom with. McCoy et al.'s goal was to decrease the volume of explicitly authored story space whilst increasing the amount explorable by the player. They reasoned the best approach was to build "a social artificial intelligence (AI) system that computationally models social space and social interaction", instead of having the author explicitly detail each possible interaction (a burdensome and effectively intractable task should the number of interactions be large enough). With their proposed AI system (CiF), the author provides "reusable and recombinable representations of social norms and social interactions" (or social rules) for autonomous agents in the model [13]. The social state existing across these agents is modified by multi-character social interactions, or social exchanges that utilise these representations. In effect, autonomous agents (the characters) perceive the global social state and attempt to alter it to accomplish their own social goals. This is achieved through social interactions, all of which follow the author-specified rules.

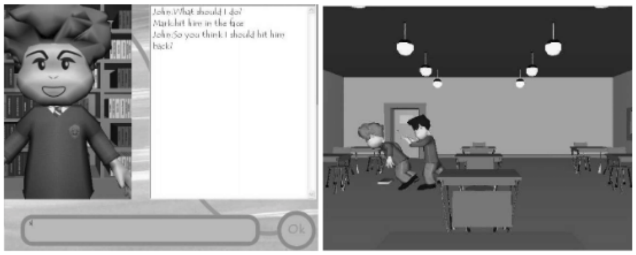
For example, if the player has the goal of attaining a date to attend the prom with, they may try to use the "flirt" social interaction on another character. This may, or may not, succeed and the social state across both characters will be altered according to the author-specified rules that dictate the social norm of that interaction **(Figure 2)**. The results of this interaction are stored in a database of social facts, acting as a sort of global memory. CiF also incorporates 'trigger rules' which "encode the cascading effects of social state change" outside of the current interaction [13]. Once an interaction has finished, these triggers are run over the social network in order to update the social state of each character.

The CiF system itself does not attempt to model the entire story world. It is a social reasoning component that aims to realistically model the rules and patterns that characters should follow during any social exchange. In contrast to Cavazza et al.'s hierarchical task network planning approach [11] (and other approaches such as behaviour trees), CiF encapsulates domain knowledge in one large rulebase that represents the social norms of a given story world. The system chooses each character's behaviour based on these rules. CiF doesn't create a static series of events, but the logic of the story world, the characters involved, and their goals. CiF is inherently driven by the simulation of social exchanges, therefore the way in which goals are met is entirely emergent and unplanned. This supports Aylett's initial description of emergent narratives as emerging from the interactions of characters, rather than being pre-planned [5][6].

An extension of the CiF system [14] was proposed by Guimaraes et al. that adapted the social physics simulator to a first-person perspective, implemented as a publicly downloadable modification in the *Elder Scrolls V: Skyrim* game engine. The results saw an extended interaction space for in-game NPCs leading to more believable characters and an improved player experience, as confirmed by the modification's popularity [15] and the results of user testing.

2.2.3 *Modelling Emotional Behaviour - FearNot! application and FAtiMA agent architecture*

Louchart et al. produced The *FearNot!* (Fun with Empathic Agents Reaching Novel Outcomes in Teaching!) storytelling system, designed for anti-bullying education and inspired by Forum Theatre [3]. *FearNot!* generates short episodes where a character is bullied and the victim asks the player for advice. The player's advice has an impact on the victim's emotional state which in turn influences the actions of that character in the next episode **(Figure 3)**. Each character is represented by an intelligent agent architecture which includes an "affective appraisal system and autonomous action selection capabilities" [3]. In addition to the character agents, an additional story facilitator agent is used for selecting characters, props, and locations for successive episodes. The story facilitator takes the player's actions and the characters' emotional state into consideration, generating an unscripted episode which is logically coherent to previously transpired events. For example, if the player "has advised the character to hit back, it may set up an episode where victim and bully confront each other directly" [3]. The story facilitator is comparable to *Virtual Storyteller*'s "virtual director" [10] which stores global story data and uses it to influence and adapt the narrative. However, *FearNot!*'s director does not perform global agent control and thus each agent remains fully autonomous. Rather than influencing characters directly, the director appraises past episodes and the current state of the story world and uses this data to generate logically cohesive, succesive content.



John:What should I do?
Mark:hit him in the face
John:So you think I should hit him back?

Figure 3: A screenshot of the *FearNot!* system showing a scenario where a child is being bullied and asks the player for advice. [3]

FearNot! takes a focus on character-based emergent narratives which Louchart et al. explain requires a "bottom-up approach in which story elements are synthesised in real time via character interaction" [3]. This contrasts branching plot-based narratives which are created top-down with the story being comprised of fixed plot elements. In *FearNot!'s* approach, characters are defined by their emotions, personalities, action tendencies, goals and emotional reactions. This is 'organic' because characters will only perform actions in-character, according to their own personality and desires, rather than requiring global action management. From the author's point of view, developing a character requires the static checking of character actions (e.g. making sure all actions have reachable preconditions) and the simulation of interaction in various contexts to confirm that the character acts in-line with the author's vision.

FearNot! uses an affective agent architecture dubbed *FAtiMA* (**F**earnot **A**ffec**TI**ve **M**ind **A**rchitecture) [19][20]. The architecture is designed to use author-specified emotions and personality to influence the behaviour of characters. It includes two essential components: a continuous planner (allowing characters to act intelligently and perform action repair and situated reasoning during unexpected events) and an emotional personality model (which effects the way characters react to events and provides a greater sense of believability). Appraisal is the assessment of any given event in relation to a character's emotions and influences their choice of actions towards their goals (world states they desire). *FAtiMA* allows for the description of appraisal rules which define the event to be appraised and the desirability of the event on the target character [3]. Using this data, the character's emotions are updated. For example, an event in which a character is bullied would have very low desirability and produce negative emotions for the victim character.

Emotions are a crucial aspect of the agent architecture and influence each character's planning process. Emotional values can be used as pre-conditions for potential actions. The use of emotional values (amongst other data such as memories and social relations) can be used by agents to perform rational processing to intelligently select actions that will help them towards their goal. However, *FAtiMA* also incorporates non-cognitive 'reactive' processing which involves "spontaneous reactions triggered by intense emotions [and is] not part of the planning process" [3]. With this feature, characters with extreme emotional states can react with appropriate actions; for example, a character with extremely negative emotions may spontanenously burst out crying. This is an attempt to allow characters to react entirely emotionally to certain situations, rather than cognitively, and adds an extra layer of believability.

FAtiMA's planning system requires actions to be defined with preconditions (such as ranges of emotional values or certain event preconditions) and effects (the outcome of the action). Characters use actions to react to stimuli (other character's actions) or to help them achieve their goals. Goals can be defined as active pursuits or interests. Active pursuit goals are defined as world states which a character strives to attain whereas interest goals are world states that the character wants to maintain. Interest goals can help characters choose certain actions over others because actions which modify desired world states are considered undesirable [3].

Due to *FAtiMA*'s flexibility and modularity, some extensions to the architecture have been proposed in order to provide greater functionality and believability. Dias et al. successfully extended FAtiMA, allowing agents to "explicitly talk about their internal thoughts and report their personal past experiences ... [using] a specific type of episodic long term memory" [21][22]. This is a promising idea because memory is an important part of the human decision making process and helps add more believability to characters.

2.3 Emergent Narrative in Commercial Games

Emergent narrative generation methods are not solely practiced in research; many commercially successful games have yielded interesting and unique methods for generating unplanned interactive storylines:

2.3.1 The Sims

The Sims [16] is possibly one of the most widely played and easily recognisable video games of the simulation genre and is a textbook example of emergent narrative creation. From the outset the player is provided with great freedom, being able to create their own character; namely their appearance, but later installments in the franchise allow for the selection of personality traits that help define unique characters and change the results of social interactions. This is interesting because it allows the player to be a partial director and their choices in the character creation stage will affect how their avatar acts and what stories emerge during the game. There is no rigid storyline to the game and no plot the player must follow. Instead, the user "plays god" and instructs their avatar to perform desired actions such as socialising with other characters or interacting with objects in the world. Typically, a narrative emerges from the results of the player's actions and the interactions of the player's avatar with other characters (e.g. neighbours or family members). For example, the player may choose to use the "flirt" social interaction with another character they meet in a bar who will respond accordingly. This could lead to an in-game relationship, marriage, and even starting a family, all of which involve a growing narrative. **(Figure 4)**.

The Sims includes the simulation of personalities, emotions, needs, and social traits, not dissimilar to *Prom Week*'s social AI system or the *FAtiMA* agent architecture's emotional personality model. Characters can also have immediate responses to emotional state, alike the *FearNot!* system's action tendencies. For example, if a character's mood is low enough they will ignore the player's actions in order to satisfy their own needs and improve their mood first.

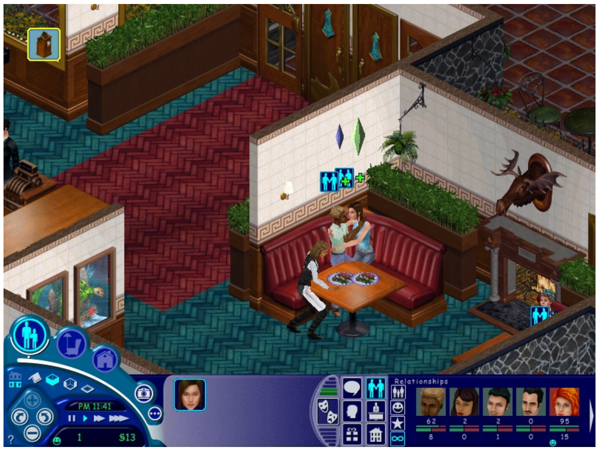


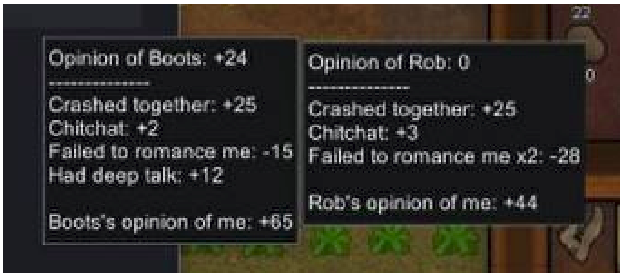
Figure 4: A screenshot of *The Sims* showing two characters bonding in a restaurant. In the bottom-right corner, relationships between characters are displayed graphically. [16]

Despite the high level of freedom given to the player, a serious limitation of *The Sims*' narrative generation is the fact that the quality of the story relies entirely on the player's choice of interesting actions. Should they desire, the player could allow their avatar to live in complete isolation and only watch TV all day, thus resulting in a lack of narrative entirely. There is no guarantee that engaging, high-quality narratives will be produced because the player acts autocratically as spectator and director.

2.3.2 RimWorld

RimWorld [17] is a "colony simulator" game that was inspired by the highly realistic simulation-based approach to game design pioneered by *Dwarf Fortress* [18], a sandbox game which sees the player manage the livelihood of a group of dwarves.

RimWorld tasks the player with managing a group of spaceship crash survivors as they construct a colony on their new home: an alien planet. The player must "manage colonists' moods, needs, individual wounds, and illnesses" [17]. The game is driven by an intelligent AI storyteller and its creator describes the game not as a strategy game, but as a story generator in its own right. At any given moment, the AI storyteller analyzes the colony's situation and decides which event will make the best story [17]. Accordingly, the AI storyteller can be considered *RimWorld*'s equivalent of *Virtual Storyteller*'s "virtual director" [10], or *FearNot!'s* story facilitator; a seperate agent that maintains story-world knowledge about the plot structure and uses it to adapt the narrative at run-time. As a result, *RimWorld* suffers less of the narrative limitations experienced in *The Sims* where the quality of the story relies wholly on how compelling a player's actions are. Because the AI storyteller acts as an agent itself, its goal can be considered the generation of a "good" story and its process of selecting suitable actions to change the narrative's course can be considered a form of



Opinion of Boots: +24

Crashed together: +25
Chitchat: +2
Failed to romance me: -15
Had deep talk: +12

Boots's opinion of me: +65

Opinion of Rob: 0

Crashed together: +25
Chitchat: +3
Failed to romance me x2: -28

Rob's opinion of me: +44

Figure 5: A screenshot of *RimWorld* showing a comparison of the relationships between two characters. Recent events and their related quantitative mood effects are shown. [17]

action repair, ensuring that player actions that divert from its goal are eventually remedied.

For example, players have the choice of upgrading the colony with weapons and defences to protect characters from periodic waves of attackers. The AI storyteller can evaluate the colony's level of defense and ensure that the next wave of attackers is strong enough to prove challenging to the player and therefore exciting enough to produce a "good" narrative.

RimWorld includes a sophisticated social simulation inspired by *The Sims.* Colonists have traits which help them react in unique ways to social interactions (e.g. "Night-Owl" colonists have a better mood if they're up at night and "Greedy" colonists have a mood loss if their bedrooms aren't impressive) and emotions which change according to various stimuli **(Figure 5)**. Similar to the *FearNot!* system [3], the state of these emotions determines which actions a character will take. Colonists can also reactively process emotions, identical to *FearNot!*'s action tendencies. This manifests as a "mental break" where extreme mood values cause colonists to disregard player instructions entirely and engage in fully autonomous, usually obstructive, behaviour. For example, if a colonist's mood is low enough they may engage in an uncontrollable psychotic episode of pyromancy, igniting the colony's buildings.

Colonists have memories and can remember the effects of actions for varied lengths of time. For example, a recently divorced colonist may suffer a mood reduction for several months. Memories can influence the decision making process, allowing disgruntled colonists to react to those that have wronged them. Such reactions depend on the character's personality; for example, a character may choose to violently attack their aggressor with a weapon or simply insult them.

3. CRITERIA FOR SUCCESSFUL SYSTEMS

By evaluating existing emergent narrative systems and their components it is possible to establish a set of criteria that successful future systems should follow. We propose the component specification shown in **(Table 1)**.

These are intended as a set of guidelines, describing what components an emergent narrative system should include, at a minimum, to successfully present users with believable, coherent, and sufficiently varied stories.

Table 1: Component Specification for Successful Emergent Narrative Systems

Component	Rationale
Emotional Personality Model	Emergent systems require a character-focused bottom-up approach to design. Modelling unique personalities and keeping track of emotions allows characters to react differently, yet believably, to each story event. This also includes the social relationships between each character.
Continuous Planner	Characters need to be able to act on their emotions and react to occuring story events. A planning component allows characters to manage personal goals that they're working towards and construct a series of executable actions to reach that goal. The planner component should also exhibit situated reasoning (allowing characters to react believably to specific events, especially unanticipated ones) and action repair (allowing characters to adapt their plans should they experience action failure) [11].
Appraisal	Appraisal is the assessment of any given event in relation to a character's emotions and influences their choice of actions towards their goals (world states they desire). Appraisal is essential for each agent's decision making process.
Memory / Knowledge Base	Characters need some way of remembering events that have transpired in the story. Memories of the positive and negative actions they have experienced should influence a character's decision making process. This could be a single global knowledge base that all agents can refer to (such as the *Comme il Faut* system's global rulebase [13]) or encapsulated inside the agent architecture itself [21]. The former assumes perfect knowledge of the storyworld for each character but the latter allows each character to have varying amounts of knowledge.
(Optional) Virtual Director	Many systems use some form of virtual director; a seperate agent that uses general storyworld knowledge about the plot structure to adapt and improve the story [3][10][17]. This shouldn't be used to perform global action control for individual agents (removing full agent autonomy), rather to facilitate the generation of interesting cohesive content (as in *FearNot!* [3]) or to constantly analyse the state of the storyworld and create appropriate story events that challenge the player (as in Rimworld [17]).

4. DESIGN

Note: *the following is a very early prototype design. It has not been fully implemented at this stage and is highly liable to change in the future.*

Here, we describe a design for an emergent narrative system, heavily influenced by existing systems (particularly *FearNot!*) and conforming to our criteria for successful systems **(Table 1)**.

4.1 Agent Architecture

The agent architecture for the proposed system mimics the modular design of the *FAtiMA* architecture [19]. It includes an emotional personality model and continuous planner (as described in the *FearNot!* system design [3]), and is augmented with an autobiographical memory module inspired by Dias et al.'s extension to *FAtiMA* [21] but using an alternate method [28] that was researched and implemented successfully with non-playable characters in *Minecraft*. Every character in the game adopts this design **(Figure 6)**.

4.1.1 Emotional Personality Model

There have been many personality models proposed in research including the OCC model [25], which is used by *FearNot!* [3] and the OCEAN model [26] which was employed in the creation of ERiSA (Emotionally Realistic Social Agents) [27], a framework for intelligent virtual agents in games, also developed at the University of Nottingham. At the time of writing, a 'best' model has yet to be decided. The following text assumes use of the OCC model. [25]

The design for the emotional personality model is very similar to the *FearNot!* system [3] and utilises the OCC model [25] which "defines 22 emotions through a set of rules: for example a character will be happy-for another character if an event occurs which is congruent with the second character's goals and the second character is liked by the first character." [3].

Just like *FearNot!* the proposed system utilises appraisal rules called "emotional reactions" which allow a character's emotion values to be changed as a result of experienced events. Non-cognitive reactive processing is applied through the use of "action tendencies" which are "impulsive actions caused by certain events and emotions."

4.1.2 Continuous Planner

The continuous planner allows agents to generate "plans", which are a sequence of actions, to accomplish a goal. Actions are author-specified and are atomic, meaning they are irreducable and independent of any other process in the system. This helps keep actions generic and usable by many characters in many situations. Actions can have preconditions including emotional values and/or prerequisite actions that must be completed before it can be selected. The action library is a collection of actions unique to a single character.

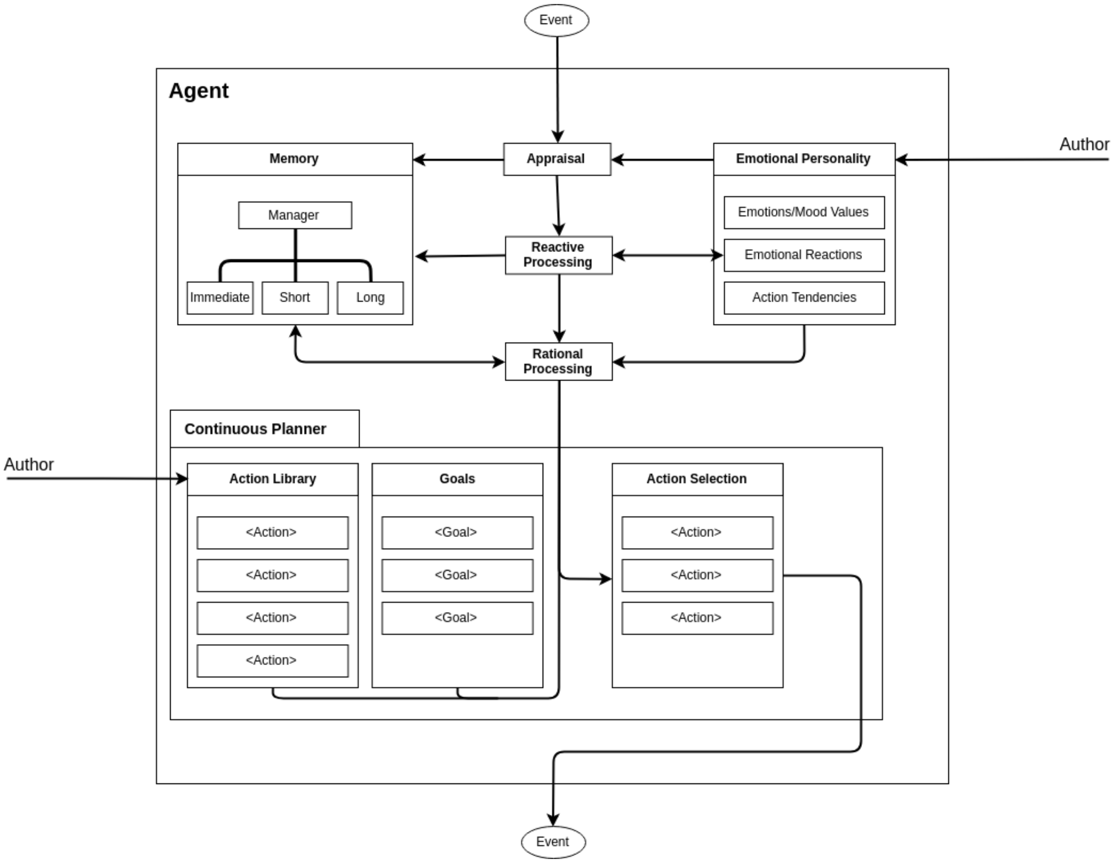


Figure 6: Agent architecture for the proposed system

Goals are generated using knowledge of a character's currently executable actions, their current emotional state, and their stored memories. For example, if a character is angry with another character and recalls a memory of them performing an action they find objectionable, they may generate a goal of starting a fight with them.

A character's plan is maintained until they either achieve their goal, or it is impossible for them to achieve their goal. The planner is designed to be "continuous" meaning that characters constantly observe the story-world state and are able to alter their plans in order to resolve challenges (action repair).

4.1.3 Appraisal

Appraisal is the assessment of any given event in relation to a character's emotions and influences their choice of actions towards their goals (world states they desire). When a character experiences a new event they must first appraise it before deciding on a response, if any. The appraisal process can be split into 4 steps:

1. **Emotional Reaction** - firstly, the character's emotional state is updated. Each event has an emotional intensity value associated with it, however, the actual amount of change for a specific emotion is dependent on each character's threshold value for that emotion. This step represents the initial reactive "emotional reaction" that someone would have to an action, before they've decided to respond.

2. **Reactive Processing** - secondly, the character's action tendencies are evaluated to see if any of them have been triggered by the new event. Action tendencies are the physical manifestations of reactive processing and represent non-cognitive, impulsive actions that a character expresses before taking their goals into consideration (e.g. crying, screaming, laughing, etc.).

3. **Rational Processing** - thirdly, the character performs rational, cognitive processing by evaluating the event with respect to their goals, existing plans, and memory of previous events before making a final decision on how to react (action selection).

4. **Commit to Memory** - finally, the character commits the event, their response, and a record of the associated emotional state changes to memory so that they can use it to evaluate future events and recall the event when communicating with other characters.

4.2 Autobiographical Memory

The idea of the autobiographical memory module is to allow characters to engage in context-dependent recall of previous events that have occurred in the game. The concept was inspired by Dias et al.'s extension to the *FAtiMA* architecture [21] but aims to utilise a different model [28] which was successfully tested using NPCs in *Minecraft* and uses a connectionist network model where the nodes are memories consisting of keywords, emotional valences, timestamps, and weights(importance). When engaging in conversation, characters can query each other's memory networks with keywords and provide a recount of related events.

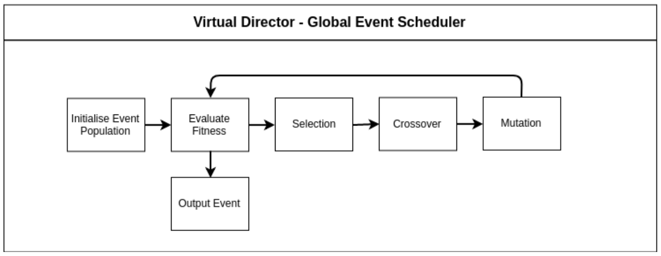

Virtual Director - Global Event Scheduler
Initialise Event Population
Evaluate Fitness
Selection
Crossover
Mutation
Output Event

Figure 7: High-level example of the processes involved in a genetic algorithm

Because memories stored in this model hold emotional valences, characters can not only remember what event occured, but also how they felt about it too. This makes the memory model useful for the appraisal of similar events (with matching keywords) and/or action selection: if the character remembers how their choices played out previously, it will help them make similar decisions.

Finally, the memory model used also allows for multiple memory "pools" including immediate, short and long -term memory. "This allows the system to differentiate between memories which were recently recalled (in the immediate memory pool) and memories which have not been recalled for a long time (in the long term memory pool)." [28].

4.3 Virtual Director

The virtual director is designed as a seperate agent that maintains general story-world knowledge about the plot structure. *FearNot!* used a director-like agent called the story facilitator to set-up each episode and ensure it was coherent and relevant to previous events in the story [3]. *Virtual Storyteller* used a virtual director to perform global action control for individual agents in order to ensure their actions were in line with a "good" plot (removing full agent autonomy) [10].

The goal of the virtual director in our system is uniform in that its purpose is to establish an interesting narrative. However, instead of global agent control or behind-the-scenes episodic creation, it aims to generate appropriate "global events", **at run-time**, that challenge the player and other characters in the game. It is inspired heavily by RimWorld's "AI Storyteller" which works in a similar fashion [17].

The virtual director maintains story world state including the player and each character. It is able to query any character to retrieve information on their emotions, plans, memories etc., but is not able to perform action selection for characters because this would violate their full-autonomy. It uses this information to evaluate the coherency of the story and how interesting it is. After evaluation, the virtual director generates a "global event" to introduce to the story world which characters then react to accordingly.

4.3.1 Potential for Genetic Algorithm

Genetic algorithms are a subset of evolutionary algorithms and are designed for metaheuristic population-based optimisation. It is reasonable to depict the set of possible global events that the virtual director can generate as a population. It is therefore evident that, included in that population, are one or more “optimal” events that will result in a more coherent and interesting story when chosen over the others. Therefore, the potential for implementing a genetic algorithm for use in the virtual director presents itself.

Genetic algorithms must involve a representation for candidate solutions. Typically this is a string of binary, integer or real values where each bit represents a specific part of the solution (e.g. target character, when to start the global event, emotional valences, etc.). A population is then initialised using this representation. Members of the population are repeatedly selected (using a selection scheme such as tournament or roulette-wheel selection) and crossover and mutation is performed between candidate solutions to perturb them. After this process, the resulting population is evaluated and, if the desired criteria is met, a solution is chosen. If not, a replacement strategy is applied and the process repeats itself until a soluton is found **(Figure 7)**.

5. IMPLEMENTATION

5.1 Unity

The prototype emergent narrative system will be implemented in the Unity game engine and written in C#. Unity was chosen for several reasons:

- **Free-to-Use** - Unity Personal edition is free-to-use and includes all the engine’s core features, enough to build the system.
- **Ease-of-Use** - Unity is designed to be simple and easy to approach. Loading and using graphics, sound and other essential game components is very easy and is abstracted with Unity’s simple GUI for the most part. This allows time spent on the project to be focused on the system’s implementation, not its presentation.
- **GPU Programming** - Unity allows developers to write compute shaders: programs that run on the GPU, but outside of the normal graphics pipeline and allow algorithms to be massively parallelised. This is potentially useful for greatly improving the run-time of large calculations (such as running the genetic algorithm’s fitness function across the search space).

5.2 Prototype System

So far, the only implemented part of the prototype system is the emotional personality model for each agent. Currently, emotions are represented by the OCC model [25] which is also used in the *FearNot!* system [3]. Each character’s emotional personality model is constructed in an XML file unique to that character and following a standard schema.

Work has also begun on constructing the continuous planner, beginning with the creation of a XML schema for generic actions and each character’s action library, though this is far from complete at this stage.

5.3 Tools for Authors

A crucial part of the system is the ability for authors to have great control over the creation of unique and varied characters. For this reason, work has also started on a suite of creation tools. Currently, this tool allows authors to specify a list of characters for the system and edit the threshold, decay and initial values for each emotion. When finished, authors can export individual XML files for each character which are easily “plugged-in” to the emergent narrative system without any additional work.

6. PROJECT PROGRESS

6.1 Project Management

The Gantt chart **(Figure 8)** is from the initial proposal for this project. **Table 2** shows each task with different colours defining the amount of progress made toward them:

- **RED** - not completed
- **AMBER** - partially completed
- **GREEN** - completed

The project is running slightly behind schedule, with 2 tasks partially completed, and 1 not started:

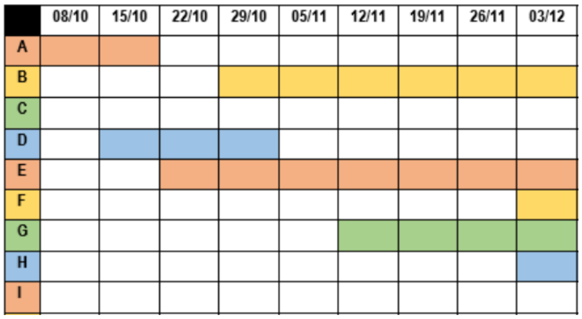
- **Task G** - work has started on implementing the prototype framework but it is not complete yet (see **Implementation** section).
- **Task H** - because the prototype framework has not been fully implemented, the testing of existing IS methods has not been conducted yet.
- **Task I** - task I was not supposed to be started until after the interim report was submitted. However, as the research from tasks D-F has been so successful, a design for a novel system has been proposed in this report.

6.2 Contributions and Reflections

I believe the slight delay in project progress is a result of the learning process. Some of the more complex concepts in this

Table 2: Project Gantt Chart - Task Key

Task ID	Task Description
A	Write Proposal
B	Write Interim Report
D	Research and evaluate existing games that are products of IS research
E	Research and evaluate the success of existing IS methods, particularly EN methods
F	Research ways of creating believable and effective AI agents
G	Implement a simple framework for a prototype where different IS methods can be tested
H	Use the prototype to test existing IS methods and identify their pros and cons
I	Develop a novel hybrid approach to IS by combining parts of successful approaches



	08/10	15/10	22/10	29/10	05/11	12/11	19/11	26/11	03/12
A	🟧	🟧							
B				🟨	🟨	🟨	🟨	🟨	🟨
C									
D		🟦	🟦	🟦					
E			🟧	🟧	🟧	🟧	🟧	🟧	🟧
F									🟨
G						🟩	🟩	🟩	🟩
H									🟦
I									

Figure 8: Gantt chart from the project proposal

project (modelling intelligent autonomous agents and understanding how the multitude of existing systems worked) were difficult to grasp initially.

I spent a long time researching existing systems and ensuring that I fully understood how they worked before I could write this report. Despite this issue, I believe I have managed my time well; the proposed criteria for creating successful systems **(Table 1)** is the output of extensive research into the field and the high-level design for the new system promises several extensions to that research (namely the autobiographical memory module and the genetic algorithm approach for building the virtual director).

At this time, I believe all the project objectives can be completed on time and I am looking forward to implementing the system design specified in this document and conducting user testing. I believe the most novel part of the project is the virtual director. Genetic algorithms are not prolific in the field of interactive storytelling and it will be interesting to discover how suitable they are for generating effective narratives. My initial task schedule will not be changed at this time. The upcoming Christmas break will give me plenty of time to further implement the prototype and hopefully the system will be near finished and ready for user-testing by the new year.

7. REFERENCES

[1] Bostan B., & Marsh T. (2012). “Fundamentals Of Interactive Storytelling”. *Academic Journal of Information Technology*, **3**(8), pp. 20-42

[2] Riedl M. O. (2010). “A Comparison of Interactive Narrative System Approaches using Human Improvisational Actors”. *Georgia Institute of Technology, College of Computing*

[3] Louchart S., Aylett R., Kriegel M., Dias J., Figueiredo R., & Paiva A. (2008). “Authoring Emergent Narrative-Based Games”. *Journal of Game Development*, **3**(1), pp. 19-37

[4] Louchart S., & Aylett R. (2003). “Solving the Narrative Paradox in VEs - Lessons from RPGs”. *In: Rist T., Aylett R.S., Ballin D., Rickel J. (eds) Intelligent Virtual Agents. IVA 2003. Lecture Notes in Computer Science, Vol. 2792*, pp. 244-248

[5] Aylett R. (2000). “Emergent Narrative, Social Immersion and “Storification””. *Proceedings Narrative and Learning Environments Conference*, Edinburgh, Scotland

[6] Aylett R. (1999). “Narrative in Virtual Environment - Towards Emergent Narrative”. *In: Proceedings of the AAAI Fall Symposium on Narrative Intelligence, 1999*

[7] Kriegel M., & Aylett R. (2008). “Emergent Narrative As A Novel Framework For Massively Collaborative Authoring”. *In: Prendinger H., Lester J., Ishizuka M. (eds) Intelligent Virtual Agents. IVA 2008. Lecture Notes in Computer Science, vol 5208*

[8] Klein S., Aeschlimann J.F., Balsiger D.F., Converse S.L., Court C., Foster M., Lao R., Oakley J.D., & Smith J. (1973). “Automatic Novel Writing: A Status Report”. *University of Wisconsin*

[9] Meehan, J.R. (1977). “Tale-Spin, an Interactive Program that Writes Stories”. *In: Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI)*

[10] Theune, M., Faas S., Nijholt, A., & Heylen, D. (2003). “The Virtual Storyteller: Story Creation by Intelligent Agents”. *In: Proceedings of TIDSE 2003: Technologies for Interactive Digital Storytelling and Entertainment*

[11] Cavazza, M., Charles, F., & Mead, S.J. (2002). “Character-based Interactive Storytelling”. *IEEE Intelligent Systems*, **17**(4), pp.17-24

[12] McCoy, J., Treanor, M., Samuel, B., Reed, A., Mateas, M., & Wardrip-Fruin, N. (2013). “Prom Week: Designing past the game/story dilemma”. *In: Proceedings of the Eighth International Conference on the Foundations of Digital Games (FDG 2013)*, pp.94-101

[13] McCoy, J., Treanor, M., Samuel, B., Wardrip-Fruin, N., & Mateas, M (2013). “Comme il Faut: A System for Authoring Playable Social Models”. *In: Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp.158-163

[14] Guimaraes, M., Santos, P., & Jhala, A. (2017). “CiF-CK: An Architecture for Social NPCs in Commercial Games”. *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pp.126-123

[15] Social NPCs Mod for Elder Scrolls V: Skyrim, available at: https://www.nexusmods.com/skyrim/mods/77792/ [Accessed 21 Nov. 2018]

[16] Maxis: The Sims. Electronic Arts (2000)

[17] Tynan Sylvester: RimWorld. Ludeon Studios (2018), available at: https://rimworldgame.com/ [Accessed 21 Nov. 2018]

[18] Adams, T., Adams, Z.: Slaves to Armok: God of Blood Chapter II: Dwarf Fortress (2006)

[19] Dias, J., Macscarenhas, S., & Paiva, A. (2014). “FAtiMA Modular: Towards an Agent Architecture with a Generic Appraisal Framework”. *Emotion Modeling* pp.44-56

[20] Dias, J., & Paiva, A. (2005). “Feeling and Reasoning: A Computational Model for Emotional Agents”. *In: Proceedings of 12th Portuguese Conference on Artificial Intelligence, EPIA 2005,* pp.127-140

[21] Dias, J., W. C. Ho., Vogt, T., Beeckman, N., Paiva1, A., & Andre, E. (2007). “I Know What I Did Last Summer: Autobiographic Memory in Synthetic Characters”. *Affective Computing and Intelligent*

Interaction: Second International Conference, ACII 2007 Lisbon, Portugal, September 12-14, 2007 Proceedings pp.606-617

[22] Ho, W.C. & Watson, S. (2006). “Autobiographic Knowledge for Believable Virtual Characters”. *In: Proceedings of the 6th International Conference on Intelligent Virtual Agents (IVA 2006)* pp.383-394

[23] Hayes-Roth, B., Brownston, L., & Sincoff, E. (1995). “Directed Improvisation by Computer Characters”. *Technical Report KSL 95-04, Knowledge Systems Laboratory, Stanford University, 1995*

[24] Perlin, K., & Goldberg, A. (1996). “Improv: A System for Scripting Interactive Actors in Virtual Worlds”. *In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)* pp.205-216

[25] Ortony, A., Clore, G.L., & Collins, A. (1988). “The Cognitive Structure of Emotions”. *New York, NY, US: Cambridge University Press*

[26] Saucier, G., & Goldberg, L.R. (1996). “The Language of Personality: Lexical Perspectives on the Five-Factor Model”. *In: The Five-Factor Model of Personality: Theoretical Perspectives* pp.21-50

[27] Chowanda, A., Blanchfield, P., Flintham, M., & Valstar, M. (2014). “ERISA: Building Emotional Realistic Social Game Agent-Companions”'. *In: Proceedings of the 14th International Conference on Intelligent Virtual Agents (IVA 2014)* pp.134-143

[28] Kope, A., Rose, C., & Katchabaw, K. (2013). “Modeling Autobiographical Memory for Believable Agents”. *In: Proceedings of the Ninth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* pp.23-29