IMPLEMENTATION OF AN EMERGENT NARRATIVE GENERATION SYSTEM FOR VIDEO GAMES

Final Progress Report

By

Fahreen Bushra

CS 4490Z, Thesis

Department of Computer Science

Western University

04/08/2022

Project supervisor: Michael James Katchabaw, Dept. of Computer Science

Course instructor: Nazim H. Madhavji, Dept. of Computer Science

1.0 Structured Abstract

Context and motivation

- Interactive story-telling in videogames offer non-linear narratives in which the story emerges from the actions of the player. An ideal emergent narrative generation system should offer the following:
  - temporal and spatial consistency
  - high freedom to the user
  - flexible granularity
  - significant level of creative control
  - pacing control
  - Domain independence

However, currently no narrative generation architecture has satisfied all evaluation metrics.[1]

Research Question:

- Can we implement a narrative generation which satisfies all the evaluation metrics?

Principal ideas:

- The principal idea of this research is based on existing research in supervisors' group, which proposes a design for a narrative generation architecture. The research also includes Vladimir Propp's morphology of folktales [2].

Research approach (or Methodology):

- Build a prototype system in unity which implements existing design in supervisors' group for a narrative generation architecture. Extend the complexity of narratives with the Vladimir Propp's morphology of folktales.

Anticipated results of your research:

- Research implements a narrative generation system which creates non-linear stories which respond dynamically to user actions. Implementation of such an architecture and the analysis of its viability will enhance narrative generation methods for video games.

Anticipated impact of results:

- Showcase a prototype which makes use of the proposed architecture from previous research. This prototype acts as a demonstration for the effectiveness of this architecture and can guide future research towards commercial use of the engine in creating games with dynamic narratives.

1.0 Introduction

A majority of commercial videogames offer interactivity solely through gameplay. More often than not, narratives in video games follow the formula of films, with predetermined cutscenes and story lines which do not respond to player input. Commercially, dynamic storytelling is often reserved for side missions, in which a player may meet new characters and foster relationships with them, however this rarely changes the pre-determined main story.[1]

Some games such as “Life is Strange”[3] and “Detroit Become Human”[4] offer multi-branching storylines, where certain outcomes have been linked with specific choices of the player. However, this is still quite limited as these games often do not differ heavily between different outcomes, and rather have a small number of possible endings at the completion of the game. A number of story generation systems have been produced in academic settings, such as:

- Lebowitz's Universe
- Turner's Minstrel
- Facade's architecture

- Oz project
- The virtual storyteller

However, none of these architectures fulfill many of the evaluation metrics for an ideal narrative generation system, such as temporal/spatial consistency, pacing control etc.[1] The existing research in the supervisor's groups proposes a design for a robust architecture for generating emergent narratives to use in video games. This architecture was also tested in a text based simulation in C++.[1] In this project, this architecture is further developed and implemented in unity. Additionally, the system was integrated into a more sophisticated video game, for which the graphics and gameplay will demonstrate the narrative architecture.

The report begins with the Background section, containing an overview of current Narrative Generations Systems that have been produced. This is followed by a breakdown of the evaluation metrics which was used to evaluate the system developed in this thesis. The Narrative systems are then compared according to this evaluation Criteria. Following this, is the results section, containing a breakdown of the architecture and implementation details of the system developed. A case study is also provided. The system is validated with the use of the evaluation metrics, Finally, the significance of the key results are discussed.

2.0 Background and Related Work

2.1 Dynamic Narrative Generation Systems

A number of story generation systems have been produced in academic and commercial settings, such as:

- Lebowitz's Universe: The system uses recurring plotlines extracted from daytime television soap operas to create intricate and intertwined stories among a large cast of characters. [5]
- Turner's Minstrel: creates stories about King Arthur and the Knights of the Round Table. The system begins with limited knowledge about the domain, and generates a number of stories using a problem solving approach. Each event is presented as a problem, which is solved by adapting a similar, previously used problem, from the system's database. [6]
- Facade's architecture: With the use of simple graphics and natural language processing, this system creates an interactive drama, where the user experiences a dinner party with friends. [7]
- Oz project: accepts pre-authored points, input to the system by the author, that make up the major transitioning points of the story. The order of plot points are evaluated by system functions. [8]
- The Virtual Storyteller: aims to create virtual environments inhabited by autonomous agents with which a user can interact. Characters can set their own goals, and act based on their internal goals and personalities. With this system, the plot is not guaranteed to be interesting.[9]

2.2 Evaluation metrics

The following sections evaluate different design aspects of narrative generation systems that must be considered when using a system to create a story. [10]

- <u>Temporal and Spatial Consistency:</u> Temporal consistency guarantees the plot occurs in the correct order. Spatial consistency requires that all locations of characters and objects are accounted for and realistic..
- <u>Granularity of Story Control</u>: the amount of control the *system* has over the story. A highly granular system has less control. For a system with low granularity, such as linear narratives, each aspect is micromanaged.
- <u>Freedom to the User:</u> the degree to which the *user* can impact the story through their actions. A high amount of freedom is desired for the ideal system.
- <u>Ease of Authoring the Story/Creative Control</u>: the amount of creative control given to a human author, for the story direction.
- <u>Pacing Control</u>: A system with good pacing control assures that events do not occur too fast or too slowly.
- <u>Domain Dependency</u>: Systems that are domain independent allows that the story can be told using a variety of media, e.g. text based game or a 3d interactive world.

2.3 Analysis and Research Gap

An ideal narrative generation system should offer: temporal and spatial consistency, flexible granularity, high freedom to the user, a significant level of creative control to the author, appropriate story pacing, and domain independence[10]. Figure 1 summarizes the evaluation metrics for each Narrative generation System discussed above. When it comes to Temporal and Spatial Consistency, all five systems are capable of satisfying the criteria. However for Granularity, it is only the Oz project[8] and the Virtual Storyteller[9] which has large Granularity, but lack in creative control and pacing. Facade[7] and Turner's Minstrel[6] have limited User freedom and small granularity. Of the five systems, only three offer different methods to control the pacing of the story.

An analysis of the six evaluation criteria reveals that currently, there exists no such narrative generation system, which satisfies all six evaluation criteria. The proposed architecture [1] combines the favorable attributes of each narrative generation system, and fulfills all six criteria, as seen on Figure 2. By implementing this design and building a prototype to prove its efficacy, this may lead to the development of an improved architecture for creating dynamic, Emergent narratives for video games.

<u>3.0 Research Objectives</u>

- O1: Implement the previously proposed architecture for a emergent narrative
- generation in Unity
- O2: Create a prototype to showcase the usability if the narrative generation system
- O3: Use functions from Vladimir Propp's Morphology of folktales to guide the prototype narrative
- O4: Evaluate the extended architecture and Prototype based on evaluation metrics

<u>Significance</u>

The anticipated results of this thesis will extend the current research in the field of dynamic storytelling in videogames. Currently, much of storytelling in games are either linear or rely on pre-written branching storylines, in which the player has little to no control [12]. Furthermore, previous work for narrative generation has many limitations such a lack of consistency or being domain dependent. This research aims to overcome these limitations by further developing a system capable of creating dynamic stories based on a player's actions. Ultimately, by incorporating a more dynamic story within games, the overall goal is to create a player experience that is more immersive.

4.0 Methodology

The design of the narrative generation system was obtained from a previously published paper by the supervisor's group [1]. This work contained a general design for a narrative system, identifying key components (e.g. Director), responsibilities of those components(e.g. The director must keep track of all possible story elements), and the relationships between each component(e.g. The Director communicates with the Controller to make changes in the story). This design proposed that if implemented correctly, it would satisfy all the evaluation criteria for a narrative generation system. With the use of this design, a software architecture diagram was created for its implementation. Unity was chosen as the engine for development because it is a common game development tool with a large amount of resources and it allows the creation of games in a large number of domains (e.g. 2D, 3D or text based games). This would allow the architecture to be domain independent.

Once the architecture was implemented, the first prototype created with the architecture was a text based simulation, which consisted of a number of buttons that would allow the user to have control over the story world values. The story written for the prototype was derived from the first 5 functions of Vladimir Propp's Morphology of the Folktale.

The second prototype implemented using the architecture was a 2D RPG style game, where the player is able to interact with the game world by moving the player with arrow keys and interacting with the characters through a dialogue system. This prototype extends the first 5 functions, by randomly generating characters, and their attributes under defined constraints. Finally, the prototype was assessed under the six evaluation criteria, through a detailed case study.

5.0 Schema Design Results

Figure 3 represents the contextual diagram of the narrative generation system within unity, the prototypes implemented, and the role of the Player, Author, and System User. The Architecture and both prototypes were built in unity with custom C# scripts, inheriting from the Unity Mono-behaviour class. The singleton design pattern is used to guarantee that only one instance of these classes can exist in the Unity Scene.

5.1 Architecture Implementation

The design for the Architecture of the narrative generation system was extended from previous research conducted by the supervisor [1]. A diagram of this system can be seen in Figure 4**.** This design was used as a basis to create the software design and implementation for the architecture. The key Components of the architecture are detailed below.

5.1.1 Plot Fragments, Preconditions and Actions

The Preconditions class for this implementation contains a Boolean value and an ID. Each instance of a precondition represents the evaluation of a specific story world value within the narrative and can only be set to true or false. The Plot Fragment class represents events that occur in the story. Each plot fragment has precondition members, which must be true in order to be presented to the player. The Fixed Precondition List contains preconditions that cannot be changed through player actions, but rather are decided at the instantiation of a new game. The Mutable Preconditions and Trigger Preconditions lists are changeable through player action. All Fixed preconditions and Mutable preconditions must be true for the system to consider the plot fragment. Once true, the system waits until at least one trigger Precondition is true. At this point, the plot fragment is presented to the player, with the use of the Action member. The Action Member is an Interface, for which every plot fragment is assigned a specific implementation for its *execute()* method. Under this method, the system user can specify how to change the game world and present the plot fragment to the player.

5.1.2 Gameworld Generator

The Gameworld Generator class is responsible for instantiating the game. All elements defined in this class define the story, such as Preconditions, Plot fragments, Actions, Characters and Roles. Plot Fragments are grouped under Author level Plot fragments, which are story plot points meant to guide the story, based on the author's intent. The order of the author-level fragments are defined in this class. This order is maintained by the system, as each author-level plot fragment uses the completion of the previous fragment as a precondition.

5.1.3 Director, Controller, Evaluator

The Director is the core component of the architecture, responsible for managing plot fragments and system goals. The Director decides which plot fragments to present to the player with the use of the Evaluator. The Evaluator class interacts directly with the game world and communicates to the director, once a precondition changes states. Once the Director has chosen a plot fragment to present to the player, it provides the Controller with the Action member. The Controller is able to directly change the Game World based on the Action instructions.

5.1.4 Character and Roles

Each character contains a prefab component which represents a game object from the Gameworld. A Role class represents theory is defined under the Character class, containing a name, gender, age, relation to hero and a pr type of effect that character has on the story, and contains a reference to a character object.

5.2 Prototype Implementation - Story

The story for the prototype was based on the first five functions of Vladimir's Propp's morphology of folktales. The first 5 functions serve to set up the villain with crucial information about the hero to forward the plot. Each function contains sub-types, representing different methods in which the function may be satisfied. Figure 4 outlines the first five functions. Each function is represented as an author level plot fragment in the prototype. Within each author level fragment, subtypes are defined as basic level plot fragments. Figure 5 shows the plot fragments written for the prototype. The Prototype defines five roles; Hero, Villain, Absentee, Interdiction Messenger, and Guardian. On instantiation of the Game, the Gameworld Generator creates a noble family with seven members, containing the hero, the mother, father, and 4 siblings. The genders and ages of the characters are chosen randomly within given ranges. In addition, Fixed precondition values of the world, such as "Family secret exists" are assigned either true or false at instantiation.

5.3 Prototype Implementation - Game world

Two Game Worlds were created for the prototype. The first prototype created was a text-based low fidelity prototype, where user input was given through buttons, which directly changed story world preconditions. The second prototype created is a 2D RPG style game. The character is able to navigate within a 2D castle and interact with items and other characters through a dialogue system. Figure 6 showcases a case study for each prototype.

5.4 System Validation

The Architecture and Prototype of the system was evaluated based on six evaluation metrics.

5.4.1 Temporal and Spatial Consistency

Spatial and temporal consistency can be achieved through preconditions, which guarantee that certain conditions must be true or specific characters must be present in story locations before the fragment is presented to the user. This metric is satisfiable by the architecture but is ultimately the responsibility of the system user and author to maintain consistency through accurate preconditions.

5.4.2 Granularity of Story Control

The proposed system's granularity is flexible and can be either low or high, based on the author's design. This is determined by the Author level plot fragments. To achieve a linear storyline, the author can decrease granularity with low level author plot fragments, with very few preconditions, easily achievable by the user. In contrast, the author may employ high level author plot fragments in order to give the user more freedom over the story.

5.4.3 Freedom to the User

The system is able to give the player a high amount of control, based on the complexity of the story world and the number of plot fragments and gameplay elements implemented in the game world generator. The prototype created for this architecture, however, does not offer a high amount of control as a small number of plot fragments were implemented due to time and resource constraints.

5.4.4 Ease of Authoring the Story/Creative Control

The system gives the author a high level of creative control based, through the design of plot fragments and author level plot fragments. However, the design of the plot fragments and director can make it quite difficult to write good narratives, especially so as user freedom increases. In the case of the prototype created, although the stories generated are diverse, it did not always create an interesting story.(Figure x)

5.4.5 Pacing Control

The architecture does not implement explicit pacing controls. However, the prototype created using the architecture implements functions such as coroutines, in order to keep track of time and push plot points on the player if the story is not moving forward.

5.4.6 Domain Dependency

The architecture is implemented in a way such that the Gameworld is abstracted from the story elements. This allowed the creation of two prototypes, without change in the plot fragments, director and other components in the architecture.

Novelty of Results

Currently no such architecture exists, for which all the evaluation metrics for an ideal narrative generation system is satisfied(figure 1). The novelty of the results lies within the system validation, for which this architecture shows potential to satisfy all of the validation criteria (figure 2). The implementation of the second prototype also showcases the efficacy of this architecture in commercial use. This acts as a demonstration for the effectiveness of the architecture and guides future research towards commercial use of the engine in creating games with dynamic narratives.

6) Discussion

Threats to the validity of the results

- The initial narrative implemented with the architecture was incredibly linear. Although the architecture allows for high granularity and high freedom for the user, this is dependent on the nature of the author plot fragments and basic plot fragments. Thus, the initial prototype did not satisfy the evaluation criteria in the research objectives. This threat was contained by increasing the quality and number of plot fragments written for the prototype.

Implications of the research results

- The implementation of this narrative generation system extends the current research in the field of dynamic storytelling in videogames. It proposes a design for an architecture that is able to satisfy all of the evaluation criteria for an ideal narrative generation system.
- This research implementation is the prototype in order to prove the efficacy of this architecture. With additional resources, this architecture has potential to be extended to be used in commercial games, providing an immersive, and dynamic narrative experience

Limitations of the results

- Although the architecture has the potential to satisfy all the evaluation criteria for an ideal narrative generation system, it does not guarantee it. Because this system gives a great deal of control to the Author and System user, it is the responsibility of those parties to write narratives and plot fragments in such a way that all the evaluation criteria are met.
- For example, the system is able to give a high degree of freedom to the user, with intricate plot fragments. However, the architecture can also be used to implement a very linear story.
- These conditions may be minimized through adding additional controls. However, in order to guarantee one evaluation criteria, another may be sacrificed.
- For example, the system can implement a feature to guarantee spatial and temporal consistency, however this may reduce author freedom as the system can prevent the author from exploring stories related to time travel or teleportation.

Generalisability of the results

- The system can be extended to develop narrative games in a large number of domains, within Unity.

8) Conclusions

The problem addressed by the research conducted was that to date, there exists no such narrative generation system, which satisfies the particular criteria. The proposed architecture [1] combines the favorable attributes of each narrative generation system and fulfills all six criteria. The goal of this thesis was to extend and implement this design by building a prototype in Unity to prove its efficacy and Usability. The development of this project may lead to the development of an improved architecture for creating dynamic, Emergent narratives for video games.

The details of the Architecture implemented in this research, can be seen in section 5.1. When the architecture implementation was judged based on the 6 evaluation criteria discussed in section 5.2, it was found that the architecture can indeed satisfy all six criteria. The results of this are summarized in figure 2. However, as discussed in section 6, the architecture does not guarantee that all six criteria are always met, but rather enables the Author and user of the system to achieve the six defined criteria. Although the prototype implemented does offer a number of storylines, it is still not incredibly diverse, in terms of possible narratives. Thus, we can conclude that the narrative generation system implemented can offer a very dynamic story within games in order to create more immersive experiences, however, it must be coupled with Authors who are able to make use of the narrative system.

9) Future Work and Lessons Learnt

- Extend the prototype to include more complex plot fragments. This will result in a more dynamic story
- Incorporate Procedurally generated characters, locations and areas at instantiation of the game.
- Enable a memory system for each character, and enable AI behavior to implement more believable characters
- Incorporate gameplay within actions in order to create a more enjoyable experience

10) Acknowledgments

I'd like to thank Dr. Mike Katchabaw for his continued guidance, support and patience for this project.

10.0 References

[1] Schudlo, Nicholas A., "Development of an Emergent Narrative Generation Architecture for Video Games" (2014). Electronic Thesis and Dissertation Repository. 2429. https://ir.lib.uwo.ca/etd/2429

[2] V. J. Propp, L. Scott, S. Pirkova-Jakobson, L. A. Wagner, and A. Dundes, Morphology of the Folktale. University of Texas Press, 1977.

[3] Life is Strange. Dontnod Entertainment, 2015.

[4] Detroit: Become Human. Quantic Dream. 2018.

[5] M. Lebowitz, “Story-Telling as Planning and Learning.,” 1985.

[6] S. Turner, The Creative Process: A Computer Model of Storytelling and Creativity. Psychology Press, 1994

[7] M. Mateas and A. Stern, “Façade: An Experiment in Building a Fully-Realized Interactive Drama,” in Game Developers Conference (GDC’03), 2003.

[8] M. Mateas, “An Oz-Centric Review of Interactive Drama and Believable Agents,” 1997.

[9] M. Theune, S. Faas, E. Faas, A. Nijholt, and D. Heylen, “The Virtual Storyteller: Story Creation by Intelligent Agents,” in Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference, 2003, pp. 204–215.

[10] B. Magerko, “A proposal for an interactive drama architecture,” Ann Arbor, vol. 1001, p. 48109, 2000.

[11] B. Magerko and J. Laird, “Building an interactive drama architecture,” First Int. Conf. Technol. Interact. Digit. Storytell. Entertain., pp. 226–237, 2003.

[12] P. Holleman, “Narrative in Games,” The Game Design Forum

Appendix

Evaluation Metric						
System	Temporal & Spatial Consistency	Granularity	User Freedom	Creative Control	Pacing Control	Domain Dependency
Oz Project	None described. Dependant on the author.	Large grain	High	Medium	No pacing control.	Domain independent.
Façade	Yes. Dependant on the author.	Medium to small grain	Limited	High	Has pacing control	Domain dependant.
Virtual Storyteller	Yes Maintained through agent models and action supervision by Virtual Director.	Large grain	High	Medium	Has pacing control.	Both domain independent & dependant
IDA	Yes Dependant on the author.	Medium to small grain	High	High	Has pacing control.	Domain independent.
GADIN	Yes	Small grain	Limited	Limited	No pacing control.	Domain independent.

Figure 1: Summary of Evaluation Criteria for Narrative Generation Systems

Evaluation Metric						
System	Temporal & Spatial Consistency	Granularity	User Freedom	Creative Control	Pacing Control	Domain Dependency
Proposed Architecture Results	Yes. Can be maintained by the author with the use of Plot Fragments and Author Plot fragments.	Large grain	High to medium. Dependant on Author	High	Can be implemented through preconditions of plot fragments, such as coroutines.	Domain independent.

Figure 2: Summary of Evaluation Criteria for proposed Architecture

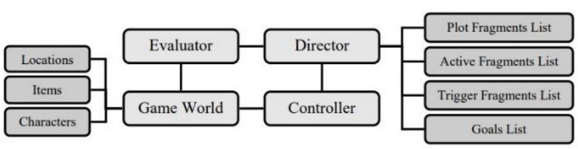

Figure 3: Overview of proposed architecture

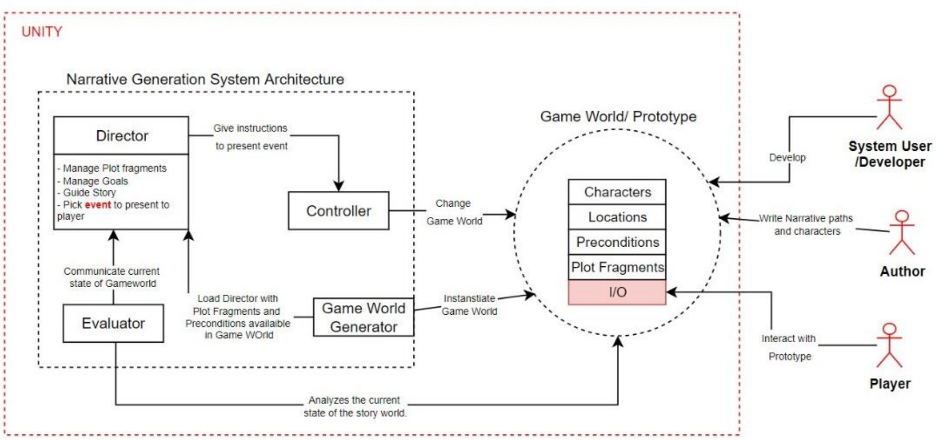

Figure 4 : Contextual schematic system diagram

Vladimir Propp's First Five Story Functions	
Function	**Function Sub-Types**
ABSENTATION(β) Def: one of the members of a family absents himself from home.	β^1 : The person absenting himself can be a member of the older generation(like parents) β^2 : An intensified form of absentation is represented by the death of parents β^3 : Sometimes members of the younger generation absent themselves(they go visiting, fishing, for a walk, out to gather berries, etc.)
INTERDICTION(γ) Def: an interdiction(the action of prohibiting or forbidding something) is addressed to the hero. A forbidding edict or command is passed upon the hero ('don't go there', 'don't do this'). The hero is warned against some action.	γ^1 : Interdiction **not to go out** may be strengthened or replaced by putting children in a stronghold. Can show up as a weaker form, such as a *request* or *advice* γ^2 : An *inverted* form of interdiction(an order/suggestion). e.g. "*Bring breakfast out into the field*", "*Take your brother with you to the woods*"
VIOLATION(δ) Def: the interdiction is violated	The forms of violation correspond to the forms of interdiction. Functions 2 and 3 form a paired element. However, The 3 can sometimes exist without 2.
	ϵ^1 : Reconnaissance. The reconnaissance has the aim of finding out the location of children, precious objects

Figure 5 : Outline of Vladimir Propp's Story Functions

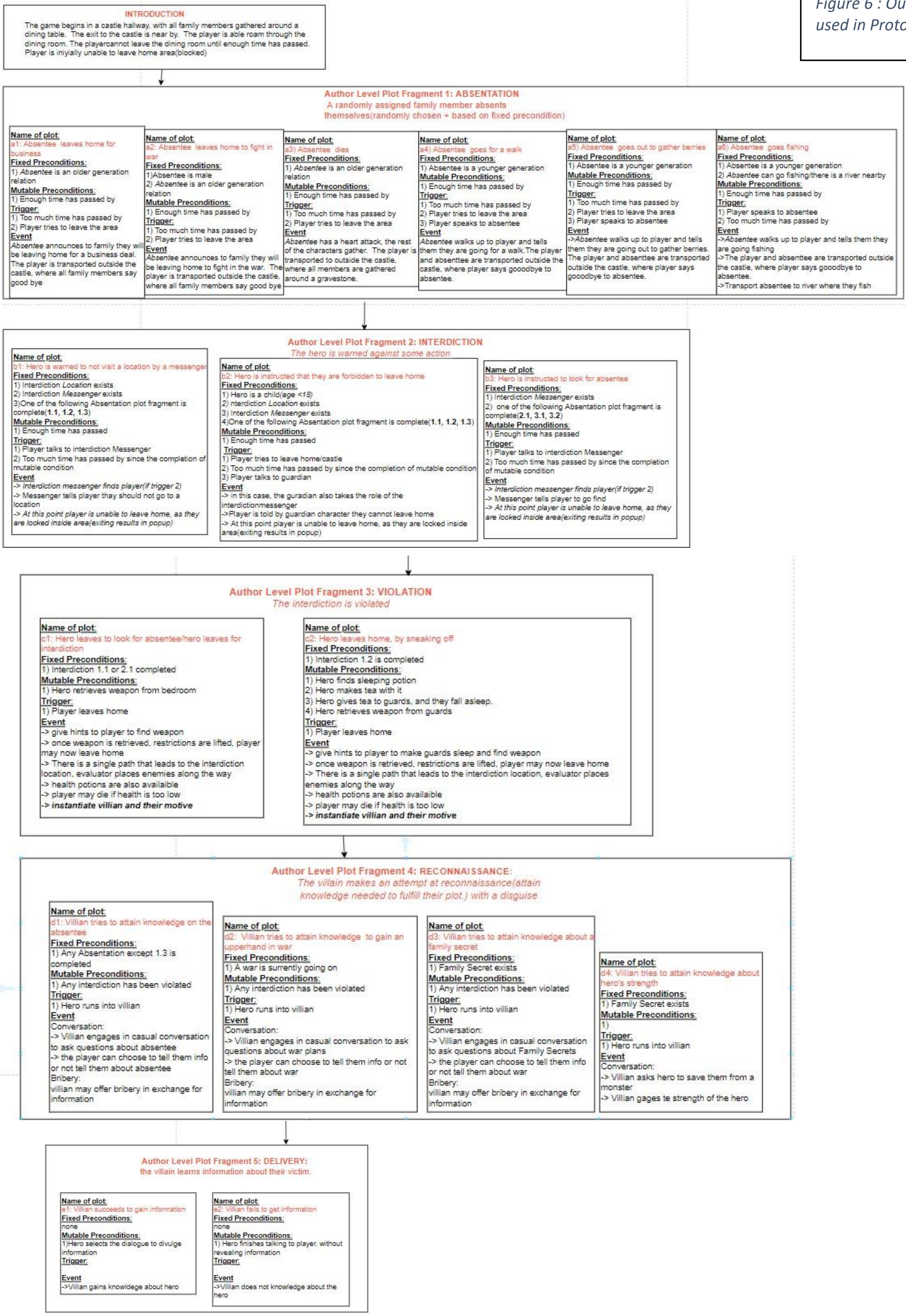

Figure 6 : Outline of Plot fragments used in Prototype

Figure 7 : Case Study conducted of two system prototypes.