

# Artificial Folklore for Simulated Religions

Jason A. Hall, Benjamin Williams, and Christopher J. Headleand  
School of Computer Science  
University of Lincoln  
Brayford Pool, Lincoln, UK  
Email: cheadleand@lincoln.ac.uk

**Abstract**—In this paper we use grammar-directed procedural content generation (PCG) techniques to develop folklore; based on the seven basic story plots, for a simulated religion. A hierarchy of values for a simulated community was first generated. Using these values, a variety of deities were procedurally generated, each with their own reflected values and persona. A Context-Free Grammar is then traversed in order to generate fables appropriate for each deities persona. The intention of this work is to generate fables which can be used to contextualize a given simulated culture’s beliefs.

## I. MOTIVATION

A realistic and contextually consistent virtual environment is a crucial component in maintaining player immersion [1]. Typically this is achieved through a process of manual design, which provides the developers total creative control, but limits the possible experiences a player can have. Procedural Content Generation (PCG) provides an alternative approach, by algorithmically generating variations in content. However, maintaining contextual consistency becomes increasingly challenging due to the stochastic nature of the approach.

Video games often have an underlying story or lore which structures many of the contextual elements in the game. This provides contextual depth to the game world, and provides underlying justification for core and ancillary content. A good example of this is *Fallout 4* and *Elder Scrolls: Skyrim* where the lore is central to the game narrative. However, there are very few examples where cultural and contextual content has been embedded into a procedurally generated virtual world.

This paper presents early stage work towards procedurally creating cultural context for a virtual world. We present a system for generating shared parables and legends which could act as the basis of simulated belief system.

## II. BACKGROUND

Fictional religions are often used as a framing device for contextual and thematic elements in games. This is often explicit, structuring significant narrative elements. Wagner [2] states that “Religion itself can work as a sort of game, shaped by an agreed-upon set of rules, defining how we should act, what’s possible, and what our goals are”. This structure could be applied to non-player characters (NPCs), implementing behavior that is culturally consistent within the game world. Johnson [3] explores this concept in the design of the “Ultima Ratio Regum” (URR) game, which rather than having religion as a background element, it contributes to the procedural generation of the NPCs. The algorithm used in URR is capable

of generating over a million detailed religions. Each religion contains information about it’s beliefs, god(s), special events, relationships with other religions, geographical areas of influence, altars, and worshipers. A commercial example, can be found in the *Elder Scrolls: Skyrim*, where an in-game religion provides a context to the behaviours exhibited by NPC’s. An example of this is the conflict between the *Imperials* and *Nords* which is of religious nature. In *Black and White*, the player assumes the role of a god, with the ultimate objective of defeating an opposing deity, in a good vs evil value narrative. However, religious elements in games are often less explicit, inferred through elements such as architecture and shrines [4].

However, while religions are often described in the broadest sense, their value and belief system is often glossed over. At the time of writing, we do not believe there are any games which include detailed fables or parables in the context of simulated religious literature. Aristotle, argued that a fable was an “instrument of persuasion” [5], and many religious stories express an underlying moral position central to the religions belief system. As noted by Turner [6] fables are often used as tools to pass down traditions and values down generations.

*Tale-Spin* [7] produces procedural short stories by generating descriptions regarding how each character goes about completing a goal. The system is supplied with inputs such as: characters, places and a objectives, by running these inputs through a rule system, stories are generated.

*Minstrel* [6] follows a similar approach, describing the process of which a character has gone through to solve a problem. However, this system uses case-based reasoning to produce new solutions to the challenges within the story. In contract to *Tale-Spin* this approach doesn’t make any inferences throughout the generation of the story, it instead takes on an author-approach, manipulating all aspects of the story in parallel.

## III. CELT FABLE GENERATION SYSTEM

The purpose of the CELT (Character-Event Lore Time-line) system is to procedurally generate appropriate parables for a given simulated religion. These fables are based on the seven basic plot types [8], and are intended to reflect the values of a simulated culture.

The system is implemented as a pipeline of processes which is illustrated in Fig. 1. The first stage of the pipeline involves generating the values of the religion, establishing what is considered to be virtuous and immoral behavior. Following the

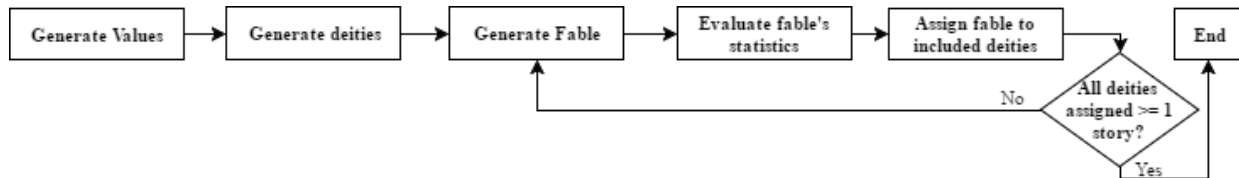


Fig. 1. Fable Generation Pipeline

generation of values, one or a number of deities is generated. For the purpose of this system, each deity is generated to reflect specific virtues or vices. Finally, the fables are created. This is achieved by procedurally traversing a story grammar that generates a time-line of fable events. The fable events are the significant elements of the story, such as a specific decision being made. The specific characters in the fable are selected based on the values that are being represented. This stage is repeated until all the deities have a minimum threshold of stories assigned to them.

#### IV. GENERATING VALUES

For the purpose of this paper an illustrative two dimensional array of values was created. The first dimension was populated with a number of commonly accepted virtues, taken from known codes of conduct and moral behavior including ‘The Seven Heavenly Virtues’ [9], ‘The Chivalric Code’ [10] and ‘Bushido’ [11], [12]. The second dimension was populated with the antonym for each virtue creating a list of vices. For example, the antonym for honest is deceitful. As there were some overlap between the behavior codes that we sampled, any duplicates or synonyms were removed. This resulted in the final array which is highlighted in the “values” section of Fig. 2. This array was then passed through two processes: Filtration and Selection.

a) **Filtration:** In the filtration stage, an integer between -1, and 1 was randomly assigned to each virtue/vice row in the array. Value combinations which were assigned 0 were removed from the list (See the “filtration” section of Fig. 2). This process results in a new list which represents the values that are specifically important to that culture. Each time the process was run, a varying number and selection of values would be kept in the system, simulating differences in values between different cultures.

b) **Value Selection:** Cultures differ based on what they consider to be morally good. This is often used as a plot device in science fiction. For this reason the value array was created to include extremes of a value belief (for example Diligent and Slothful). In the previous stage, each remaining row was assigned either a -1 or 1. If the row had been assigned a 1, the left hand cell of the row was considered to be the virtue, conversely, if the row was assigned a -1 the right hand cell was used.

The array was then randomly shuffled. This created a list of values, which were ordered from most to least important (see “selection” section of Fig. 2). It is common to have a hierarchy

within a value system as this allows for moral conflicts to be resolved through priority.

#### V. RELIGIOUS STRUCTURE

The establishment of the structure affects the number of deities that exist in the system. In the CELT system, there are two possible structures that can be generated:

- **Monotheist:** A community which worships and believes in only one deity.
- **Polytheist:** A community which worships and believes in 2 or more deities.

For the purpose of the example in this paper the structure is randomly selected. An atheist structure has also been considered for future developments.

When the religion type is Polytheist, multiple deities will be generated. In order to identify how many deities are to be generated, a number between two and a maximum threshold is randomly created.

1) **Deity Name:** The name of the deity is generated by selecting a random prefix and suffix from a name database<sup>1</sup> and concatenating them. In the polytheist structure a check is made to ensure that all deities have a unique name.

2) **Values:** Each deity is given a persona based on the values that they reflect. In the monotheist structure the sole deity is assigned all the virtues established in the previous section.

In the polytheist structure we first establish whether each generated deity represents either good or evil. Based on this assignment a random number of values are assigned to that deity to define its persona. If the deity is good, then it will bias towards virtues, and if evil, it will bias towards vices. To reduce the chance of generating deities with the same value sets, once all of the values have been assigned to the deity, the order of priority of these values is shuffled.

3) **Relationships Between Deities:** It is a common trope within polytheistic stories that some deities have adversarial relationships with each other [13]. With this in mind, the system establishes if any of the deities have been generated with opposing values. The more that two deities oppose, the more likely they are to be adversarial.

The process of identifying potential disagreements among deities is done for all of the deities once all of the deities have been initially created. This is to ensure that all of the other deities are considered when looking for potential disagreements.

<sup>1</sup><http://www.fantasynamgenerators.com/god-names.php>

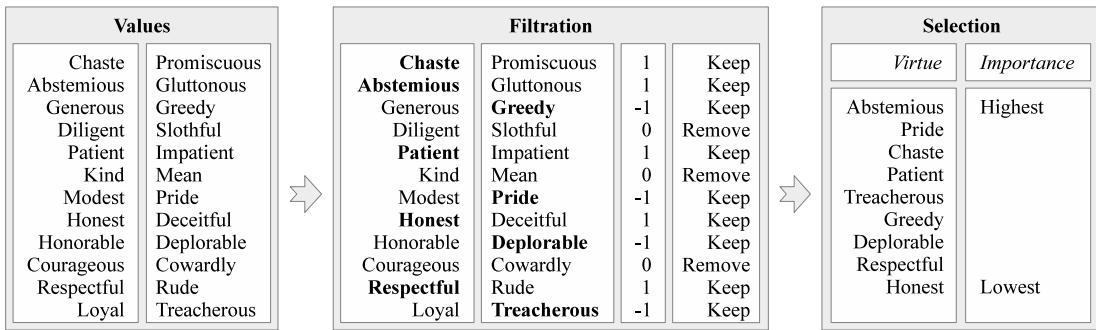


Fig. 2. The value system generation pipeline

## VI. FABLE GENERATION

The fables are generated through the use of a Context-Free Grammar, and a refinement process.

A Context-Free Grammar is a set of recursive production rules, containing terminal and non-terminal symbols which is used to generate string patterns [14]. By passing different seed values through the same production rules, different output can be produced.

$$\begin{aligned} \langle P \rangle &::= \langle X \rangle \langle X \rangle \\ \langle X \rangle &::= \text{“}y\text{”} | \text{“}z\text{”} \end{aligned}$$

Fig. 3. A basic example of a Context-Free Grammar.

Figure 3 shows an example of a Context-Free Grammar comprised of two non-terminal symbols ‘<P>’ and ‘<X>’, and two terminal symbols *y* and *z*.

With ‘<P>’ being replaced with two instances of ‘<X>’ and each instance of ‘<X>’ being replaced with either a “*y*” or a “*z*”, this grammar will result in one of the four following outputs:

- “*yy*”    • “*zz*”    • “*yz*”    • “*zy*”

CELT works by implementing the 7 basic story types [8] as context-free grammars. The grammar produces a story as a set of parallel character time lines which captures significant events in the narrative. This allows a basic story to be produced with placeholder characters which can then be refined in the second stage of the fable generation process using a state machine of possible deities and heroes.

As previously described, each placeholder character is represented as a time-line of their actions in the fable. Each action is scored based on the values it reflects, for example, defeating a foe could represent the value of heroism.

Each placeholder character’s time-line is evaluated and scored based on the values that have been reflected. If two extremes of a value system are represented, then one is subtracted from the other to produce a final score. For example, if respectful was given a score of 5, and rude was given a score of 3, then the resulting score would be respectful:2.

The values represented in each time line are then ordered from highest to lowest value. A pattern matching exercise is then conducted to evaluate which deity or hero should replace the placeholder character.

At any stage an allocation check’s condition is satisfied, the first deity to satisfy the allocation check replaces all instances of that particular placeholder character in the fable. This deity adds the fable to it’s list of fables and is then removed from a list of possible replacements for the remaining placeholder characters.

The first allocation check is to see if there are any deities available which are a ‘perfect match’ to the placeholder character being checked. A perfect match is when the deity has the same values, with the same priorities.

A partial match allocation check is undertaken if a perfect match is not found. This looks at the principle value reflected by the placeholder, and replaces with one of the deities or heroes that reflect that value strongly.

As an addition, a decision was made that deities (good or evil) should be deflatable but not killable. For this reason, a mortal hero with the values of the initially chosen deity would be generated and selected if the placeholder character died instead of the initially chosen deity.

Finally, the system can access the current state of variables as the grammar is being parsed. This enables the appropriate pluralization and punctuation to be applied before moving onto the next portion of the fable. This also enables the maintenance of story variation, and character consistency. If a hero is killed in one story, they are removed from the state machine, preventing their inclusion in future fables.

## VII. CONCLUSION

In this paper we have described a method of procedurally generating fables for a given simulated religion. Values and deities were first procedurally generated. A fable template in the form of character time lines is first generated through use of a Context-Free Grammar. This output is then refined with placeholders being replaced by appropriate characters through a value comparison process.

This paper presents early-stage work in fable generation. The next phase of this project will be to conduct a large scale in-game evaluation.

For the next development of the system we aim to replicate story variance between different non-player narrators. The timeline system will be able to be passed through individualized parsing processes when the story's are retold by different non-player agents, retelling the fables with different levels of emphasis. As the perceived personality of non player characters has been shown to influence player behavior [15], we are interested in the role that personalized narrative elements may play. Furthermore, the ultimate goal of this project is to produce believable in-game religious organizations. These generated fables will be used to contextualize the simulated cultures beliefs.

#### REFERENCES

- [1] P. Berger, "There and back again: Reuse, signifiers and consistency in created game spaces," in *Computer Games as a Sociocultural Phenomenon*. Springer, 2008, pp. 47–55.
- [2] R. Wagner, *Godwired: Religion, ritual and virtual reality*. Routledge, 2012.
- [3] M. R. Johnson, "Modelling cultural, religious and political affiliation in artificial intelligence decision-making," 2015.
- [4] G. P. Grieve and H. A. Campbell, "Studying religion in digital gaming. a critical review of an emerging field," *Online-Heidelberg Journal of Religions on the Internet*, vol. 5, 2014.
- [5] F. R. Adrados, *History of the Graeco-Latin fable*. Brill, 1999, vol. 201.
- [6] S. R. Turner, "Minstrel: a computer model of creativity and storytelling," 1993.
- [7] J. R. Meehan, "Tale-spin, an interactive program that writes stories," in *Ijcai*, vol. 77, 1977, pp. 91–98.
- [8] C. Booker, *The seven basic plots: Why we tell stories*. A&C Black, 2004.
- [9] K. Dahlsgaard, C. Peterson, and M. E. Seligman, "Shared virtue: The convergence of valued human strengths across culture and history," *Review of general psychology*, vol. 9, no. 3, p. 203, 2005.
- [10] M. H. Keen, *Chivalry*. Yale University Press, 2005.
- [11] T. Doi and J. Bester, *The anatomy of dependence*. Kodansha International Tokyo, 1973, vol. 101.
- [12] O. Benesch, "Bushido: the creation of a martial ethic in late meiji japan," Ph.D. dissertation, University of British Columbia, 2011.
- [13] I. Nielsen, "Cultic theatres and ritual drama in ancient greece," *Proceedings of the Danish Institute at Athens*, vol. 3, no. 3, pp. 107–134, 2000.
- [14] S. Ginsburg, *The Mathematical Theory of Context Free Languages.[Mit Fig.]*. McGraw-Hill Book Company, 1966.
- [15] C. J. Headleand, J. Jackson, B. Williams, L. Priday, W. J. Teahan, and L. Ap Cenydd, "How the perceived identity of a npc companion influences player behavior," in *Transactions on Computational Science XXVIII*. Springer, 2016, pp. 88–107.