

Two Families: Dynamical Policy Models in Interactive Storytelling

Fabio Zambetta
School of CS&IT, RMIT
University
GPO Box 2476V
Melbourne, Victoria, 3001
fabio@cs.rmit.edu.au

Adam Nash
School of Creative Media,
RMIT University
GPO Box 2476V
Melbourne, Victoria, 3001
adam.nash@rmit.edu.au

Paul Smith
School of Creative Media,
RMIT University
GPO Box 2476V
Melbourne, Victoria, 3001
smitty20001@hotmail.com

ABSTRACT

In this paper we introduce a mathematical model of conflict that enhances Richardson’s model of Arms Race accounting for interactive scenarios, such as the ones provided by CRPGs (Computer Role Playing Games). Such an improvement translates the model into an HCP (Hybrid Control Process). We also provide a sneak peek at the multi-disciplinary project *Two Families (A Tale of New Florence)*, set up to illustrate the applications of the model.

Two Families will result in a Neverwinter Nights 2 module featuring non-linear interactive storytelling, and a substantially different user experience based on complex political interaction between in-game factions and the overall plot.

Keywords

Game Design, Interactive Storytelling, System Dynamics.

1. INTRODUCTION

Originally a pure form of entertainment, nowadays video games are establishing their identity as a complete art medium, and supported by complex technological infrastructures they allow for diverse and heterogeneous forms of expression.

For instance, serious games are pushing the boundary of computer games much further than entertainment as their areas of application include education, training, health, and public policy [12].

Mainstream games often tend to ignore these issues, with rare exceptions such as Chris Crawford’s Eastern Front, Balance of Power, and Balance of the Planet [6].

Crawford’s career initially focused on war games that found, and still find, a large space in the video games industry (for instance tactical First Person Shooters, strategic games, etc.), but his biggest achievement probably remains Balance of Power, “an *unwar game*, a game about the prevention of war” [6]. Crawford observed that a fun game does not necessarily need a continuous display of violence, whereas it definitely needs **conflict**: War is the most violent expres-

sion of conflict, but it is by no means the only. A game on how a major conflict can be avoided would convey a very relevant message that goes beyond the realm of entertainment. Our own interest in conflict lies on one hand in exploiting dynamical models to achieve a more structured approach to the design of (story-driven) games [10], and on the other on pushing the boundary of interactive storytelling. We envisage an extended story-driven approach, where not only can players influence the game story, but also the story itself can change under the pressure of political balance. Stories and drama are generated by conflict, as detailed by Aristotle a very long time ago [8]. To this aim we designed a model that can generate political balance and conflict between factions of PCs (Player Characters) and NPCs (Non-Player Characters) in a CRPG, and we will offer a first glimpse at the results obtained during the development of a prototype implemented in the renowned NWN 2 (Neverwinter Nights 2) computer RPG [1].

Our work is rooted in Richardson’s dynamical model of Arms Race [16], devised to analyze the causes of international conflicts, initially applied by Richardson to a World War I scenario. The improvements that our modified Richardson’s faction model brings are numerous when compared to the standard faction models currently used in RPG games. First and foremost, in our approach more options are available to RPG designers that enable the creation of different types of stories that integrate policy in the plot itself and extend the usual story driven approaches. Secondly, by simply varying the basic parameters of the core model, many scenarios can be created that correspond to a different political *status quo* (e.g., tense relationships, truce, initial friendly relations, etc.) that can lead over time to different types of equilibria. Finally, players’ choices will impact the in-game political balance, but at the same time the plot will evolve under the pressure of political events, giving rise to an evolution of existing gameplay styles. The scenario we are designing to expand the current basic prototype has been dubbed **Two Families: A Tale of New Florence**. Two Families aims to create a historically plausible game world wherein stories can emerge in a unique and meaningful way. Two Families draw its primary inspiration and mood from the historical events surrounding the rise of the Italian middle class and its popular institutions during the 12th century, most notably in Florence [5] (see Figure 1).

Furthermore, it is inspired by the bloodshed undermining medieval Florence around the years 1289-1301, the famous Black vs. White Guelfs conflict whose cruel details are



Figure 1: Concept art portraying a rich bourgeois, a peasant, and an armored knight.

vividly narrated by Dante in some cantos of his masterpiece, the *Divina Commedia* [2]. Players will take the side of one of two influential families in the fight for supremacy in a fictional city, and decide whether they want to further their faction's political agenda or act as a maverick. The principal story takes place over the course of a single day, set in a fictional recreation of a medieval Tuscan city (such as Florence or San Giovanni). The Player character relives the early life of an exiled alchemist, narrated through a long conversation with his apprentice. The PC enters the medieval city during its period of utmost splendor, where development had been boosted by the trade of agricultural products from the fertile neighboring hills.

The remainder of the paper is organized as follows: Section 2 introduces dynamical models, and describes the modified Richardson's model used to compute a political balance amongst factions, Section 3 details the design principles which will constitute the foundation for our NWN 2 module, Section 4 analyzes the numerical results generated by the model, and Section 5 outlines our future work.

2. BEYOND RICHARDSON'S MODEL OF ARMS RACE

A **dynamical system** is a mathematical abstraction of a real world system governed by a fixed rule describing the time dependant change in its state (denoted by a collection of real numbers). The dynamical system is deterministic, and generally described by differential equations. Dynamical systems have been applied to diverse fields such as natural sciences, and many computing and engineering disciplines. Richardson's Arms Race model was developed by Lewis Fry Richardson to predict whether an arms race between two alliances was to usher in a conflict. The original model consists of a system of two linear differential equations that can be generalized to a multi-dimensional case

[9]. Richardson's assumptions about the model are given below:

- Arms tend to accumulate because of mutual fear;
- A society will intrinsically oppose a constant increase in arms expenditures;
- There are factors independent of expenditures which conduce to the proliferation of arms.

The equations describing the behaviour are given as:

$$\begin{aligned} \dot{x} &= ky - ax + g \\ \dot{y} &= lx - by + h \end{aligned} \quad (1)$$

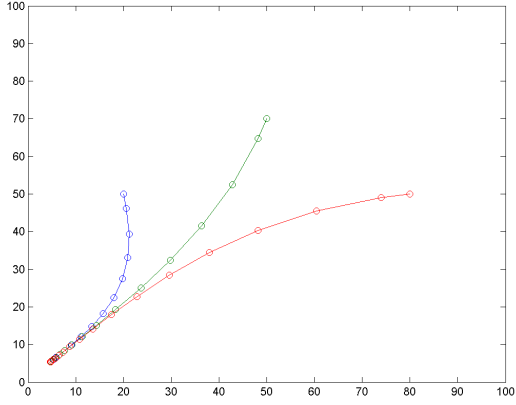
The values of \mathbf{x} and \mathbf{y} indicate the accumulation of arms for each nation. Rewriting the equations in matrix form yields, with appropriate substitutions:

$$\begin{aligned} \dot{z} &= Az + r \\ A &= \begin{pmatrix} -a & k \\ l & -b \end{pmatrix} \\ z &= \begin{pmatrix} x \\ y \end{pmatrix} \\ r &= \begin{pmatrix} g \\ h \end{pmatrix} \end{aligned} \quad (2)$$

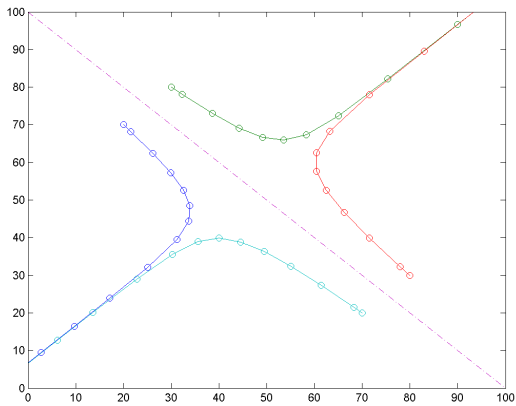
The solutions of the system of linear ODEs (Ordinary Differential Equations) [3] do not depend mostly on the relative magnitude of the constants involved, and the signs of \mathbf{g} and \mathbf{h} . The constants \mathbf{k} and \mathbf{l} are named *fear constants* (mutual fear), \mathbf{a} and \mathbf{b} are the *restraint constants* (internal opposition against arms expenditures), and as already mentioned, \mathbf{g} and \mathbf{h} are the *grievance terms* (independent factors, which can be interpreted as grievance against rivals). Note that only \mathbf{g} and \mathbf{h} are allowed to assume negative values. When analyzing the model, we will take into account the optimal lines (where the first derivatives of \mathbf{x} and \mathbf{y} equal $\mathbf{0}$), the equilibrium point $\mathbf{P}^*=(\mathbf{x}^*, \mathbf{y}^*)$ where the optimal lines intersect, and the dividing line \mathbf{L}^* for cases where equilibrium depends on the starting point. Trajectories heading towards positive infinity are said to be going towards an unlimited armament or a *runaway arms race*, whereas the ones going towards negative infinity are said to be going towards *disarmament*. There are two cases that can occur in the general assumption that $\det \mathbf{A} \neq 0$:

- All trajectories approach a stable point (stable equilibrium, see figure 2(a)).
- Trajectories depend on the initial point: They can either deviate towards positive/negative infinity or approach a stable point if they start on the dividing line (unstable equilibrium, see figure 2(b)).

If $ab > kl$, we will reach **stable equilibrium**: An equilibrium point is considered stable (for the sake of simplicity



(a) The system trajectories converge to an equilibrium point.



(b) The system trajectories depend on the initial point, and can lead to different outcomes. The dividing line separates trajectories heading towards positive or negative infinity.

Figure 2: Possible equilibria for the system.

we will consider asymptotic stability only) if the system always returns to it after small disturbances. If $ab < kl$, we will reach **unstable equilibrium**: The system moves away from the equilibrium after small disturbances. Our modified version of the model can produce alternating phases of stability and instability, yielding *variable* but *non-chaotic* results: This can give rise to a richer simulation of faction dynamics, as alliances can be broken and conflict be ceased temporarily.

Our investigation has involved three main steps: Reinterpreting the model semantics to fit our intended game context, modifying the model to produce a satisfactory representation of interaction among factions, and finally converting the model output to the input used by the Neverwinter Nights 2 faction system.

2.1 Reinterpreting the semantics of Richardson’s model

Table 1: The reinterpreted parameters semantics.

Parameters	Semantics
k	Faction X belligerence factor
l	Faction Y belligerence factor
a	Faction X pacifism factor
b	Faction Y pacifism factor
g	Friendliness of X towards Y
h	Friendliness of Y towards X

Whilst Richardson’s model was designed with a very coarse level of granularity in mind, our goal was to give designers the freedom to finely reshape a game’s story over time. We will start our analysis by naming two factions **X** and **Y**, and by reinterpreting **x** and **y** as the (greater than or equal to zero) level of **cooperation** of faction **X** and **Y** respectively. The parameters of the model are listed in Table 1: The meaning of all the parameters is not very different in our version of the model, but increasing values will lead to **cooperation** instead of **conflict**. This change aligns with the convention used by the NWN 2 faction system

The level of cooperation of each faction will lead either to a **stable equilibrium** point **P***, or **unstable equilibrium** that will drive the system towards increasing levels of *competition/cooperation* (**decreasing cooperation** indicates **competition**). Without loss of generality, we will concentrate on a restricted context of unstable equilibrium: Richardson’s model will be modified in order to obtain a rich behaviour, and at the same time cater for the interactive scenarios found in modern videogames. Also, we will assume that **g** and **h** are negative (indicating that the two factions harbour **resentment** towards each other).

2.2 Modifying Richardson’s model

The standard formulation of Richardson’s model in the unstable equilibrium case implies that the final state of the system will be dictated by the initial conditions of the system. The initial condition of the system, a point **P** in the cooperation plane depicted in figure 2(a) and 2(b), will be such that:

- If **P** lies in the half-plane *above* the dividing line **L***, then the system will be driven towards infinite **cooperation**.
- If **P** lies in the half-plane *below* the dividing line **L***, then the system will be driven towards infinite **competition**.
- If **P** lies *on the dividing line* **L***, then the system will be driven towards a stable condition of **neutrality**.

This model is uninteresting in an interactive scenario, even though it apparently contains all the main ingredients required to produce a rich behaviour. Once an ODE solver [3] starts approximating the solution of the model from its initial condition, only a single outcome in any given run will be possible (any of the three listed above, depending on the initial position of **P**). To cater for scenarios where PCs and NPCs interact with each other and the game world, we developed a *stop-and-go* version of Richardson’s model: The

solution of the system will be initially computed by our ODE solver until an external event is generated in-game. When that happens, the parameters of the model listed in Table 1 are conveniently recomputed, leading to a possible change in the equilibrium of the system, because moving the dividing line L^* alters the direction of motion of the current system trajectory. Recalling (3) we have:

$$\begin{aligned} A_{new} &= \lambda A_{old} \\ \lambda &> 0 \end{aligned} \quad (4)$$

Now we want to see how scaling A will influence the equilibrium of the system. To do so, let's first compute the equation of L^* , which is the locus of points where both the derivatives in our system will go to zero. The equation of L^* will result in:

$$\begin{aligned} \dot{x} + \dot{y} &= (ky - ax + g) + (lx - by + h) \\ &= (l - a)x + (k - b)y + (g + h) \\ &= 0 \end{aligned} \quad (5)$$

The effect of scaling on A will yield:

$$\begin{aligned} \dot{x} + \dot{y} &= \lambda(l - a)x + \lambda(k - b)y + (g + h) \\ &= 0 \end{aligned} \quad (6)$$

Thus, we will finally have:

$$(l - a)x + (k - b)y + \frac{(g + h)}{\lambda} = 0 \quad (7)$$

Three distinct cases will be possible then:

- $0 < \lambda < 1$: L^* is moved in its original upper half-plane, giving rise to a possible decrease in cooperation.
- $\lambda = 1$: The scale factor does not change A (there is no practical use for this case, though).
- $\lambda > 1$: L^* is moved in its original lower half-plane, giving rise to a possible increase in cooperation.

To test these claims, the reader needs only to take a look at Figure 3, where the case $0 < \lambda < 1$ is depicted. The dividing line is initially L_1 , and the point describing the trajectory of the system is P : The ODE solver generates increasing values of cooperation stopping at P_1 , because an external event has just occurred. At this stage, A gets scaled and as a result of that, the new dividing line becomes L_2 : The new dividing line brings P_1 in the lower half-plane, leading to decreasing values of cooperation (increasing competition). Generalizing the considerations inferred from this last example, suppose that initially $L_1 \cdot P > 0$ (increasing cooperation) and that $0 < \lambda < 1$. Then we will have three alternatives when an external events occurs:

- $L_2 \cdot P_1 > 0$: The level of cooperation keeps on increasing.
- $L_2 \cdot P_1 < 0$: The level of cooperation starts to decrease.
- $L_2 \cdot P_1 = 0$: The level of cooperation will move towards a stable value.

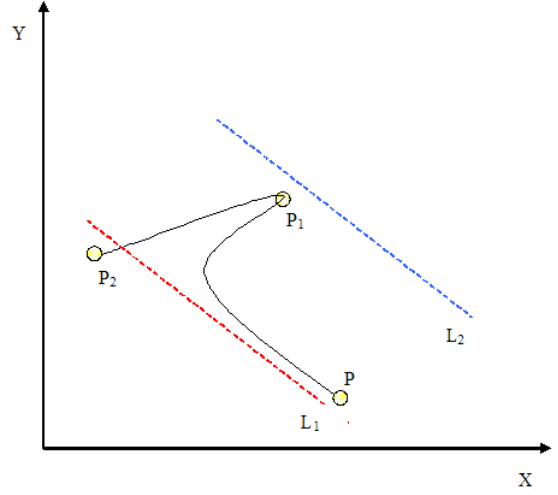


Figure 3: Effect of scaling A.

Clearly, if $L_1 \cdot P > 0$ and $\lambda > 1$ then $L_2 \cdot P_1 > 0$. Similar conclusions can be drawn in the case $L_1 \cdot P < 0$. Hence, any application using our model will need to provide a set (or a hierarchy) of events, along with a relevance level $\lambda_j, j \in \{1..M\}$ that could be either precomputed in a lookup table or generated at runtime. Obviously, all the events having $\lambda_j > 1$ will correspond to event that increase cooperation, whereas events having $0 < \lambda_j < 1$ will exacerbate competition. The effect of the λ -scaling is to change partitioning of the first quadrant, giving rise from time to time either to a bigger semi-plane for cooperation or for competition.

Finally, we want to stress that the improved Richardson's model presented here can be characterized in terms of an HCP (Hybrid Control Problem) [4]. An HCP is a system involving both continuous dynamics (usually modelled via an ODE) and controls (generally incorporated into a Finite State Machine). The system possesses *memory* affecting the vector field, which changes discontinuously in response to external *control commands* (our in-game *events*).

2.3 Converting to the Neverwinter Nights Faction System

Converting the to the NWN 2 faction system is straightforward once the proper values of cooperation have been computed: A few function calls are available in the NWN Script API to adjust the reputation of a single NPC (e.g., **AdjustReputation**). In NWN 2 faction standings assume a value in the $[0, 100]$ range per each faction: Values in $[0, 10]$ indicate competition (in NWN 2 *hostility*), whereas values in $[90, 100]$ represent cooperation (in NWN 2 *friendship*). The most straightforward conversion possible would simply use \mathbf{x} and \mathbf{y} as the faction standings for each faction: \mathbf{x} would indicate the way NPCs in faction \mathbf{X} would feel about people in faction \mathbf{Y} and vice versa, clamping the values outside the $[0, 100]$ range.

We are also evaluating the possibility of introducing a scaling factor which could represent the relative importance of each NPC in a faction: It is reasonable to expect that more hostility or friendship would be aroused by people in command positions. Hence, if we split a faction (say X for explanatory purposes) in \mathbf{N} different ranks, then we will have some

	Player	Hostile	Comm	Mercha	Defend
Hostile	0	100	0	0	0
Commoner	100	0	100	50	50
Merchant	100	0	50	100	100
Defender	100	0	100	100	100

Figure 4: The NWN 2 Faction Editor.

coefficients ϵ_i , with $i \in \{1..N\}$ such that:

$$x_{NWN} = x * \epsilon_i \quad (8)$$

3. NON-LINEAR NARRATIVE

In our model, the actions performed by the player within the game influence the level of cooperation/competition between factions, broadly at the level of the overall game narrative, and locally at the level of player-NPC interaction. In turn, our model influences the game world perceived by the player. Thus, a kind of feedback loop of influence is established between the player and the world. This can create a more sophisticated level of engagement between the player and characters within the game. In the long term, this has implications for the generation of non-linear gameplay. As an immediate application, however, we aim to employ this model to the generation of random encounters with NPCs in a module designed with the Neverwinter Nights 2 toolset.

3.1 Generating random encounters

Currently, in RPGs, random encounters with NPCs often tend to appear random or without reason to the player, since the NPC comes out of nowhere. Because our model can modulate values of cooperation/competition over time, these can be used as cues to generate random encounters. We expect that these encounters will seem more meaningful to the player. For example, if a player has joined faction X, and the level of competition between X and Y is high, the encounters will be harder and more frequent, whereas if the level of cooperation is high, the player will encounter NPCs more likely to negotiate truces or alliances. Because our model will be applied at the level of the NPCs, we expect the encounters will be able to have more depth and unpredictability than a simple competition/cooperation dichotomy. “A game, like a story, must have a conflict” [6]. Similarly, characters in a game, like characters in a story, must experience conflict to some degree. Tension drives all decisions, and decision making is the key to having NPCs that are able to interact with the PC to a degree that seems meaningful to the player in the context of the PC’s current position in the game, which is a result of the sum total of decisions the player has made to that point. In

other words, NPCs need to be a kind of local version of the overall conflict scheme of the game narrative, based on our model of competition and cooperation. We expect that each NPC can become a “web of obligations” [6] informed by the strains of arguments and alliances. In the case of Two Families, this means that each NPC needs to basically be experiencing an internal conflict similar to the conflict experienced by the PC. A layered set of options is required for each NPC, with each layer influencing the priorities of each other layer in a kind of tension network. These layers may consist of sets of simple *narrative modules* that may be combined to create pseudo complex responses. The first layer may be the faction alliance of the NPC. Another layer may be the personality disposition of the NPC. This layer can be broadly classified along the lines of dualities like loyal/treacherous, aggressive/passive, thoughtful/impetuous, selfish/selfless, kind/mean etc. All of these dualities can be seen as modes of conflict, arrived at through application of our model. Another layer may be the social position of the NPC. The higher the position, the more predictable the response should become along the lines of the broad level of competition and cooperation between factions. For example, a highly ranked soldier will be less susceptible to ideological conversion than a soldier of low rank. However, the highly ranked soldier may be susceptible to bribery on account of a selfish or material personality. Another layer may be directly influenced by the relevant skill level of the player character in any given area. For example, if the player has developed a high skill ranking in rhetoric, the player may be able to more easily influence an NPC. Conversely, the player with underdeveloped rhetoric skill may come undone at the hands of a loyal, high ranking NPC *pretending* to be treacherous.

3.2 Navigating non-linear game narrative

The narrative of the overall game, therefore, is constructed from these smaller narrative *components*, made up of nodes where a choice is possible. We will attach scripting logic to each of these nodes, so that alternative paths may be taken based on the current level of cooperation/competition. This will allow players to experience a multitude of different subplots generated as a result of the player’s interaction with characters in-game, rather than a predetermined duality.

From a practical point of view, exponential growth of non-linear structures has to be kept under control due to resource implications: A widespread game structure used to preserve non-linear design without leading to unbearable resource consumption, is a *convexity* [15]. Each of the nodes containing scripting logic will incorporate fuzzy rules [20], describing what action should be taken based on the value of fuzzy predicates. We could theoretically use classic logic to express these conditions, but fuzzy logic is very good at expressing formal properties using quasi-natural language. For instance, we might have some scripting logic like below:

```
IF cooperationX IS LOW THEN Action1
OR:
IF cooperation IS AVERAGE THEN Action2
```

The interested readers will be able to find more details on how fuzzy logic is being integrated in our system in [21]. Ultimately, the goal we have in mind is to render a new game genre viable i.e., RPS (Role-Playing Strategic). The best of both worlds, Role-Playing Games and Real Time Strategic,

is pursued here as a blending of the classic story-driven approach familiar to RPG players with strategic considerations that can substantially influence the gameplay.

3.3 A tool to create non-linear stories

A tool to create non-linear stories would allow game designers to both interactively script the game structure, and make changes to the structure itself. In order to restructure the game narrative it is foreseen that a more complex language will be needed that not only will be able to describe the choices occurring in the storyline, but also script more generic game events. The simplest (and probably most effective) idea we have been thinking about would see the fuzzy rules systems incorporated through an API exposed by a more generic games-friendly scripting language (e.g., Python, Lua, Javascript, etc.).

An example of a language used to script narrative content is given by ABL, a reactive-planning language used to script the beats (dramatic units) in the interactive drama *Façade* [11]. Even though ABL did a good job in scripting *Façade* dramatic content, it clearly falls short in terms of complexity of the scriptable actions: All in all, *Façade* is a piece of interactive drama with a quite sketchy 2D interface, and not a real game (which is what we are really interested in).

Also, people at the University of Alberta proposed an approach based on software patterns to help game designers in story building [7]: Scriptease, the tool they produced, can be used to automate to some extent the scripting of typical narrative and interaction patterns in *Neverwinter Nights*. The concept of a formal structure underpinning a story is not new at all, as it was first analyzed at large by Propp in relation to traditional Russian folktales [14]. Despite some criticism to Propp's work, it is our intention to incorporate the core of its arguments to be able to recombine essential story elements in multiple ways: This could lead to the generation of new storylines, which can then be manually refined by game designers and writers with less effort. Ideal candidates for this task are represented by evolutionary algorithms, whose power of recombination driven by an automatic or semi-automatic fitness procedure has been applied to music [13] or graphics [18] and animation [17]. Of course, building a tool to forge non-linear stories is a far-reaching goal outside the scope of our current research, but an intention in our future work.

4. EXPERIMENTAL RESULTS

The prototype we have implemented has been used to conduct functional testing of the model, and it will be integrated in *Two Families*.

The ODE solvers implemented have been based on the midpoint method (or Runge-Kutta order 2) [3], as it represents a good compromise between runtime efficiency and accuracy. The scripts solving the ODE can be hooked up to act as a proper event handler of the module *OnHeartbeat*, which is invoked by the game engine every six seconds of real-time. We have not yet implemented a full NWN 2 module integrating our model, but we are able to describe how the model behaves over time. Using a minimalist scenario, we can simulate external events that will be generated by PCs and NPCs in a real game scenario instead. We have tested a number of different parameter settings in the simulator, and we present some of the interesting results below: For a thorough analysis of our model parameters setup, the inter-

ested reader is referred to [22].

Trying to use the model in any generic NWN 2 scenario is not of much help, because the context of the story would not present the features we are seeking. On the other hand, a simulator can inform us about some of the patterns that will likely arise when implementing our own NWN 2 module in the future. This idea is clearly in line with our game design philosophy, which sees formal models and techniques as a way to produce more predictable games. Another formal methodology, which stresses the role of *game dynamics* is gaining increasing acceptance and consideration [10]. From a practical point of view, three main set of parameters seem to exert a clear influence on the system equilibrium:

- The probability distribution of external events;
- The λ -values;
- The original parameters of Richardson's model.

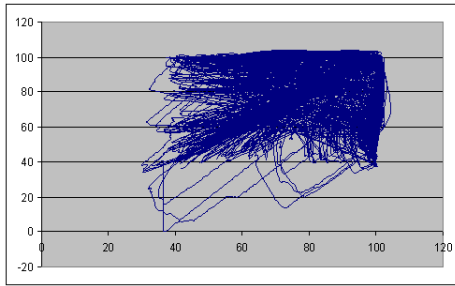
A few assumptions have been made that we summarize below:

- External events are generated both by PCs and NPCs;
- A simple probability distribution including a single class for cooperative events, and one more for competitive ones;
- The portion of the cooperation/competition plane selected has been $I = [0, 120] \times [0, 120]$. The logic of Richardson's model has been reversed to fit the NWN 2 convention (lower numbers indicate *competition*), and the model output scaled down to fit $J = [0, 100] \times [0, 100]$, which is once again dictated by NWN 2.

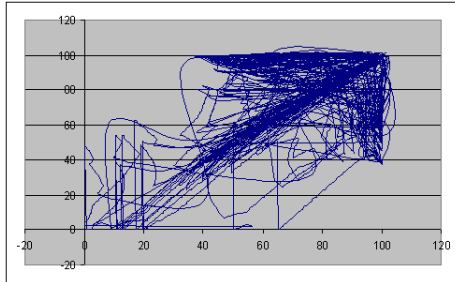
Figure 5(a) exemplifies the case of a probability distribution $P = \{0.5, 0.25, 0.25\}$, respectively the probabilities of a cooperative event, a competitive event and of no remarkable event. Clearly, the system trajectories gather in the area of cooperation, but that is only true on average because the system can also generate very distinctive levels of competition. Figure 5(b) shows the role played by the set of values $\lambda = \{0.25, 20\}$: In our current understanding of the model the probability distribution acts as a set of weights for the λ -values. Finally, the original parameters play the role described in Section 2: Their order of magnitude plays the most relevant role, and the sign of \mathbf{g} and \mathbf{h} . Reported in Table 2 are the values used to generate Figure 5(c). It should be noted that by altering the values provided here, it is very easy to generate different patterns which result to be rotated, mirrored, scaled, etc. with obvious repercussions on the gameplay.

5. CONCLUSIONS AND FUTURE WORK

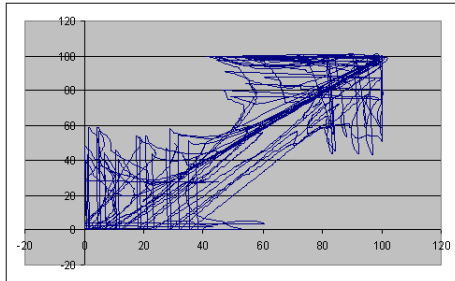
We introduced our modified version of Richardson's model that, based on a *stop-and-go* variant, provides game designers with a tool to introduce political scenarios in their story-driven games and game mods. We have discussed the formal properties of the model (that can be more formally regarded as a Hybrid Control Problem), and analyzed some stochastic patterns generated by the factions behaviours that are



(a) Arrow-shaped system trajectories.



(b) Arrow-shaped trajectories with higher variance.



(c) Double arrow-shaped system trajectories.

Figure 5: Some typical patterns that emerge when simulating the system behaviour.

likely to emerge during the production of **Two Families**, our Neverwinter Nights 2 module.

Two Families will incorporate both random encounters (as described in this paper), and a non-linear story that can be navigated via the output generated by our model. Having the opportunity to test the model in a real-world scenario will likely affect its current formulation. Along with a deeper investigation of the formal properties of the model, substantial efforts will be put in the custom content creation process. The NWN2 game includes 3D models, textures, animation data and sound. Whilst some of these assets may be of practical use, many will certainly need to be extended or redeveloped. We will be dispensing with weapons of fancy and settle on the modification of existing NWN2 weapons or custom creation of in-game weaponry. Our inspiration will be drawn from the fascinating array of historically plausible weapons such as those codified in [19].

6. REFERENCES

- [1] Bioware's Neverwinter Nights web site.
<http://nwn.bioware.com>.

Table 2: The value of the original parameters used in the experiments.

Parameters	Values
k	15
l	15
a	1
b	1
g	-400
h	-600

- [2] D. Alighieri. *La Divina Commedia*. Mondadori, Milano, 2005.
- [3] W. Boyce and R. DiPrima. *Elementary Differential Equations and Boundary Value Problems*. John Wiley & Sons, Hoboken, 2004.
- [4] M. Branicky, V. Borkar, and S. Mitter. A unified framework for hybrid control: Background, model, and theory. In *Proceedings of the 11th International Conference on Analysis and Optimization of Systems*, June 1994.
- [5] F. Cesati. *The Medici Story of a European Dynasty*. La Mandragora, Firenze, 1999.
- [6] C. Crawford. *Chris Crawford on Game Design*. New Riders, Berkeley, 2003.
- [7] M. Cutumisu, C. Onuczko, D. Szafron, J. Schaeffer, M. McNaughton, T. Roy, J. Siegel, and M. Carbonaro. Evaluating pattern catalogs - the computer games experience. In *Proceedings of the 28th International Conference on Software Engineering (ICSE '06)*, May 2006.
- [8] G. Freytag. *Freytag's Technique of the Drama*. Griggs & company, Boston, 1995.
- [9] J. Goulet. Richardson's arms model and arms control. In *Proceedings of the SIAM Conference on Discrete Math and Its Application*, June 1983.
- [10] R. Hunicke, M. LeBlanc, and R. Zubek. MDA: A formal approach to game design and game research. In *Proceedings of the AAAI-04 Workshop on Challenges in Game AI*, July 2004. Available online at <http://www.cs.northwestern.edu/hunicke/pubs/MDA.pdf>.
- [11] M. Mateas and A. Stern. Structuring content in the Facade interactive drama architecture. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, June 2005.
- [12] D. Michael and S. Chen. *Serious Games. Games That Educate, Train, and Inform*. Osborne/Mc Graw, Boston, 2005.
- [13] E. R. Miranda and A. Biles, editors. *Evolutionary Computer Music*. Springer-Verlag, Nwy York, 2007.
- [14] V. Propp. *Morphology of the Folktale*. University of Texas Press, Austin, 1968.

- [15] S. Rabin, editor. *Introduction to Game Development*, chapter 2.1, pages 71–97. Charles River Media, Hingham, 2005.
- [16] L. Richardson. *Arms and Insecurity*. Boxwood Press, Pittsburgh, 1960.
- [17] K. Sims. Artificial evolution for computer graphics. In *Proceedings of the SIGGRAPH '91 Conference*, July 1991.
- [18] K. Sims. Evolving virtual creatures. In *Proceedings of the SIGGRAPH '94 Conference*, July 1994.
- [19] G. Stones. *A Glossary of the construction, Decoration and Use of Arms and Armor in All Countries and in All Time*. Dover Publications, New York, 1999.
- [20] L. Zadeh. Fuzzy sets. *Journal of Information and Control*, 8:338–353, 1965.
- [21] F. Zambetta. *Computational Intelligence in Multimedia Processing: Recent Advances*, chapter Modelling Interactive Non-linear Stories. Springer-Verlag, Berlin, 2007. To appear.
- [22] F. Zambetta. *AI Game Programming Wisdom 4*, chapter Implementing Story-Driven Games with the Aid of Dynamical Policy Models. Charles River Media, Hingham, 2008. To appear.