# Emerging archetypes in massive Artificial Societies for literary purposes using Genetic Algorithms

R.H. García-Ortega[1], P. García-Sánchez[2] and J.J. Merelo[2]

[1] Fundación I+D del Software Libre, Granada, Spain
[2] Dept. of Computer Architecture and Technology, University of Granada, Spain
rhgarcia@fidesol.org, pgarcia@atc.ugr.es, jjmerelo@atc.ugr.es

**Abstract.** The creation of fictional stories is a very complex task that implies a creative process where the author has to mix characters, conflicts and plots. This work presents a simulated environment with hundreds of characters that allows the study of coherent and interesting literary archetypes (or behaviours), plots and subplots. We will use this environment to perform a study about the number of profiles (parameters that define the personality of a character) needed to create two emergent groups of archetypes: "natality control" and "revenge". A Genetic Algorithm will be used to find the fittest number of profiles and parameter configuration that enables the existence of the desired archetypes (played by the characters without their express knowledge). The results show that parametrizing this complex system is possible and that these kind of archetypes can emerge in the given environment.

## 1 Introduction

In video games, NPCs (Non Player Characters) are a type of characters that live in the game world to provide a more inmersive experience. Modern RPGs (Role Playing Games), such as The Witcher™or Skyrim™count with hundreds of NPC characters. The effort to create a good interactive fiction script is directly proportional to the number of these characters. That is the reason this kind of agents usually counts with limited behaviours, such as wandering in the villages, selling groceries or guarding the cities. Also, they usually offer scripted conversations, for example, for buy and sell objects to the player. In other cases they interact with the player depending of the player's behaviour: for example, if the player steals something a city guard would attack him. However, these characters do not interact among them, only with the player, and their activities are only guided with this purpose. In a world with such a number of characters, their collective interactions could improve the gaming experience, leading to a richer and more inmersive world. For example, hungry inhabitants could become thieves, guards could pursuit the thieves, villagers could fell in love with others or different war alliances could emerge.

These facts have motivated us to develop a multi-agent system called MADE (Massive Artificial Drama Engine) to model a self-organized virtual world where their elements influences each other, following a cause-effect behaviours in a coherent manner. This system needs to be a suitable environment for the plot of a specific literary work, being also interesting for the player/spectator. A set of probabilities and states are associated to agents' actions, and these probabilities are optimized by means of an Evolutionary Algorithm (EA) to match with a specific literary archetype, defined by the fiction creator. The archetypes are behaviours and patterns universally accepted and present in the collective imaginary [?], that allows empathize with the characters and immerse yourself in the story (for example, the well-known "hero" archetype).

In this work, several experiments have been carried out to answer following questions:

- Is it possible to model a virtual environment inhabited by hundred of characters with interesting auto-generated behaviour based on literary archetypes?
- Could the personality of the agents be parametrized to obtain different behaviours?
- How many profiles (groups of parameters that define a personality) are necessary to generate emergent quality sub-plots?
- Could a Genetic Algorithm be used to find the fittest parameter values that allow the creation of this kind of sub-plots?

In this paper we prove that EAs, together with a proper design of literary patterns, can be used to find the parameters that promote the generation of drama plots and sub-plots in a multi agent based environment.

The rest of the work is structured as follows: after the state of the art, the developed system is presented in Section 3. Then, the experiments conduced with the EA are showed (Section **??** 5). Finally, conclusions and future works are discussed.

## 2 State of the art

Auto-generated interactive fiction research is mainly focused in methods to create the process of a story generation [?]. Story generation can be divided in two areas: interactive and non-interactive. In the first area, and according to [?], an Interactive Drama is defined in a virtual world where the user has freedom to interact with the NPCs and objects in a dramatically interesting experience, different in each execution, and adapted to the interactions of the user.

The generation of interactive dramas can also be based in script structure [?], where each possibility in the story must be previously defined, so there is a limited number of possible plot combinations. There exist other techniques, not based in plot structure, such as...

On the other side, in non-interactive plot generation systems the user does not take control as the protagonist. For example, in the system presented by Pizzi et al. [?] the user can interact with the characters, changing their emotions, but making the user an spectator, rather as an actor.

As opposed to those concepts, MADE is focused in Artificial Non Interactive Drama, because its aim is the massive generation of plots for secondary characters, to provide a context for the writer and the player to perceive a virtual world as coherent, detailed and enriched. The story generation (that is, the narrative) is not addressed by MADE, but it has been studied in the systems presents in the survey by Arinbjarnar et al. in [**?**].

Previous works define the plot as an emergence for the behaviour of the agents that follow a set of rules. In MADE, the agents' behaviour is product of its personality and the environment. That is, the agents does not follow the plot, but they generate the plot itself.

Futhermore, the previous works generate plots in worlds with a limited number of characters. This restriction does not exist in MADE, where the number of characters to create is unlimited.

Following the ideas of the work of Epstein and Axtell [**?**] an environment based in *Sugarscape* has been developed, with concepts such as food, metabolism and vision. This environment, uses the elements by Gershenson [**?**]: a virtual world, agents who born, grow, interact, reproduce and dead; resources (food), mediators, and relations of rivalry (friction) and cooperation (synergy). The actions of these agents are parametrized according the work of Nairat [**?**], based in the existence of profiles, and will be mapped into a chromosome to be used in a Genetic Algorithm (GA). This system allows the definition of behaviour patterns (or archetypes, and using a multi-objective fitness function to measure the presence of the desired archetypes.

## 3 The MADE Environment

The MADE enviroment is a virtual place where different agents play their artificial lives. Its functions are:

- **Create an initial set of agents:** MADE environment initializes a set of just born orphan agents, each with a profile sequentially assigned. These agents must compete or collaborate in order to survive.
- **Place agents in the map:** the environment has a squared map, formed by cells that can be occupied by one (and only one) agent. The environment allows the agents to discover and interact with other agents in the neighbourhood.
- **Start and control the time:** after the creation of the initial set of agents, the MADE environment starts the timer, day by day until a maximum date is reached.
- **Execute each agent during a time unit (a day):** In each iteration the list of agents is randomly reordered, and after that following the new order, each agent perform an iteration of its life-cycle, and the dead agents are removed from the grid.
- **Perform as an external agent that changes th environment:** In each iteration in the MADE environment food rations are placed in random cells.

An agent only can eat if it is over a cell with a ration, so agents could move the other forcibly.
- **Offer services to the agents:** MADE environment allow the agents to check which closer cells have food, are occupied, who agents are in a near position or which positions can be occupied.
- **Decide the profile of the agents:** MADE allows the existence of different agent profiles. A profile is a set of characteristics which governs the agent's behaviour.

The MADE environment can be configured by using the following parameters:

- Number of agents initially placed
- Map square grid dimension
- Number of rations randomly placed in the grid each day
- Duration (in virtual days) of the environment execution

### 3.1 MADE Agent

A MADE Agent lives in a MADE Environment, occupies a cell in the grid, moves around looking for food or mate and interacts with other agents.

A very simple agent has been designed for this study: a virtual rat. We have modelled 4 states for it (be alive, be hungry, look for mate and be pregnant), 7 actions (move, eat, attack, defend, escape, find mate and have offspring) and parameters that define its characteristics and probabilities to perform actions depending on the state. It's important to remark that no "feelings" and no memory have been modelled in the MADE agent.

Figure 1 illustrates the nature of a MADE Agent.

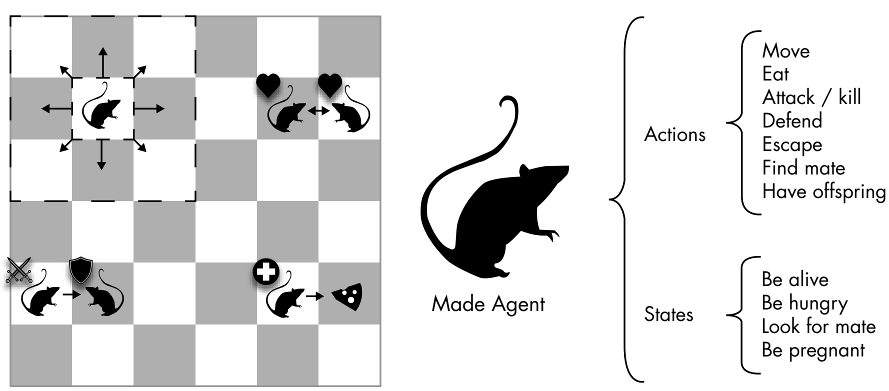


Fig. 1: Actions and states modelled in the MADE Agent.

The life cycle of a MADE Agent illustrated in Figure **??** represents on day in its life. Every decision made by the agent is based on its state and its characteristics (probabilities to perform different actions).

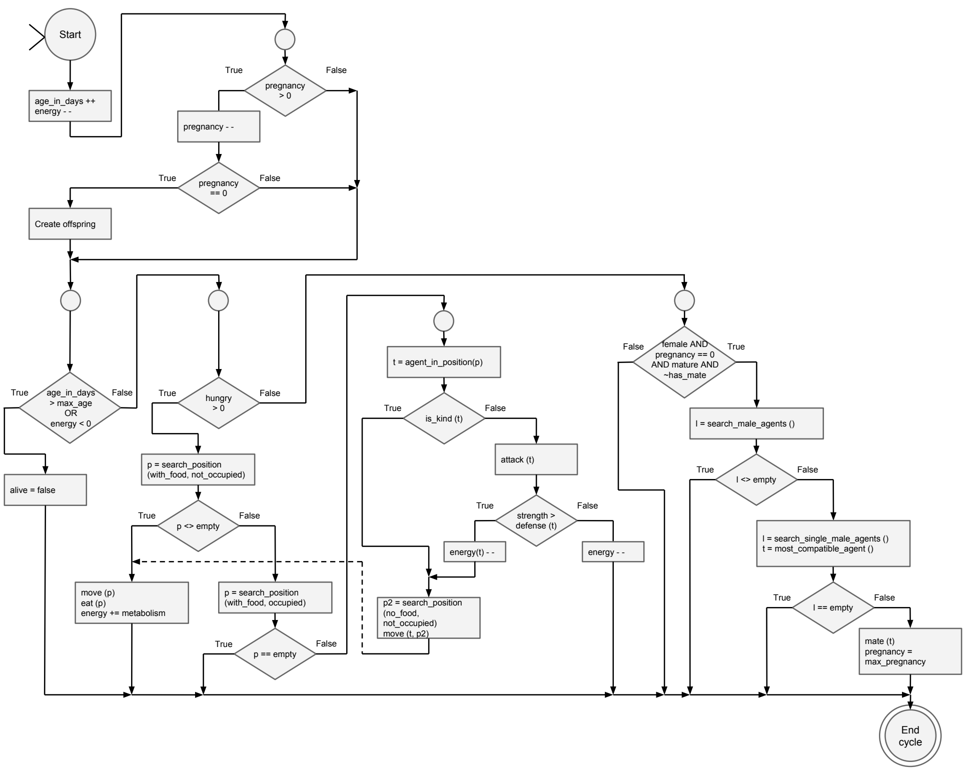


Fig. 2: MADE Agent's life cycle

## 3.2 MADE Execution

Every agent has a diary that stores all the relevant events in its life in a simple format. Each line of the log indicates the day, the event in a short readable format and some extra information. Every agent's diary in the MADE Environment in coherent with all others' diaries. Many plots are being created, with a structured format that can be read by game engines or natural language processors, but, given the simplicity of the modelled agents, the stories could be seen by the reader as non-interesting. In the next section we propose a method based on EA to let the author of a story promote different behaviours in the MADE agents that could be seen as literary archetypes (usually associated with human feelings and high level cognitive and memory abilities).

## 4 Experimental set-up

The MADE Agent is created using 12 parameters that define its base features and probabilities to make the decisions presented in the life cycle. The execution of an agent is dynamic, and depends on the internal probabilities and states but also on the neighbourhood, and the map configuration. Even so, we can say that these initial parameters define in some way the possible situations where the agent could be involved.

Thanks to the agents' diaries, we can know every event (internal and external) of their lives, and evaluate their interest. In this work, we have implemented a method based in regular expressions with backreferences. The proposed technique puts annotations in every agent whose diary matches a complex regular expression able to find emerging high level behaviours, not implemented in the life cycle.

In this proposal, the parameters used to define an agent will be mapped to a chromosome, and a Genetic Algorithm will be used to evolve the solution. The fitness function will be expressed in terms of:

- **Regular expressions applied to the diary of each agent in the environment:** An agent is tagged when a regular expression matches its diary.
- **A numeric function over the number of tagged agents for each archetype:** the fitness of the solution is increments with the returning value.

The concept of agent profile in introduced to assign different parameters to different agents. If only one profile is used, all the agents will be created with the same parameters, evolved by the Genetic Algorithm. If more profiles are used, they will be assigned to the agents in order of appearance in a loop. Our assumption is that some archetypes could emerge using one profile and other will need more (those that require two roles clearly differentiated). The more profiles are used, the harder the solution converge, since the number of alleles of the chromosome are multiplied by the number of profiles.

This work will perform a study about the number of profiles (parameters that define the personality of a character) needed to create two emergent groups of archetypes: "natality control" and "revenge". The Genetic algorithm will use the parameters described in the Table **??**.

### 4.1 Experiment 1: Natality control archetype

This experiment agglutinate different sample archetypes where many factors must be taken into account. It will be used to answer the following questions:

Given a map size, an initial population and the rate of food available per day, what values are optimal to:

Table 1: Parametrization of the Genetic Algorithm

| Parameter | Value |
|---|---|
| Codification | 12 alleles per profile |
| Fitness function | Average of 10 executions. Patterns are defined in sections 4.1 and 4.2 |
| Natural selector | Original Rate: 0.9 |
| Crossover operator | Rate: 35% |
| Mutation operator | Desired Rate: 12 |
| Stop condition | 100 executions |
| Generations | 30 |
| Population size | 30 |

- Assure that, after 1000 virtual days, the alive population will be the 30% of the total population.
- Emerge the *downtrodden* archetype in the 15% of the population. An agent will be considered as a *downtrodden* or *defender* if it has been attacked at least two times and has defended the position.
- Emerge the *warrior* archetype in the 15% of the population. An agent will be considered as a *warrior* if it has satisfactory attacked at least five times.
- Emerge the *helpless* archetype in the 15% of the population. An agent will be considered as a *helpless* if it has been attacked at least ten times and hasn't defended the position.
- Emerge the *bad warrior* archetype in the 15% of the population. An agent will be considered as a *bad warrior* if it has unsatisfactory attacked at least ten times.

### 4.2 Experiment 2: Revenge archetype

This experiment will be used for a more complex memory based behaviour:

Given a map size, an initial population and the rate of food available per day, what values are optimal to:

- Emerge the *revenge* archetype in as many agents as possible after 1000 days. An agent (a) will be considered as a *revenger* if it has been attacked by other agent (b) and after that, in a moment in its life, it has satisfactory attacked the agent b, in revenge.

## 5 Results and discussion

### 5.1 Experiment 1: Natality control archetype

Figure **??** shows the average of the best fitness and the average of the average fitness for 30 executions of each configuration. Every test has been proved from

1 to 5 profiles. The evolution of the best fitness for each number of profiles is shown in Figure **??**.

Table 2: Results for 30 executions of each configuration for 1 to 5 profiles

| Number of profiles | Best fitness (average) * | Standard deviation | Average fitness ** | Standard deviation |
|---|---|---|---|---|
| 1 | a.aa | a.aa | a.aa | a.aa |
| 2 | a.aa | a.aa | a.aa | a.aa |
| 3 | a.aa | a.aa | a.aa | a.aa |
| 4 | a.aa | a.aa | a.aa | a.aa |
| 5 | a.aa | a.aa | a.aa | a.aa |

* Average of the best fitness in the last generation of each execution
** Average of the average fitness in the last generation of each execution

### 5.2 Experiment 2: Revenge archetype

Figure **??** shows the average of the best fitness and the average of the average fitness for 30 executions of each configurations. Every test has been proved from 1 to 5 profiles. The evolution of the best fitness for each number of profiles is shown in Figure **??**.

Table 3: Results for 30 executions of each configuration for 1 to 5 profiles

| Number of profiles | Best fitness (average) * | Standard deviation | Average fitness ** | Standard deviation |
|---|---|---|---|---|
| 1 | a.aa | a.aa | a.aa | a.aa |
| 2 | a.aa | a.aa | a.aa | a.aa |
| 3 | a.aa | a.aa | a.aa | a.aa |
| 4 | a.aa | a.aa | a.aa | a.aa |
| 5 | a.aa | a.aa | a.aa | a.aa |

* Average of the best fitness in the last generation of each execution
** Average of the average fitness in the last generation of each execution

## 6 Conclusions

## Acknowledgements

This work has been supported in part by FPU research grant AP2009-2942 and projects EvOrq (TIC-3903), CANUBE (CEI2013-P-14) and ANYSELF (TIN2011-28627-C04-02).