Imperial College of Science,

Technology and Medicine

(University of London)

Department of Computing

Dynamics of Human Behaviour

by

Camille Guérillot

Submitted in partial fulfilment

of the requirements for the MSc

Degree in Advanced Computing of the

University of London and for the

Diploma of Imperial College of

Science, Technology and Medicine.

September 2005

Abstract

Human social behaviour is one of the most intricate phenomena studied by science. If we take a human society as a system and we try to analyse it, its high complexity prevents us from deducing its evolution even in a near future. The usual method to understand a system is to divide it into small parts, then to understand the behaviour of these small parts, and finally to deduce the whole behaviour from what we know about the small parts. However, the different parts of a society (that is to say the human beings, but also their environment, their history, their economy, etc…) are so interwoven that we are compelled to look at a human society as a whole in order to understand its dynamics. For example, emergent phenomena such as culture change, social organisation or even the integration of social norms are the effects of a group dynamics: it is thus impossible to predict their evolution by analysing an isolated human.

Therefore, social theorists need theoretical and computational tools developed to study emergent phenomena in complex systems in order to develop a unified body of knowledge which would help us to shed light on the long lasting question of social humanity. With this intent in mind, I tried to develop a tool which could be used in order to understand better the phenomenon of social norms. This software is a model of a human society developed in the form of a Multi Agent System (MAS). Based on the Talk or Fight system, this MAS is a simple abstraction of a society. However, the results obtained even with this quite crude representation are sometimes quite near from what can be observed by sociologists working with real people.

In this paper, I wish to explain the model which was used in order to run the simulations as well as present the results obtains.

Acknowledgements

I would like to thanks Professor Abbas Edalat (who supervised this project) for his support. Thanks also to Kelvin Au and Rishi Aggarwal for their help at the beginning of the project and to Ali Khanban for his precious advices.

Contents

Introduction ____ *4*

Background ____ *6*

1. Definition of a complex system ____ **6**

2. Examples of complex systems ____ **8**

3. Properties of complex systems ____ **9**

3.1. Non determinism ____ 9
3.2. Limited functional decomposability ____ 9
3.3. Distributed character of information ____ 10
3.4. Short-range non-linear relationships ____ 11
3.5. Emergence ____ 11

Chapter 1: Description of the model ____ *13*

Introduction ____ **13**

1. The model in a nutshell ____ **14**

2. The frameworks ____ **16**

2.1. The city ____ 16
2.2. Schools and offices ____ 16
2.3. The households ____ 17

3. The agents ____ **18**

3.1. Parameters ____ 18
3.2. The contact list ____ 21
3.3. From birth to death, the life of an agent ____ 22
3.4. Transmission of norms ____ 25

4. Interactions ____ **26**

4.1. Sum up of the process ____ 26
4.2. First step: selection of a partner ____ 27
4.3. Second step: selection of an interaction ____ 27
4.4. Interlude: modification of a parameter ____ 29
4.5. Third step: applying the interaction itself ____ 32
4.5.1. Talk ____ 32
4.5.2. Fight ____ 33
4.5.3. Give ____ 34
4.5.4. Steal ____ 35
4.5.5. Seduce ____ 35
4.6. Fourth step: feedback and adjustment of the norm ____ 36
4.6.1 Personal payoff ____ 36
4.6.1 External feedback ____ 37
4.6.3 Adjustment of the norm ____ 38

5. Human-machine interface and visualisation of the results ____ **39**

5.1. Overview ____ 39
5.2. Controls ____ 40
5.3. Visualisation tools ____ 41
5.4. Settings ____ 43

Remarks and conclusion ____ **44**

Chapter 2: Autonomous society ______ *47*

Introduction ______ **47**

1. Patterns obtained with default settings ______ **48**

1.1. Awareness and Integration ______ 48
1.2. Sociability/Misanthropy and Pacifism/Aggression ______ 51
1.3. Generosity/Selfishness and Honesty/Dishonesty ______ 54
1.4. Flirtation/Puritanism ______ 57
1.5. Overview ______ 59

2. The weight of norms in the decision making process ______ **59**

3. Influence of external feedback ______ **61**

4. Transmission of norms ______ **62**

Remarks and conclusion ______ **65**

Chapter 3: Society with external influence ______ *67*

Introduction ______ **67**

1. Additions to the model ______ **68**

1.1. Teachers ______ 68
1.2. Mass trauma events ______ 69

2. Patterns obtained with teachers ______ **70**

2.1. Same teachers in all districts ______ 70
2.2. "Clash" of cultures ______ 71
2.2.1. Prosocial behaviour versus antisocial behaviour ______ 72
2.2.2. Other experiments ______ 77

3. Influence of mass trauma events ______ **78**

3.1. Annoyance or cataclysm, the effect of intensity ______ 78
3.2. Resistance to trauma through self organisation ______ 79
3.3. Influence of the parameters ______ 80

Remarks and conclusion ______ **81**

Conclusion and future works ______ *83*

Bibliography ______ *85*

Annex ______ *87*

Introduction

Human social behaviour and culture change are amongst the most complex phenomena studied by science. This results from the large number of interacting entities within a society or the different mechanisms underlying human social interaction (amongst other factors). Thus, it is impossible to find an analytical solution to the "equations" of social dynamics. Dividing a society into small fragments and analysing them would not help us to predict its evolution. Supposing that it could be possible, even the complete analysis of an isolated human being would not be enough to understand social behaviour. When we analyse a society, we are compelled to look at it as a whole.

Thus, one way to obtain results and to deal with complexity when we want to analyse a society is to run computerised simulations. The Talk or Fight system is an example of such a simulation. In this system, each human being is represented by an agent which is defined by 2 parameters: its awareness and its integration. A positive awareness means a good level of education while a negative one will represent an indoctrinated person; a positive integration will represent a person who is confident in his emotions, while a negative integration will be used for a traumatised person. The agents can only interact in two ways: they can talk of fight. Each interaction will have different effects on the awareness and integration of both agents.

If we run the Talk or Fight system for some thousands iterations, some patterns emerge: we can differentiate regions of talkative people and regions of aggressive ones, each of them trying to expand its influence. Thus, we obtain the emergence of two "cultures" in our virtual society, beginning with only two parameters and two simple interactions.

The objective of my work was to expand the Talk or Fight system in order to create a tool which could help us to study the phenomenon of social behavioural norms. By increasing the complexity of the initial model, I tried to understand the factors involved in the emergence and the transmission of behavioural norms. When I created this tool, I did not want it to be an ultra realistic model. On the contrary, the idea was to keep the initial simplicity of the Talk or Fight model, increasing its complexity just enough in order to have relevant results but keeping the number of parameters low in order to exploit easily these results. Therefore, I do not pretend having developed a tool which can foresee the evolution of a society. However, the model obtained is rich enough to give us results which are quite near from what can be observable in real societies.

I will begin this paper by giving some background information about complex systems. Then I will be able to present the model I chose to represent my societies. The two final parts of this paper will expose the results obtained with two different kinds of simulations: the autonomous ones, where the society evolves all by itself, and the ones involving external influence, where the user is able to simulate phenomenon as indoctrination or mass trauma.

Background

The system I have developed is a social system, that is to say a system formed by a society of agents. I have already said that the behaviour of such systems was unpredictable: the number of elements in that system and the nature of the interactions between them prevent us from predicting its evolution. We could say the same thing about ecological systems, economical ones, the universe, the brain, etc… All these systems are formed by small parts interacting in non-linear ways which makes their behaviour unpredictable. They are called complex systems.

The notion of complex systems was born around 1890 with H. Poincaré. In 1887, the king of Norway and Sweden, Oscar II, initiated a mathematical competition to celebrate his 60th birthday in 1889. Poincaré decided to work on the equations used to predict the trajectory of planets. He showed that it was mathematically impossible to find a solution to these equations even for a system as simple as three planets interacting in a non-linear fashion. This work is stayed in the literature with the name of 3-Body problem. For the first time, the idea that a totally causal system could have unpredictable results was proposed. Even if he had failed to solve this problem, Poincaré received the prize. Later, the 3-Body problem was ignored and put aside. It is only far later, with the appearance of computerized simulations, that the scientific community realised that Poincaré had predicted chaotic motion and complex systems.

1. Definition of a complex system

"Formally, a system starts to have a complex behaviour the moment it consists of parts interacting in a non linear fashion" (Pavard – Dugdale). This definition, however correct, is quite insufficient to understand clearly what a complex system is, but the following quotes give us a flavour:

"…a system that is complex, in the sense that a great many independent agents are interacting with each other in a great many ways." (Waldrop, 1993)

"…to understand the behaviour of a complex system, we must understand not only the behaviour of the parts but how they act together to form the whole." (Bar-Yam, 1997)

"…you generally find that the basic components and the basic laws are quite simple; the complexity arises because you have a great many of these simple components interacting simultaneously. The complexity is actually in the organization–the myriad possible ways that the components of the system can interact." (Stephen Wolfram, quoted in Waldrop, 1993)

"A complex system is a system for which it is difficult, if not impossible to restrict its description to a limited number of parameters or characterising variables without losing its essential global functional properties." (Pavard, 2000)

"Complex adaptive systems consist of a number of components, or agents, that interact with each other's behaviour in order to improve their behaviour and thus the behaviour of the system they comprise." (Stacey, 1996)

"…the complex whole may exhibit properties that are not readily explained by understanding its parts. The complex whole, in a completely nonmystical sense, can often exhibit collective properties, "emergent" features that are lawful in their own right." (Kauffman, 1996)

"The task of formulating theory for CAS [complex adaptive systems] is more than usually difficult because the behaviour of a whole CAS is more than a simple sum of the behaviours of its parts; CAS abound in nonlinearities…" (Holland, 1995)

"…complexity is not located at a specific, identifiable site in a system. Because complexity results from the interaction between the components of a system, complexity is manifested at the level of the system itself. There is neither something at a level below (a source), nor at a level above (a meta-description), capable of capturing the essence of complexity." (Cilliers, 1998)

It is important to make the difference between a complicated system (such as a plane or a computer) and a complex system (such as ecological or economics systems). The former are composed of many functionally distinct parts but are in fact predictable whereas the latter interact non-linearly with their environment and their components have properties of self-organisation which makes them non-predictable beyond a certain temporal window.

2. Examples of complex systems

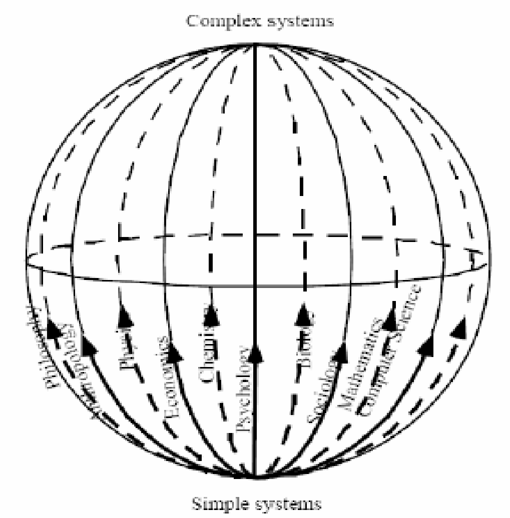
Examples of actual systems that are complex systems:

- Governments
- Families
- The human body
- A person, from a psychosocial point of view
- The brain
- The eco-system of the world
- The weather
- A corporation
- The universe

Examples of artificial complex systems:

- Cellular automata
- Social simulation
- Artificial life

Those examples belong to different fields: physics, chemistry, biology, mathematics, computer science, sociology, economics, etc… The following illustration represents this state of fact:



There are two approaches to the study of a particular complex system: from the bottom of the sphere, using our knowledge of simple systems, or from the top of the sphere, considering common properties of complex systems.

3. Properties of complex systems

3.1. Non determinism

As shown by H. Poincaré with the 3-Body problem, even a totally causal system can have a non deterministic behaviour. For example, if you study the iterative map:

$$\begin{cases} u_{n+1} = a \times u_n (1 - u_n) \\ -1 < u_0 < +1 \end{cases}$$

you will find chaotic behaviour for a > 3,6. This chaotic behaviour appears because of the non-linearities existing in the system.

That is why a complex system will be fundamentally non deterministic. Considering that a complex system "consists of parts interacting in a non linear fashion" and considering the high number of these interactions inside such a system, we can understand that chaotic behaviour quickly emerges in a complex system. This chaotic aspect makes any complex system unpredictable and, therefore, non deterministic.

3.2. Limited functional decomposability

According to the traditional analytical approach, the behaviour of a system can be guessed by knowledge of the functions of its sub-components. For example, if we know the function of each element of a car (wheels, engine, breaks, etc…) we are able to deduce the final behaviour of the car. However, in the case of a complex system, we cannot divide it into well-defined functional components. Indeed, each small part of the system can change its behaviour and evolve due to an internal or external change: the constituent functions of the system are fluctuating. Self-organisation mechanisms as well as the interactions with the environment prevent us from decomposing the system into functionally stable parts.

This property is very interesting since complex systems can reorganise rapidly their internal structure depending on the context (internal or external events). This modification of behaviour can occur without having been programmed at a central level.

If we think about the human brain, this notion can be easier to understand. It is well none that a great part of our neurones are usually never used. However, these unused neurones can be very useful in case of brain damage. It has been observed on many patients having received brain damaged that the brain is able to reorganise its internal structure and to activate those "idle" neurones in order to replace the damaged ones. By changing its structure, the brain is able to function again, even after an important damage.

3.3. Distributed character of information

A system is said to be distributed if its resources are distributed on various sites. In complex systems, information is distributed. Indeed, elements in the system cannot "know" what is happening in the system as a whole: they are not sufficient to describe it. Otherwise, all the complexity of the system would have to be present in that element, and since complexity arises from the relationships between the elements, this is impossible. A corollary of that is that no element in a complex system can control this system.

As their resources are distributed, complex systems are very robust. It is impossible to localize precisely the information since it is more or less uniformly distributed between all the objects of the system. So even with a small part damaged, a complex system is able to behave correctly, having lost only a small part of the information. This is particularly true in a society. If a part of a society in destroyed, for example because of a natural disaster, a trauma will appear in the society but it will manage to reorganize itself in order to overcome this trauma and to get back to a stable situation.

3.4. Short-range non-linear relationships

The relationships in a complex system are usually short-range: information is normally received from near neighbours. As there are a lot of channels of communication within a complex system, this information will be transmitted to the whole system. However, because of the number of elements involved in the transmission of the information, this information is likely to be modified in the way.

Moreover, the parts interact in a non-linear way, so a small stimulus may cause a large effect or no effect at all. Very often, the non linear relationships contain feedback loops: the effects of an element's actions are fed back to the element, and this will affect the way the element behaves in the future.

3.5. Emergence

Emergence is maybe the most important property for a complex system. A property is emergent when it cannot be predicted from the behaviour of the components of the system. Emergence is thus the process of deriving some new and coherent patterns, structures or properties in a complex system, and these patterns are observable at a "macroscopic" scale (that is to say at a scale a bit lower than the one of the system). However, they results from activity at a "microscopic" scale (that is to say at the scale of the components). What emerges is a new pattern, at a higher level of the system from the elements which created it, and this new pattern can feedback down to influence the further development of these very elements.

Nigel Gilbert differentiates two kinds of emergence, in order to compare emergent behaviour in human social organisations and emergence in non-human social systems (like an ant colony for example). The differentiation between these two emergent behaviours lies in the ability to reason. Indeed, he explains that emergence in human societies is more complex than in non-humans ones. When reasoning is not implied, an emergent behaviour will be called a first-order emergence. However, for a human social organisation, the "parts" of the system are able to feel the emergence and this perception can make them change their behaviour in order to modify this emergence. Quoting Nigel Gilbert: "people have the ability to recognize, reason about and react to human institutions, that is, to emergent features. Behaviour which takes into account such emergent features might be called second-order emergence."

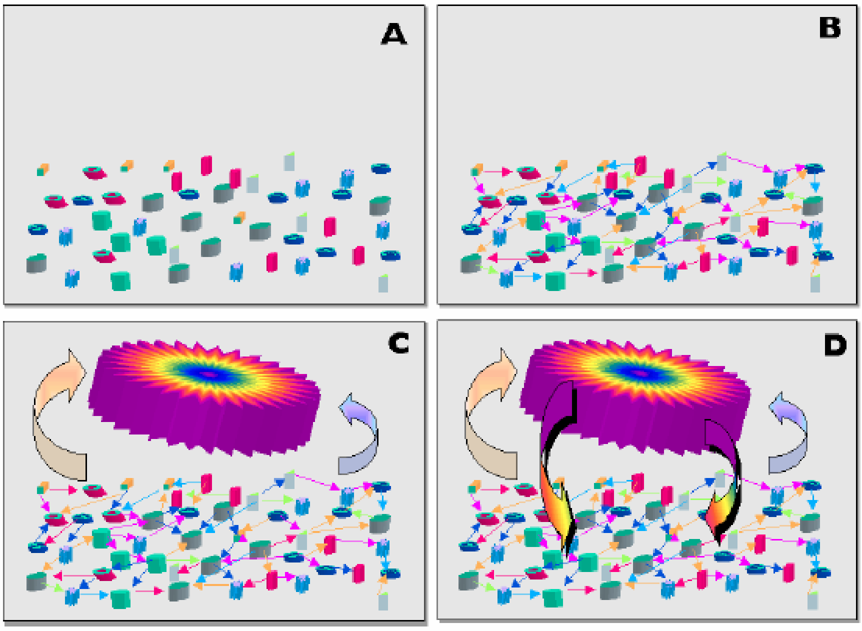


Fig. 1: A high number of elements (A), richly connected (B) give birth to an emergent pattern (C) which feeds back down into the system (D).

Emergence of patterns can lead to very interesting behaviours: Per Bak and his colleagues have shown that a state of "self organized criticality" can emerge from a complex system. When this occurs, the system, through self organization, has reached such an unstable state that changes of all sizes can occur as a result of small external stimuli. However, their size and frequency follow a power law: there are a lot of small changes, fewer medium-sized changes and very few large ones.

As emergent patterns are observable at a macroscopic scale, they are very useful to describe and study complex systems. For example, they can show us the major attractors and/or repellers of the system, or be subject to general properties that help to describe the global dynamics.

Chapter 1: Description of the model

Introduction

The model aims at representing a society of individuals. This society has the form of a city. In this city, people live in family, they work (or go to school), they try to find a mate, they raise their children if they have any, and, finally, when they become too old, they die.

Each individual of my society believes (or not) in several behavioural norms and will interact with other inhabitants of the city depending on his opinion about these norms. They will define the "natural" scheme of behaviour of an individual but will not be followed blindly when a person interacts with another. Indeed the individuals also make their decisions depending on their mental state and on the one of their partner at the time of the interaction.

The norms are transmitted from parents to children. People make also their opinion about them depending on their own experiences. Moreover, the opinion of their family, neighbours, colleagues or friends is also important. Each time an individual interacts with another, this interaction can be witnessed and the individual can be criticized for this interaction. If he receives too many critics by people he likes (or at least that he does not despise), he will feel shame and will strongly change of opinion about his behavioural norms.

During the simulations, we will be very interested in the dynamics of the parameters representing the beliefs in these various norms. By looking at their value, it will be possible to delimit patterns of people who have the same behavioural norms. Since people belonging to the same pattern have almost the same behaviour and the same beliefs, these patterns can be seen as different emergent "cultures" in the city. We will thus follow the evolution of these cultures, looking at how they gain or lose influence on the city.

1. The model in a nutshell

The software I developed is an object-oriented, multi agent system. Each agent is a simple abstraction of a human being. To sum up, a human being will be represented by his level of "awareness", by his level of "integration", and by his beliefs (or disbeliefs) about five social behavioural norms. All these parameters take values between -1 and 1, and have a symmetric meaning depending on the sign of their value:

- Awareness: This parameter represents how aware about its environment an agent is. An agent with a positive awareness has a good level of education: he understands the people around him and is able to teach what he believes. On the contrary, an agent with a negative awareness will be indoctrinated: he will easily follow others' will.
- Integration: This parameter represents the mental state of the agent. An agent with a positive integration will have no troubles with his feelings while an agent with a negative one will be traumatized.
- Behavioural norms: Each norm represents a value that an agent can decide to follow or to reject. The norms considered (and their symmetric meanings) are the following: sociability (and misanthropy), pacifism (and aggression), generosity (and selfishness), honesty (and dishonesty) and flirtation (and puritanism).

All the agents live in the same city. They share a household with their direct family. By using "direct" I mean that in the same house will live the "father", the "mother", and the "children" (if any). Once he is born, an agent will chronologically:

- Be a baby, and interact essentially with its "mother".
- Become a child and go to school.
- Become a teenager
- Become an adult and go to the office
- Leave the household
- Marry another agent (or not)
- Have up to three children
- Die

There are no other frameworks than the city, the house, the school and the office. Each week (that is to say one iteration of the algorithm) each agent will perform six interactions with people from these diverse neighbourhoods.

The agents can only perform five kinds of interactions. They are able to "talk", to "fight", to "give", to "steal" and to "seduce". Each one of these interactions is associated with one of the behavioural norms, which are transmitted from the parents to the children:

talk → sociability/misanthropy
fight → pacifism/aggression
give → generosity/selfishness
steal → honesty/dishonesty
seduce → flirtation/puritanism

An interaction can be a success or a failure. Its outcome is non deterministic and will influence the parameters of both agents interacting. An agent can also witness an interaction and give its opinion about this kind of behaviour: it can criticize or praise one of the protagonists. This will induce shame or pride.

At the end of an interaction, the active agent compares the payoff he has just received performing this interaction and the external feedback he has received from the witnesses. This information can make him modify his point of view regarding the behavioural norm associated with the interaction he has just performed.

After this quick summary, we will now enter into the details. I will begin by presenting the various frameworks first. I will then describe the representation of an agent and I will go on by explaining in details the process of an interaction. Finally, I will describe the visualisation options given by the tool. The diagram shown in appendix 1 can be useful to understand this part.

2. The frameworks

2.1. The city

The city is the main framework. It contains everything that exists in the system. It is an array of 60 by 60 households and is divided into 16 districts. When an agent live in a district, he will study (resp. work) in a school (resp. office) located in this district. This way, two agents of the same district can meet each other ever if they don't live in the same direct neighbourhood.

But the main contacts that exist in the city are direct neighbourhood relationships: the agents will often interact with one of their direct neighbours. We thus have mainly short range relationships.

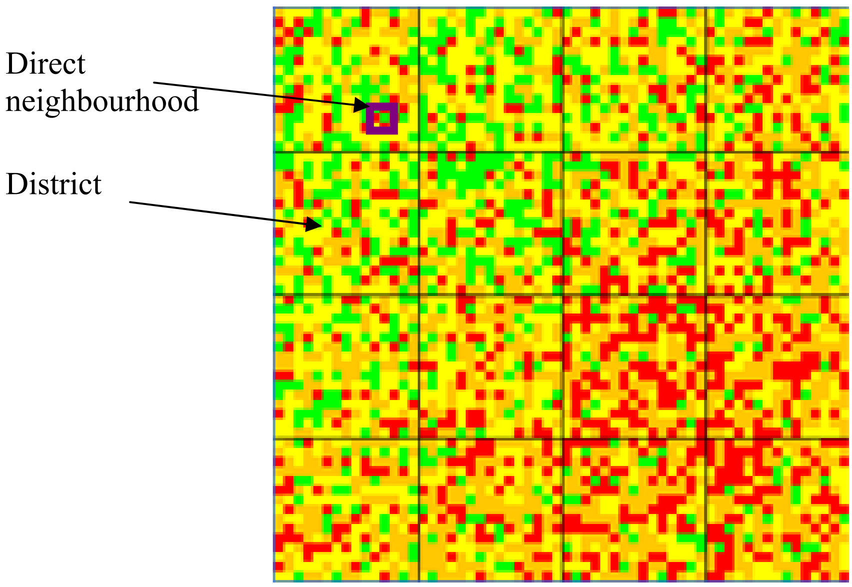


Fig. 2: illustration of a city

2.2. Schools and offices

Each school is divided into 8 sections, each section corresponding to a certain class of age: a child is registered at school at the age of 155 iterations and moves to a new section at the ages of 258, 363, 467, 571, 675, 762 and 849 iterations. He will leave school at 936 iterations (one iteration represents about a week of the life of the agent). There is one school per district.

When an agent is old enough to leave school, he will begin to work in an office. Here again, each office is divided into 8 sections, but this time the sections correspond to a class of awareness: from -1 to -0.75, from -0.75 to -0.5, from -0.5 to 0, etc… An agent will work in the office until he dies; however, depending on its current awareness, he can move from one section to a higher (or lower) one. Here again there is one unique office per district, so every adult belonging to the same class of awareness and living in the same district can interact with each other.

2.3. The households

Up to five members of the same family can live in the same household (2 parents and 3 children). A household can also be empty. This place is very important in the system since the interactions within a family are determinant when the education of children is concerned. During the simulation, an important part of the interactions performed will concern two agents of the same family.

Sometimes, an agent will have to leave his household and find a new one. This happens when the agent becomes an "adult", or when he finds a mate. In these cases, the agent will always try to stay in the neighbourhood of his former house. Therefore, when a search is performed in order to find a new household, the city is covered with a spiral.

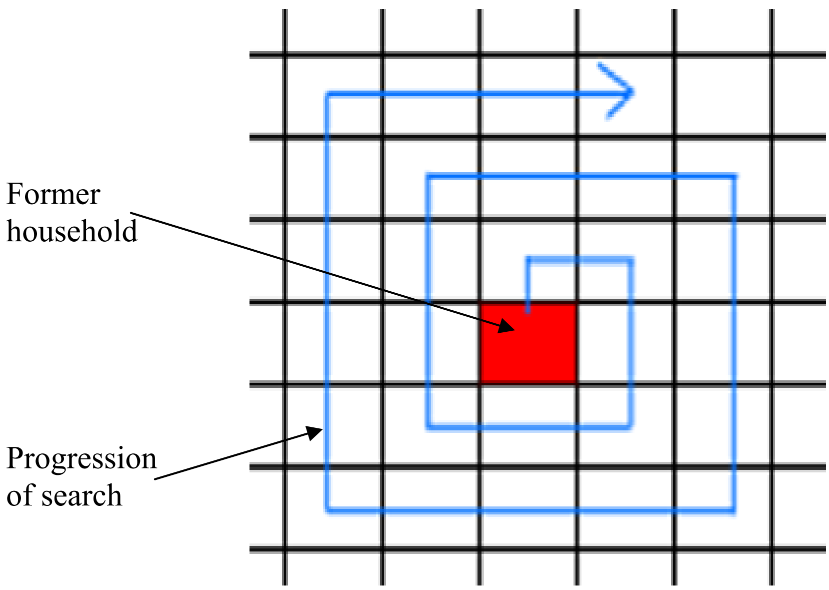


Fig 3: Algorithm of search in the city

This way, even when they move out, people usually stay in the same neighbourhood.

There are three important parameters for a house: its district, whether or not it is empty, and whether or not the house is inhabited by a single agent (in this case, the sex of the single agent living there is accessible).

Finally, when an agent is created, the only persons he knows (see below the description of an agent) are the members of his family, that is to say the agents living in the same household.

3. The agents

3.1. Parameters

The agents are simple abstractions of human beings. They are first defined by the following internal parameters:

- Id
- Sex
- Age
- Awareness
- Integration
- Satisfaction
- Behavioural norms
- Strength
- Life span expectancy

These parameters are created and modified automatically and can only be read by the user. They cannot be modified by an external action. Here is a precise definition for each of them:

¤ <u>Id:</u>

Type: integer

Value: from 0 to infinity

Comments: Id is the name of the agent. Each agent has a unique identification number which will be used to identify and recognize it.

¤ <u>Sex:</u>

Type: integer

Value: 0 or 1

Comments: Sex equals 0 if the agent is a woman or 1 if the agent is a man.

¤ <u>Age:</u>

Type: integer

Value: from 0 to 5700

Comments: The age of the agent is the number of iterations that have passed since its birth. An iteration corresponds to a week.

¤ <u>Awareness:</u>

Type: double

Value: from -1 to 1

Comments: This parameter represents how aware about its environment an agent is. An agent with a positive awareness has a good level of education: he understands the people around him and is able to teach what he believes. On the contrary, an agent with a negative awareness will be indoctrinated: he will easily follow others' will. If we compare this parameter to the human brain, the awareness value represents the conscious zone of the brain (reason)

¤ <u>Integration:</u>

Type: double

Value: from -1 to 1

Comments: This parameter represents the mental state of the agent. An agent with a positive integration will have no troubles with his feelings while an agent with a negative one will be traumatized. If we compare this parameter to the human brain, the integration value represents the unconscious zone of the brain (feelings, state of mind, etc…).

¤ <u>Satisfaction:</u>

Type: double

Value: between 0 and infinity

Comments: The satisfaction is a special kind of resources for an agent. It represents the global well being. The total amount of satisfaction in the city is not constant. Prosocial behaviours will increase this global amount while anti-social behaviour will decrease it in exchange of an increase in the personal amount.

¤ <u>Behavioural norms:</u>

Type: Array of doubles

Values: All the elements of the array are between -1 and 1

Comments: This array represents the opinion of the agent regarding the five social norms. These norms are the following ones:

- Sociability: the first element of the array will be used in order to know if the agent likes to be together with other individuals. The more this value is, the more often an agent will try to talk with his neighbours. It will be articulate and seek the company of others. However, a negative value for this parameter will represent a misanthropic agent which tries to avoid communication with the others.

- Pacifism: the second element of the array will show us whether the agent is aggressive or not. A high value for this parameter will represent an agent which avoids the fights whereas a negative value will represent an agent with a propensity for aggression.

- Generosity: the third element of the array is the level of generosity of the agent. A positive value will represent an agent who is ready to share its satisfaction with the persons he likes. On the contrary, an agent with a negative generosity will be very selfish.

- Honesty: the fourth element of the array represents the honesty of the agent. With a positive value for this parameter, an agent will almost never cheat and will behave honestly with his neighbours. On the contrary, an agent with a negative honesty will be a thief.

- Flirtation: the fifth and last element of the array will show us the point of view of the agent when it comes to courtship and seduction. A positive value will represent a flirtatious agent: he will enjoy seducing the partners he meets. On the contrary, a negative value will reflect a puritanical behaviour.

¤ Strength:
Type: double
Value: from -1 to 1
Comments: The strength is a genetic parameter: its value will depend on the strength of the agent's father and the one of its mother. It will be used to determinate the success or failure of any "fight" interactions.

¤ Life span expectancy:
Type: int
Value: a Gaussian distribution of mean 3900 and variance 1000
Comments: When the age of the agent becomes bigger than its life span expectancy, an agent dies (see part 3.3 for more details). Since an iteration equals a week, the average age reached before dying is around 75 years old.

There are also five parameters editable at any time by the user: the importance of norms in the decision-making process (see part 4.3.) and the amplitude of a modification when a change of awareness, integration or behavioural norms occurs (the user can enter a different value for each of these three parameters, see part 4.4), and the importance of the judgement of others for the agent (see part 4.6).

Apart from these parameters, each agent has also access to various objects: a household, a school or an office, the city, and, above all, a contact list.

3.2. The contact list

The contact list is an abstraction of the agent's knowledge. Inside this object, an agent will be able to find information about a partner if he has already met him in the past. When an agent is created, its contact list only contains the members of its family but whenever he meets a new partner, the contact list will be updated.

Thanks to its contact list, an agent can find two pieces of information about another one: the relationship between them (if they are friends or rivals) and, if the two agents are in the same family, the familial bonus of the contact.

The relationship is a double between -1 and 1. Initialised at zero, it represents the level of affection existing between both agents. This value is symmetric: if agent A has a relationship of R with agent B, B has also the relationship R with A. If the agents like (resp. hate) each other, the value will be positive (resp. negative).

The familial bonus appears when two agents of the same family interact with each other (see part 4). It is also a double between -1 and 1. It is initialised at 0.9 for a son or a daughter, at 0.75 for a companion, a father or a mother, and at 0.5 for a brother and a sister. This parameter reflects the peculiar link existing between the members of a family. This bonus will change with time. For example, if a child is beaten by his father, the familial bonus of the father from the point of view of the child will decrease: the familial link between them is weakened. When this bonus becomes negative (it is now a malus), this means that not only the agent does not like this family number, but also that, because of the latter's behaviour, it utterly despises him and do not consider him from being part of its family.

Because of memory requirements, the contact list is limited to a finite number of known agents. Because of this, a contact which has not been met for a long time will be forgotten.

3.3. From birth to death, the life of an agent

An agent can be created in two different circumstances. The most obvious is when he is born. In that case, its age, awareness, integration and norms are initialized at 0. His strength (resp. satisfaction) will be the average of his parents' strength (resp. satisfaction).

But an agent can also be created before the beginning of the simulation. In that case, the values of awareness, integration and the five behavioural norms are chosen at random between -0.9 and 0.9 following a Gaussian distribution centred in 0. Age, satisfaction and strength are then uniformly distributed.

Just after being born, an agent is considered as a baby. Each iteration, a baby will:

- Interact three times with members of its family (anyone living in the same household),

- Interact three times with its mother,

- Receive a teaching from its mother,

- Receive a teaching from its father (for more information about teachings and interactions, see parts 3.4 and 4),

- Increase its age.

At the age of 155 iterations (three years old), the agent is considered as a child. He will now go to school and meet people from the outside. Each iteration, a child will:

- Interact three times with members of its family,

- Interact twice with other students at school,

- Interact once in the direct neighbourhood,

- Receive a teaching from its mother,

- Receive a teaching from its father,

- Increase its age and change school if necessary.

At the age of 675 (about 13 years old), the agent becomes a teenager. Its timetable will change a little and it will be able to seduce other agents. Here is what a teenager does during one iteration:

- It interacts twice with members of its family,

- It interacts twice with other students at school,

- It interacts twice in the direct neighbourhood,

- It receives one teaching from its mother or its father,

- It increases its age and changes of school if necessary.

At the age of 936, our agent finally becomes an adult. It leaves the household and begins to live alone. It will no longer go to school but will work in an office. It is still single for the moment and will have to find a mate in order to have children. During each interaction it will:

- Interact three times with colleagues at the office,

- Interact three times in the direct neighbourhood,

- Have a probability of getting married (2% chance),

- Increase its age and change of office if necessary. It dies if its age becomes bigger than its life spam expectancy.

When he finally gets married, our agent will have a new timetable. During each iteration it will:

- Interact three times with members of its family,
- Interact twice with colleagues at the office,
- Interact once with the direct neighbourhood,
- Have a probability of having a child if it is a female, is younger than 2800, and has less than three children (0.02% chance).
- Increases its age and change of office if necessary. It dies if its age becomes bigger than its life spam expectancy.

Finally if the agent is a female and just had a baby, it is considered as a "young mother". It will:

- Interact three times with members of its family,
- Interact twice with its baby,
- Interact once with the direct neighbourhood,
- Increases its age and change of office if necessary.

When its baby has become a child, the young mother becomes a regular adult again.

Remarks: ¤ When an agent gets married, he does not choose his companion: this one is imposed by the society. However, the future companion will have about its age (the difference must be < 10 years). The future companion is found thanks to the search algorithm described figure 3.

¤ An agent cannot die if ever he still has a child to take care of. However, this rule is no more taken into account as soon as this child becomes an adult.

¤ If the two parents of the household die and if there are still single adults (the sons of the deceased agents) living in the household, the older single becomes the owner of the house. He continues to live with its brothers until they get married at last.

¤ There must be at least 2,1 children per woman in a society for its population to be renewed: that is why each woman can have up to three children. When you limit the household to two children only, the population of the city is bound to decrease and disappear after a certain lapse of time (usually several "centuries", ≈20000 iterations).

3.4. Transmission of norms

People follow internal norms when they value certain behaviours for their own sake, in addition to or despite the effects these behaviours have on their personal fitness and/or perceived well-being. Norms are emergent phenomena: if we study a single human being, there is nothing about him that suggests the notion of norm. Norms emerge from the group, are passed to the next generation of individuals by parents and other influential elders, and then are internalised by these individuals: internalisation moves norms from external constraints that one can treat instrumentally towards maximizing well-being, to norms that are valued as ends rather than means.

In the model, the norms will be transmitted to the children by the parents thanks to an action we will call teaching (later, we will introduce other agents called "teachers" that will be also be able to transmit some norms to the children when they are at school; see chapter 2). When a parent teaches something to his child, he will try to make his child have the same opinion than him regarding a particular norm. This norm will be chosen at random from the five possible ones and the parent will try to transmit it.

The child can then internalise it or not. If he is totally indoctrinated (awareness < -0.8), the child will accept blindly the teaching (the internalisation of what has been taught is automatically successful). Otherwise, the probability of internalizing it will depend both on the awareness of the parent and on the absolute difference between the opinion of the teacher and the one of the student regarding this norm.

If the internalisation is successful, the opinion of the child regarding the norm taught will be modified in order to be closer to the opinion of the teacher; otherwise, nothing happens:

```
double s = 2*teacher.awareness +
            (2 - |behaviouralNorms[topic] - teacher.behaviouralNorms[topic]|) ;
double t = -2 + TalkOrFight_v1.random.nextDouble()*6 ;
 if (t <= s || awareness <-0.8) {
        behaviouralNorms[topic]+= 0.01 * ( teacher.behaviouralNorms[topic]
                                            -  behaviouralNorms[topic] ) ;
 }
```

4. Interactions

There are five possible interactions: talk, fight, give, steal and seduce. The latter is not available for children and babies. Each interaction is associated with a behavioural norm:

talk → sociability(/misanthropy)

fight → pacifism(/aggression)

give → generosity(/selfishness)

steal → honesty(/dishonesty)

seduce → flirtation(/puritanism)

The outcome of "talk", "fight" and "seduce" is non deterministic: these interactions can be failed or successful. "Give" and "steal" however are totally deterministic.

4.1. Sum up of the process

During an interaction, there is always an active agent and a passive one. The active agent will be the one which decide what kind of interaction will be performed, whereas the passive one has no other choice but to be subjected to this iteration. When I said in part 3.3 that an agent will "interact twice at school" for example, it means in fact that he will twice be active in an interaction at school. It is possible (and likely) that he will be implied in more interactions, but this time he will be the passive agent.

When an agent must interact with someone (he is thus active) he will have to:

¤ 1. Find a partner to interact with (the passive agent)

¤ 2. Decide which interaction to perform depending on this partner and the active's agent parameters.

¤ 3. Perform this interaction, which will modify the parameters of both interacting agents.

¤ 4a. Evaluate its personal payoff after this iteration

¤ 4b. Receive an external feedback (other agents may have witnessed the interaction and they may give their opinion about it).

¤ 4c. Adjust the behavioural norm associated with the interaction he has just performed depending on its payoff and feedback.

4.2. First step: selection of a partner

This step is the simplest one. The partner will always be selected at random within the current neighbourhood. The definition of "neighbourhood" will change depending on where the interaction takes place:

- During an interaction with a member of the same family, any person living in the same house can be selected as a partner.

- During an interaction at school (resp. at the office), any student (resp. colleague) working in the same section of the school (resp. office) can be selected as a partner.

- During an interaction with the "direct neighbourhood", any agent living in one of the 8 houses surrounding the active agent's household can be selected as a partner.

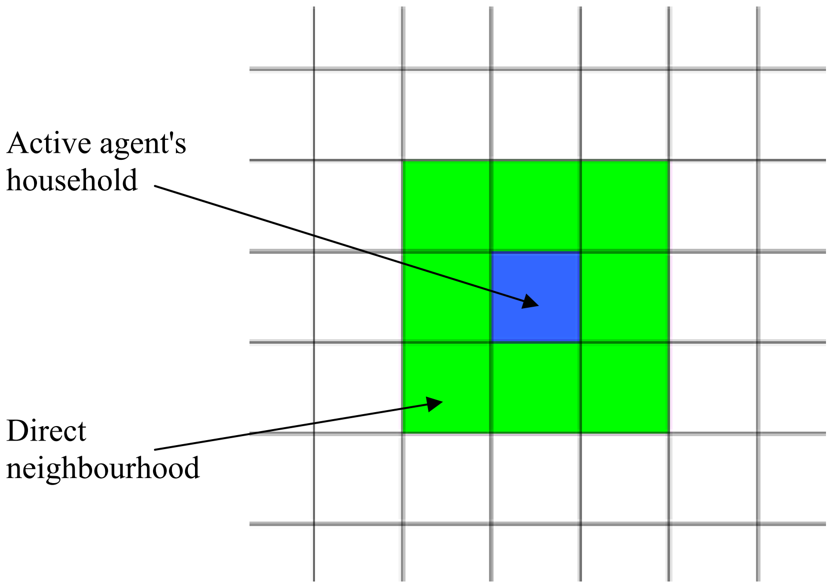


Fig. 4: direct neighbourhood

The only exception to these rules is the case of the mother/baby relationship. In this case and this case only, the partner is chosen in a deterministic way.

4.3. Second step: selection of an interaction

This step is one of the most important: it represents the decision-making process of the agent. Before making its choice, the agent will first perceive the situation. This situation will depend on its own mental state, the one of its partner, the relationship between them, the eventual familial bonus and finally the difference of satisfaction between them.

$$\vec{S} = \begin{bmatrix} awareness \\ integration \\ relationship \\ partner.awareness \\ partner.familial\ bonus \\ \dfrac{satisfaction\ -\ partner.satisfaction}{\max(satisfaction,\ partner.satisfaction)} \end{bmatrix}$$

(All the elements of ***S*** are between -1 and 1).

Moreover, each interaction is associated with a vector of weights ***I(interaction)***. These vectors are I and normalized. They represent the various interactions in the space of the situations.

$$\overrightarrow{I(talk)} = \begin{bmatrix} 0.398 \\ 0.398 \\ 0.199 \\ 0.1 \\ 0.796 \\ 0 \end{bmatrix} \quad \overrightarrow{I(fight)} = \begin{bmatrix} -0.234 \\ -0.468 \\ -0.468 \\ -0.117 \\ -0.702 \\ 0 \end{bmatrix} \quad \overrightarrow{I(give)} = \begin{bmatrix} 0.196 \\ 0.196 \\ 0.392 \\ 0 \\ 0.392 \\ 0.704 \end{bmatrix}$$

$$\overrightarrow{I(steal)} = \begin{bmatrix} 0 \\ -0.426 \\ -0.213 \\ 0 \\ -0.213 \\ -0.853 \end{bmatrix} \quad \overrightarrow{I(seduce)} = \begin{bmatrix} 0 \\ 0.577 \\ 0.577 \\ 0.577 \\ 0 \\ 0 \end{bmatrix}$$

These values are not randomly chosen but represent the ideal situation for each interaction. Thus, if the agent is aware and well integrated, and is facing a member of its family that he likes, the situation is nearly ideal for a little talk. Or maybe his partner will be far poorer in satisfaction than him and he will decide to be generous. Or the partner will be someone he does not like and he will decide to fight. Etc…

Each available interaction will receive a mark, defined by:

$mark(Interaction) = (1-\alpha) \times \vec{S} \bullet \overrightarrow{I(Interaction)} - \alpha \times Norm(Interaction)$ if Interaction = "steal" or "fight".

$mark(Interaction) = (1-\alpha) \times \vec{S} \bullet \overrightarrow{I(Interaction)} + \alpha \times Norm(Interaction)$ otherwise

α is what I called earlier the importance of norms in the decision making process and is a double between 0 and 1. This parameter will be editable by the user in order to simulate more or less conservative societies. The value of the norm is counted negatively in the cases of "fight" and "steal" because the interactions and their associated norms (pacifism and honesty) have opposed implications.

The scalar product ***S.I(interaction)*** will tell us how well the interaction evaluated is adapted to the situation. Seen geometrically, it is the projection of the current situation on the direction defined by the interaction. The higher this projection, the more an interaction is adapted to this situation. Since all the ***I*** vectors are normalised, the value of the scalar product will only depend on the angle existing between the current situation and the direction representing the interaction. Indeed, ***S.I*** = |***S***|*|***I***|*cos(***S,I***) = |***S***|*cos(***S,I***) and |***S***| is a constant since the situation is the same for all interactions.

After that all the interactions have been evaluated, the one which has received the best mark will be performed.

4.4. Interlude: modification of a parameter

Almost all the parameters (awareness, integration, norms, familial bonus and relationships) will evolve following hysteresis cycles. A hysteresis cycle is a function often found in the field of magnetism. The idea is simple: to increase (resp. decrease) the value of a parameter to a certain point this value will follow a certain function. However, if you want the parameter to be restored to its initial value, then it parameter will not follow the same function as the one he followed during the increase (decrease).

The hysteresis cycles I use in my tool will have the following behaviour:

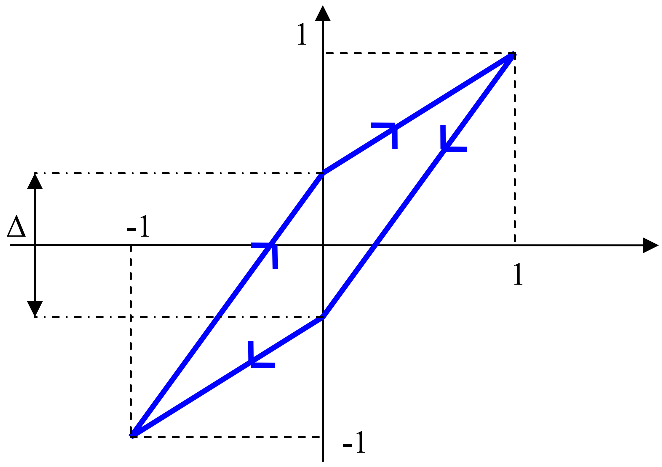


Fig. 5: hysteresis cycle

Δ is what I called earlier the amplitude of the modification. This is a parameter which can be editable by the user for a modification of awareness, integration and behavioural norms. The default values are the following:

Δawareness = 0.1%　　　Δintegration = 0.1%

Δnorms = 0.05%　　　Δrelationships = 1%

Δfamilial bonus = 1%

When a parameter is modified, its value will increase (or decrease) by a number of discrete steps. The size of a step is computed in the following way: let v_n be the current value of a parameter and v_{n+1} its value after a modification of one step.

¤ If $v_n > 0$ and we want to increase it, $\mathbf{v_{n+1} = \Delta + (1 - \Delta)v_n}$.
¤ If $v_n > 0$ and we want to decrease it, $\mathbf{v_{n+1} = -\Delta + (1 + \Delta)v_n}$.
¤ If $v_n < 0$ and we want to increase it, $\mathbf{v_{n+1} = \Delta + (1 + \Delta)v_n}$.
¤ If $v_n < 0$ and we want to decrease it, $\mathbf{v_{n+1} = -\Delta + (1 - \Delta)v_n}$.

Graphical construction:

If this definition seems a bit abstract, the following graphical construction is a good way of representing the dynamics of the parameters. Its values v_n are read on the abscissa. We use the straight line y = x as a way of reporting the new value of the parameter on the abscissa. The example here shows us a graphical construction for the following sequence: an increase of 2 steps followed by a decrease of 5 steps. In order to see what happens precisely, I used here a very high Δ. In the model however, the Δ values as well as the size of the steps are lower.

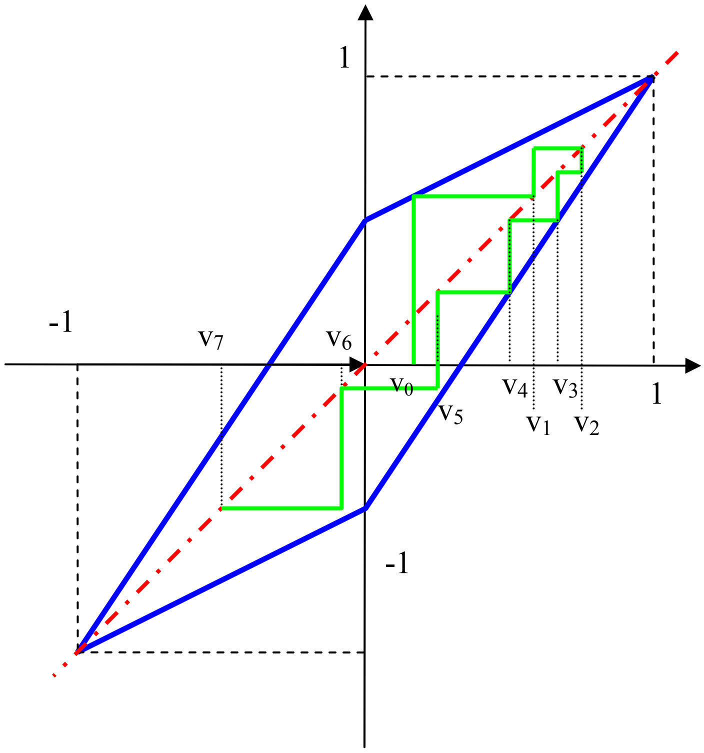


Fig. 6: increase of 2 steps followed by a decrease of 5 steps.

As you can see on this graphics, it takes more effort to lower v_n around its initial value once it has increased. That is what I have chosen the hysteresis cycles as functions in my model: these cycles keep a history of the former states of the parameter. This will reflect quite accurately the actual situation: once a human being has become totally indoctrinated for example, it takes a lot of energy to make him overcome this indoctrination, more energy in fact that the one used during the indoctrination. We are indeed trying to change behavioural patterns which have been instilled since childhood…

The theory of iterative maps gives us a few properties about the dynamics of the parameters: the values will oscillate between -1 and 1 because the system has two attractive fixed points: (1,1) and (-1,1). When the parameter increases (resp. decreases), its value is attracted towards 1 (resp. -1) but it will never have the exact value 1 (resp -1).

4.5. Third step: applying the interaction itself

Now that the agent has a partner and that he has chosen an interaction, this interaction will have effects on both agents' parameters. In the following, "$agent_1$" will represent the active agent and "$agent_2$" the passive one.

4.5.1. Talk

Talking is a prosocial interaction. From dialog, global awareness and cooperation increase. However, sometimes a dialog can become boring or even degenerate into an oral fight. The other person can also decide to ignore you and not to talk to you. That is why, in my model, a talk can be successful or failed. A successful interaction represents a dialog which was interesting and fruitful. On the contrary, a failed one represents a dialog that was found uninteresting or fruitless (failed negotiation, argument, boring face-to-face, etc…). The active agent does not manage to learn anything from a failed talk; moreover, since he was the one who started the discussion, he will be a little traumatised not to manage to communicate with the others. In the worst case, if the active agent was already indoctrinated, this failed communication will

To decide whether a "talk" is successful or not, we will use the following rule:

¤ *Let a = relationship($agent_1$, $agent_2$) + $awareness_1$ + $integration_1$ + familial $bonus_{2/1}$.*
¤ *Let b be a number taken at random in [-3,3] (uniformly).*
¤ *If b < a: success.*
Else: failure.

In case of success, we have the following effects:

- $awareness_1$ increases by one step,
- $awareness_2$ increases by one step,
- $integration_1$ increases by one step,
- $integration_2$ increases by one step,
- $satisfaction_1$ increases by 0,1%,
- $satisfaction_2$ increases by 0,1%,
- relationship($agent_1$, $agent_2$) increases by one step.

In case of failure:

- if $awareness_1 < 0$, $awareness_1$ decreases by one step,
- $integration_1$ decreases by one step,
- relationship($agent_1$, $agent_2$) decreases by one step.

Thus, a successful talk will be totally beneficial for both agents, whereas a failed one will have negative effects for the active agent.

Remark: talking is the only way to increase in awareness.

4.5.2. Fight

Fighting is an anti social situation. By starting a fight, an agent excludes himself from the society. He will try to hurt its partner, which in term of our parameters, will be represented as a decrease of integration and satisfaction. After a fight, the global amounts of satisfaction, integration and awareness in the system decreases.

There is a winner or a loser in a fight. This will be determined by the strength of the two opponents. The rule is the following:

¤ *Let* $a = strength_1 – strength_2$.
¤ *Let b be a number taken at random uniformly between -1 and 1.*
¤ *If* $b < a$: $agent_1$ *wins.*
Else: he loses.

When $agent_1$ wins, we have the following effects:

- $integration_1$ decreases by two steps,
- $integration_2$ decreases by three steps,
- $satisfaction_1$ increases by 0,2%,
- $satisfaction_2$ decreases by 0,5%.

Otherwise we have:

- $integration_1$ decreases by three steps,
- $integration_2$ decreases by one step,
- $satisfaction_1$ decreases by 0,5%.

Moreover, whoever wins the fight, there are always the following effects:

- $awareness_1$ decreases by one step,
- relationship($agent_1$, $agent_2$) decreases by two steps,
- if $agent_1$ and $agent_2$ are from the same family, familial $bonus_{1/2}$ decreases by two steps.

What happens is that a fight between two agents will always traumatise them both, and the loser will receive even more trauma. If the active agent is victorious, its satisfaction will increase: it wanted to fight and it won the battle, so now it "feels better". However, because of the nature of the interaction itself, an agent will always become more indoctrinated and more traumatised after a fight than before.

There is a special focus on the familial environment here: if an agent attacks someone from its family, its familial bonus from the point of view of its partner (and victim) can decrease and even become negative. This simulates what can happen in cases of domestic violence.

4.5.3. Give

The act of giving is a deeply prosocial interaction. For the active agent, giving will mean offering a part of its satisfaction to its partner. This interaction is automatically successful and will have the following effects:

- $integration_1$ increases by two steps,
- $integration_2$ increases by two steps,
- $satisfaction_1$ decreases by 20%,
- $satisfaction_2$ increases by 20% of $satisfaction_1$,
- relationship($agent_1$, $agent_2$) increases by two steps,
- if $agent_1$ and $agent_2$ are from the same family, familial $bonus_{1/2}$ increases by one step.

Remark: Giving is the only way to increase the familial bonus. Sometimes a gift can strengthen familial links.

4.5.4. Steal

Stealing is an antisocial interaction. When this interaction happens, the global amounts of satisfaction and integration in the system decrease. This interaction is automatically successful: since there are neither policemen nor law in my society, a dishonest person cannot be arrested. However, by this act, he will suffer a decrease of integration as will his victim (because of the trauma implied in the act of being stolen). Here are the effects of a "steal":

- $integration_1$ decreases by two steps,
- $integration_2$ decreases by one step,
- $satisfaction_2$ decreases by 10,5%,
- $satisfaction_1$ increases by 10%,
- relationship($agent_1$, $agent_2$) decreases by two steps,
- if $agent_1$ and $agent_2$ are from the same family, familial $bonus_{1/2}$ decreases of one step.

Here again we have a focus on the familial environment: when your family members are dishonest with you, it is far more difficult to trust them.

4.5.5. Seduce

The last interaction, seducing, can be catalogued neither into the prosocial interactions nor into the antisocial ones. However, sexuality and seduction play an important role in our social behaviour. Some societies encourage sexual liberty while others, more puritans, will prefer to adopt more rigorous norms, as virginity before wedding or abstinence except for procreation.

An attempt to seduce another agent can succeed or fail. This is defined by the following rule:

¤ *Let a = relationship($agent_1$, $agent_2$) + $awareness_1$ + $integration_1$.*
¤ *Let b be a number taken at random in [-3,3] (uniformly).*
¤ *If b < a: success.*
Else: failure.

In case of success we have the following effects:

- If $agent_1$ is committing adultery, $integration_1$ decreases by one step,
- Else, $integration_1$ increases by two steps,
- If $agent_2$ is committing adultery, $integration_2$ decreases by one step,
- Else, $integration_2$ increases by one step,
- $satisfaction_1$ increases by 0,2%,
- $satisfaction_2$ increases by 0,2%,
- relationship($agent_1$, $agent_2$) increases by two steps.

And these ones in case of failure:

- If $awareness_1 < 0$, $awareness_1$ decreases of 1 step,
- $integration_1$ decreases of one step,
- $integration_2$ increases of one step (it is valorising to be courted),
- $satisfaction_1$ decreases by 0,1%.

Remark: You have to be at least a teenager in order to be able to seduce. Moreover, it is impossible to seduce your children or one of your parents.

4.6. Fourth step: feedback and adjustment of the norm

After having performed the interaction he had selected, our agent will have the opportunity to adjust his belief about the norm associated with this interaction. In order to decide how to adjust it, the agent will look at two parameters: its personal payoff and the external feedback it receives from its neighbourhood.

4.6.1 Personal payoff

The personal payoff is computed using the following formula:

$$PP = \Delta integration + \Delta satisfaction_{rel} + Norm(interaction\ performed)$$

Its first and second terms are respectively the amount of integration and relative amount of satisfaction (that is to say $\frac{satisfaction_{after} - satisfaction_{before}}{satisfaction_{before}}$) earned thanks to the interaction.

The last term represents the prosocial emotions guilt and comfort. Indeed, when someone acts according to its norms, he will feel comfortable with its actions. Thus, they will seem more interesting to him. On the other hand, when someone act contrary to what he thinks is right, he will have a feeling of guilt which will lower the personal payoff.

4.6.1 External feedback

Up to three agents in the neighbourhood can have witnessed the interaction. They will then give their opinion about the interaction they have just seen. The external feed back is given by the following formula:

$$EF = \sum_{witnesses} opinion(witness, interaction) \times (1 + familialBonus_{witness/agent_1})$$ where:

- If interaction = "fight" or "steal":

  opinion(witness, interaction) = -1 if Norm(interaction) > 0,5.

  opinion(witness, interaction) = 0 if -0,5 < Norm(interaction) < 0,5.

  opinion(witness, interaction) = 1 if Norm(interaction) < 0,5.

- Otherwise:

  opinion(witness, interaction) = 1 if Norm(interaction) > 0,5.

  opinion(witness, interaction) = 0 if -0,5 < Norm(interaction) < 0,5.

  opinion(witness, interaction) = -1 if Norm(interaction) < 0,5.

(here again we are compelled to differentiate the two kinds of interactions because of the antisocial values of "fight" and "steal").

Remark: if the witness giving its opinion is despised by the active agent (that is to say in terms of parameters relationship(witness, $agent_1$) < -0,8), this opinion will be ignored by the agent (that is to say opinion(witness, interaction) = 0).

This external feedback represents the prosocial emotions shame and pride. An agent will be proud of what he has performed if he is congratulated and ashamed if he is criticized (especially by members of his family). This will have effects on the final adjustment.

4.6.3 Adjustment of the norm

The final payoff is computed in the following way:

$$payoff = (1-\beta) * PP + \beta * EF$$

β represents the importance of others' judgement for the agent. This parameter is a double between 0 and 1 and is editable by the user. However, its value should not be too low in order to have a realistic simulation. Indeed norms emerge from the group and are followed as ends rather than means. Thus, the personal payoff does not have to play an important part in the evaluation of an interaction: norms are not adopted because of the perceived well-being they offer. By default, the value of β is 2/3.

Once this final payoff has been computed, the belief in the norm associated with the interaction performed is finally adjusted in the following way:

- If payoff > 1,4: the belief is increased by 8 steps,
- If 1,1 < payoff ≤ 1,4: it is increased by 6 steps,
- If 0,9 < payoff ≤ 1,1: it is increased by 4 steps,
- If 0,6 < payoff ≤ 0,9: it is increased by 3 steps,
- If 0,4 < payoff ≤ 0,6: it is increased by 2 steps,
- If 0,1 < payoff ≤ 0,4: it is increased by 1 step,
- If -0,1 < payoff ≤ 0,1: it is not modified,
- If -0,4 < payoff ≤ -0,1: it is decreased by 1 step,
- If -0,6 < payoff ≤ -0,4: it is decreased by 2 steps,
- If -0,9 < payoff ≤ -0,6: it is decreased by 3 steps,
- If -1,1 < payoff ≤ -0,9: it is decreased by 4 steps,
- If -1,4 < payoff ≤ -1,1: it is decreased by 6 steps,
- If payoff ≤ -1,4: it is decreased by 8 steps.

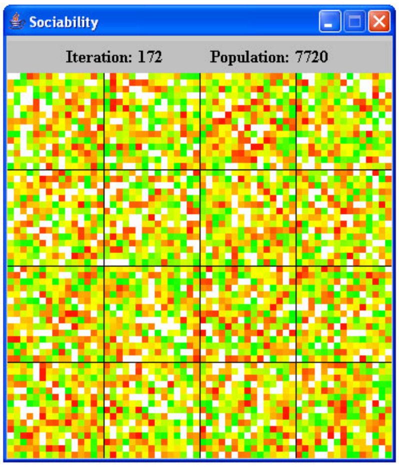
This concludes the interaction process.

5. Human-machine interface and visualisation of the results

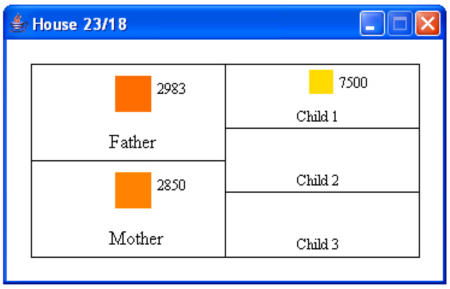
5.1. Overview

The human-machine interface in composed of six windows:

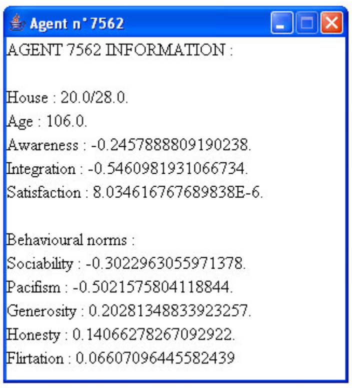
1. The main visualisation window which represents the city. Here, we are looking at the agents' level of sociability:

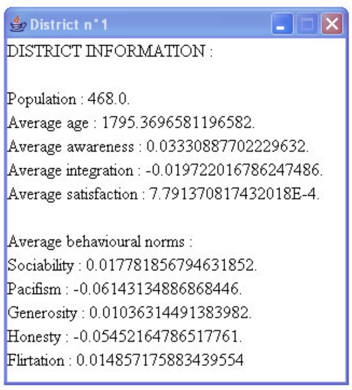


2. A second visualisation window will allow the user to look inside a household:

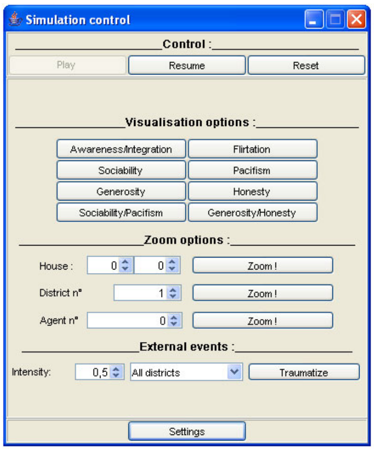


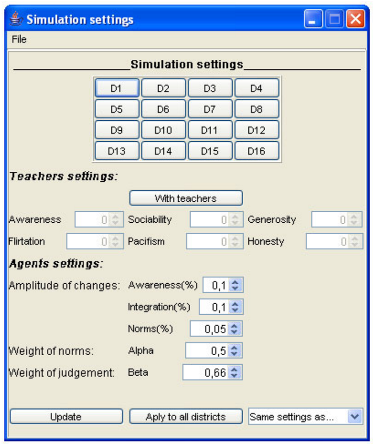
3/4. A window which give the user some information about an agent and a similar one for some information about a district:





5/6. A window which controls the whole software and another one to change the settings of the simulation:





Only the control window and the main visualisation one are opened when the software is launched. The four others ones are popup windows which can be opened later by the user.

I will now explain how to use each of these windows.

5.2. Controls

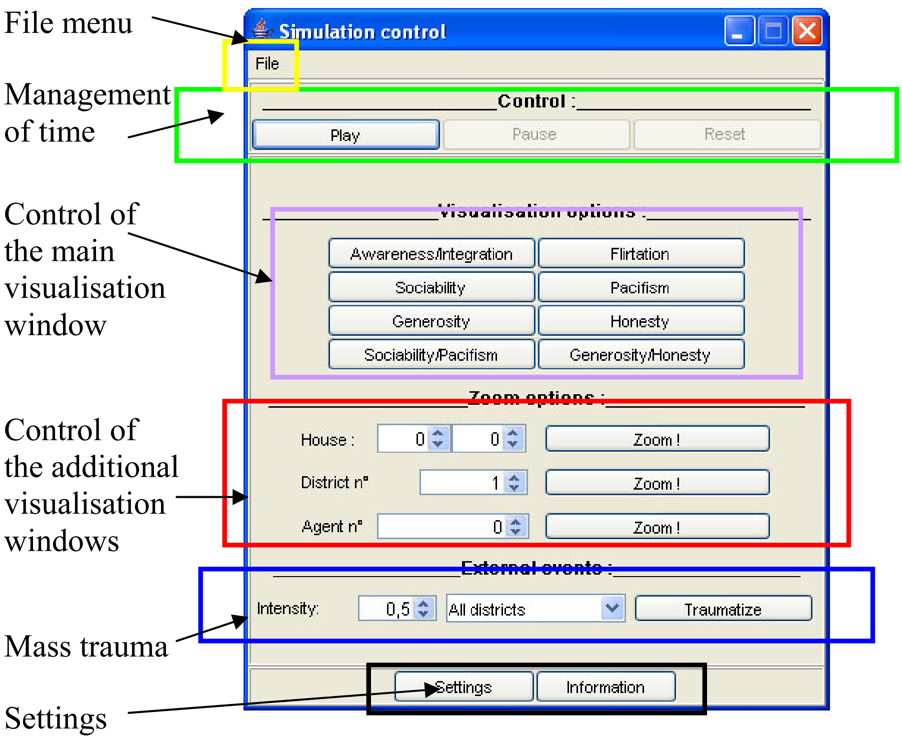


Fig. 7: The "control window"

Thanks to the file menu, the user can save and load the situation of a city.

At the top of this window, we find the three buttons which will control the flow of time in the system. With them, we can start a simulation, interrupt it in order to resume it later, or reset the system and start with a brand new city.

The second group of buttons are used in order to choose the parameter to be displayed in the main visualisation window. The user can choose to look at the values for the five behavioural norms or to visualise the awareness and integration of his agents. He can also choose to visualise the correlation existing between sociability and pacifism as well as between generosity and honesty.

The third group of buttons will open the additional visualisation windows (house, information on district and information on agents). The house to zoom on will be selected by its coordinates, the district by its district number and the agent by its id.

The fourth group of spinners + button will trigger events which will traumatise the agents. This functionality has not been defined yet and does not belong to the initial model. See chapter 3 for more details.

Finally, the isolated button at the bottom of the window will open the settings window.

5.3. Visualisation tools

The user will use the main visualisation window in order to visualise the results of the simulation. Thanks to this window, he will be able to observe in a simple glance the evolution of the internal parameters (behavioural norms and awareness/integration) as well as have some information about the state of the system (the age of the city and its population).

This main window will give us an aerial view of the city. What we see represents households, and not agents. Each square represents a house and the district boundaries are displayed. When the user observes a parameter, he will see the average value of this parameter for each household: the corresponding squares will be coloured depending on this value. The window does not use the same coloured code whether the user observes one or two parameters.

If we observe only one parameter (case of behavioural norms), the colour system is simple: the greener (resp. redder) a house, the higher (resp. lower) the average value of this parameter. When the value is around 0, the house is coloured in yellow/orange shades.

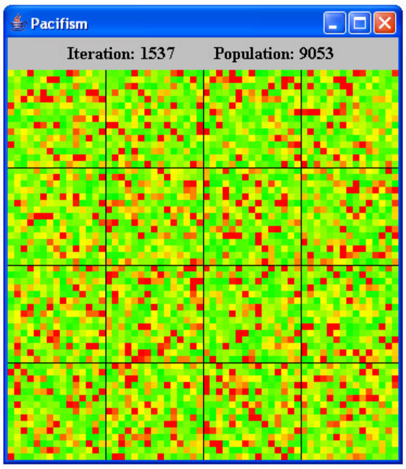


Fig. 8: Visualisation of one parameter

On the other hand, when the user will look at two parameters at the same time, the following code will be used:

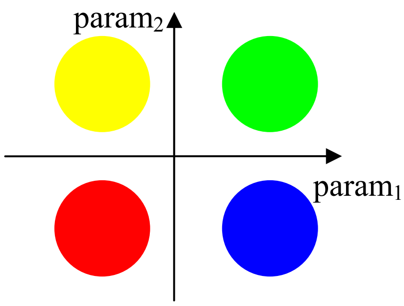


Fig. 9: Coloured code for two parameters

We will thus obtain this kind of screen (here with $param_1$ = awareness and $param_2$ = integration):

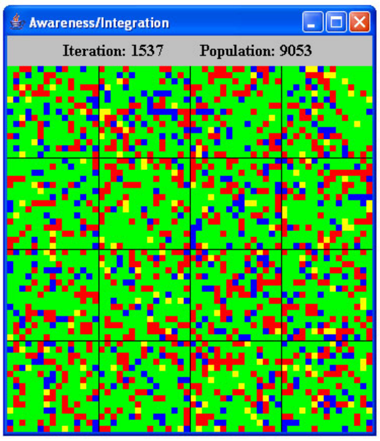


Fig. 10: Visualisation of two parameters

If the user wants to have more detailed information about the society, he will have to look inside the houses. A house is represented in the following way:

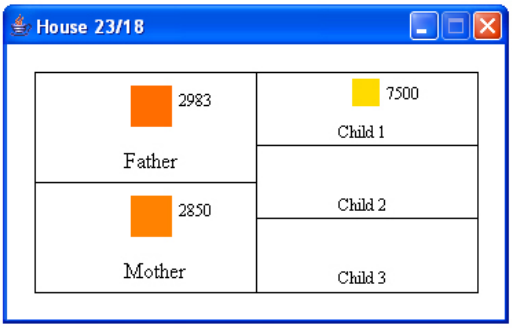


Fig. 11: The inside of a house

The parameters observed are the same as the one selected for the main visualisation window, but this time we have directly access to the agents' parameters, and not to an average. The number written at the right of the squares representing the agents are these agents' ids. In this example, we can see that agent #2983 married agent #2850 and that they had one child, #7500. This window can be opened by clicking on a house in the main visualisation window, or by entering the coordinates of the house in the control one.

Finally, if the user wants quantitative results, he can open information windows about an agent, a district, or the whole city (illustrations of these windows can be seen above).

5.4. Settings

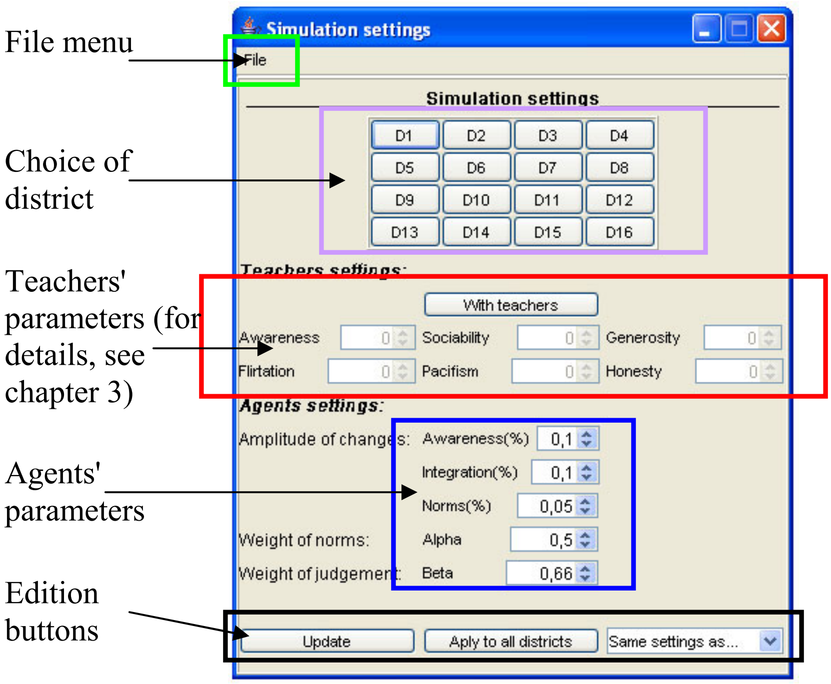


Fig. 13: The settings window

This window will be used in order to change the settings of the simulation for each district (it is possible to have different settings for the different districts of the city). At its top, a file menu can be found. This menu gives the opportunity to the user to save his settings or to load previously saved ones.

Below, there is a group a 16 buttons, each one of them representing a district. They are disposed as their corresponding districts in the city, which facilitates the navigation into the settings. When the user clicks on one of them, he will access the settings of the associated district.

Six spinners and a button are located below the district buttons. They will be used to modify the parameters of the teachers (this concept has not be defined yet and does not belong to the initial model; see chapter 3 for more details).

After that, there is a group of spinners used to modify Δawareness, Δintegration and Δnorms (parameters of the hysteresis cycles), as well as α, the importance of norms in the decision-making process.

Finally, the last panel will be used in order to edit the parameters in a friendlier way. The "Apply to all districts" button will apply the settings currently displayed to the 16 districts. The "Same settings as…" combo box will be used to copy the settings of a district into another district. Finally, the "Update" button will transmit the settings to the system and close the settings window.

Remarks and conclusion

I would like to add a few comments about this model. First of all, it may seem at first that some of the five behavioural norms are incompatible with each other. How can someone be sociable and aggressive (=negative pacifism) at the same time? Well, sometimes people lose their mind. Even a very sympathetic person can become violent depending on the situation (if he feels threatened or furious for example). At the scale of the society, whole groups of usually calm and humanist persons can suddenly become aggressive if they feel oppressed: that is what happens during revolutions or civil wars.

If we look back at the model, an agent which will have a high (positive) value for sociability and a low (negative) one for pacifism will treat both interactions as two possible strategic choices and will choose between these two choices depending on the situation: if it faces a friend or someone it has not seen before, he will talk to him; if it does not like its partner (negative relationship), it will fight him.

There are no more incompatibilities between generosity and dishonesty (=negative honesty). Very often, dishonest people like to share their loot with their family or their friends. An extreme example would be Robin Hood which was dishonest in order to be generous (at least in the legend).

I would also like to come back on the prosocial emotions modelled (shame/pride and guilt/comfort). I decided to involve these emotions in the model because, without the prosocial emotions, we would all be sociopaths, and human societies would not exist. Sociopaths have no mental deficit except that they can hardly (or even not at all) experience shame, guilt or remorse. They comprise only 3% of the male population in the USA, but account for approximately 20% of the United States' prison population. The following example, proposed by Gneezy and Rustichini in 2000, will illustrate the influence that prosocial emotions can have on social behaviour:

Parents are sometimes late in picking up their children at day care centres. In Haifa, a fine was imposed for lateness at some randomly chosen centres while in a control group of centres, no fine was imposed. The natural expectation for this experiment was that punctuality would improve at the first group of centres. However, the contrary happened: the parents responded to the fine by even greater lateness (the number of people picking up their children late more than doubled). After 16 weeks, the fine was revoked, but the increased tardiness persisted. Meanwhile, in the control group of centres, the behaviour of the parents had stayed the same for the whole duration of the experiment.

The authors of the study explain that the fine converted lateness from the violation of a moral obligation which might have occasioned the feeling of guilt, to a choice with a price that many parents were willing to pay. Revoking the fine did not restore the initial situation but rather lowered the price of lateness to zero. Because of the absence of guilt in the interaction, the level of cooperation between the parents and the employees of the day-care centres decreased. Other experiments done by Bowles and Gintis in the context of public good games have shown that shame is also an emotion that increases the cooperation between individuals.

Guilt and shame are probably the most important emotions contributing to prosocial behaviour. They both involve the violation of a norm. However, shame is induced by others knowing about the violation and making their displeasure known by the violator whereas guilt is an internal feeling of discomfort caused by the fact of having done something wrong by one's own norm. They both function as the basic emotion "pain" in providing guidelines for action that bypass the cognitive optimising process. They induce a simple message: whatever you did, undo it if possible and never do it again.

With this in mind, by modelling shame and guilt (with their opposites, pride and comfort), my goal was to increase the cooperation in my system as well as making my agents' reactions more realistic.

In the next chapter, I will expose the results obtained with this model. We will study emergent patterns and some experiments will show us the influence of the model parameters on the behaviour of the system. We will also be interested in what happens during the education of a child and in how behavioural patterns are transmitted generation after generation.

Chapter 2: Autonomous society

Introduction

Now that I have exposed how the model works, it is time for me to show the results obtained with it. In this chapter, we will work with the model described in chapter 1. The user has this far no way to interact with his virtual society except by observing it. In chapter 3, we will add additions to this model, in order for the user to simulate "external events" and to influence the evolution of his society.

But for the moment, the societies we will simulate will evolve in a completely autonomous way. Of course, I will change some parameters of the model in my experiments in order to show their influence on the society's evolution. However, I will always make these changes before launching the simulation, while the population is still randomly generated. Once the simulation is launched, there are no more external perturbations.

At first, I will show and explain the emergent patterns observed with the default settings. Then, the influence of α, the weight of norms in the decision-making process (see Chapter 1, Part 4.3), will be studied: by modifying this parameter, we will be able to work with agents which follow more or less blindly their norms. After that, we will have a closer look on the influence of what we called earlier the external feedback (other agents' judgement about an interaction just performed). Finally, I will finish in focusing on the relationship between parents and children: we will look inside the households. We will follow families in order to see how the behavioural patterns are transmitted.

1. Patterns obtained with default settings

In this part, we will look at the results obtained if we run the simulation with default settings (see Chapter 1). We will look at the patterns observed in this case in order to use them as a reference later when we will change the parameters. We will use screenshots taken from a simulation (which is quite representative of the usual behaviour of the system) in order to illustrate our description of the dynamics. Each simulation is unique and the screenshots used do not show what *will* happen in the system but what *is likely* to: variations may appear with other experiments.

1.1. Awareness and Integration

During the first hundreds of iteration, the evolution of these two parameters will be quite chaotic. Indeed, there are a lot of free houses available to the agents: they will move a lot in the system. Moreover, the parameters are chosen at random during the creation of the city. Therefore, we will observe very different people living near from each other and this high gradient of behaviour will imply important modifications of the parameters.

Around 500 iterations (that is to say after ten years in virtual time), we begin to observe the first hints of emerging cultures:

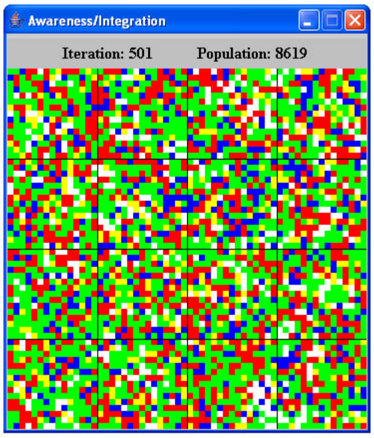
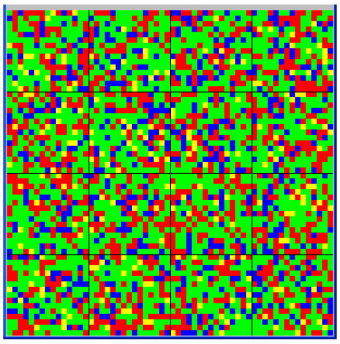
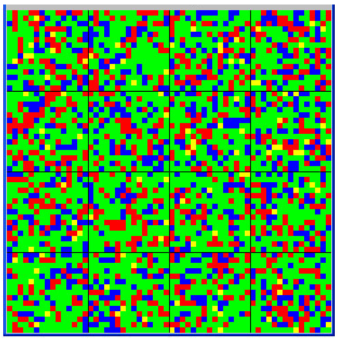
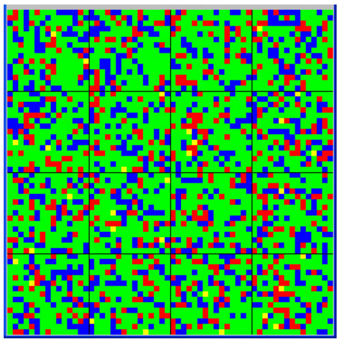
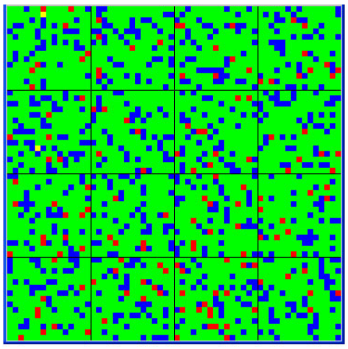
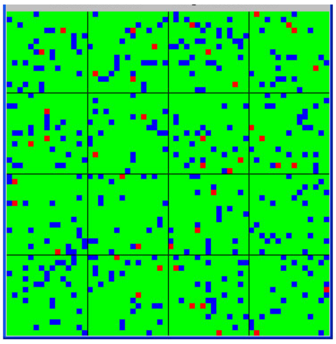


Fig. 14: Awareness/Integration at iteration ≈ 500.

We can see at once that the "green" culture, that is to say the set of people both aware and integrated, is expanding while the "yellow" one (people indoctrinated but well integrated) is disappearing. This slow disappearance is easily explainable by the model equations. Indeed, the more people become integrated, the more they are likely to talk with their interaction partners. Thus, since the "talk" interaction has positive effects on the awareness of both agents talking, the more agents are integrated, the more they talk, the more they become aware.

However, it will take some time before these yellow households all disappear. Some isolated yellow households will stay and resist during thousands of iterations until the culture completely disappear. Meanwhile, the green patterns will continue to expand themselves, but in a slower fashion. Indeed, now that the yellow patterns are becoming scarcer and scarcer, the green patterns must absorb households being currently blue or red. Absorbing red households is tougher: "red" people have a state of mind completely opposed to the "green" ones (a red colour represents a household the inhabitants of which are, on average, indoctrinated and traumatized while a green one is used for a good average awareness and integration). Moreover, "red" people have often a totally anti social behaviour, and this behaviour will have a strong influence on "green" people, making them more indoctrinated and traumatized. Thus, what will happen is that green and red patterns will compete for the domination of the system. When an agent goes from a red pattern to the green one (and respectively), he will have first to become blue (intermediate state) before changing totally his state of mind. We will thus have two main cultures ("green" and "red") fighting for complete domination of the system, a very present third one (blue) which represents the agents not yet belonging to one of the main cultures (this culture is often found at the boundaries between red and green patterns), and finally a very marginal culture (yellow) which regroups only a few individuals.

Here is the evolution of these parameters from iteration 1000 (twenty years) to iteration 8000 (one century and a half):

What we can see is that the system converges towards a situation where everybody in the system becomes very aware. The level of integration is also quite high since only a minority of agents are traumatized. This global behaviour is a bit simple if we compare with what can be observed in real life. However, we must remember that the model is a very simple abstraction of reality. Thus, even if we have not a very realistic asymptotic result (in order to be realistic, we should have kept a non negligible proportion for each colour at the end), this one is however good enough: we have to remember that the society we model will evolve in a very privileged way. Indeed, it evolves in an autonomous fashion, without external events, and without any contacts with other societies (no wars, no global economy, etc…). With this in mind and the fact that the simulation has been launched with the same parameters for all districts, it is finally quite logical to encounter such a result. Indeed, the uniformity of results comes from the uniformity of parameters, while an attraction towards awareness and integration is after all the aim of all societies…

However crude, this result is thus plausible. We will see in Chapter 4 some ways of simulating more realistic conditions of evolution for the society, conditions which will give us less homogenous asymptotic behaviours, but for now we will continue in analysing the patterns obtained with autonomous simulations and have a closer look to the first group of behavioural norms: sociability/misanthropy and pacifism/aggression.

1.2. Sociability/Misanthropy and Pacifism/Aggression

Here again, during the first hundreds of iterations, these parameters will have a very chaotic behaviour (the reasons are the same as the one given in part 1.1). The situation begins to stabilize itself around iteration 500:

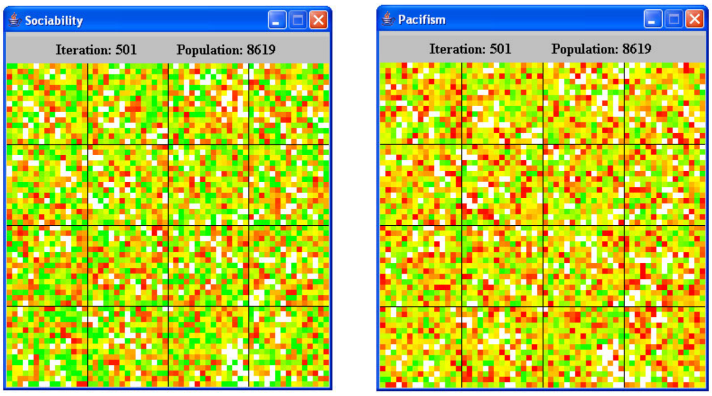


Fig. 15: Sociability and pacifism at iteration ≈ 500

The global level of sociability will increase as a result of the phenomenon described in the previous parts. Indeed, I explained in the previous part that, indoctrinated but well integrated people were inclined towards talking and that, by talking, their awareness would increase. However, this has also a retroactive effect on the sociability: the more people become aware, the more they talk (and the more they are likely to communicate successfully with others). Thus, the sociability of these agents will increase.

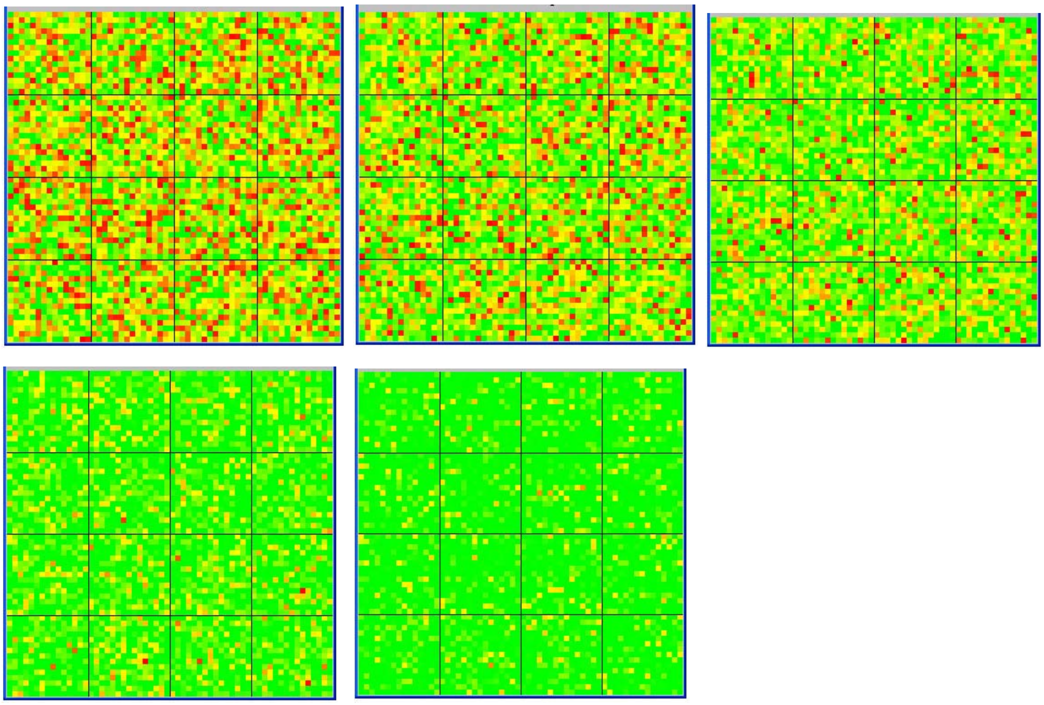
Moreover, aware and integrated agents have generally no problem in talking and they often have a sociable behaviour. This way, the families living near such agents will be attracted towards a sociable behaviour as well because of the external feedback effects. And since the number of aware and integrated agents increases at the beginning, so does the global sociability.

The evolution of pacifism however will be quite different. Clusters of aggressive people will appear here and there and they will expand themselves but the situation is still very heterogeneous and we will have to wait in order to know which culture is going to dominate the others.

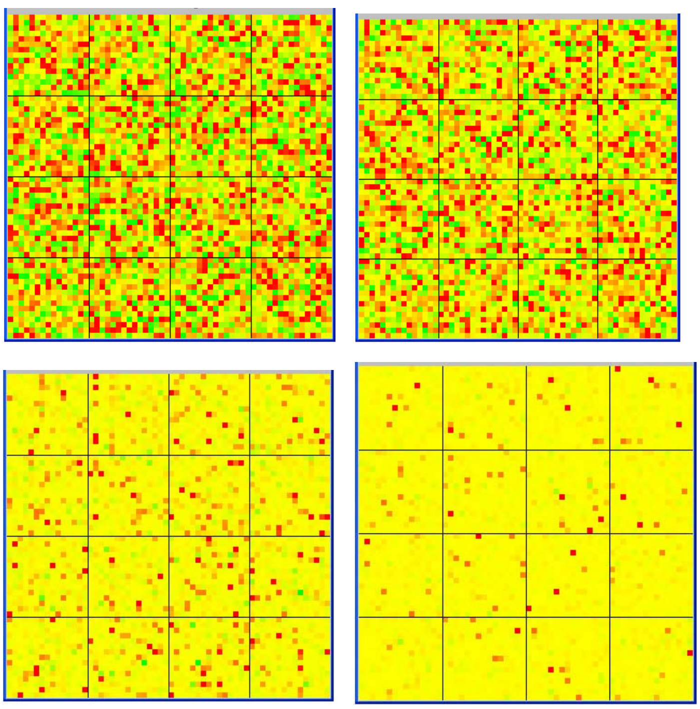
If we look at the compared visualisation, we can see that the majority of people are at least a bit aggressive (red and blue patterns) but half of these aggressive people are also sociable so the situation may quickly change.

Here is the evolution of these parameters from iteration 1000 (twenty years) to iteration 8000 (one century and a half).

Sociability:



Pacifism:



At the first glance, we can see that sociability quickly increases and that, asymptotically, most of the agents will be totally sociable. Only a few misanthropic people stay in the system. That means that the agents will often consider talking as being a very good strategic choice when they choose an interaction. Their default behaviour when they meet a stranger will be to talk to him, and not to fight him. This is good, since it means that, naturally, the city evolved towards a state where people communicate enormously with each others and try to meet new people.

However, aggression does not disappear completely from the system. Even if our agents are more and more aware and integrated, they do not reject aggression. Most of them do not adopt it either: there are generally few "red" agents after tens of thousands of iterations and these agents live isolated from each other. This result is also quite near from what can be observed in real life. Indeed, even in the most civilised societies, violence is never totally rejected: military power is still a very important factor during diplomatic negotiations for example. Furthermore, we can observe some isolated red and green plots remaining in the system until the end. This is again quite interesting: in all societies, whatever the global level of education and civilisation, there are always some violent individuals as well as some fervent pacifists who would absolutely refuse to fight even if their country was at war. Thus, having a society which is quite neutral towards pacifism with some isolated extremes seems quite realistic.

If we look at the "final" situation visualising sociability and pacifism at the same time, we obtain the following result:

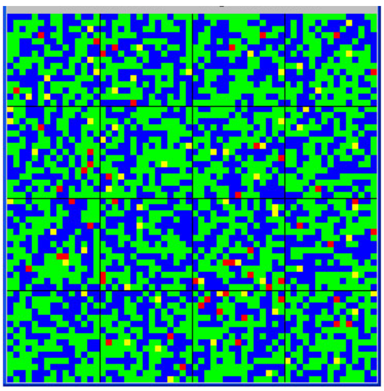


Fig. 16: Sociability/Pacifism at iteration = 8000.

We have two main groups of agents: the sociable and (a bit) pacifist people and the sociable and (a bit) aggressive ones. But if we look at the results for pacifism only, we can see that there are not important differences between these two groups. Thus, we cannot describe these groups as "cultures". They are two groups belonging to the same culture but having slightly different opinions: we could compare them to two political parties.

With some other experiments made in the same conditions and with the same parameter, one could have found a more pacifist or a more aggressive society. The final average value of this parameter varies from a simulation to another but its absolute value never exceeds 0,2. However, in the case of sociability, we will always have this convergence towards one.

1.3. Generosity/Selfishness and Honesty/Dishonesty

After the now usual chaotic sequence of movements inside the city, the situation is the following at 500 iterations:

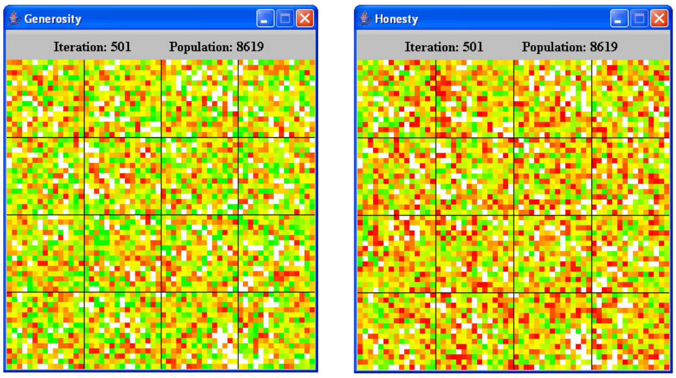


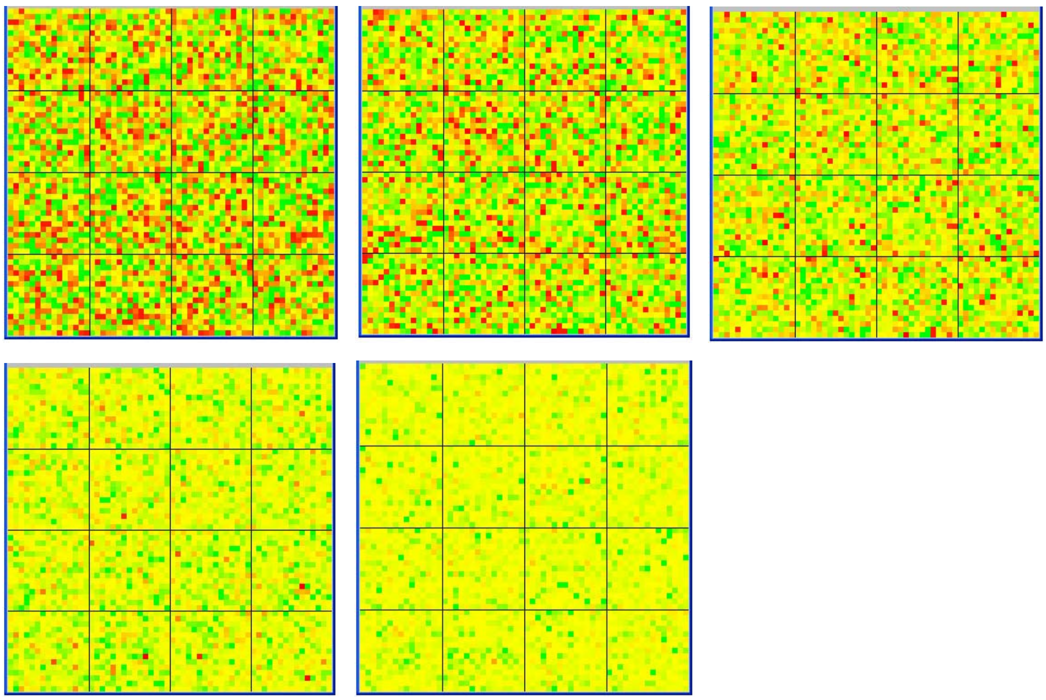
Fig. 17: Generosity and honesty at iteration ≈ 500

After 500 iterations, generosity is still a very heterogeneous norm: the generous people seem to be dominant but the average behaviour is neutrality. However, the average value for generosity slowly increases. This is again a consequence of the global increase in awareness. The agents being more conscious of what surrounds them, they will help people more often.

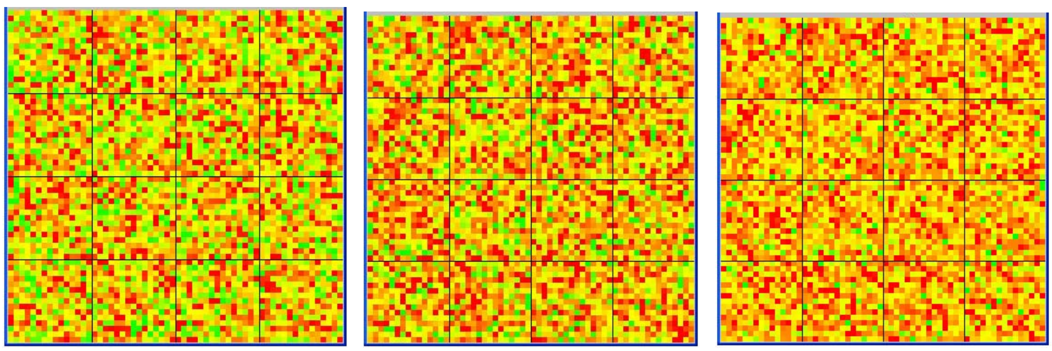
On the other hand, the evolution of honesty is quite clear: the agents are becoming more and more dishonest. This is not incompatible with the increase of awareness since the fact of choosing to steal instead of any other interaction will be more driven by integration than by awareness.

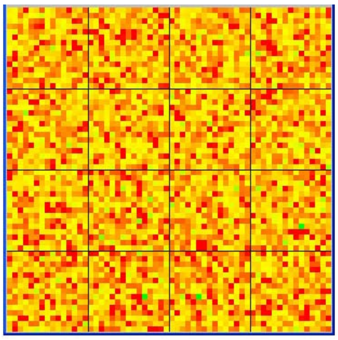
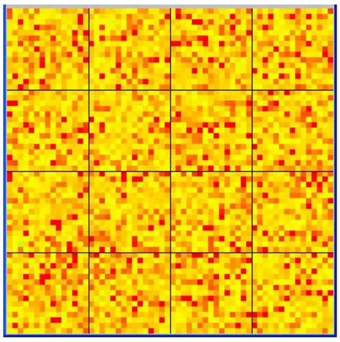
Honesty will continue to decrease rapidly and we will soon obtain a dishonest society. However, what is paradoxical is that people are not selfish: we will obtain a lot of agents which are both generous and dishonest. Here is the evolution of these parameters between iteration 1000 and 8000:

Generosity:



Honesty:



If the results obtained for generosity are quite realistic and compatible with the ones observed for awareness and integration, it is surprising to find so many dishonest agents in the system. It will be shown in chapter 3 that with the help of teachers and education, it is possible to make the agents behave honestly. However, during autonomous simulations, this quite low honesty always appears. In fact, the problem with dishonesty is that it is a very contagious phenomenon. When the global level of honesty is high, the society will not degenerate in a lawless situation. However, as soon as there are too many dishonest people, dishonesty begins to spread everywhere in the system and it is very difficult to stop the propagation.

At the scale of the model, this phenomenon is quite logical. Indeed, when agents repeatedly steal they will:

1. Have bad relationships with their neighbours
2. Become richer and richer
3. Decrease more and more the satisfaction and the integration of their neighbours

Now, suppose that an agent who has been stolen in the past has to interact with a "thief". The agent will be very likely to steal him because he is interacting with somebody he does not like (bad relationship) and who is richer than he is. That is why dishonesty propagates very quickly in the system.

This situation is maybe crude but reflects a sociologic fact: in order for a group of people to be honest with each other, they must trust each other. As soon as one individual cheats the others, there is no more trust within the group and the situation deteriorates quickly.

After 8000 iterations, almost everyone is the system is at least a bit dishonest, but the majority of them are generous as we can see on the Generosity/Honesty window:

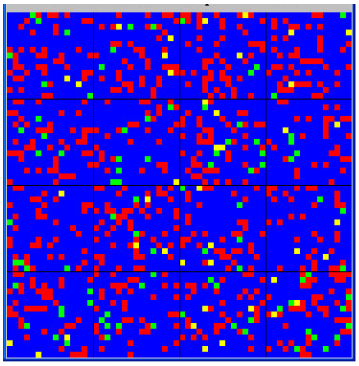


Fig. 18: Generosity/Honesty at iteration = 8000.

1.4. Flirtation/Puritanism

<u>Situation after 500 iterations:</u>

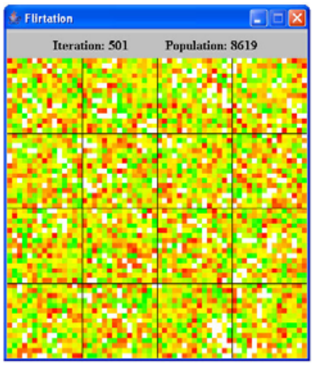
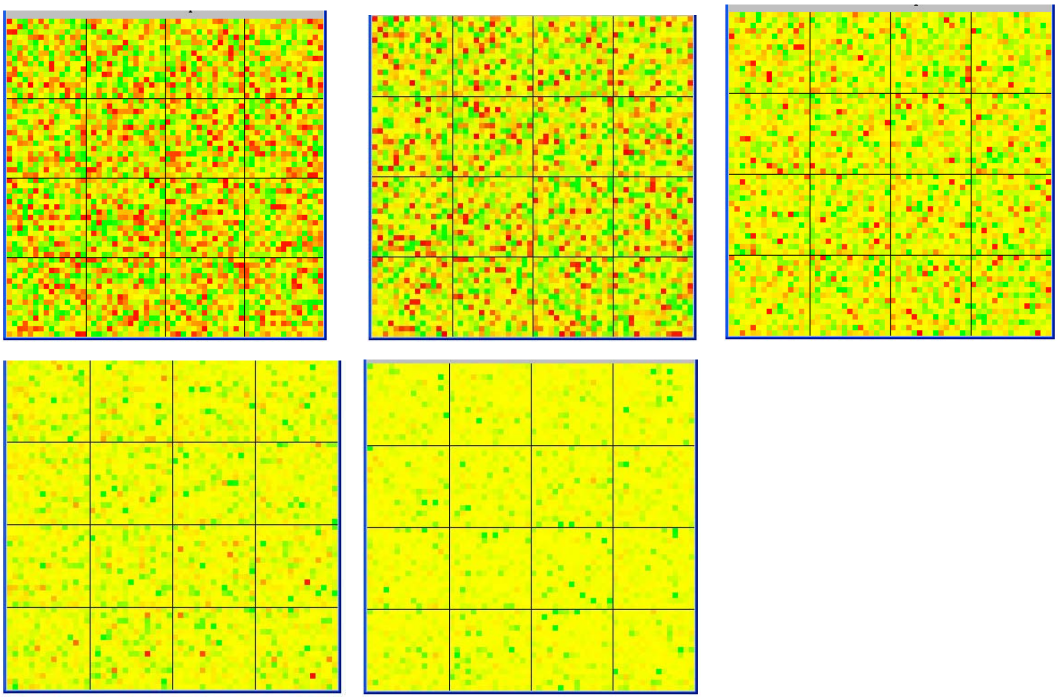


Fig. 19: Situation after 500 iterations for flirtation.

We can see at once that the level of flirtation is increasing in the system at the beginning. Moreover, what's very interesting is that there are very few totally puritan people. Even the redder spots at the creation are decreasing in intensity. Most agents are quite neutral (yellow or orange), but there is also a quite high number of flirtatious agents. There is not any dominant culture at 500 iterations but there is clearly one culture which is far weaker than the others: the totally puritan agents.

Evolution between iteration 1000 and iteration 10000:



This evolution shows that our first impression was good: the puritans are indeed decreasing in number and the society converges towards a state of moderate flirtation. The final result found here is a bit different from what generally happens with other experiments. It is not scarce to find an average value a bit higher than what we have here.

This result is compatible with the other results found for this simulation. Indeed, almost everybody is aware and integrated in the system, which is a situation which encourages seduction. However, seduction can bring trauma if the agent is committing adultery. Thus, the agents will be attracted towards flirtation until they get married and after that, their level of flirtation will often decrease since this interaction will imply bad effects if performed with the wrong person. However, flirtation will often stay positive, because performing it with one's companion has positive effects for both agents. That is why we have this final state of moderate flirtation.

1.5. Overview

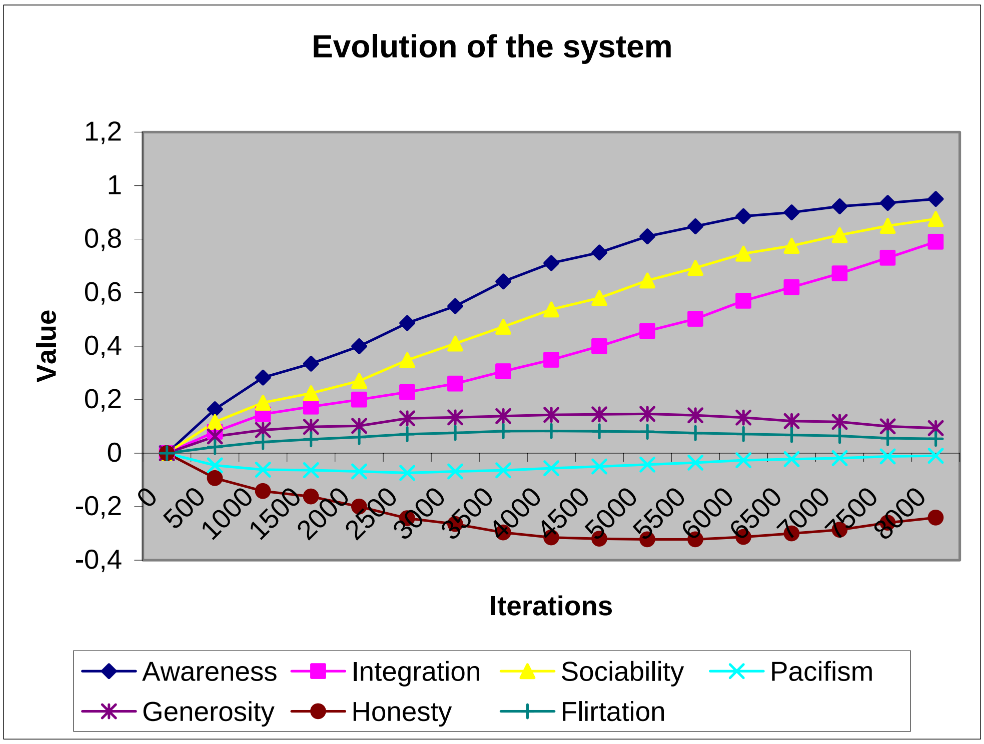


Fig. 20: Evolution of the parameters of the system during the experiment.

2. The weight of norms in the decision making process

When an agent interacts with someone, he decides which interaction to perform by giving a mark to each of them and performing the one having received the best mark. The mark received will depend on two parameters: the compatibility between situation he perceives and the interaction evaluated, and the opinion of the behavioural norm associated with this interaction. The weight which is given to the norm in this evaluation is given by α, a parameter editable by the user for each district (for more details about the decision making process, see Chapter 1, part 4.3).

When α is low ($\alpha < 0,2$), it is the compatibility between interaction and situation which will be the most important parameter in the decision-making process. Metaphorically speaking, the agents will react more by instinct than by reason. As a result, their behaviour will change often and their evolution will be quite chaotic. At the scale of the system, the patterns are less stable than with the default settings (α = 0,5) and the boundaries between them fluctuates quickly.

When α is high (>0,8), the behaviour of an agent will be driven by its norms. Its action will be very repetitive: he will often follow blindly the norm he prefers until the last of his days. As a result on the system, the patterns will be more stable but smaller, the whole system being more heterogeneous. Since the agents follow blindly their norms, it will be very difficult for a culture to "absorb" new elements, that is to say to make them change their mind. It is not scarce to observe agents which are fanatics about one norm, and which will not change their beliefs until they die.

The value of α has also an effect on the parameters of the model. Indeed, when α increases:

¤ Integration increases more slowly in the system,
¤ Sociability increases more quickly,
¤ Honesty decreases more quickly.

Moreover, when α is low, the global level of flirtation will increase in the society; when α is high however, there is no side-effect on flirtation (see fig. 25).

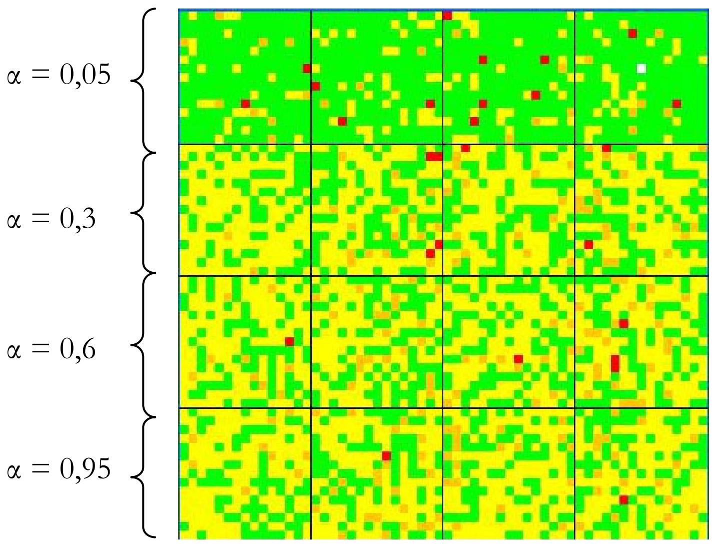


Fig. 25: Flirtation level in the society for different values of α.

This influence of alpha on parameters is an emergent phenomenon which cannot be deduced from the equations of the model. However, thanks to alpha, we are able to control the stability of our system, which will be very interesting when we will add external events to the simulation (see Chapter 3).

3. Influence of external feedback

After having performed an interaction, an agent evaluates the result of this interaction. If he is happy (resp. unhappy) about what he has just done, the level of the norm associated with the interaction performed will increase (resp. decrease). In order to evaluate the results of the interaction, two parameters will be used: the payoff for the interaction and the judgement of eventual witnesses about what the agent has just performed (or "external feedback"). The weight which is given to others' judgement is called β, this is an editable parameter (for more details about norm adjustment, see Chapter 1; part 4.6). By default, β is quite high (= 0,66). In this chapter we will see what happens if this value is decreased.

I have already said that it would be unrealistic to set β to a low value. Indeed, integration of norms should be quite independent from the perceived well-being. The reason is that people value norms as ends rather than means, which involves that they do not adopt norms for the well-being they bring. However, setting β to a low value will have other effects on the system. The external feedback is an important process in the model which will increase the global stability of the system.

Indeed, the external feedback works has a retroaction. Retroaction is often used in order to stabilize a system. For example, in the human body, the feminine hormonal system is strongly regulated by retroactive hormonal messages: the producers of hormones send even messages to themselves in order to control their rate of production. Furthermore, in the field of automatics, retroaction filters are a common solution in order to stabilize a system:

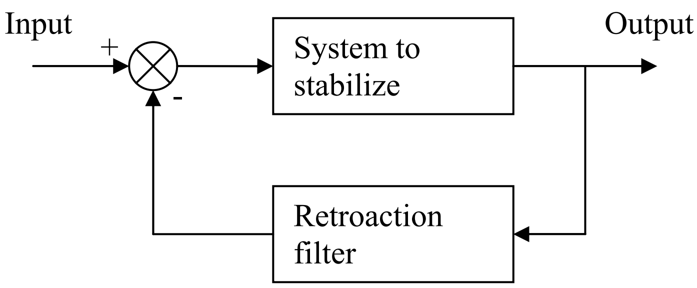


Fig. 26: Stabilisation of a system by retroaction filter.

Therefore, when we decrease the weight of external feedback the system will become less stable and some chaotic behaviour can appear at the boundaries between two important cultures. Moreover, the external feedback is a way of modelling shame and pride. These two prosocial emotions increase the cooperation inside human societies. What happens with a low β is that the patterns observed are smaller and that the city is very heterogeneous. The cooperation between agents decreasing, it takes more time for cultures to emerge.

Thus, in a (virtual) society where individuals do not care to the judgement of others, each agent behaves in a very egocentric way. Their beliefs often change and are completely independent from the ones of their neighbours. This phenomenon reflects well what would happen to a human society without shame. This experiment shows the importance of prosocial emotions in human social organisations. Without them, the cooperation between individuals strongly decreases.

4. Transmission of norms

Now that we have seen what happens at the scale of the city during a simulation, we will change our scale of observation and look at what happens inside a household. It appears that the behaviour of the children will be directly influenced by the behaviour of their parents: it is very scarce to find children which behave totally differently from the way their parents do.

The examples below show us the evolution of the levels of sociability and honesty in a family. As we can see, the parents will easily transmit their norms to their children. However, the latter are not the exact copy of their parents: their behaviours are similar, but different.

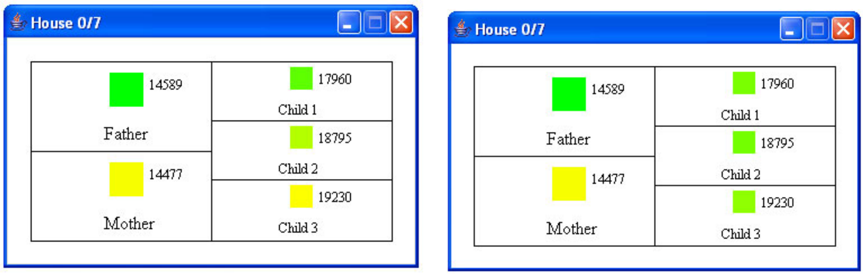


Fig. 27: Level of sociability in a family at 4900 (left) and 5400 (right) iterations

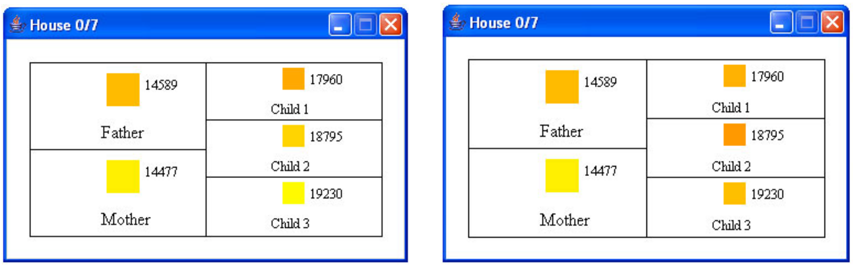
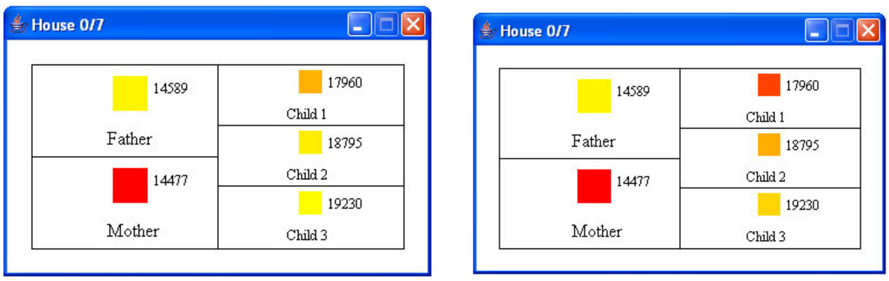


Fig. 28: Level of sociability in a family at 4900 (left) and 5400 (right) iterations

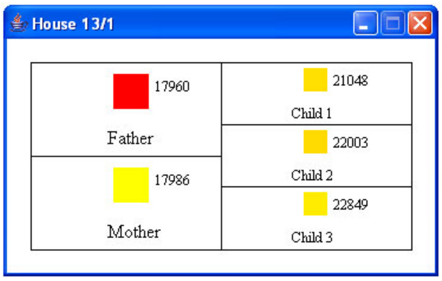
I followed this family for some time. Indeed, the mother was beating her children and I wanted to know if this behaviour was to be reproduced by the children. What happened was quite near from what can be observed in reality: this behavioural was indeed followed by agent 17960, which began to beat his children once he had grown up. The following screenshots show us the evolution of the level of pacifism/aggression in this family:

¤ Iteration 4900 to 5400. Agent 17960 still lives with its parents and is beaten by his mother:



We can see that, as a result of the bad treatment inflicted by the mother, the aggression of the children increases.

¤ Iteration 5800 to 6800. Agent 17960 has left home and is now living with his wife and children. He will reproduce the bad treatments received in his youth and beat his kids:



¤ However, thanks to the influence of their mother, the children of 17960 will not reproduce this pattern.

In this example, the transmission of aggressive behaviour was quickly stopped. However, this is not always the case. Agent 18795, the brother of 17960 was also violent with his children. In his case, the transmission of violent behaviour lasted longer and I had to wait for four generations before it disappeared totally in this family.

On average, the transmission of a specific behaviour like children-beating will last two or three generations. After this period, the agents obtained are usually different enough from their ancestors for having different patterns of behaviour. Moreover, this transmission of behaviour will often happen with one or two of the children, but scarcely with all of them. Therefore, when behaviour is transmitted in the family tree, only some of the branches will be affected, not all of them.

When a behavioural pattern disappears from a family tree, it can of course reappear later. However, this will not be because of heredity but because of external factors (that is to say, factors having their origin outside the family).

Remarks and conclusion

When we let the simulations run in an autonomous, we obtain a society of aware and integrated agents. These agents are sociable, moderately generous and flirtatious, neither pacifist nor aggressive, and quite dishonest. By changing the weight of norms in the decision making process, we are able to control the stability of the system and we have seen that the system is stabilized by the phenomenon of shame and pride modelled at the end of an interaction. Finally, we have seen how norms were transmitted from a generation to another and how hereditary behavioural patterns would appear in family trees.

However, the results we obtain are a bit crude. They lack diversity in order to be totally realistic. Moreover, it seems quite optimistic to obtain societies where almost every individual is perfectly aware and integrated. Indeed, our virtual societies have evolved so far in perfect situations: no perturbation, no tyranny of a powerful ruler, no real difference between the different agents, etc… Even if we observe some interesting phenomena with our initial model the behaviour of the system is not rich enough for us to simulate less utopian situations. For example, in the simulations we ran so far, the conditions of living are the same in all the districts, which is unlikely to happen in reality.

Moreover, the user still does not have enough control on his virtual societies. The parameters he can access are not sufficient to attract the agents toward a kind of behaviour he has chosen. If we want to obtain more results than the ones we have described, we are compelled to provide other functionalities in order to interact with the population of the city.

In order to simulate less simple situations and to have more control on the evolution of the system, I will now improve the model described in Chapter 1. I will add functionalities to the software in order to simulate external events and local diversity. With these new tools, we will obtain societies the dynamics of which are richer, with several dominant cultures coexisting asymptotically. The user will also be able to control better their evolution and to attract the agents toward a particular behaviour.

Chapter 3: Society with external influence

Introduction

In the last chapter, we have seen that if we let the society evolve in a totally autonomous fashion, we get very uniform results. In order to bring more diversity in the system, we will now let the user have an influence on it. He will be able to simulate external events and to control the evolution of the society in a far more powerful way.

Two new concepts will thus be added to the initial model: the concept of teachers and the one of mass trauma events. Teachers will try to attract the social behaviour towards a certain state. They will have long lasting and long term effects. On the contrary, a mass trauma event will have immediate effects. These events aim at simulating a state of global trauma within the society, because of a disaster. For example, the terrorist attacks in the public transports in London this year would be modelled in my system by a mass trauma event.

With these two new concepts, the simulations will give us new results. The teachers will play a very important role in the evolution of the society, as will be shown in part 2.

Finally, we will see how the simulated societies react after a mass trauma event. Depending on the parameters of the simulation, the consequences of such an event will be far different: some societies will quickly overcome the trauma while others will suffer greatly from it.

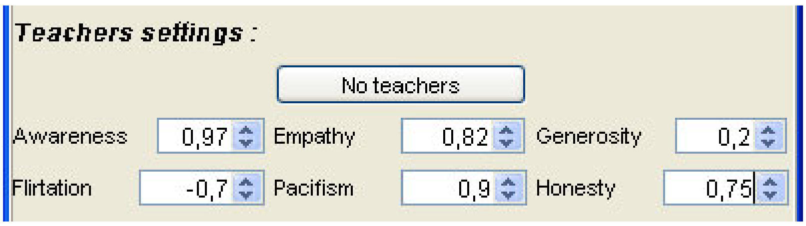
1. Additions to the model

1.1. Teachers

Teachers will play a role in the system by teaching the children when they are at school. Thus, when they are involved in the system, parents are not the only one to transmit their norms to their children anymore. The education of a child will be made not only at home, but also at school.

Once a week (twice for a teenager), the agents that still go to school receive a teaching from the teacher. This teaching works exactly in the same way that what was done previously when parents taught their children a norm (see Chapter 1; part 3.1). The only difference is that the parameters of the teachers are editable by the user.

Indeed, thanks to the settings window, the user will be able to define, for each district, the parameters of the teachers:



The awareness parameter will be used to define the power of persuasion of the teacher (remember that the more an agent is aware, the more he is likely to successfully transmit its norms). The other ones represent the norms that the teacher will try to transmit to its students.

By involving teachers he has "created" in its virtual society, the user will be able to "attract" the members of this society towards a certain behavioural pattern. Teachers can encourage students to have any behaviour (pro or anti social). This way, the user can decide if his teachers will be some gurus preaching violence and revolution or some wise men preaching sociability and pacifism. Any kind of behaviour can be modelled.

However, teachers will not totally control the dynamics of the society: they are attractors but will not be followed blindly (see part 2).

1.2. Mass trauma events

Sometimes, external disasters can happen in a society, and these events will traumatize the whole population. They can be natural (earthquake, tsunami, flood, epidemic, etc…) or not (terrorist attack, declaration of war, nuclear accident, etc…). In both cases, they will imply a quick reaction of trauma which takes place almost instantly. Then, the society will gradually overcome this trauma and the situation will slowly evolve back to an integrated society. However, during a non negligible period, the whole population will behave differently from usual and this will affect the whole dynamics of the social system.

In order to simulate such an event, the user has only to choose a few options and then click on a button:

The user first chooses the intensity of the trauma inflicted. Each agent will be modified in the following way:

¤ *Let bonus = max(0, integration/2).*
¤ *integration = integration – (1 – bonus)*intensity.*

Thus, as we can see from this equation, the more traumatized an agent is before this external event, the more trauma he will receive. This behaviour reflects what happens in real societies: such a phenomenon has been observed and proved by sociologists. The influence of these external events on the behaviour of the system will be detailed in part 3.

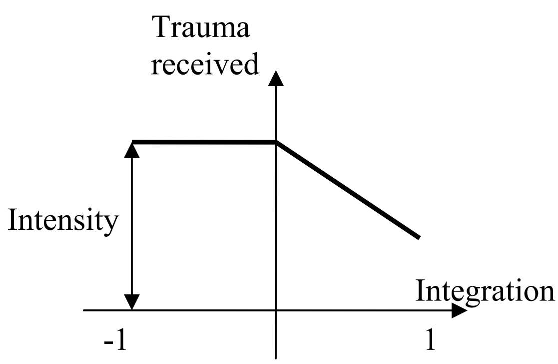


Fig. 29: Trauma received during an external event

2. Patterns obtained with teachers

The addition of teachers in the model will dramatically change the behaviour of our system. The societies simulated are more realistic and their dynamics are richer. Thanks to screenshots of simulations, I will show and comment how my societies evolve when they are influenced by the user.

First, I will present the results of experiments where the whole city is under the same influence. At the scale of the model, that means that the settings of our teachers will be the same in every district.

Then, I will present experiments where the parameters of the teachers will be different from a district to another. We will thus see what happens when various cultures coexists in the same system.

2.1. Same teachers in all districts

Since the teachers are attractors in my system, they are a way to control its evolution. Indeed, our teachers will transmit their norms to every children (and teenagers) in the system. When these children are grown up, their behaviour will be altered by the regular contacts they had with the teachers, even if they evolve differently from the behavioural model proposed by the teachers. This alteration in their norms will be transmitted to their children, who will go to school and be altered even more and then transmit this alteration to their children, etc… Thus, generation after generation, the norms of the agents in the system should gradually converge towards the ones of their teachers.

Therefore, if the user wants to obtain a society with specific values for each norm, one option for him is to put teachers (whose norms are set to the desired values) in each district of the city. By modifying the awareness of these teachers, he will be able to control the speed of convergence of the system. Indeed, the more aware teachers are, the more often they succeed in transmitting their norms.

This method for controlling the evolution of our societies works quite well. However, it is not perfect and some conditions must be respected in order to obtain the desired results. First of all, the teachers' awareness must be high enough. If the teachers are too indoctrinated, they will often fail to transmit their norms, and the effects of the transmission teachers/students will become negligible compared to the transmission parents/children. As a consequence, if a teacher has an awareness of -1, he will not influence the system at all (this will be used later in order to model "neutral" districts, that is to say districts not influenced by teachers).

Moreover, the power of teachers is limited. In order to be able to attract a society towards a certain model of behaviour, they must work with students (and with societies in general) whose norms are compatible with the ones they must teach. Therefore, if the user begins influencing its society since iteration 0, he will be able to attract his society towards almost any situation, because the teachers will influence a city where the average value for each norm is 0. However, if you bring a society towards a point where everybody is sociable and puritan, you will not be able to turn it back towards misanthropy and flirtation even by changing the parameters of the teachers. The transmission of norms teachers/students is powerful enough to control a neutral society but becomes negligible if applied in extremist ones.

Finally, some states are unstable in the system and they cannot be reached. For example, teachers with a sociability of 0 will not attract the society towards a neutral opinion about this norm: the global level of sociability will increase anyway. In order to reduce sociability, the only way is to reject it by using teachers having -1 as a value for this parameter. Even with this, the asymptotic behaviour of the city will not be totally misanthropic and as soon as we stop the action of the teachers, sociability will increase again.

Even with these restrictions, teachers are a powerful way to control the behaviour of our agents. By modifying the settings of teachers we will be able to obtain almost any kind of global behaviour.

2.2. "Clash" of cultures

We will now use a variation of the method described in the previous part. Instead of applying the same influence to the whole city, we will divide the city into groups of district. Each group will be influenced by different kinds of teachers. Thus, if we divide the society into three groups, we can have one part of the city were teachers are pacifist (pacifism = 1), a second part of the city where they are aggressive (pacifism=-1), and a last one where the teachers have no influence at all. Usually, we will use the following configurations in order to divide the city:

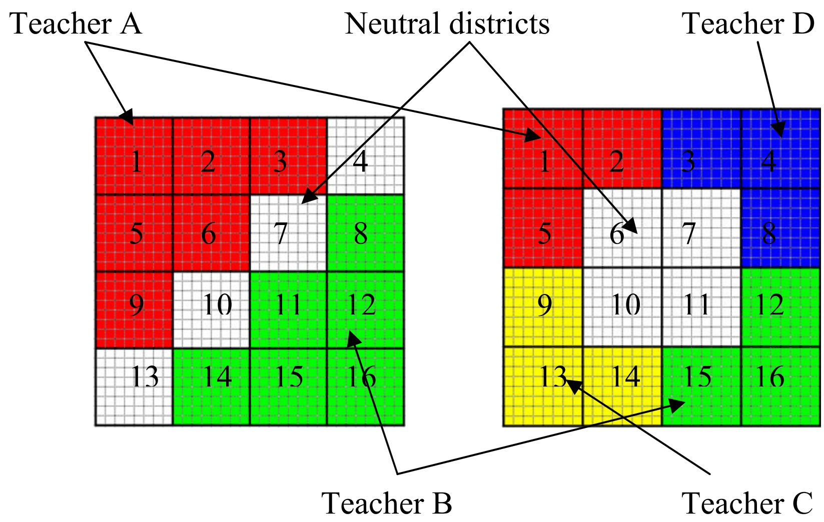


Fig. 30: Usual configurations for three and five groups

2.2.1. Prosocial behaviour versus antisocial behaviour

For the first experiment with teachers, we will use the configuration for three groups (group A, group B and neutral group). The teachers will have the following settings:

Group A:

Awareness = 1	Sociability = 1	Generosity = 1
Flirtation = 1	Pacifism = 1	Honesty = 1

Group B:

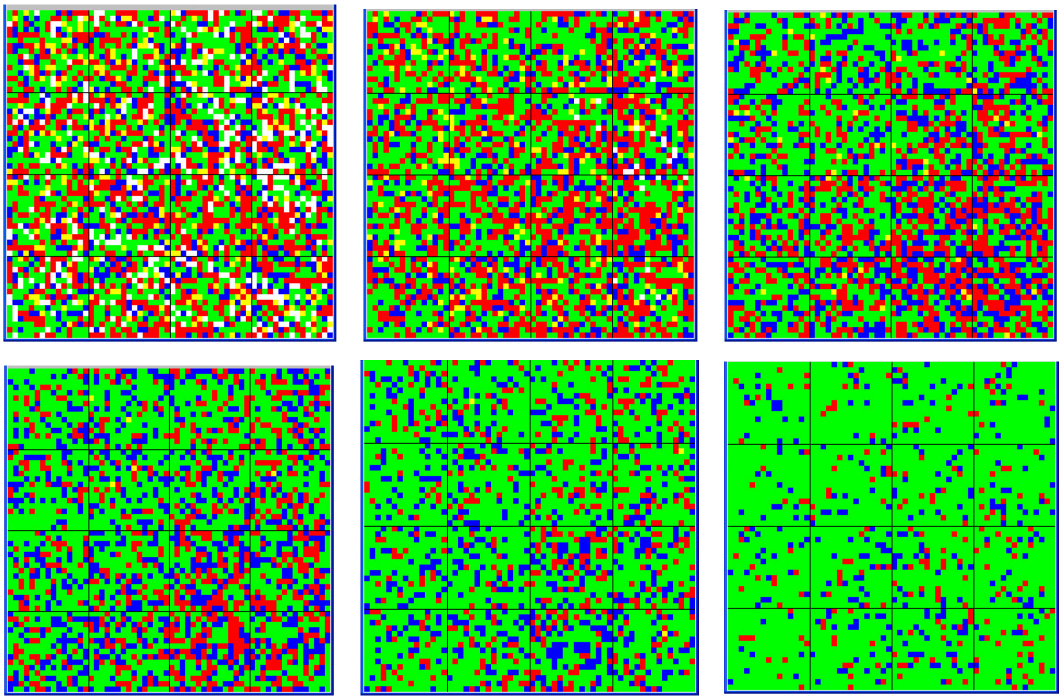
Awareness = 1	Sociability = -1	Generosity = -1
Flirtation = -1	Pacifism = -1	Honesty = -1

Neutral group:

Awareness = -1	Sociability = 0	Generosity = 0
Flirtation = 0	Pacifism = 0	Honesty = 0

Thus we will have a clash of culture between a society where the behaviour taught is completely prosocial (in the upper-left corner) and another one where the behaviour taught is totally antisocial (in the bottom-right one). Between them, on the diagonal, neutral groups will be a first place for our two cultures to expand themselves. Let's now have a look at the evolution of the system between iteration = 500 and iteration = 10000:

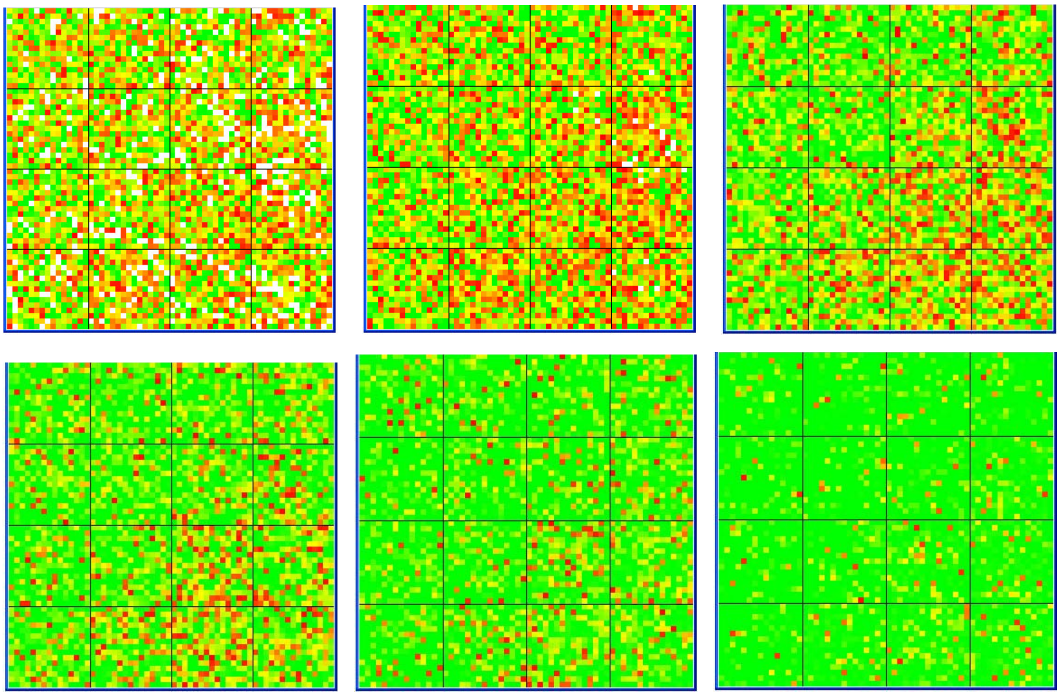
¤ Awareness/Integration:



This evolution is quite representative of what happens when a prosocial culture and an antisocial one coexist in the same city. The awareness and integration of the agents living in the prosocial part of the city will quickly increase. In the bottom-right part, the awareness and integration of the people will decrease, but in a slower fashion. Therefore, the neutral group of districts will be attracted towards awareness and integration, and the green culture will become dominant.

It will increase its dominance more and more and begin to enter into the initial territory of the antisocial culture (beginning with districts 8 and 14). Finally, asymptotically, the system evolves towards a state where almost every agent is aware and well integrated. However, the city is not perfectly uniform: there will always be isolated red or blue households here and there in the city, because of the effects of the teachers in the districts of group B. These effects are not strong enough to keep traumatized and indoctrinated an important number of agents, but they are sufficient to prevent the "green" culture to absorb the whole system.

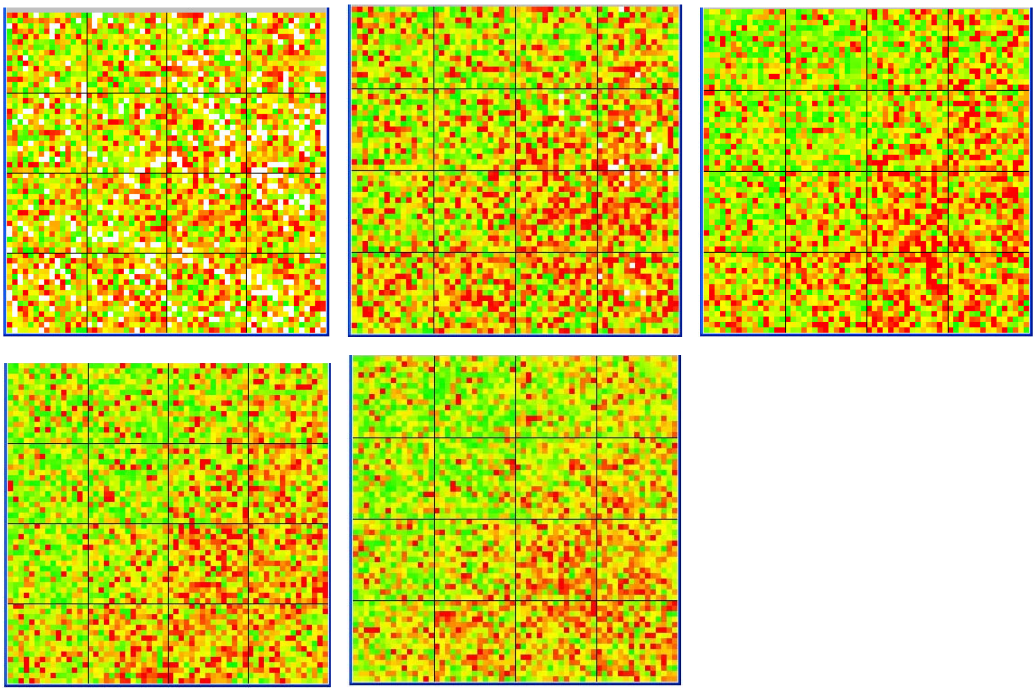
¤ Sociability:



When a misanthropic culture is opposed to a sociable one, it will always be attracted towards sociability. This is not surprising since the natural evolution of the system is towards sociability. Thus, rejecting sociability can be compared as swimming against the current whereas encouraging it would be swimming with the current. In these conditions, it becomes too difficult for the teacher of group B to maintain a low level of sociability. The group B will be rapidly be influenced by the group A. The presence of antisocial teachers slows down the expansion of sociability in the bottom-right corner but this is not sufficient. Finally, the system converges towards total sociability.

However, here again, there will never be uniformity in our system. Some isolated families of agents living in districts of group B will keep a misanthropic behaviour for the whole simulation because of the teachers' influence. Generally, the level of sociability in the districts of group B will be 10% lower than the average level in the city.

¤ Pacifism:

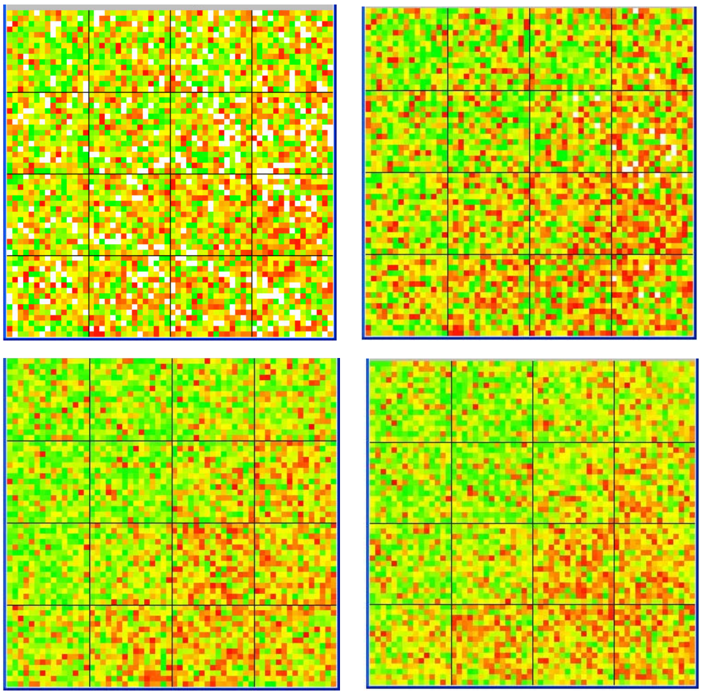


The case of pacifism is far different from the one of sociability. Here, we will not have a convergence towards a totally pacifist city. Indeed, the pacifist culture (in green) and the aggressive one (in red) have about the same strength of influence on the city. We can see in the screenshots that the pacifist culture is however a bit more powerful. Indeed, asymptotically, the dominant culture in the neutral group is the pacifist one: there is almost no observable difference between the agents living in the neutral group and the ones living in group A when it comes to pacifism.

However, the influence of the teachers in group B will be sufficient to keep a very low level of pacifism (that is to say a very high level of aggression) in the bottom-left corner of the city. Moreover, we can see that aggression never disappears from the districts of group A: the teachers of group B are powerful enough to prevent uniformity to occur in group A.

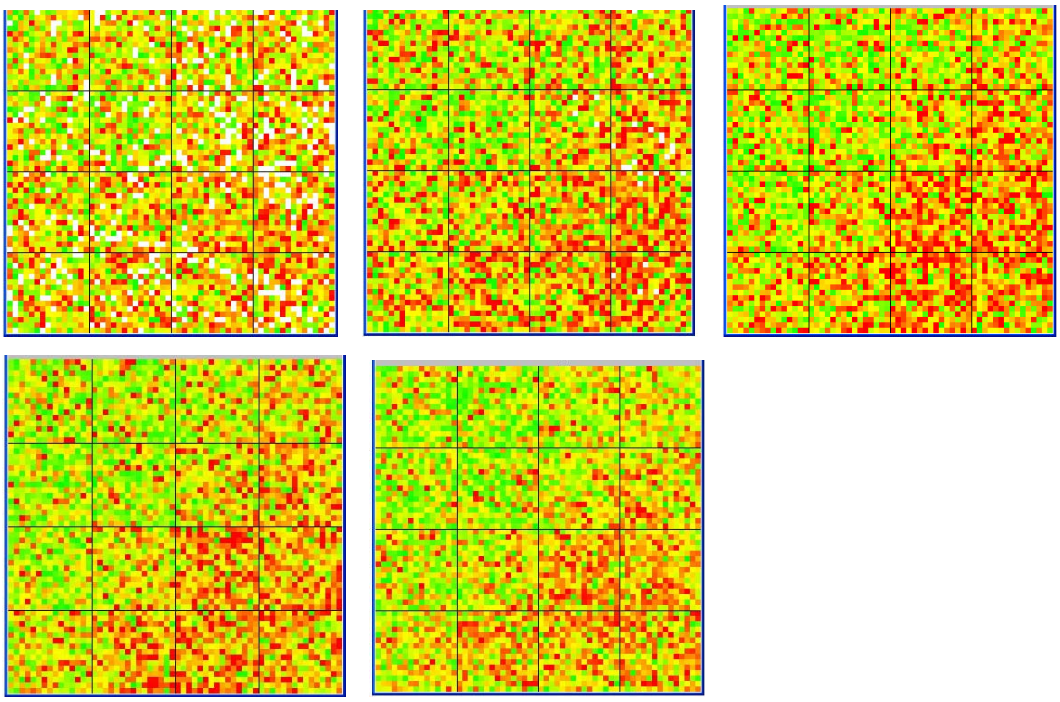
Therefore, we obtain at the end 10 districts where the dominant culture is pacifism but where aggressive agents can be found, 2 districts (8 and 14) where both cultures have equal influences, and 4 districts (11, 12, 15, 16) where the dominant culture is aggression but where pacifists agents can be found.

¤ Generosity:



Here, we have the same situation than with Pacifism/Aggression, but the power of generosity compared to selfishness is even greater than the one of pacifism compared to aggression. As a result, the level of generosity in the group B will be a bit lower. Apart from that, the general behaviour is the same.

¤ Honesty:



Here again, we have the same behaviour than with pacifism/aggression. Asymptotically however, the average level of honesty in the city is lower than the average level of pacifism. Indeed, in group A and in the neutral group, there will be more dishonest agents than aggressive ones. Dishonesty is a phenomenon that spreads quite quickly in the city and the situation in group A will be strongly influenced by the effects of teachers in group B.

¤ Flirtation:

Here, the evolution of the city is exactly the same than the one observed for generosity/selfishness. If the user skips from the visualisation of one norm to the other, the patterns observed will have almost exactly the same locations and intensities (we will not put the screen shots here, look at the screenshots for generosity in order to have an idea of the evolution). Therefore, asymptotically, we obtain a society dominated by flirtation, except in the bottom-right corner were the main culture is moderate puritanism.

Finally, we see that when a prosocial culture is opposed to an antisocial one, the prosocial culture will often become the dominant one in the system at the end of the simulation. However, the antisocial culture will never disappear completely and, as a consequence, the final state of our city will be quite homogeneous compared to what we had obtained with autonomous simulations.

2.2.2. Other experiments

I made other experiments with other settings for teachers and other configurations for groups. The following results appear:

¤ First, the power of a culture is directly linked to the awareness of the teachers transmitting it. The more aware the teachers are, the more powerful the culture is.

¤ Then, when four different cultural groups compete for influence in the city, there are often only three surviving cultures asymptotically, sometimes two. The other one(s) will be present in the system but its influence will be negligible.

¤ If we put too many different groups in the city, it becomes difficult to differentiate the different cultures. Indeed, each culture influencing the other ones and their initial territory being quite small (since there are many different cultures), the state we obtain at the end of the simulation is quite uniform: the cultures will merge with each other and what we obtain at the end are a few dominant cultures which are mixes of the initial cultures we put in the system.

¤ If the initial territory of a group is smaller than the others, the culture of this group is likely to disappear or, at least, not to expand itself outside the initial territory of this group: the bigger a group is, the more chances it has that its culture becomes dominant in the city.

3. Influence of mass trauma events

I will now explain what reaction can be observed in the system after a mass trauma event. Depending, on the intensity of the trauma inflicted, on the state of the system and on the parameters of the simulation, a society can be mentally destroyed by such an event, overcome it more or less easily.

3.1. Annoyance or cataclysm, the effect of intensity

By modifying the intensity of the external events inflicted to his society, the user will be able to simulate more or less important disaster. If the intensity of the trauma inflicted is below 0,2, the society will always overcome the event and the situation will be stabilized quite soon. The effects on the behaviour of the agents are negligible, and the perturbation is not important enough to change the way the system behaves asymptotically.

If we increase the intensity, between 0,2 and 0,8 the trauma inflicted will be important enough to change the way our agents behave during one generation. However, this trauma will be slowly forgotten with time and after two generations (40 years, 2000 iterations) the situation will generally be back to normal.

With an intensity above 0,8, the effect will depend a lot on the actual state of the society. If the average level of integration was high at the moment of the event, the society will overcome the trauma with time. However, if the average level of integration was already low, the society can evolve towards a situation where the agents have very limited mental abilities and have an antisocial behaviour. This situation can last some thousands of iterations before than the average level of integration returns back to a more suitable level, but it can also last forever if other mass trauma events are repeatedly inflicted.

3.2. Resistance to trauma through self organisation

When some mass trauma event is applied to a society, its evolution after the event will also strongly depend on its state at the moment of the perturbation. Some states are very resistant to trauma, and an external event applied to such a society is often quickly overcome. This is the case for example if we have a high average sociability in the society, with opinions at least neutral about pacifism and honesty. What happens in this case after a mass trauma event is that the agents, even traumatized, will continue to have a prosocial behaviour. Thus, their integration will increase quickly and return back to its former value. The contrary situation will also be very resistant to trauma: if we consider a society where the average behaviour is highly antisocial, such an event will only increase antisocial behaviour and the global dynamics of the system will not be strongly influenced.

However, interesting phenomena can be observed when we apply trauma to a less stable situation. For example, applying a high trauma just at the beginning of the simulation can lock a society to a state of trauma and antisocial behaviour. Indeed, since there are no emergent cultures at the beginning, the stabilizing effect of external feedback is quite low. At this moment, it is possible to perform important changes in the system. Therefore, by applying a trauma just at the beginning, we will put the system in a state where it will strongly be attracted towards antisocial behaviour. Sometimes, the natural evolution of the system will be sufficient to overcome the initial trauma and to obtain prosocial behaviour at the end; sometimes, the system will stay traumatized.

Other states are quite unstable: moderate honesty (the value for honesty decreases quickly after a mass trauma event if it was not high enough before the event), neutral pacifism and sociability, etc… However, what is interesting is that none of them can be accessed without the use of teachers. The societies obtained without teachers in the model resist generally very well to perturbations. By self organisation, the model will always evolve towards a state where it is resistant to a decrease of integration.

3.3. Influence of the parameters

By changing the parameters of the model, we will be able to modify the stability of these various states in order to make them more are less resistant to trauma. The first parameter which will have a strong effect on mass trauma events is α, the weight of norms in the decision making process.

Indeed, the higher α is, the more resistant to trauma the society will be. Agents with a high α are so driven by their norms that their behaviour will not change after a mass trauma event. Indeed, when they evaluate each interaction in order to choose one of them to perform, the trauma received will have a negligible effect on the mark given to each interaction. Thus, such agents will continue to behave as if nothing had happened in the system and their level of integration will return to their former value very quickly. I had already shown that α was a way to control the internal stability of the system (see Chapter 2 part 2). We see now that its stabilizing effects are more general: α also controls the resistance to perturbations. When α is too low ($\alpha < 0{,}2$) and if we inflict a mass trauma event of high intensity, the system can be locked in a state of antisocial behaviour and never return to a "normal" situation.

The other parameter to influence the evolution of a society after a mass trauma event is the amplitude of the modifications of the integration parameter (this parameter is called Δintegration). Indeed, the higher this parameter, the wider the hysteresis cycle associated with a modification of integration. Therefore, when Δintegration is high, it will be easy for an agent to gain a lot of awareness quite quickly and the system will return to its former state very quickly. However, if we gain some stability against external perturbations by increasing Δintegration, the internal stability will decrease until the behaviour of the system becomes chaotic for Δintegration > 50% (since the parameters evolve rapidly in this case, the behaviour of the agents is always changing and the system is never stable). Thus, the value for this parameter should always stay reasonably low.

Remarks and conclusion

With these two new tools, we are now able to simulate richer and more realistic societies. The number of scenarios to experiment is quasi infinite, and we have enough control on the system to bring the society to almost any state, with the condition that the scale of observation of this state is just a bit smaller than the scale of the city.

However, if we can attract the society towards a particular global state, our control on the system is not sufficient to influence directly a small neighbourhood. We will not be able to make quick modifications of the state of the city either: before that a significant change of behaviour occurs, a society must be influenced for a long period of time.

Finally, the various experiments presented in this chapter show that, if the conditions of evolution are not too extremes (reasonable amounts of trauma inflicted and teachers that are not totally evil), the self organisation process of the system favours prosocial behaviour and stability against external perturbations. This behaviour is realistic: human beings are, after all, social beings. We seek the communication with others and we cooperate with them in order to reach goals that would be unattainable by a human being alone.

Conclusion and future works

Human social organizations are one of the most complex systems studied by science. Societies are formed by human beings, whose behaviour is driven by a brain, which is formed by cells, and all these components are complex systems. Therefore, it is impossible to predict the evolution of a society by mathematical analysis. With the help of computerized simulations, however, a new way of studying societies is possible. With a rich enough model, it would be possible in theory to measure the reaction of a society to certain stimuli and therefore, to predict the future. Of course such a model only exists in science fiction (Isaac Asimov calls it "psychohistory" in its famous Foundation), but my project proves that even with a simple abstraction of reality and not a ultra realistic model of a human society, it is possible to obtain quite realistic results. Thanks to my system, we can obtain good first order approximations of actual situations whereas the model used for the simulation is very simplified.

In its present state however, my system clearly cannot be use in order to predict the future of a real society. Indeed, too many concepts are forgotten in my model: economics, nations, religions, political parties, scientific progress, artefacts (books or tools for example), law, etc… Moreover, the intelligence of our agents, their environment and their liberty of action are very limited. All we can obtain is some information about the way norms emerge and how they are transmitted from generation to another.

In order to use my system in more practical applications, I must therefore improve my model. First of all, in order to increase the performances of the simulator, one idea to improve the system would be to parallelize the algorithm used. This way, we could work with bigger societies. For example, we could modify the model in order to work with several cities interacting with each other, each city being managed by a different CPU.

Then, my real objective will be to add the notion of groups in the model. Indeed, my model considers that social norms have a very global meaning: agents do not make any distinctions between people when it comes to social norms. In reality, this will not be the case. Indeed, the norms are context-dependant. Depending on their partner's cultural group (its sex, age, race, nationality, religion, political ideas, etc…) they will not have the same behaviour. Religious extremists for example can be totally sociable with people of the same religion but are always violent with the ones they consider as "heretics".

I had the opportunity to discuss with Kelvin Au, who also worked on an expansion of the Talk or Fight system. His project, however, did not focus on the behavioural norms but on the dynamics of hierarchical organisations. He managed to model in a quite realistic fashion the relationship between leaders and followers and obtained virtual societies composed of groups governed by a leader. My idea would be to merge Kelvin's model and mine. The agents would still behave according to the model explained in Chapter 1, except that they will belong to a group governed by a leader. The behavioural norms would not be absolute anymore: the default behaviour of an agent would depend on his partner's group. For example, if we consider a world with three groups, we would be able to find agents who are pacifist and sociable with people belonging to their own groups, whereas they are violent and misanthropic with people from the two other ones (in ancient Rome or Greece for example, a strong difference was made between the citizens of the empire and the "barbarians", or foreigners). With this improved tool, we could even simulate wars between two groups and work on the effects of resentment and forgetfulness between cultures. When this new model is developed and analysed, I will increase its complexity by adding another notion like resources and economy, and then improve it again, etc… Each step towards realism will be more difficult than the former one: the complexity of a society is superior to the sum of the complexity of its parts. However, each step towards realism will help us to understand who we are: since we are social beings, understanding better human societies will teach us a lot of things about us.

Bibliography

¤ Yaneer Bar-Yam, *Dynamics of complex systems*, Addision-Wesley publishing, 1997.

¤ Bernard Pavard and Julie Dugdale:
An introduction to complexity in social sciences, available at:
http://www.irit.fr/COSI/training/complexity-tutorial/complexity-tutorial.htm
The contribution of complexity theory to the study of socio technical cooperative systems

¤ Bernard Pavard, *Complexity Paradigm as a framework for the study of Cooperative Systems*

¤ Samuel Bowles and Herbert Gintis:
Prosocial Emotions, 2002
The origins of Human Cooperation, 2002

¤ Richard Seel:
Complexity and organisation development, 2004
Culture and complexity: New Insights on Organisational Change, published in Organisations & People Vol.7 No 2, 2000.

¤ Jessica C. Flack and Frans B. M. de Waal, *Being Nice is not a Building Block of Morality,* published in Journal of consciousness studies Vol. 7 No1-2, 2000.

¤ Melanie Mitchell, *Can Evolution Explain How the Mind Works?*

¤ John M. Ingham, *Psychological anthropology reconsidered,* Cambridge University press, 1996.

¤ M. Mitchell Waldrop, *Complexity: The Emerging Science at the Edge of Order and Chaos,* Viking, 1993.

¤ Ralph Stacey, *Complexity and Creativity in Organizations,* Berrett-Koelher, 1996.

¤ Stuart Kauffman, *At Home in the Universe: The search for Laws of Complexity,* Penguin, 1996.

¤ John H. Holland, *Hidden Order*, 1995.

¤ Paul Cillers, *Complexity and Postmodernism: Understanding Complex Systems,* Routledge, 1998.

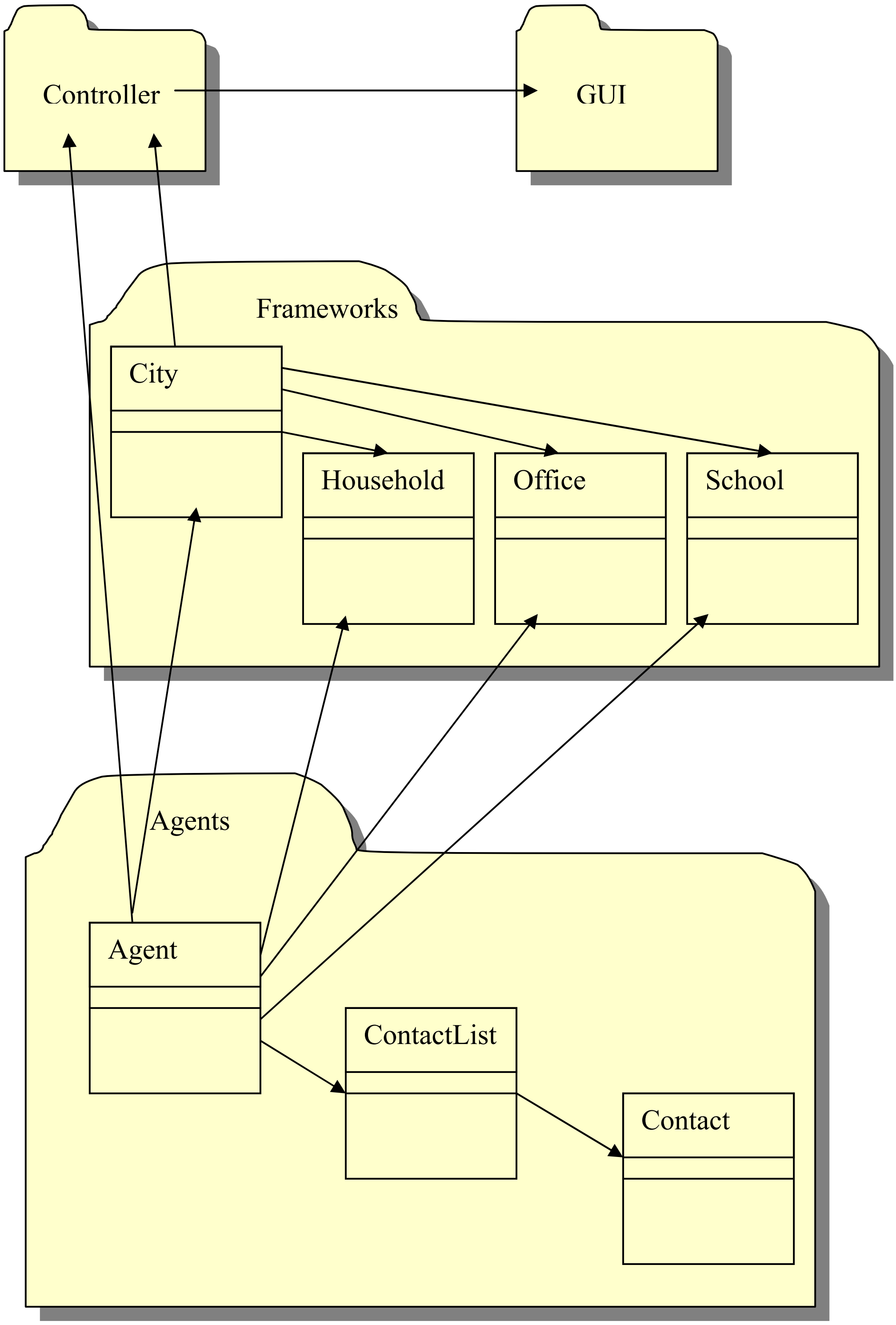
¤ Camille Guérillot, *Complex Systems and Human Societies,* 2005.

¤ Kelvin Au, *Dynamics of Human Societies: Evolution of Hierarchical Groups*, MEng Dissertation, Department of Computing, Imperial College London, June 2005.

¤ Rishi Aggarwal, *Dynamics of Human Societies: Talk or Fight*, MEng Dissertation, Department of Computing, Imperial College London, June 2005.

Appendix

Appendix 1: Architecture.



```java
/*--------------------------------------------------------------------
            Appendix 2: Code for an agent
 --------------------------------------------------------------------*/

/*********************************************************************
 *                              Class Agent                          *
 *                                                                   *
 *********************************************************************/

package Agents ;

import java.lang.Math.* ;
import java.util.* ;
import Framework.* ;
import Controller.* ;


public class Agent {

    //Main parameters:
    public int id ;
    public int age ;
    public int sex ;
    public double awareness ;
    public double integration ;
    public double satisfaction ;
    double strength ;
    int lifeSpamExpectancy ;
    public double[] behaviouralNorms = new double[5];

    //A contact list:
    public ContactList contacts ;

    //Four known frameworks:
    public Household house ;
    public School school ;
    public Office office ;
    City city ;

    //Familial state:
    public boolean singleAdult ;
    public boolean youngMother ;
    boolean seduce ;

    //Editable parameters:
    public double normDrivenFactor ;
    public double beta ;
    public double hystAwa ;
    public double hystInt ;
    public double hystBN ;

    //Internal parameters:
    int currentFramework ;
    double talkVector [] = new double[6] ;
    double[] fightVector = new double[6] ;
    double[] giveVector = new double[6] ;
    double[] stealVector = new double[6] ;
    double[] seduceVector = new double[6] ;



    /*****************************************************************
     *                  Construction of an agent                     *
     *****************************************************************/

    // Constructor for a new born baby:
    public Agent(int id0, Household house0, City city0) {
        initialize() ;
        id = id0 ;
        age = 0;
```



```java
        sex = TalkOrFight_v1.random.nextInt(2) ;
        awareness = 0 ;
        integration = 0;
        for (int i=0; i<5; i++) {
            behaviouralNorms[i] = 0 ;
        }
        house = house0 ;
        school = null ;
        office = null ;
        city = city0 ;
        normDrivenFactor = city.settings[house.district][6] ;
        hystAwa = city.settings[house.district][7] ;
        hystInt = city.settings[house.district][8] ;
        hystBN = city.settings[house.district][9] ;
        beta = city.settings[house.district][10] ;
        singleAdult = false ;
        youngMother = false ;
        seduce = false ;
        strength = (house.father.strength + house.mother.strength)/2 ;
        satisfaction = (house.father.satisfaction + house.mother.satisfaction)/2 ;
    }

    //Constructor for a random agent
    public Agent(int id0, int age0, int sex0, Household house0, City city
                boolean single, boolean mother) {
        initialize() ;
        id = id0 ;
        age = age0 ;
        sex = sex0 ;
        awareness = Math.min(0.9, Math.max(-0.9,TalkOrFight_v1.random.nextGaussian())) ;
        integration = Math.min(0.9, Math.max(-0.9,TalkOrFight_v1.random.nextGaussian())) ;
        satisfaction = 50+5*TalkOrFight_v1.random.nextGaussian() ;
        strength = -1+2*TalkOrFight_v1.random.nextDouble() ;

        for (int i=0; i<5; i++) {
            behaviouralNorms[i] = Math.min(0.9, Math.max(-0.9,TalkOrFight_v1.random.nextGaussian())) ;
        }
        house = house0 ;
        school = null ;
        office = null ;
        city = city0 ;
        singleAdult = single ;
        youngMother = mother ;
        if (age >= 675) { seduce = true ;}
        else { seduce = false;}
        normDrivenFactor = 0.5 ;
        hystAwa = 0.001 ;
        hystInt = 0.001 ;
        hystBN = 0.0005 ;
        beta = 0.66 ;
    }

    //Initialization of parameters
    void initialize(){
        talkVector[0] = 0.398;
        talkVector[1] = 0.398;
        talkVector[2] = 0.199;
        talkVector[3] = 0.1;
        talkVector[4] = 0.796;
        talkVector[5] = 0;

        fightVector[0] = -0.234 ;
        fightVector[1] = -0.468 ;
        fightVector[2] = -0.468 ;
        fightVector[3] = -0.117 ;
        fightVector[4] = -0.702 ;
        fightVector[5] = 0 ;
```



```java

        giveVector[0] = 0.196;
        giveVector[1] = 0.196;
        giveVector[2] = 0.392;
        giveVector[3] = 0;
        giveVector[4] = 0.392;
        giveVector[5] = 0.784;

        stealVector[0] = 0;
        stealVector[1] = -0.426;
        stealVector[2] = -0.213;
        stealVector[3] = 0;
        stealVector[4] = -0.213;
        stealVector[5] = -0.853;

        seduceVector[0] = 0;
        seduceVector[1] = 0.577;
        seduceVector[2] = 0.577;
        seduceVector[3] = 0.577;
        seduceVector[4] = 0;
        seduceVector[5] = 0;


        lifeSpamExpectancy = 3900 + Math.round(1000*(float)TalkOrFight_v1.random.nextGaussian()) ;
        contacts = new ContactList() ;
        currentFramework = 0;
    }



    /****************************************************************
     *                          Cycles                              *
     ****************************************************************/

    //The method called by the city in order to launch an agent cycle
    public void performClick() {
        if(singleAdult){
            cycleSingle();
            return ;
        }
        if(youngMother){
            cycleMother();
            return ;
        }
        if(0<= age && age < 155) {
            cycleBaby();
            return ;
        }
        if(155<= age && age < 676) {
            cycleChild();
            return ;
        }
        if(676<= age && age < 936) {
            cycleTeenager() ;
            return ;
        }
        if(age >= 936){
            cycleAdult();
            return ;
        }
    }

    //Cycle for a single adult
    void cycleSingle() {
        currentFramework = 3;
        for (int i=0; i<3; i++){
            Agent partner = office.choosePartner(id) ;
            interact_with(partner) ;
        }
        currentFramework = 1 ;
```



```java
        for (int j=0; j<3; j++){
            Agent partner = city.choosePartner(house) ;
            interact_with(partner) ;
        }
        age ++;
        if (house.mother!=null){
            if ((sex == 0 && house.mother.id != id)){
                house = city.findNewHome(this) ;
            }
        }
        if (house.father!=null){
            if ((sex==1 && house.father.id != id)){
                house = city.findNewHome(this) ;
            }
        }
        double roll = TalkOrFight_v1.random.nextDouble() ;
        if (roll < 0.02 && house.children.size()==0){
            house = city.marry(this) ;
        }
        if (age > lifeSpamExpectancy ){
            city.die(this) ;
        }
    }

    //Cycle for a young mother
    void cycleMother(){
        currentFramework =0 ;
        for (int i=0; i<3; i++){
            Agent partner = house.choosePartner(id) ;
            interact_with(partner) ;
        }

        Agent baby = contacts.getBaby() ;
        interact_with(baby) ;
        interact_with(baby) ;

        currentFramework =1 ;
        Agent partner2 = city.choosePartner(house) ;
        interact_with(partner2);

        age ++;
        if (baby.age == 154) {
            youngMother = false ;
            contacts.baby = null ;
        }
    }

    //Cycle for a baby
    void cycleBaby(){
        for(int j=0; j<3; j++){
            Agent partner = house.choosePartner(id) ;
            interact_with(partner) ;
        }

        Agent mother = house.getMother() ;
        for(int i=0; i<3; i++){interact_with(mother) ;}

        receivesTeach(mother) ;
        Agent father = house.getFather() ;
        receivesTeach(father) ;

        age ++ ;
        if (age == 155){
            school = city.registerAtSchool(this) ;
        }
    }

    //Cycle for a child
    void cycleChild(){
        currentFramework =0 ;
        for (int i=0; i<3; i++){
```

```java
            Agent partner = house.choosePartner (id) ;
            interact_with (partner ) ;
        }
        currentFramework =2 ;
        for (int j=0; j<2; j++){
            Agent partner = school.choosePartner (id) ;
            interact_with (partner ) ;
        }

        if(!city.noTeachers ){
            Agent teacher = school.getTeacher () ;
            receivesTeach (teacher ) ;
        }

        currentFramework =1 ;
        Agent partner2 = city.choosePartner (house ) ;
        interact_with (partner2 );

        Agent mother = house.getMother () ;
        receivesTeach (mother ) ;
        Agent father = house.getFather () ;
        receivesTeach (father ) ;

        age ++ ;
        if (age == 258){
            school = city.nextSchool (id, school ) ;
        }
        if (age == 363){
            school = city.nextSchool (id, school ) ;
        }
        if (age == 467){
            school = city.nextSchool (id, school ) ;
        }
        if (age == 571){
            school = city.nextSchool (id, school ) ;
        }
        if (age == 675){
            school = city.nextSchool (id, school ) ;
            seduce = true ;
        }
    }

    //Cycle for a teenager;
    void cycleTeenager (){
        currentFramework =0 ;
        for (int i=0; i<2; i++){
            Agent partner = house.choosePartner (id) ;
            interact_with (partner ) ;
        }
        currentFramework =2 ;
        for (int j=0; j<2; j++){
            Agent partner = school.choosePartner (id) ;
            interact_with (partner ) ;
        }

        if(!city.noTeachers ){
            Agent teacher = school.getTeacher () ;
            receivesTeach (teacher ) ;
            receivesTeach (teacher ) ;
        }

        currentFramework =1 ;
        for (int i=0; i<2; i++){
            Agent partner = city.choosePartner (house ) ;
            interact_with (partner );
        }

        Agent parent ;
        boolean choice = TalkOrFight_v1 .random .nextBoolean () ;
        if (choice ) {parent = house.getMother ();}
        else {parent = house.getFather ();}
```



```java
        receivesTeach (parent ) ;

        age++ ;
        if (age == 762) {
            school = city.nextSchool (id, school ) ;
        }
        if(age == 849) {
            school = city.nextSchool (id, school ) ;
        }
        if (age == 936) {
            school.removeStudent (id) ;
            school = null ;
            office = city.registerAtOffice (this) ;
            house = city.findNewHome (this) ;
            singleAdult = true ;
        }
    }

    //Cycle for an adult
    void cycleAdult (){
        currentFramework =0 ;
        for (int i=0; i<3; i++){
            Agent partner = house.choosePartner (id) ;
            interact_with (partner ) ;
        }
        currentFramework =3 ;
        for (int i=0; i<2; i++){
            Agent partner = office.choosePartner (id) ;
            interact_with (partner ) ;
        }
        currentFramework =1 ;
        Agent partner = city.choosePartner (house ) ;
        interact_with (partner );

        age++;
        office = city.allocateOffice (this);

        if (sex==0 && house.children .size() <3 && age<2800 ){
            double n = TalkOrFight_v1 .random .nextDouble () ;
            if (n < 0.002){
                city.procreate (this) ;
            }
        }

        if (age > lifeSpamExpectancy ){
            boolean canDie = true ;
            Enumeration enum = house.children .elements () ;
            while(enum.hasMoreElements ()){
                if(! ((Agent )enum.nextElement ()).singleAdult ){
                    canDie = false ;
                }
            }
            if(canDie ){
                city.die(this) ;
            }
        }
    }

    /**************************************************************
     *                        Interactions                        *
     **************************************************************/

    //Selection of an interaction
    void interact_with (Agent partner ){
        int result =0 ;
        double bestMark = 0 ;

        if(!contacts .isKnown (partner .id)){
            contacts .add(partner .id, age) ;
        }
        if(!partner .contacts .isKnown (id)){
```



```java
            partner .contacts .add(id, partner .age) ;
        }

        double [] situation = new double [6] ;
        double relation = contacts .getRelation (partner .id, age) ;
        double classBonus = contacts .getClass (partner .id) ;
        double deltaSat = (satisfaction -
    partner .satisfaction )/Math .max(satisfaction , partner .satisfaction ) ;

        double sNorm =
    Math .sqrt (Math .pow(awareness ,2)+Math .pow(integration ,2)+
                            Math .pow(relation , 2)+Math .pow(partner .awareness
                            Math .pow(classBonus , 2)+Math .pow(deltaSat ,2)) ;

        situation [0] = awareness /sNorm ;
        situation [1] = integration /sNorm ;
        situation [2] = relation /sNorm ;
        situation [3] = partner .awareness /sNorm ;
        situation [4] = classBonus /sNorm ;
        situation [5] = deltaSat /sNorm ;

        for (int i=0; i<6; i++){
            bestMark = bestMark + talkVector [i]*situation [i] ;
        }
        bestMark = normDrivenFactor *behaviouralNorms [0] +
    (1-normDrivenFactor )*bestMark ;

        double mark = 0;
        for (int i=0; i<6; i++){
            mark = mark + fightVector [i]*situation [i] ;
        }
        mark = normDrivenFactor *behaviouralNorms [1] +
    (1-normDrivenFactor )*mark ;
        if (mark > bestMark ){
            bestMark = mark ;
            result = 1;
        }

        mark = 0;
        for (int i=0; i<6; i++){
            mark = mark + giveVector [i]*situation [i] ;
        }
        mark = normDrivenFactor *behaviouralNorms [2] +
    (1-normDrivenFactor )*mark ;
        if (mark > bestMark ){
            bestMark = mark ;
            result = 2;
        }

        mark = 0;
        for (int i=0; i<6; i++){
            mark = mark + stealVector [i]*situation [i] ;
        }
        mark = normDrivenFactor *behaviouralNorms [3] +
    (1-normDrivenFactor )*mark ;
        if (mark > bestMark ){
            bestMark = mark ;
            result = 3;
        }

        if(seduce ) {
            if ((!house.isInHousehold (partner .id) ||
    contacts .isCompanion (partner .id))&& sex != partner .sex) {
                mark = 0;
                for (int i=0; i<6; i++){
                    mark = mark + seduceVector [i]*situation [i] ;
                }
                mark = normDrivenFactor *behaviouralNorms [4] +
    (1-normDrivenFactor )*mark ;
                if (mark > bestMark ){
                    bestMark = mark ;
```



```java
                    result = 4;
                }
            }
        }

        switch (result ){
            case 0 : talk(partner ) ; break ;
            case 1 : fight(partner ) ; break ;
            case 2 : give(partner ) ; break ;
            case 3 : steal(partner ) ; break ;
            case 4 : seduce (partner ) ; break ;
            default : System .out.println ("Error in choosing interaction
"
                                            +id+".") ;
                      System .exit(0) ; break ;
        }
    }

    //Talk function
    void talk(Agent partner ){
        double formerInt = integration ;
        double rel = contacts .getRelation (partner .id, age) ;
        double s = rel + awareness + integration +
  contacts .getClass (partner .id) ;
        double t = -3+6*TalkOrFight_v1 .random .nextDouble () ;
        if (t<=s){
            modifyAwareness (1);
            partner .modifyAwareness (1);
            modifyIntegration (1) ;
            partner .modifyIntegration (1) ;
            satisfaction = satisfaction * 1.001 ;
            partner .satisfaction = partner .satisfaction * 1.001 ;
            modifyRelation (partner .id, 1) ;
            partner .modifyRelation (id, 1) ;
            adjustNorms (0, 0.001, integration -formerInt );
        }
        else{
            if (awareness < 0){
                modifyAwareness (-1) ;
            }
            modifyIntegration (-1) ;
            modifyRelation (partner .id, -1);
            partner .modifyRelation (id, -1);
            adjustNorms (0, 0, integration -formerInt );
        }
    }

    //Fight function
    void fight(Agent partner ){
        double formerInt = integration ;
        modifyAwareness (-1) ;
        modifyRelation (partner .id, -2) ;
        partner .modifyRelation (id, -2) ;

        if(house.isInHousehold (partner .id)){
            partner .modifyClass (id, -2) ;
        }

        double s = strength - partner .strength ;
        double t = -1+2*TalkOrFight_v1 .random .nextDouble ();
        if(t<=s){
            satisfaction = satisfaction *1.002 ;
            partner .satisfaction = partner .satisfaction *0.995 ;
            partner .modifyIntegration (-3);
            modifyIntegration (-2) ;
            adjustNorms (1,0.002, integration -formerInt );
        }
        else{
            satisfaction = satisfaction *0.995 ;
            modifyIntegration (-3);
            partner .modifyIntegration (-1) ;
```

```
            adjustNorms (1,-0.005, integration -formerInt );
        }
    }

    //Give function
    void give(Agent partner ){
        double formerInt = integration ;
        modifyIntegration (2) ;
        partner .modifyIntegration (2) ;
        partner .satisfaction = partner .satisfaction + 0.2*satisfaction ;
        satisfaction = 0.8*satisfaction ;
        modifyRelation (partner .id, 2) ;
        partner .modifyRelation (id, 2) ;
        if(house .isInHousehold (partner .id)){
            partner .modifyClass (id, 1) ;
        }
        adjustNorms (2,-0.2, integration -formerInt );
    }

    //Steal function
    void steal(Agent partner ){
        double formerInt = integration ;
        double formerSat = satisfaction ;
        modifyIntegration (-2) ;
        partner .modifyIntegration (-1) ;
        partner .satisfaction = 0.895*partner .satisfaction ;
        satisfaction = satisfaction + 0.1*partner .satisfaction ;
        modifyRelation (partner .id, -2) ;
        partner .modifyRelation (id, -2) ;
        if(house .isInHousehold (partner .id)){
            partner .modifyClass (id, -1) ;
        }
        adjustNorms (3, 0.1*partner .satisfaction /formerSat , integration -formerInt );
    }

    // Seduce function
    void seduce (Agent partner ){
        double formerInt = integration ;
        double s = contacts .getRelation (partner .id, age) + awareness + integration ;
        double t = -3+6*TalkOrFight_v1 .random .nextDouble () ;
        if (t<=s){
            if(contacts .companion != partner ){
                modifyIntegration (-1) ;
            }
            else{
                modifyIntegration (2) ;
            }
            if(partner .contacts .companion != this){
                partner .modifyIntegration (-1);
            }
            else{
                partner .modifyIntegration (1);
            }
            satisfaction = satisfaction * 1.002 ;
            partner .satisfaction = partner .satisfaction * 1.002 ;
            modifyRelation (partner .id, 2) ;
            partner .modifyRelation (id, 2) ;
            adjustNorms (4, 0.002, integration -formerInt );
        }
        else{
            if (awareness < 0){
                modifyAwareness (-1) ;
            }
            modifyIntegration (-1) ;
            partner .modifyIntegration (1) ;
            modifyRelation (partner .id, -1);
            partner .modifyRelation (id, -1);
            satisfaction = satisfaction *0.999 ;
            adjustNorms (4, -0.05, integration -formerInt );
```



```
        }
    }

    //Adjustment of norms
    void adjustNorms (int norm, double deltaSat , double deltaInt ){
        double mark = 0 ;
        double x1 = deltaSat + deltaInt + behaviouralNorms [norm] ;
        int nbWitnesses = 0;
        for (int i=0; i<3; i++){
            if(TalkOrFight_v1 .random .nextDouble () <= 0.2){
                nbWitnesses ++ ;
            }
        }
        if (nbWitnesses == 0){
            mark = x1 ;
        }
        else{
            double x2 =0 ;
            Agent[] witnesses ;
            switch(currentFramework ){
                case 0 : witnesses = house .getWitnesses (id, nbWitnesses ) break ;
                case 1 : witnesses = city.getWitnesses (house, nbWitnesses break ;
                case 2 : witnesses = school .getWitnesses (id, nbWitnesses break ;
                case 3 : witnesses = office .getWitnesses (id, nbWitnesses break ;
                default : witnesses = new Agent[0];
                          System .out .println ("Erreur dans le framework courant ") ;
                          System.exit(0) ;
                          break ;
            }
            for(int i=0; i<nbWitnesses ; i++){
                if(!contacts .isKnown (witnesses [i].id)){
                    contacts .add(witnesses [i].id, age) ;
                }
                if(!witnesses [i].contacts .isKnown (id)){
                    witnesses [i].contacts .add(id, witnesses [i].age) ;
                }
                if (contacts .getRelation (witnesses [i].id, age) > -0.8) {
                    x2 = x2 + witnesses [i].criticize (norm)*(1 + contacts .getClass (witnesses [i].id)) ;
                }
            }
            mark = (1-beta)*x1 + beta*x2;
        }


        if(mark > 1.4) modifyBeNorm (norm, 8) ;
        if(mark > 1.1 && mark <= 1.4) modifyBeNorm (norm, 6);
        if(mark > 0.9 && mark <= 1.1) modifyBeNorm (norm, 4);
        if(mark > 0.6 && mark <= 0.9) modifyBeNorm (norm, 3);
        if(mark > 0.4 && mark <= 0.6) modifyBeNorm (norm, 2);
        if(mark > 0.1 && mark <= 0.4) modifyBeNorm (norm, 1);
        if(mark > -0.1 && mark <= 0.1) modifyBeNorm (norm, 0);
        if(mark > -0.4 && mark <= -0.1) modifyBeNorm (norm, -1);
        if(mark > -0.6 && mark <= -0.4) modifyBeNorm (norm, -2);
        if(mark > -0.9 && mark <= -0.6) modifyBeNorm (norm, -3);
        if(mark > -1.1 && mark <= -0.9) modifyBeNorm (norm, -4);
        if(mark > -1.4 && mark <= -1.1) modifyBeNorm (norm, -6);
        if(mark <= -1.4) modifyBeNorm (norm, -8);
    }

    //Judgement of a witness about an interaction
    public double criticize (int i){
        if (behaviouralNorms [i] >= 0.5) return 1 ;
        if (behaviouralNorms [i] <= -0.5) return -1 ;
        return 0 ;
    }
```



```

    /*******************************************************************
     *                    Transmission of norms                        *
     *******************************************************************/

    //Teachings management
    void receivesTeach (Agent teacher ){
        int topic = TalkOrFight_v1 .random .nextInt (5) ;
        double s = 2*teacher .awareness +(2 - Math.abs(behaviouralNorms [topic] -
                    teacher .behaviouralNorms [topic])) ;
        double t = -2 + TalkOrFight_v1 .random .nextDouble ()*6 ;
        if (t <= s || awareness <-0.8) {
            behaviouralNorms [topic]+= 0.01 * (teacher .behaviouralNorms [topic]
                                    - behaviouralNorms [topic]);
            if (age>155 && teacher == school.getTeacher ()) modifyIntegration (1) ;
        }
    }


    /*******************************************************************
     *                    Hysteresis Cycles                            *
     *******************************************************************/

    //Modication of a parameter
    void modifyAwareness (int steps ){
        awareness = hysteresis (awareness , steps, hystAwa );
    }
    void modifyIntegration (int steps ){
        integration = hysteresis (integration , steps, hystInt );
    }
    void modifyBeNorm (int norm, int steps ){
        behaviouralNorms [norm] = hysteresis (behaviouralNorms [norm], steps hystBN );
    }
    void modifyRelation (int id, int steps ){
        double newRel = hysteresis (contacts .getRelation (id, age), steps, 0.01) ;
        contacts .setRelation (id, newRel ) ;
    }
    void modifyClass (int id, int steps ){
        double newClass = hysteresis (contacts .getClass (id), steps, 0.01)
        contacts .setClass (id, newClass ) ;
    }

    //Hysteresis function
    double hysteresis (double un, int steps, double width ){
        double result = un ;
        if (steps > 0) {
            for(int i=0; i<steps; i++){
                if(result >=0){
                    result = width + (1-width )*result ;
                }
                else{
                    result = width + (1+width )*result ;
                }
            }
        }
        else{
            for(int i=0; i>steps; i--){
                if(result >=0){
                    result = (1+width )*result -width ;
                }
                else{
                    result = (1-width )*result -width ;
                }
            }
        }
        return result ;
```



```
    }
}
```