# THÈSE DE DOCTORAT
## en Informatique

# Atteinte de consensus basée sur l'échange de préférences entre agents rationnels et autonomes : application à la formation d'alliances

VAUVERT Guillaume

Numéro officiel
Université Paris Nord

Preliminary version – January 15, 2004

# Thèse de doctorat

Spécialité : Informatique

**présentée par**

VAUVERT Guillaume

**pour l'obtention du titre de**

Docteur en Informatique

**sur le sujet**

*Atteinte de consensus basée sur l'échange de préférences entre agents rationnels et autonomes : application à la formation d'alliances*

**directeur de thèse**

EL FALLAH - SEGHROUCHNI Amal

**soutenue le 15 Janvier 2004 devant le jury composé de :**

Suzanne PINSON
Richard EMILION
Philippe DAGUE
Jean-Daniel ZUCKER
Patrick TAILLIBERT
Henry SOLDANO

# Contents

# Chapter 1

# Introduction

## 1.1 The need of powerfull and independent components

**In computer sciences** At the very beginning of computer sciences, programs were build directly with processor level instructions. This kind of programming was very powerful, because one can control exactly what happens. However, it was difficult to build large programs with this kind of languages.
Then, macros and later functions allowed to reuse and to organize the code. Programs begin to be designed as a set of independent components. Since the size of code that human beings are able to understand is bounded, programs become more and more great. To postpone limits, components must be as few dependent as possible: software engineers can then be concentrate on a component, and pay few attention to others. The top-down functional approach is replaced by a bottom-up object oriented approach [SB86, Mey97] (although, it is not completely removed), because objects are more stable than operations. Instead of procedures grouped by functionality, the O.O.P approach put an object and the functions that process on it together. The run is no more regarded as calls of function that call sub-function and so on, but as sendings of message to objects. Objects react to them by calling a method. The method's specifications are known by the object that sends the message, but the way to abide by them is not: the object is timidly autonomous, because he decides how to obey.
In the same time, programming languages provide more and more complex functional components. They allow to think hard about specific parts of the program, reusing already built code.
Finally, science computing tends to more autonomous and powerfull components.

**Over networks** In the other hand, as shown in [Kra97, SLA$^+$99], the communication infrastructure is growing, at several scales, from intranets (localized in small areas) to internets (connecting subnets over the world). More and more processes are enable to interact: synchronization by basic signals (flags, semaphores), value of several types (integer, string), exchange of data or knowledge, complex formulae that formalized cognitive states.
Initially, computers were dedicated to solving numerical problems. They were isolated machines that used their resources to make calculations. Nowerdays, more and more applications make computation at the operative decision making level. Moreover, they

interact with other applications over the world.
Each application:

- runs concurrently with other applciations;
- has a large amount of private informations (knowledge, strategies) that it wants to keep secret;
- owns some skills peculiar to its enterprise;
- uses its intelligence capacity to make decisions in an autonomous way.

Distributed Systems (DS) have been developed in order to support heterogeneous components and communications, to take the environment and the evolution of the system into account, and adapt the system to the events [RG02]. The main problems of this field are to model the knowledge, to use autonomous components and to deal with interactions.
The limitation of the object approaches have been identified [WJ99, Jen00, JCC]:

- the model of knowledge doesn't take into account the others entities in the system;
- objects don't decide, they execute (no initiative, no decision making process);
- interactions among objects are basic: no semantic level, no protocol, no negotiation or planning.

Electronic commerce is a typical example of domain supported by autonomous and powerfull applications.
Both cases show the need of powerfull and autonomous components.

## 1.2 Multi-Agent Systems

Originally, computer sciences has been used to make complex computations, what is no more than a process of symbols. Since decades, a new field of research has emerged: the Artificial Intelligence (AI) [Tur50]. Its goal is to build softwares that tend to have the same mental capabilities than human beings. Most of time, large monolithic programs are built, but they are hard to maintain, because these growing intelligent systems have to deal with a more and more large amount of knowledge. A branch of AI – called Distributed Artificial Intelligence – has been created in order to take advantage of concurrent computational resources. The knowledge and the runs are distributed, among several processes, but the control is almost centralized.
Another approach does exist: instead of considering that intelligence is totally embedded in a single process, Multi-Agent Systems (MAS) propose to build separate intelligent programs (named agents) that interact with each other, without central control. The most part of the system's intelligence arises from these interactions.
Together, DAI, OOP and DS allow to create a new domain. DAI brings intelligence and concurrency, OOP 's contribution is the decentralized control, since DS brings a large support at interaction level.
Others concepts appears:

- interactions among intelligent entities are more rich, because of complex information exchanges: believes, knowledge, mental states, plans, *etc.*;
- a large amount of entities may be hard to control when considering them at the same level: to organize them becomes essential;
- outside the system, a world does exist, called the environment.

Thanks to their cognitive capabilities, agents are able to improve problem solving in complex environments (dynamic, open, unpredictible, *etc.*). Objects decide how to carry out a method; agents go further: they may accept *or refuse*, they may decide *when* to execute it. Here, the message sending has its true meaning: agents don't send orders to execute a function, but send messages that are red and interpreted by the receiver. They could be more efficient if they can make any decision without the direct intervention of humans or other agents, *i.e.* if they are autonomous (deeply studied in chapter 2 on page 19).
Finally, an agent may be define as computational entity that:

- are artificially intelligent in the classic means of AI ;
- run concurrently;
- may communicate with others (generally using a common language);
- may have knowledge about organization;
- may interact with an environment.
- are autonomous

## 1.3 MAS in electronic commerce

In fact, the commonly found definition of autonomy is the definition of a partial autonomy: we could define the total autonomy as the right to make any decision. This one is often regarded as impossible to reach, because of the impossibility to obtain the desirable global behavior [Cas95, Ld00].
At a contrary, we think that it is possible to design and to control a system of totally autonomous agents.
In one hand, the need of interaction among several intelligent entities in order to carry out decisions among applications; in the other hand, the emergence of a new concept, called agent, that seems to be very well suitable for this problem.
Depending on the context of use, agents may be regarded either as intelligent objects (small agents), either as complex intelligent components (middle agents), or as intelligent applications (large agents). We will focusse on electronic commerce, where agents are regarded as complex stand-alone applications (large agents).
In this context, our agents have particular properties.

- often, they have time to reason (however humanely acceptable) and to make a decision;

- they are regarded as totally autonomous entities (see chapter 2 on page 19) by agents that interact with; they may be autonomous processes that reason and send formatted messages, or a program that simply sends messages as decided by a human being.

### Autonomy in an electronic commerce context

#### Total autonomy

Most of characteristics of agents are issued from several previously existing fields: (artificial) intelligence, concurrency, separated entities... Among them, only one is really new: the autonomy. It may be defined as the freedom from intervention in the making decision process by others, agents or people (see chapter 2 on page 19 for a deeper study). This property, that makes agents special concepts, is very interesting and promising, because:

- Agents don't need to wait human orders to react to environment changes. They may take the initiative in acting, thus increasing the system's efficiency improved in many situations.

- Autonomy increases the independence of agents' developers : each agent may be programmed more independently than objects, because agents' decisions are less dependent on others' decision in the specification (even if they may be very dependent during the run).

- One of the main difference between people and softwares is that the first may take unpredictable decisions, *i.e.* are totally autonomous. Considering agents as totally autonomous too allows to regard people as agents. Of course, people need to use an interaction medium in order to be able to communicate with a MAS, but this medium doesn't necessarily bound their decision making. Thus, autonomy makes the integration of people in computer systems easier.

However, it is very difficult to increase the level of autonomy of a system. The more a system is autonomous, the more it is hard to be controlled: autonomy allows agents to make decision without direct intervention, but this freedom makes difficult to predict the system's behavior. In one hand, a system that does exactly what we want because it is wired-programmed, but that cannot adapt its behavior to new conditions; in the other hand, a system that may change its behavior to be efficient in a large amount of circumstances, but that is hard to control. It is a fight of freedom against control.

**Why is total autonomy necessary?** In a context of electronic commerce, softwares and people interact *via* an electronic medium. A protocol that allows everybody to interact cannot assume that people is not totally autonomous, because they are really autonomous. A solution could be to constrain people's behavior using an electronic marketplace, but it has several drawbacks (see subsection 2.2.2 on page 23 for more details):

- the centralization of communication is costly;

- agents must be in trust with the platform.

So, all agents (electronic and human) may be regarded as totally autonomous.

**Infringements** Despite the increasing weight of the specification part during the design of a complex system, the total control is difficult to reach, especially if the software is programmed by many developers, geographically far, during a long time, in several steps, with modifications, extensions, additions, *etc.*In that case, software components (namely agents) may not behave as decided *a priori* . Moreover, in an electronic commerce context, economically rational agents are often anonymous, what makes easier to cheat.
So, intentional and not intentional infringements must be taken into account in the design of protocols.

**Heterogeneity** In the previously described context, it is difficult to impose and to ensure that agents will abide by the constrains. So a protocol cannot make assumptions on the agents' behavior and their computational capabilities: agents have to be regarded as heterogeneous.

## 1.4 Totally autonomous agents reach a consensus

Our definition of total autonomy (chapter 2 on page 19) produces two precepts:

1. A protocol must not make assumptions on not observable data.
2. Agents must choose their partners of interaction.

In this context, three problems arise:

- How to design a protocol that makes no assumptions on not observable data?
- What kind of visible data (exchanged messages) ?
- How to allow totally autonomous agents to choose their partners ?

To solve the third problem, agents have to be able to dynamically change their partners of interaction, because : 1) the reasons of the collaboration may also change quickly; 2) the organizational structure is unpredictible, because it is based on unknown agents preferences (total autonomy doesn't allow to know internal informations). We propose to call *alliance* this kind of set of agents in collaboration that may be created or removed according to agents' preferences changes. Usually, in MAS, the term *coalition* is used instead of the term *alliance*. Our reaching consensus protocol requires another kind of group, which looks better like a coalition: agents rally *against* other groups. Moreover, in our context of totally autonomous agents, the concept of usually called *coalition* is closer to the notion of *alliance*: agents regroup to take of advantage of the synergy of skills. Possibly, they may ally against some other teams, but it is not always the case. We hope that this exchange of words keeps the thesis clear.

Generally, no optimal organizational structure exists (*i.e.* a structure that satisfied all agents). Since a structure chosen using an arbitrary criteria may always be critized by some agents, the choice of the structure have to be made by the agents themselves. The main difficulty is thus to find a protocol that allows totally autonomous agents to reach a consensus about the organizational structure.
So, how to allow totally autonomous agents to reach a consensus, since each of them is free to make its own decision (for the choice and during the protocol)? The main idea to solve the first problem is to allow agents to evolve their positions by exchanging informations until a consensus is reached (chapter 3 on page 41).
The second problem of visible data appears again: what to exchange? We cannot assume that agents will send their true opinion[1], because we cannot be sure that they will not lie. But we can demand to agents to exchange their current position, because a position (see chapter 4 on page 65) is a given information which cannot be regarded as true or false.

**Validation** In order to validate our work, we have made experiments about the opinions and about the protocol of alliance formation.
We compare the expressivity of our formalism of opinion with another ones, and the results of elections of our aggregation operator with the classic vote system.
In order to test our protocol of alliance formation, we have developed a platform that allows to quickly implement and test protocols by describing the protocol as a Petri net (chapter 6 on page 111).
To test our protocol of consensus reaching, we need a problem to which apply it; of course, we have chosen the alliance formation problem. In the context of totally autonomous agents, the measure of agents' satisfactions is not interesting, because the goal of our protocol is not to obtain a solution that satisfies agents in a certain mean (see subsubsection 3.1.1 on page 42), but to allows them to choose together their solution. In this case, some agents could be disappointed, but the result depends on agents' strategies, what cannot be controlled (due to agent's autonomy). However, we have tested several strategies and measure the complexity.

## 1.5 Plan

The thesis is organized as follows:
First, two kinds of autonomy are introduced (chapter 2 on page 19): autonomy at agent level (section 2.2 on page 20) and autonomy at organizational level (section 2.3 on page 30).
Considering a total autonomy constrains the way to design a MAS (section 2.5 on page 32): protocols (subsection 2.5.1 on page 32) and organizations (subsection 2.5.2 on page 34).
Then (in chapter 3 on page 41), based on the constrains defined before, a protocol of consensus reaching is given (section 3.2 on page 44 and section 3.3 on page 49). Complexity

[1] Usually, the word *preference* refers to the comparison among all choices, while we think that it must refer to the comparison among two choices, without pay attention to other comparisons. So, in this thesis, the word *preference* refers to a level of preference between two choices, while the term *opinions* points out the set of preferences. We hope that this unusual use of words keeps the thesis clear.

and termination are studied in section 3.4 on page 56.
The protocol below is based on an exchange of opinion. Next chapter (chapter 4 on page 65) is about opinion formalism (section 4.2 on page 66, section 4.3 on page 71) and operators on opinions (section 4.5 on page 78, section 4.6 on page 79, section 4.7 on page 81, section 4.8 on page 82). Some experimental results are given in section 4.9 on page 84.
The next chapter (chapter 5 on page 93) is dedicated to the consequence of the organizational autonomy, the formation of alliances. Alliances (section 5.2 on page 94) and alliances formation (section 5.3 on page 96) works in the literature are criticized. Then, we define our context (section 5.4 on page 101) in which our protocol takes place (section 5.5 on page 101).
In order to validate the proposed protocol, we developed a platform that allows to quickly test it (chapter 6 on page 111). The results of experiments are given in chapter 7 on page 119.
Finally, chapter 8 on page 131 concludes this thesis and outlines our future works.

# Bibliography


[Cas95] Christiano Castelfranchi. Guarantees for autonomy in cognitive agent architecture. In Springer-Verlag, editor, *Intelligent agents*, pages 56–70. M. Wooldridge and N. Jennings, New York, N.Y., 1995.

[JCC] P. Glize J-C. Casteran, M-P. Gleizes. Des méthodologies orientées multi-agent. In *Proceedings of JFIAD/SMA 2000*.

[Jen00] N. R. Jennings. On agent-based software engineering. *Artificial Intelligence*, 117(2):277–296, 2000.

[Kra97] S. Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94(1-2):79–98, 1997.

[Ld00] Michael Luck and Mark d'Inverno. Autonomy: A nice idea in theory. In *Agent Theories, Architectures, and Languages*, pages 351–353, 2000.

[Mey97] Bertrand Meyer. *Object-oriented Software Construction*. Prentice-Hall, Inc., New York, N.Y., second edition edition, 1997.

[RG02] Éric Rigaud and Franck Guarnieri. Apports des systèmes multi-agents à l'ingénierie des systèmes d'information distribués. Technical report, Séminaire sur les Systèmes Distribués et Connaissances, 2002.

[SB86] M. Stefik and D.G. Bobrow. Object-oriented programming : themes and variations. *AI Magazine*, 7(4):40–62, 1986.

[SLA+99] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohme. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 1999.

[Tur50] pp.. Turing, A.M. 1950. . Mind 59. Computing machinery and intelligence. In *Mind 59*, pages 433–460, 1950.

[WJ99] M. Wooldridge and N. R. Jennings. Software engineering with agents: Pitfalls and pratfalls. *IEEE Internet Computing*, May/June 1999.

# Chapter 2

# Autonomy

## Contents

## Figures



## Tables

## 2.1 Introduction

The MAS field may be defined as a cross between several domains:

- Object Oriented Programming (OOP ): data and behaviors are encapsulated into a component, called an agent;
- Concurrent Programming (CP ): agents run concurrently;
- Distributed Systems (DS ):
- Artificial Intelligence (AI ): agents reason, learn, *etc.*

Together, these characteristics make agent a new and very interesting concept, because they promise to take the best from each field:

- OOP and CP : a promising design of encapsulated entities running concurrently;
- DS : model of knowledges about interactions, autonomy of components, deal with interactions;
- AI : intelligent system may solve complex problem in complex environments.

In fact, agents are more than that. As in AI , they are intelligent: they reason to solve a problem or to decide which act to perform [JCd], they learn to improve their efficiency or to solve new problems [VBS98], *etc.*However, agents do not work alone; thus, their acts may have consequences on others. They may influence and reciprocally may be influenced by others: interactions exist, because several agents run concurrently.
In order to improve their efficiency, they are interested in what others are doing and in the consequences of their own behavior on others. A traditional reasoning at task level (to decide what to do) is not enough to efficiently reason: agents must not only reason and act, but also reason about their reasoning and their acts.
This meta-level of reasoning and the attendance of others require from the agents a certain awareness of the existence of others. Then, the question is: how do agents influence and how are they influenced by others ? What does mean it to influence and to be influenced ? If we agree that agents blindly accept others' orders, agents may be called objects. So we assume that agents can choose to be more or less influenced, what defines the level of autonomy. Agents are not objects partly because they are autonomous.
MAS may also be defined as an extension of DAI , but, actually, there is an important difference between these two concepts. In DAI , the artificial intelligence is distributed: entities are parts or components of the system as a whole. In MAS, several artificial intelligent beings interact: agents are autonomous entities put together.
The autonomy modifies the way to design and to control the system [VE01, VE00]. Before showing how, we should explain what autonomy is.
Three kind of autonomy are described: at agent-level (section 2.2), at organizational-level (section 2.3 on page 30) and at system-level (section 2.4 on page 32). The next section (section 2.5 on page 32) points at the consequences of autonomy on the design of protocols (subsection 2.5.1 on page 32) and of organizations (subsection 2.5.2 on page 34).
The economic context provides some constrains; among them, the heterogeneity and the rationality are those that have the more consequences.
Finally we are going to consider a subset of Multi Agent Systems: Systems with Autonomous and Rational Agents with Heterogeneity (SARAH).

## 2.2 Autonomy at agent-level

Even if autonomy has often been promoted as a key concept of MAS, no definition of it has been universally accepted [Pet96].
The most consensual definition is Castelfranchi's [Cas95] (cited in [JW95, MF95]) called *goal-autonomy*:

> "Agents should be able to perform the majority of their problem solving tasks without the direct intervention of humans or other agents and they should have a degree of control over their own actions and their own internal state."

This definition shows how difficult it is to define this fuzzy notion:

- "*the* majority *of their problem*": Does really $50\%$ of problems be sufficient to consider that an agent is autonomous ? Why did the author choose this bound ?

- "*the* direct *intervention*": What does *direct* exactly means ? Who can decide that an intervention is direct or not ?

- "*a* degree *of control*": Which value ? Does a little control be enough ?

This definition uses terms with fuzzy signification, but it allows each one to define its own level of autonomy.

### 2.2.1 Definitions

In order to clarify the complex concept of autonomy, we are going to dissect several definitions.

#### Freedom from intervention

Most of definitions agree on the fact that autonomy includes freedom from intervention during the decision making process. In [WJ94], agents are autonomous if they operate without direct intervention or guidance of humans. In [BM99], a deeper study of autonomy results in a more precise definition:

> an active agent uses its capabilities to pursue some goals without intervention, oversight, or control by any other agent.

Within this definition, an autonomous agent is free from intervention not only by humans but also by other agents. The same idea can be found in [MF95]: an autonomous agent must have its own dispositions to behave in a certain way. In [GB99], autonomous agents are defined as an extension of active objects: autonomous agents are "*able to perform a number of functions or activities without external intervention, over extended time periods*". This definition adds a limited duration of non-intervention.

**What kind of allowed intervention ?** The difficulty is thus to define the kind of intervention allowed. In [BM99], three types of intervention have been identified:

1. modification of the environment;

2. influence over an agent's beliefs;

3. intervention in an agent's decision-making process.

In the first case, an agent may constrain another agent to move toward a place by putting walls around him. This kind of intervention restricts the agent's freedom of action, but not his capacity to reason. Since autonomy is defined at the reasoning level, this kind of intervention doesn't bound the autonomy.
Using the second type, an agent may try to convince, to influence, to bribe, to threaten another agent. Whatever the signification of the message chosen by the sender is (strong order or simple advice), it is the receiver's choice to blindly obey, to negotiate, or to refuse [tFS00]. An informational message can be more or less taken into account, depending on

the receiver.
Finally, only the third type of intervention is forbidden for an autonomous agent: nobody else can choose which decision will it reaches.

**Autonomy of action** *vs* **autonomy of decision** Generally, autonomy is defined as the autonomy of decision, but sometimes, what is considered is the autonomy of action. In [Gad98], the autonomy is defined as "*the capability to execute tasks according to its own information, preferences and knowledge/beliefs, not limited by any external requests, neither by how these tasks have to be executed*". In this definition, the important point is the capability to execute tasks: the agent must be free to choose how they will be executed. In fact, the way of the performance depends on:

1. the intervention on the environment: certain external rules make some decisions not possible;
2. the intervention in the decision-making process: before acting, an agent decides to act. He may accept others' arguments or not.

Moreover, the choice of which tasks to execute is important too.
The autonomy of action is an illusion: agents are not really free to choose their act, because it depends on their skills, on the environment, on the others' acts. In a contrary, agents may be free to reason, to decide, to choose, because it doesn't depend on external constrains (even if computational limitations bound his reasoning process capabilities). If pushed, an agent may decide to do something, even if it is not realizable.

**Relative freedom**

Even if most authors define autonomy as the freedom of making decision process, most of them make some restrictions upon this freedom in order to be able to control the behavior of the system.
These restrictions can be applied at different levels:

- at task-level: agents are allowed to perform a subset of the tasks they have to do: this subset may be large or reduced to one task.
- at goal-level: agents have achievable goals (a goal to pursue) or homeostatic goals (a state to reach) [CL91];
- at protocol-level: agents must abide by a protocol or play a role;
- at motivation-level: a type of rationality or a motivation is assumed.

Some definitions may combine several levels: in [BM99], the control is made possible by the respect of the goals (goal-level) and the agent's active use of capabilities (motivation-level).
In fact, at all these levels, autonomy is limited:

- at task-level: during some periods, the agents are assumed not to try to perform some acts; so their autonomy is bounded by the limitation of the authorized acts;
- at goal-level: a given goal (achievable or homeostatic) is assumed to be pursued, and it is assumed to be the only reason of every action on the agent's part; the autonomy is bounded because the existence of a goal is assumed;
- at protocol-level: agents are assumed to follow the rules; autonomy is bounded because agents are assumed not to try to cheat;
- at motivation-level: agents are not free to reason, they are only allowed to think in the frame of a rationality; the autonomy is bounded by an imposed frame of reasoning.

**Remark 1.** *Here, I must make an important remark. We assume that agents are entirely designed and ready before they run. These agents are not autonomous towards the designer or the systems analyst, because agents' reasoning processes are decided, designed and programmed by humans before agents' run. However, they are autonomous from other agents and humans during the runtime on a computer.*
*This remark becomes false when considering genetic based agents. In this case, the programmer doesn't design the reasoning process, but a frame which allows agents to design themselves a kind of making decision process. During this time, this making-decision process is used by the agents to choose the actions they will perform. This second approach, while theoretically very promising, seems to be very time-expensive and so is limited to specific applications.*

Now, we could define an autonomous agent as an agent free from intervention in his decision-making process; but this freedom is not absolute. By the way, why cannot we consider partial freedom ?

### 2.2.2 What's the problem with partial autonomy

Most authors bound agents' autonomy in order to bound individual behaviors and then to obtain the expected behavior from the system.
So, why don't we accept to deal with partially autonomous agents ? Because sometimes, the agents we designed have to interact with others *who have to be considered as totally autonomous*. The important point here is not that agents are more or less autonomous, but that they have to interact with others *as if* they were totally autonomous. Typically, in electronic commerce, we have no control on decision making process of other agents. We may *hope* that they will abide by the constraints, that they will reason and act according to their motivation or rationality and thus that the system will attain its objective. But in a SARAH, these assumptions are not realistic due to the following reasons:

1. **Heterogeneity $\Rightarrow$ not expected behavior:** Their internal architecture (*e.g.* cognitive or not, bounded rationality) is not known and is not observable. A motivation may have very different results with these heterogeneous agents. For example, two economically rational agents may make very different choices due to different strategic choices: short-dated *vs* long-dated incomes, monopoly of an area *vs*

worldwide setting up. So even if an agent has the assumed motivation, he will not necessarily have the expected behavior.

2. **Economically rational and anonymous ⇒ intentional infringement** Agents interact with others whom they have never met and who might have only acquired a temporary identity. These agents may then be considered as anonymous, which prevents it from trusting them. Thus, we cannot rely on their reputation to hope that they will abide by the constraints (*e.g.* rationality, motivation). Hence, agents may try to cheat in order to earn more money. Even if the constraints are very easy to understand and to implement, the agent's owner may be interested on not abiding by these constraints. Since an agent is a black box for the others (informations about internal decision making process are strategic and thus must be not revealed), honesty cannot be insured *a priori*.

3. **Bug or hardware failure ⇒ unintentional infringement** An implementation mistake or a network failure may cause agents to infringe the constraints unintentionally.

To summarize, agents may infringe the constraints and even if then don't, the arising behavior may be far from the expected one. In this case, to impose constraints is unrealistic. A way to compel agents to abide by them is to use a kind of marketplace, where only some actions are allowed [SD01]. This platform doesn't constrain decision making processes, but only the communicative acts. It guarantees that the agents will follow the protocols, but it introduces several problems. First, the centralization of the interactions may overload the network. The second problem is that firms require to rely on the platform: they must be sure that it is safe, against piracy attacks as much as internal disclosure of strategic information by the owner (intentionally bribing the owner or unintentionally giving access to private information).
Finally, total autonomy is the more realistic autonomy in this context.

### 2.2.3 Total autonomy

#### How to deal with total autonomy

For the authors of [Cas95], total autonomy is a synonym of isolation, and hence is irrelevant and uninteresting for MAS. They understand the freedom from intervention as an interdiction to agents to interact with each other. From our point of view, an agent may try to influence another agent, because the second agent can decide to be influenced or not.
In [Ld00], the total autonomy leads to the impossibility to control autonomy externally. We are going to show that the control, even if more difficult, is possible.
We have just shown that in some situations, partial autonomy is unrealistic and that thus total autonomy is indispensable. But how can we deal with these totally autonomous agents ? How can we obtain the desirable system behavior ?
Let us remind the limitations of the partial autonomy:

1. constraints on decision making process may be not followed by agents (and we have no mean to verify if they are or not);

2. they may lead to an unexpected behavior (which is observable).

The main idea of the proposed solution is:

1. to remove all constraints from the decision making process;

2. to put them only on observable data.

No assumption is made upon the decision making process: agents are totally autonomous. But, in order to obtain the expected system's behavior, we need to constrain the agents' behavior. However, these constrains are not made upon internal data (the decision making process) that are not visible, but upon observable data (the communication acts for example).
What difference does it make from the traditional approach ? First, with this kind of constraints, we are able to check if agents abide by the rules by observing the data and if they don't, to punish them. Then, we just have to choose punishments severe enough to obtain the desirable behavior. The pair (control;sanction) is a way to influence agents, yet, it still respects their autonomy.
Then, since the desired system's behavior is constituted by observable data, we can know if the real behavior is the one expected.
To summarize, we can make no assumption on agents' behavior because of their total autonomy. So, we cannot force them to abide by a protocol, but, at least, we can check if they follow the rules or not. It implies that rules must be checkable; *e.g.* the protocol can make no assumption on agent's internal state. In our case, the only observable data is the messages sent.
Usually, autonomy is based on agent's decision-making process. For us, it is based on interaction protocols: in a SARAH (in which agents are totally autonomous), protocols' specifications are based only on observable data. *An agent is not autonomous by himself, but in respect to what others (agents and designers) expect from him*.

**A totally autonomous agent**

A totally autonomous agent is defined by:

- an internal part that is able to make decisions by taking received and perceived informations into account (more or less according to internal decisions); it may contain beliefs, learning, planning, etc.

- an interface part that allows:

  to send and to receive messages

  to act on and to perceive the environment

Then the agent must know the expected behavior and what will happen if he doesn't abide by the rules.
Often, the obligation to take initiatives is considered an important property of agents. In fact, it bounds their autonomy. Autonomous agents are able to take initiatives, they are not obliged.

As seen above, the main difficulty is to build a protocol (a set of interaction rules) that doesn't make assumption on the unobservable part of the system. For example, the internal part of agents is not visible. But it is also the case of certain acts: if an agent moves a block and all other agents are blind (e.g. cannot perceive the action), this act is not observable.

### Total autonomy: why and when

Why and when must we consider this kind of autonomy? Why and when are we going to consider a SARAH (instead of a another MAS)?.
In an electronic commerce context, interacting agents may be electronic or human. Of course, they need an electronic medium of interaction in order to be able to communicate and to understand the others. If we assume that the software used to communicate is designed by the user based on the specification of the protocol, this software doesn't bound the possible communicative acts of people. In other words, users can freely reason and decide (because people are autonomous), and act. So, agents whether they are electronic or human may be regarded as totally autonomous from a protocol point of view. But other circumstances lead to consider this class of agents.
In fact, large scale systems require many humans to program agents in the same time. These systems change during their life: some modules are added or removed, some are modified. It is then difficult to guarantee and to maintain the homogeneity of the agents. So, for a large software with a large amount of agents developed by different persons at different times, it is difficult to be sure that agents actually respect the specified constraints. In a SARAH, the consequences of heterogeneity and possibilities of bugs or hardware failure may also be taken into account: agents have to be considered as autonomous.
In other cases, if a small number of agents is designed by a single person, if the system does not have to evolve, agents may be weakly autonomous (except in human system simulation case). But in fact, this design of agents is closer to Distributed Problem Solving than Multi Agent Systems.
More generally, the autonomy is often considered as a desired property of agents. In [BD92], "*autonomy allows the design of agents flexible enough to function effectively and efficiently in a sophisticated world*". The more the agents are autonomous, the more the system takes advantage of it.

**Example 1.** *A good example of strong autonomy is given in [SLA$^+$99]. Agents are assumed to be insincere, and the goal is then to motivate self-interested agents to follow the desired search method in order to reach a socially desirable outcome, as a consensus. For example, enforcement mechanisms will motivate the agents to search exactly what they are assigned. To motivate agents to follow the protocol, [SLA$^+$99] uses penalties: agents who find a cheater are rewarded by the penalty paid by this cheater. Agents have also to be motivated to search cheater. But if the penalty is high enough, the supervisor is motivated to search them and then agents are not motivated to cheat; supervisors will then neither be motivated to search: a Nash equilibrium can be attained (there is Nash equilibrium when each agents strategy is a best answer to the strategies of the others [MCWG95, Nas50]).*

**Human beings and autonomy** In our society, humans are educated and have a reputation; so they try to behave according to written rules (the law, regulations) and tacit rules (morals, ways and customs, rules of etiquette, code of honor). Further more, explicit punishments exist when some rules are not abide by. Sanctions are chosen in order for cheaters to be at a disadvantage compared to honest agents. For example, humans must go in jail or are fined, because freedom and money are assumed to be important for humans. In MAS, when considering anonymous agents, reputation doesn't play its role of incitement to obey the rules. Their decision-making process is not controlled by other agents: thus, they must be considered autonomous. We adopted the same reasoning to define a protocol among totally autonomous agents.
Fundamental questions have been asked by philosophers about these concepts. Do humans have truly free will ? Is their future predetermined ? Is autonomy an illusion ? Thought no definitive answer has been given to these questions, laws and rules exist and allow societies to exist. They are based upon the assumption that humans are autonomous, even if nobody knows if it is true or not. We can do the same for agents: though the subjectivity of autonomy [S97], we may consider that agents are autonomous, even if in fact, they truly are not. Under this assumption, we design roles, rules and protocols that allow agents to interact with another one. We just have to know what is important for agents (given by its rationality) in order to replace jail and penalties.
To conciliate total autonomy and system control in our society, each (autonomous) citizen has to obey the law. If somebody infringes the law, he will be punished, what requires that the infringement has been previously detected. In French law, it is not forbidden to have racist thoughts; but to deliver racist remarks is prohibited. The former is not observable, while the later is. In order to control a system, the designer may make assumptions on the agents' behavior, but these assumptions must be verifiable. Since we consider that the internal state of agents is a black box, only observable acts and sent communications can be checked.

### 2.2.4 Salesmen as totally autonomous agents

In an e-commerce context, enterprises' salesmen are replaced by agents. These agents contain strategic information, this incites each enterprise to design and develop these agents by themselves, what leads to heterogeneous agents.
Of course, they are economically rational, and thus may try to cheat. However, they may use electronic signature in order to prove their identity, what allows to use reputation to incite agents to abide by the protocol in the long-term. But it is difficult to prove that an infringement is intentional (cheating) or not (due to bug or hardware failure). The unintentional ones may be reduced by using low-level protocols of communication (that rectify some hardware failures) and by proving that software are correct.

#### What is special in e-commerce MAS ?

First, agents are purely communicative; so the only visible acts are the exchanges of messages.
Secondly, private conversations are not visible (and thus checkable) by the system, but

only by the receiver of the message. If something goes wrong, no agent is authorized to examine the communication. In a collaborative MAS, agents trust each other; then if one of them detects an infringement, the others believe him. On the contrary, in an e-commerce context, agent may lie. So an infringement detected by a lone agent is not enough to be sure that the accused agent is guilty. Two solutions are conceivable:

1. To add some policemen-agents which role is to control that the protocol is abided by worker-agents. All acts expected by the protocol are observable and observed by policemen-agents. But this solution add new problems: how to trust these new agents ? where are they running ?

2. The actions on which the protocol is based must be visible by *all* the agents of the system, what means that any message must be sent to all the agents. In this way, an agent on his own cannot cheat. In this case, the policeman-role is played by all the agents.

Thirdly, the protocol must be egalitarian: it might not favor any enterprise.

### 2.2.5 What weak autonomy allows

In a SARAH, agents are totally autonomous. But if they are not, things are simpler.
To consider weak autonomous agents allows the design of protocols that are in fact decentralized algorithms: they compute a solution that maximizes personal but known criteria. But generally, it is not possible to maximize all criteria; so, a particular rationality is assumed. It could be :

- a global rationality [BdV97]: agents tend to maximize the satisfaction of the whole system;

- a group rationality [BdV97]: agent tend to maximize the satisfaction of groups;

- a happiness on average [Sha53]: on average, agents are satisfied the best as possible.

In some circumstances, these individual rationalities allow the system to behave as expected. But, of course, agents are weakly autonomous.

### 2.2.6 Autonomy is not independence

Sometimes, autonomy is mistaken for independence. In [Ld00], independence is defined as a weak view of autonomy, since the strong view is called absolute. For a clear identification, we choose to use the terms *autonomy* to name the absolute autonomy, and the term *independence* to name the weak autonomy.
An agent is dependent if he cannot do something without the help from some other agent. He may be more or less dependent, depending on the number of solutions leading to the expected result, on the part of solutions (and their quality too) that require the help of others. For instance, a group of agents has to move desks. If furniture are difficult to move, an agent that is not enough strong to move a desk alone may be dependent on others. Another one may be able to work alone, but less efficiently: so, he will be less dependent

than the first.
So, an independent agent is able to reach his goal alone, even if the use of others may help him (*e.g.* to be more efficient). The independence is a property of the set of plans able to achieve the goal; it depends on the problem to solve and on the others' competences. The autonomy is a property of the decision making process: it forbids direct intervention of others.
Agents may combine independence/dependence and autonomy/non-autonomy. A team of furniture remover agents may be more or less dependent according to their skills (as seen above), or the data of the problem (the weight of furniture may oblige agents to cooperate). These agents may be non-autonomous if they always wait orders from a human or a leader agent.
These difference between the two terms has been pointed out in [aSD95]: even if agents are to be considered autonomous, it is not reasonable to suppose that they are also auto-sufficient, because an agent would have to be able to perform all the actions and would have control over all the resources needed in a plan in order to achieve a goal he is committed to.

### 2.2.7 Autonomy and rationality

In a SARAH, agents are autonomous and rational. But the rationality doen't bound the autonomy.
However, often, an agent is said to be rational if he behaves to maximize a criterion (utility, income, *etc.*). Since [Sim72] and [Rus94], the economical rationality have been replaced by the bounded rationality. They shown that an agent doesn't spend all his time optimizing his criterion, because of his limited resources: he stops when he is satisfied. Bounded rationality has been used to design agents and protocols that take into account bounded resources [SL97, SL95, DeV96].
In fact, motivations might be so complex that it is impossible to predict agents' behaviors. Thus, we will consider a weak rationality: an agent is rational if he is inclined to act according to his assumed motivations. His behavior is thus estimated (with more or less uncertainty) but never predicted.
The legitimacy of the solution is also important. To randomly choose the alliance structure among all possible alliance structures is not a very legitimate solution. To legitimate the solution, many works choose an external criteria of satisfaction based on an assumed rationality [SSJ97, SK95a, SK95b, SK96]. The solution may then be accepted, because it satisfies globally as best as possible the agents. But first, the agents should prefer to negotiate in hope of winning more, taking the risk to lose more; then this solution needs agents criteria to be public.

In most works, an agent is autonomous if he has his own goals; these goals are known and are used to design protocols that satisfy the agents as best as possible. In [SK96], protocols (called regulations) are incorporated into every agent, but each one of them chooses its strategy for the interaction individually and joins an alliance only if it increases its personal payoff: he seems autonomous because he can choose his strategy, but this autonomy is bounded because he must only join beneficial alliances.

In real-world systems, an agent can be a human interface, which complexity prevents him to be modeled by a goal to be reached. Even if an agent is not a human interface, a definition of strong autonomy leads to consider that agents goals and behavior cannot be known. We say that an agent is totally autonomous if he can decide and act as he wants.
Nevertheless, a set of crazy agents will rarely produce the expected results. Thus, a protocol is given to them and agents that don't follow it will be punished. Now, the question is: which sanctions could incite agents to follow the protocol ? For example, in an economic context, agents are considered to be economically rational, and so to try to maximize their benefits. So, such an agent may be sanctioned by paying a penalty. Although, for some reasons (strategic, technical or unknown), they may prefer to lose money (at least in limited time) than to follow the rules.
The rationality is used to decide which penalty is the most appropriate to sanction the agent.

## 2.3 Autonomy at organizational-level

We have just defined autonomy at agent-level by considering freedom in the making decision process.
But in MAS, agents interact with others and the autonomy also influence interactions.
In [dL96], "*autonomy allows for no artificially imposed rules of behavior; every behavior must be a consequence of the understanding and processing capabilities*". For the authors, at agent-level, an agent is autonomous if its goals are not provided by external sources and if he will only adopt a goal that favors him; at organizational-level, the effects of an interaction cannot be guaranteed, the intention of others cannot always be recognized and an agent can only know about himself. We argue that the autonomy is not a property of the agent: an agent may have goals provided by himself and/or by external source, he may decide to adopt goal for his personal reason. What is important is that the other agents consider him autonomous and then assume that the effects of interaction are not guaranteed. Moreover, a static definition of decision-making interaction can limit agents' ability to take initiatives or can enforce the communication overhead. That leads to a need of adjusting collaborative decision making as needed:

- either by working alone: in general, either goals cannot be reached, or they can be, but less efficiently;

- either by forming new alliances: the number of achievable tasks is increased and more efficiently.

An alliance is a group of agents that decide to work together. They can do so because working alone doesn't allow to perform certain tasks, or because these tasks are performed less efficiently.
The formation of an alliance is decided by and only by the future members of this alliance.
Agents are autonomous because they decide with whom they will cooperate: it is not decided by an external source. In the same way, an alliance is dissolved only if its members decide that.
This dynamic choice of partners enable to quickly react to an environment change and to adapt the organizational structure.

As autonomy at agent-level enables agents to react more efficiently, autonomy at organizational-level allows organizational structure to be more adaptive with regard to the environment changes. To conclude, high autonomy leads to the formation of alliances.

### 2.3.1 Autonomy and organizational theories

Organizational theory works address the problem of the diversity of real-world organization structures. Usually [Sch90], these works are classified in three categories: management theories, sociological theories and psychological theories. The main difference between these points of view is the definition of the autonomy.

**Goals of organizational theories** The goal of management theories is to analyze, to distribute and to coordinate activities ensuring that the activities are processed with the minimum cost in money and/or time. This theory must produce results in order to advise leaders to design an efficient and durable structure.
Sociological theories study why structures are different, why they are more or less stable and more or less efficient. Relations between structure and efficiency are deeply studied: the goal is not to find the secret of success, but to understand how things really happen. In fact, this is very complex: two organizations with opposite or same structures might be equally efficient or not.
Psychological theories study which organization characteristics influence the labor in firms. Agents' reasoning models are based on human internal mental processes, but these models are rarely used in MAS, and if they are, we cannot consider that it will be always the case. We must use enough general theories applicable to all situations. So, results in psychological theories will not be used here, because they are too human-dependent to be used in virtual organizations.

**Autonomy in organizational theories** Management theories try to predict the behavior of organizations and to advise managers where sociological theories are explanatory.
The first are obliged to use simple models in order to enable large scale computations; but these models are unrealistic indeed: humans' behaviors are generally the result of very complex processes. Agents are regarded as weakly autonomous, blindly driven by their (generally economic) rationality.
The second are more precise and often very realistic in order to be as close as possible to the reality; but they are not completely useful because of their complexity. In these theories, agents are totally autonomous and they show that most of the time, the real behavior of an organization is completely different from provided for. This fundamental result incited me to assume that agents are totally autonomous and to take care to what agents could do effectively.

## 2.4 Autonomy at system-level

We have seen what total autonomy means at agent-level and at organizational-level. The next and last level is the system-level. I have not really work on this level, but I think that the first ideas I have had are very promising and should be carried on in the future.
Sometimes, a Multi-Agent System has to make a decision. To do it, it generally uses a process given by the designer of the system (most of the time a protocol of negotiation). Agents, using the protocol, make the decision. But this protocol is a part of the system, it is imposed by the designer. However, the choice of a protocol depends on the state of the system when it is used. Some properties cannot be satisfied simultaneously, for example:

- A legitimate solution needs lengthly negotiations, when a random or centralized choice may be very quick. If the system is small, we can indulge in a complex process, but if it is large, we are obliged to choose a less legitimate but faster process. The size of the system may vary during the process and it can be very difficult to predict its evolution. So, you will better do to let agents' system choose the protocol of negotiation.

- If the agents know each other, if they work often together, they can trust each other. So simple and quick protocols can be used. But if a lot of anonymous or unknown agents enter the system, or if the system is electronically attacked, the system should better change its protocol in order to secure itself.

Since each protocol has its own properties and since they cannot all be satisfied, the designer has to make a choice before running the system. The other solution is to let the agents decide which protocol to apply, depending on the situation.
This idea of a dynamic change of the protocols has already been proposed in [Kon02]. The main reason given is – as usual – the reactivity of the system towards an environment change. But partial proposed solutions have same defaults than the usual solution to deal with autonomous agents at agent's level: the autonomy is weak and leads to unrealistic and unusable solutions.
Today, we use pre-compiled highly-structured "social laws" to coordinate agent activity [ST92, MT93]. But agents have decided to follow the social laws because they were designed to do so and not because they benefit individually from following these laws: the designers should agree in advance which regulations the agents will use.

## 2.5 Autonomy: consequences

In a SARAH, the agents' total autonomy has several consequences on the design of protocols and organizations of agents in alliances.

### 2.5.1 Consequences on the design of protocols

#### Required properties

To obtain coherent collective behaviors, it is necessary to lay down rules, but to respect personal freedom. These rules should constraint only *perceptible* data. In the event of

fraud, agents can always be sanctioned; sanctions incite them to follow the given rules.
An observable data is an act that is directed towards the others. It may be:

- A speech act sent by an agent to another one, and so observable. So, if an agent sends an unauthorized speech act or if doesn't send one that he should have sent, the mistake can be detected by the receiver. However, the receiver that lodges a complaint against the sender should be able to prove the fraud. So, specific acts must be added. In electronic commerce, several protocols have been developed in order to prove that a message has been sent or not.

- Physical acts that happens when considering robotic agents. It is not our purpose, but similar problems arise: only acts that are observable and which observation is provable may be used by a protocol.

**Heterogeneity** In a SARAH, agents' heterogeneity prevents from assuming and from imposing complex computational capabilities [KSE98], even if the agents may not have the time or the ability to do inferences [Kon86, NKP93].

**Properties** In our context, a protocol must be:

1. universal: in particular it assumes nothing on the individual choices, nor on individual rationalities, thus respecting the agent's freedom;

2. egalitarian: it doesn't support particular classes of agents;

3. distributed: centralization brings the usual problems (overload of the network and the central agent, weakness with the breakdowns...), but in addition, centralization increases the power of the central agent enabling him to cheat; although one can make the assumption of an impartial agent, the integrity of an agent could always be blamed (external influences, corruption), which would harm the legitimacy of the solution.

### The use of the rationality to incite agents to cooperate

In [Ś97], though autonomous, an agent must influence or adopt other agents' goals, and thus, he must be capable of being influenced by and adopting goals. Hence, he may interact with others, what leads to a limited social autonomy. We agree that if agents don't change their position, they have no chance to reach a consensus, but we refuse to limit their autonomy.
Even if agents may influence each other, they cannot force others to change their will. In the same way, the protocol doesn't force agents to change their opinion, but it can be designed in order to favor flexible agents (i.e. agents that accept to change their preferences). So, first, the designer of the protocol must know the rationality of the agents to know how they may be satisfied. Then, using rewards based rationality, he must find how to incite agents to collaborate. So agents are autonomous, because they choose to cooperate or not, but, since flexible agents are rewarded, most agents will be most of the time flexible enough.

### 2.5.2 Consequences on the design of organizations

The freedom of the agents also modifies the perception which one can have of the organizations, which are indeed sometimes regarded as entities themselves, depending certainly on their members, but having nevertheless their own attributes like a certain amount of autonomy. In fact, an organization is a set of dependent agents (by contract, punctually), but which each one of them is always free.
Each agent plays a certain role there, but for the same reasons, role and behavior aren't identical. A role is a set of rights and duties, codified in a rule, in the hope to obtain a coherent overall behavior. The rights are represented by a set of possible actions at each stage and a duty by a set of actions awaited by the other agents. But any agent keeps its freedom and thus, can respect the rule or not, and it is thus necessary to take into account this eventuality.

## 2.6 Conclusion

The notion of agent may be regarded as an encounter of several concepts, but he owns a new and promising property: the autonomy. It allows systems to deal with softwares and human beings in the same way, it facilitates the design of large systems and it increases the efficiency when the environment's changes are not precisely defined.
However, many definitions exist, with several degrees and kinds of autonomy. To deal in an electronic commerce context forces us to consider every agent as totally autonomous. Moreover, some other constrains are imposed by the context: heterogeneity, anonymity, economic rationality (without forgetting autonomy), infringements. This define a subset of MAS, the SARAH (System of Autonomous and RAational Agents with Heterogeneity).
The autonomy at agent level has consequences on the design of protocols. Actually, some known solutions cannot be used: centralized control, assumptions on agent's internal states or architectures, *etc.*Finally, a protocol must abide by a set of rules:

- a protocol must be distributed, egalitarian and universal;
- no assumptions on agents' behaviors;
- no complex capabilities required;
- assumptions only on checkable (*i.e.* observable) data;
- infringement implies the use of supervisors;
- motivations to follow the rules based on the rationality.

Autonomy also changes the way to design organizations. In fact, agents may chose his partners, what leads to the formation of dynamic organizations, the alliances.
These rules are summarized in figure 2.1 on the next page. Of course, the protocol of consensus reaching presented in the next chapter follows them.

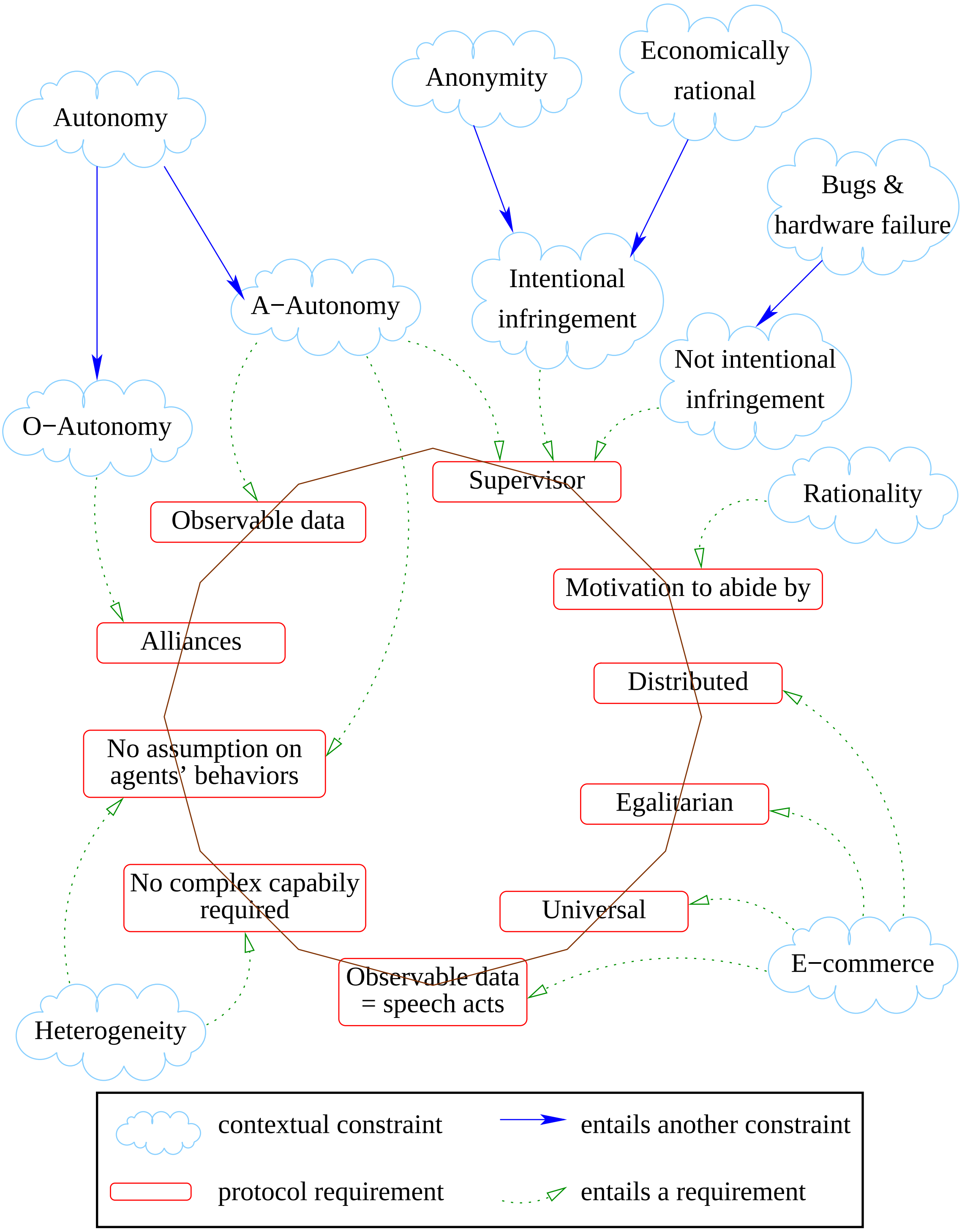


Figure 2.1: Autonomy to protocol in a SARAH

# Bibliography


[aSD95] Jaime Sim ao Sichman and Yves Demazeau. Exploiting social reasoning to deal with agency level inconsistency. In Victor Lesser, editor, *Proceedings of the 1st International Conference on Multi-Agent Systems*, pages 352–359. MIT Press, 1995.

[BD92] Olivier Boisier and Yves Demazeau. A distributed artificial intelligence view on general purpose vision systems. In E. Werner and Y. Demazeau, editors, *Decentralized AI3: Proceedings of the Third European Workshop on Modelling Autonomous Agents in Multi-Agent World*, pages 311–330. Elsevier, 1992.

[BdV97] Magnus Boman and Walter Van de Velde, editors. *Multi-Agent Rationality: Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, volume 1237. Berlin, springer verlag, lecture notes in economics and mathematical systems volume edition, May 13-16 1997.

[BM99] K. S. Barber and C. E. Martin. Agent autonomy: Specification, measurement, and dynamic adjustment. In *Proceedings of the Autonomy Control Software Workshop at Autonomous Agents*, pages 8–15, 1999.

[Cas95] Christiano Castelfranchi. Guarantees for autonomy in cognitive agent architecture. In Springer-Verlag, editor, *Intelligent agents*, pages 56–70. M. Wooldridge and N. Jennings, New York, N.Y., 1995.

[CL91] A. Covrigaru and R. Lindsay. Deterministic autonomous systems. *AI Magazine*, 12:110–117, 1991.

[DeV96] A. DeVany. Information, bounded rationality, and the complexity of economic organization. *Taïwan Journal Political Economy*, 1, 1996.

[dL96] Mark d'Inverno and Michael Luck. Understanding autonomous interaction. In W. Wahlster, editor, *Proceedings of the 12th ECAI*. John Wiley and Sons, Ltd., 1996.

[Gad98] Adam M. Gadomski. Personoid's organizations framework: an approach to autonomous software systems architecture based on intelligent agents organization. Technical report, ENEA, Italian National Agency for New Technology, Energy and the Environment, Rome, 1998.

[GB99] Zahia Guessoum and Jean-Pierre Briot. From active objects to autonomous agents. *IEEE Concurrency*, 7(3):68–??, 1999.

[JCd] I. Jarras and B. Chaib-draa. Modélisation du raisonnement multiagent : une approche basée sur les labels. In *Actes des 6èmes Journées Francophones en IA distribuée et Systèmes Multiagents, JFIAD-98*, Nancy. Hermès.

[JW95] N. R. Jennings and M. Wooldridge. Applying agent technology. *Applied Artificial Intelligence*, 9(4):351–361, 1995.

[Kon86] K. Konolige. *A Deduction Model of Belief*. Pitman, London, 1986.

[Kon02] Jean-Luc Koning. Dynamic choice of agent interaction protocols. In *Intelligent Information Processing*, pages 167–176, 2002.

[KSE98] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.

[Ld00] Michael Luck and Mark d'Inverno. Autonomy: A nice idea in theory. In *Agent Theories, Architectures, and Languages*, pages 351–353, 2000.

[MCWG95] A. Mas-Colell, M. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford, 1995.

[MF95] D. Moffat and N. Frijda. Where there's a will there's an agent. In M. Wooldridge and N. R. Jennings, editors, *Intelligent Agents: Theories, Architectures, and Languages*, volume LNAI 890, pages 245–260. Springer-verlag edition, 1995.

[MT93] Y. Moses and M. Tennenholtz. Off-line reasoning for on-line efficiency. In *IJCAI 93*, pages 490–495, 1993.

[Nas50] J. Nash. Equilibrium points in n-person games. In *Proc. National Academy of Sciences*, volume 36, pages 48–49, 1950.

[NKP93] M. Nirkhe, S. Kraus, and D. Perlis. Situated reasoning within tight deadlines and realistic space and computation bounds. In *Proc. of the 2nd Symposium On Logical Formalizations Of Commonsense Reasoning*, 1993.

[Pet96] Charles Petrie. Agent-based engineering, the web, and intelligence. In *IEEE Expert*, volume 11, pages 24–29, December 1996.

[Rus94] S. Russell. *Artificial Intelligence : a Modern Approach*. Prentice Hall Engineering, Science & Math, 1994.

[S9́7] J. Alfredo Sánchez. A taxonomy of agents. Technical report, ICT, Deparment of Computer Systems Engineering, México, January 1997.

[Sch90] J-C. Scheid. *Les Grands Auteurs en Organisation*. Dunod, 1990.

[SD01] Carles Sierra and Frank Dignum. Agent-mediated electronic commerce: Scientific and technological roadmap. In *AgentLink*, pages 1–18, 2001.

[Sha53] L. S. Shapley. *Contributions to the Theory of Games*, chapter A Value for N-Person Game. Princeton University Press, 1953.

[Sim72] H. A. Simon. *Theories of Bounded Rationality*. Decision and Organisation. 1972.

[SK95a] O. Shehory and S. Kraus. Coalition formation among autonomous agents: Strategies and complexity. *Lecture Notes in Artificial Intelligence*, From Reaction to Cognition, C. Castelfranchi and J. P. Muller (Eds.)(957), 1995.

[SK95b] O. Shehory and S. Kraus. Task allocation via coalition formation among autonomous agents. In *Proc. of IJCAI-95*, pages 655–661, Montreal, August 1995.

[SK96] O. Shehory and S. Kraus. A kernel-oriented model for autonomous-agent coalition-formation in general environments : Implementation and results. In *Proc. of AAAI-96*, pages 134–140, Portland, Oregon, August 1996.

[SL95] T. W. Sandholm and V. R. Lesser. Coalition formation among bounded rational agents. In *Proc. of IJCAI-95*, pages 662–669, Montréal, 1995.

[SL97] T. Sandholm and V. Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, Special issue on Economic Principles of Multiagent Systems(94(1)):99–137, 1997.

[SLA+99] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohme. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 1999.

[SSJ97] O. M. Shehory, K. Sycara, and S. Jha. Multi-agent coordination through coalition formation. In M. Singh A. Rao and M. Wooldridge, editors, *Lecture Notes in Artificial Intelligence*, volume Intelligent Agents IV - 1365, pages 143–154. Springer, 1997.

[ST92] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proc. of AAAI 92*, pages 276–281, 1992.

[tFS00] Christian Lemaître and Amal El Fallah-Seghrouchni. A multiagent systems theory of meaning based on the habermas/bu"hler communicative action theory. In *Proceedings of IBERAMIA-SBIA*, pages 116–125, 2000.

[VBS98] Laurent Vercouter, Philippe Beaune, and Claudette Sayettat. Apprentissages dans les sma. In *Actes des 6èmes Journées Francophones en IA distribuée et Systèmes Multiagents, JFIAD-98*, Nancy, 1998. Hermès.

[VE00] G. Vauvert and A. El Fallah - Seghrouchni. Coalition formation among egoistic agents. In *Proc. of MAMA'2000*, Wollongong, Australia, December 11-13 2000. International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce. Awarded "One of the three best papers" (among 92 accepted papers).

[VE01] G. Vauvert and A. El Fallah - Seghrouchni. Coalition formation among strongly autonomous and weakly rational agents. In *Proc. of MAA-MAW'2001*, Annecy, France, May, 2-4 2001. Modelling Autonomous Agents in a Multi-Agent World. 10th European Workshop on Multi-Agent Systems. Acceptation rate of papers : 13/73=18%.

[WJ94] Michael Wooldridge and Nils Jennings. Towards a theory of cooperative problem solving. In J.-P. Müller Y. Demazeau and J. Perram, editors, *Proceedings of the 6th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 15–26, 1994.

# Chapter 3

# Consensus among Totally Autonomous Agents

## Contents



## Figures

**Tables**



# 3.1 Introduction

In chapter 2 on page 19, we explained why we need totally autonomous agents and how to deal with them. A totally autonomous agent has its proper opinion and decides to be influenced or not by the others. So, other agents cannot assume that he will accept (or refuse) the content of the communicative act. Hence, possibly, no agent may change its opinion.
However, often, agents interact with each other: by working on the same data, by communicating, by collaborating, by competing, *etc.*Sometimes, a common decision needs to be taken by all agents (or at least several agents), because this decision is a concern of all of them. But, how can they come to an agreement ?
Of course, if only one possibility exists, or if there is a possibility that satisfies all agents, the agreement is easy to be reached. Unfortunately, most of times, agents' preference are not the same and the situation is conflicting. There is a need of a mean to reach a consensus.

## 3.1.1 Ways to make a decision

**To impose a choice**

The simplest way is to impose a choice to the agents. The decision function is statically decided by the system designer. This can be a good way if:

1. A best solution exists; else some agents should prefer an another solution and thus would be disappointed by this choice.

2. This best solution can be statically computed, or at least, how to compute this best solution can be known before the system runs; however, it is not always the case.

**A best solution doesn't exist** Generally, an optimal solution doesn't exist (of course, if it exists, agents should choose it). Some criteria of sub-optimality have been proposed in order to satisfy agents "in a certain manner":

- The Shapley-optimality ([Sha53]) is a very general criteria: since all agents cannot be satisfied, agent could be satisfied "on average" (in the mathematic mean). For each agent $a$, the system knows his satisfaction function: $f_a : \mathcal{S} \rightarrow \mathbb{R}$, where $\mathcal{S}$ is the set of choices. Then, a solution $\sigma^\star$ is Shapley-optimal if $\frac{1}{|A|} \sum_{a \in A} f_a(\sigma^\star)$ is maximal. This expression is the average of agents' satisfaction values.

- The Pareto-equilibrium [CAP02]: no solution may satisfy more all agents (the solution is dominant). All agents are not satisfied, but at least, all agents are not unsatisfied.

- The Nash-equilibrium [Nas50]: no agent may change his strategy alone, because he would always lose.

These criteria have some constrains:

- The satisfaction requires to be representable as a numerical value, which is often difficult.

- Generally (*e.g.* [ZR94, SK95]) the computation uses internal informations of agents: a utility function, inmost preferences, *etc.*In centralized computation case, several problems arise: overhead of the network, risk of hardware failure, and most of all, the agents must trust the central agent. In every case, these strategic informations must not be given to other, because agents are totally autonomous. They can deliver a position, but nobody may be sure that it is their true opinion.

- The choice is computed using an external operator, which is not perfect (*e.g.* the integral of Cochet in [Akn00]). So, it should not convince all the agents, and then, the choice will not be legitimate. If a protocol leads to a solution that is not legitimate, agents will not want to apply it. For example, using the Shapley Value [Sha53] is not acceptable because some agents should prefer to try to earn more, even if they risk to lose more.

These criteria constrain the internal representation, force agent to give strategic informations and are not necessarily legitimate.

### Voting systems

In MAS , agents sometimes must make a common decision between several alternatives. Since each agent has his own criteria, they generally don't agree on the alternative to choose. A wide variety of voting methods have been proposed: several of ones take into account only each voter's first choice, some take into account complete preference orderings, and some take into account the intensity of preferences. Some select the alternative most preferred by the most people, some select a compromise alternative. But, as shown in [Arr63], there is no method of aggregating individual preferences over three or more alternatives that satisfies conditions of fairness and always produces a logical result. Some other scholars have attempt to relax some of these conditions of fairness and logic, but no satisfactory voting system has been found. M. Condorcet discovered the paradox of voting over 200 years ago, but it has been more deeply studied by Kenneth Arrow in [Arr91].

**Argumentation**

No optimal solution exists. So agents must modify their opinion in order to reach a consensus. One of the most promising and interesting way is argumentation, because agents can be totally autonomous. Generally, agents exchange arguments and counter-arguments in order to convince the others to modify their opinion. In our case, argumentation is not used to reason with inconsistent knowledge, because the concept of inconsistency does not mean anything here even if it may exist inside the agents: agents don't try to find what is true and what is false. It is neither used to handle uncertainty: agents know what they desire. Most of approaches [LS89, Pol92, PL92, Dun95] deal with inconsistency either with uncertainty.
However, argumentation could be used to convince others to adopt a solution. The persuasion has been used in several contexts, but most of time it uses formal logic and assumes that agent architecture contain some specifics modules [DDKV01, Dun95].
Agents must understand messages in order to be able to answer in an efficient way: building counter-counter-arguments, attack using a new argument, *etc.*Argumentation requires complex process capabilities: to infer on complex logic formula, to use a large amount of knowledge (the ontology). This way can be used with agents with strong process capabilities, and obliges them to adopt a common internal architecture (or at least the same language). It cannot be used with totally autonomous agents, because of their heterogeneity and their freedom of internal design.
Moreover, to prove that a consensus is reached, logic based approaches require to assume that agents reason in a certain way [DDKV01], arguing that agents are rational. But these assumptions are not allowed with totally autonomous agents.

## 3.2 Totally autonomous agents reaching a consensus

We focus on self-interested agents acting in an economic context. They have individual goals (to increase their incomes) and might be pure software agents or interface for human, and then no strategy is assumed and rationality is bounded [VE01b, VE01a].
The protocol we propose is assumed to be known and accepted by the agents, but they are completely autonomous: protocols take into account possibilities for agents to try to cheat.

### 3.2.1 Our approach

Totally autonomous agents can use argumentation, but it requires strong computational capabilities and constrains internal agents' architecture. Our purpose is:

1. To simplify as much as possible the argumentation process in order to allow all the agents to be able to use it. But what does a simplification mean ? Symbolic

processes have to be no more required and all exchanged symbolic representations must be replaced by numerical ones. In fact, numerical representations generally require less computational capabilities than symbolic ones.

2. Don't make assumption about internal architecture. For instance, no symbolic computational capabilities have to be required.

### 3.2.2 Protocol requirements

The protocol is intended to be used by totally autonomous agents; thus, it must respect some constrains:

1. No assumption about internal architecture.

2. Legitimacy of the solution. Generally, no optimal solution exists, so the chosen solution doesn't satisfy all agents. However, there are several ways to choose this solution. Often, an external operator is used: maximization of the Shapley's value, random choice, *etc.*but generally it doesn't satisfy all agents.

### 3.2.3 Principles

**Arguments**

We propose a protocol (see figure 3.1 on page 48) that allows agents to reach a consensus. Since agents may have different opinions, they should change their positions. But they will not change without reason; so, agents must interact to influence others in the hope that they will come round to their opinion. How to interact ? Several solutions may be considered:

1. Agents can send resources: money, gifts [RcB01], or anything else that may be regarded as interesting by the receiver, depending on its rationality. They can bribe, corrupt or try to gain others' favors.

2. Instead of presenting gifts, agents may promise something in the future: they can undertake to retaliate the collaboration by being reimburse in the future the loss of resources.

3. They can exchange informations that are regarded as potential influencing arguments by the sender. Argumentation is often used to convince the others by giving new informations as a rational way to reach a consensus.

Each of these solutions can be criticized:

1. The goal is to reach a consensus in an egalitarian way (see subsubsection 2.5.1 on page 32), that is to say that the result of the process must not depend on external facts. In this case, the richer agent could always win. Moreover, generally, it is not easy to convert a service into a resource.

2. Commitments may help to reach a consensus more quickly: an agent may accept to make concessions if he is certain to be rewarded later. But, commitments require no-anonymity (not guaranteed in our context) in order to allow agents to keep a long term contact with others; moreover, agents don't always work together during a long time. As seen in [caCP96], they may increase the system's efficiency.

3. As seen before (subsubsection 3.1.1 on page 44), they require processes that are too complex.

**Remark 2.** *Since agents are totally autonomous, their ways to influence others are not limited. In the same time that they use the protocol, they can bribe and promise. However, it is not required by the protocol.*

In MAS literature, some paradigms have been proposed:

- auctions [VJ98] are simple and usable by heterogeneous agents, but they assume that the only criteria is money and only one resource may be allocated at one stage;

- game theory approaches [RZ94]: it is a more powerful approach, but agents are assumed to have perfect information and perfect rationality; since we consider agents as weakly rational, it is not acceptable (see subsubsection 2.2.3 on page 26 or [VE01a]);

- most of other powerful paradigms need complex process capabilities.

Usual argumentation processes are very complex and not usable by heterogeneous agents. But, why does an agent argue ? Because he wants to change other's opinion. How does he argues ? By giving reasons to change. In the case of a consensus reaching problem, he tries to come the other round to his opinion. So given reasons look like: "you should come close to my position, because ...". The goal we pursue is to simplify arguments. The simplest argument is to say: "you should come close to my position". Finally, agents will exchange only their current position:

1. It's a very simple information, which can be understood and processed by heterogeneous agents.

2. Agents don't need to reveal their internal utility function (according to autonomy constraints), even if knowing it should favor speed convergence. For example, [DJJT01] induces hypothesis about the agents' preferences automatically, but with uncertainty.

Opinions are used in three ways:

i as proper preferences (what I really prefer);

ii as a representation of the current position (voluntarily influenced by other opinions);

iii as a way to influence other agents (you may take my position into account in order to modify your opinion).

**To incite to evolve positions**

Since agents are motivated to satisfy their own preferences first, they have no interest to change their positions; so we must add a process that incite them to do so, especially if we need to guarantee that agents reach a consensus.
The second question is how to incite agents to modify their positions. That may be the time cost: if agents don't choose quickly, they will earn less; but money must be the only criteria used by agents to estimate a choice.
Due to the reasons given above (subsubsection 3.1.1 on page 44), argumentation cannot be used as the main tool to induce evolution in agents' positions. However, if they are able to use argumentation (to argue and to understand arguments), they could use it to speed up the convergence: our protocol allows agents to use other procedures to convince others.
In [VE01b], we choose to allow agent to make evolve their preferences, but when all the agents give the same opinion twice (a cycle is detected), that means that the process may never end. Initially, each agent is a coalition that contain one agent, himself. Coalitions have then the possibility to merge with other coalitions. A coalition is a group of agents that decide to behave as an entity; the opinion of the coalition is computed using an aggregation operator on opinions of its members (that are hidden to outside the coalition). Agents are free to choose to merge their coalition or not, but if nobody decides to merge, then the two coalition that are nearest are forced to merge. We need two operators to choose the two nearest coalitions (a coalition may contain only one agent) and to allow computation of a coalition's opinion.
In each coalition, a member has a particular role: a representative role. He communicates with other representatives (sending coalition preferences). Formally, a coalition may be defined by:

**Definition 1 (Coalition).** *A coalition $\lambda \in \Lambda$ is defined by $\lambda = \langle \mathcal{A}, a_{rep} \rangle$, where $\mathcal{A} \subset A$ and $a_{rep} \in \mathcal{A}$ a coalition member with a representative role, with the constraint that an agent can belong to only one coalition.*

**To decide when a consensus is reached**

The third question concerns the reaching of a consensus. We could use a vote to decide if there is a consensus or not, but that will make the process excessively complex (a new decision process included in an other decision process). Moreover, convergences could be very slow, because agents that are not satisfied by the result of the consensus may vote so that consensus is not reached.
Thus, we decide to use an operator based on agents' positions (see section 4.4 on page 75).

**Summarize**

Basically, our protocol of consensus reaching (proposed in [VE01b]) may be described as follows:

1. an exchange of current positions to influence others' opinions;
2. a process that incites agents to make evolve their opinion (the coalition merging);
3. stop when a consensus is reached.

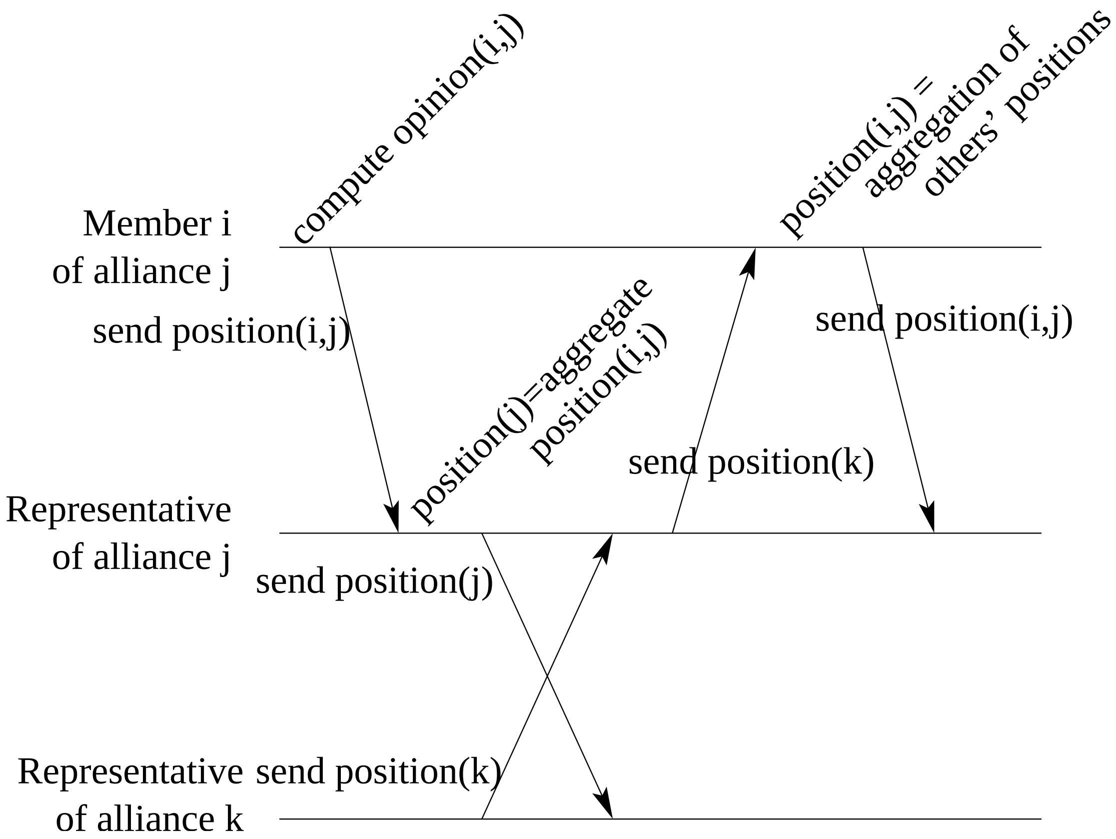


Figure 3.1: Protocol

To summarize, our protocol requires:

1. a formalism to represent and to handle agents' opinions;
2. a cycle detector to recognize a cycle;
3. a nearest opinions operator that chooses the two nearest opinions, with respect to some criteria;
4. an aggregation operator that computes the opinion of a coalition (or more generally the opinion of a group);
5. a consensus operator able to decide if a consensus is reached;
6. a chooser operator that chooses the most preferred choice once a consensus is reached.

## 3.3 The protocol in details

Conceptually speaking, in our protocol, four roles are distinguished (even if the same agent may play the two last roles):

- the organizer who sends data and manages inscriptions and turns;
- the supervisor who prevents from agents cheating, *i.e.* to send different preferences to each agent (information can not be used by an agent before the others thanks to a parallel diffusion[VE00]) by asking agents what preferences they have sent and received (penalty may be paid by culprits);
- the member who participates to the poll: he receives and sends his preferences when asked by the representative;
- the representative of a coalition who plays the role of interface between his coalition and the other coalitions, *i.e.* he receives opinions from his coalition's members, computes the aggregated opinion and sends it to the others.

To begin with, each agent creates a coalition with cardinality $1$. So, he supports two roles: the only member and the representative of the coalition.
Then, coalitions may increase (some members are added), and so some agents may loss their representative role.

### 3.3.1 Parallel diffusion

Agents may cheat by sending different informations to other agents. In [ZR94], the authors proposed the following solution. An agent $a$ wants to send an information $\theta_a$:

1. he generates a random private key $K_a$;
2. he encodes its information $\theta_a$ with this key $K_a$: $\theta_a^\star$;

Table 3.1: Parallel Diffusion of a data $\theta$ in $\mathcal{A}$

**for all** $a \in \mathcal{A}$ **do**
  $\theta^* \leftarrow Encrypt(\theta, key)$
  $\mathcal{B} \leftarrow \mathcal{A} \setminus \{a\}$
  $a$.**broadcast**($\theta^*$,$\mathcal{B}$)
  $a$.**receipt**($\theta^*$,$\mathcal{B}$)
  $a$.**broadcast**($Ack$,$\mathcal{B}$)
  $a$.**receipt**($Ack$,$\mathcal{B}$)
  $a$.**broadcast**($key$,$\mathcal{B}$)
**end for**

3. he broadcasts its encoded information $\theta^\star_a$;

4. he waits to receive all encoded informations $\theta^\star_b$ from the others;

5. he broadcasts the key $K_a$;

6. he receives all keys $K_b$ from other agents;

7. he decodes all information $\theta^\star_b$ with the keys $K_b$.

Steps $1$, $2$ and $3$ are sequential but may be made in parallel with $4$. Step $5$ and $6$ are sequential, and cannot begin before $4$ ends.
Using this diffusion, agent can cheat: in a system with three agents $a, b$ and $c$, $a$ sends its preferences to $b$ and $c$, $b$ to $a$ and $c$, but $c$ only to $a$; $a$ has received all the preferences, it sends its key to $b$ and $c$, which makes it possible $c$ to send to $b$ another preference.
The authors assume that agents did not cheat too much, and improve this mechanism to prevent any fraud sending and receiving information in the same process.
We propose that agents send their encoded preferences with a private key, then when each one has received all the preferences, they diffuse an acknowledgment of delivery. Each agent diffuses his key only when he has received all acknowledgments. This diffusion solves the problem, because if one carries out the same scenario here, the cheating agent will not be able to change its encoded message any more. It could of course send a wrong key, but the fraud would be visible.

### 3.3.2 Strategy of agents

**Member's strategy**

The agent's strategy depends on his preferences computation:

i **Independent Positions Computation** $IPC$: computation of the first positions without knowing those of the others;

ii **Dependent Positions Computation** $DPC$: computation of positions of next turns. The $DPC$ is computed using previous received position or any else information.

These strategies depend on the domain. Several examples will be given subsection 5.5.3 on page 104 in the case of alliance formation.

Table 3.2: Coalition Member ($CM$): Main

position $\omega = private\_position(CM)$
$CM$ sends his position $\omega$ and in the same time receives the positions of other agents
**while** a consensus is not reached **do** $\{\not\bowtie\}$
  $CM$ computes his new position $\omega$
  $CM$ sends his position $\omega$ to his representative
**end while**

Table 3.3: Coalition Member ($CM$): Main

**receive**("$CR$",$Organizer$) $\{CR$ = Coalition Representative$\}$
$\omega = IPC$ {first positions computation}
**send**("$\omega$",$CR$) {$CM$ sends his position $\omega$ to his representative}
**receive**("$\{\omega_a, a \in A\}$",$CR$) {he receives the positions of other agents}
**while** $\not\bowtie$ **do** {a consensus is not reached}
  $\omega = DPC$ {new positions computation}
  **send**("$\omega$",$CR$) {$CM$ sends his position $\omega$ to his representative}
  **receive**("$\{\omega_a, a \in A\}$",$CR$) {he receives the positions of other agents}
**end while**

#### Representative's strategy

**Definition 2 (Representative Strategy).** *Representative agent has particular procedures that define his strategy:*

- **Releasing Switch-over Proposal Criterion** $RSPC$*: criterion used to decide when to propose to release to switch-over mode. A* $RSPC$ *is an application* $H \to \{False, True\}$.
- **Releasing Switch-over Acceptance Criterion** $RSAC$*: criterion used to decide to accept or not to switch to release mode. A* $RSAC$ *is an application* $H \to \{False, True\}$.
- **Coalition Merging Proposal Criterion** $CMPC$*: gives a list of coalitions to which to propose to merge. A* $CMPC$ *is an application* $H \times A \to \{False, True\}$.
- **Coalition Merging Acceptance Criterion** $CMAC$*: allows to answer to coalition merging propositions. A* $CMAC$ *is an application* $H \times A \to \{False, True\}$.

### 3.3.3 Role of a coalition's member (CM)

#### Simplified view

The algorithm 3.2 gives a good idea of the expected behavior of a coalition's member.

#### Detailed view

The algorithm 3.2 is a more detailed version with formal operators.

**Example 2.** *The figure* **??** *on page ?? shows the initial state of a set of four agents (number $1$ to $4$). Each agent is the lone member of its alliance, and he wears two hats (two roles), representative (the hat $R$) and member (the hat $M$). Opinions are represented by geometric shapes.*
*On figure* **??** *on page ??, after the exchange of their opinion, agents evolve, but no consensus has been reached.*

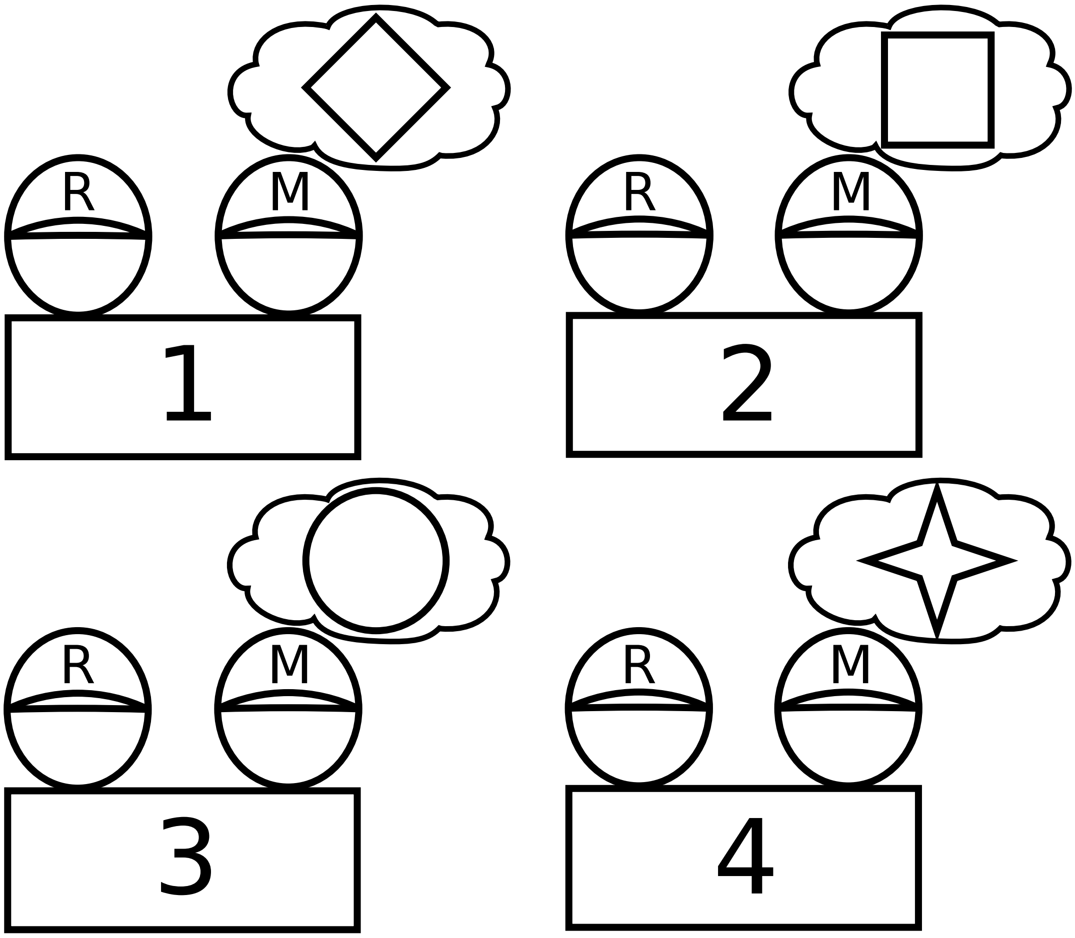


Figure 3.2: Protocol - step 0

*Agents $3$ and $4$ have near opinions; they decide to form an alliance (figure* **??** *on page ??). The agent $4$ is chosen to play the representative role. The opinion of the alliance is computed by merging the opinions of its members. The process continues. After several steps, a consensus is reached. The agent $2$ and the alliance $\{3, 4\}$ have the same opinion. The prefered solution – according to this opinion – is chosen.*

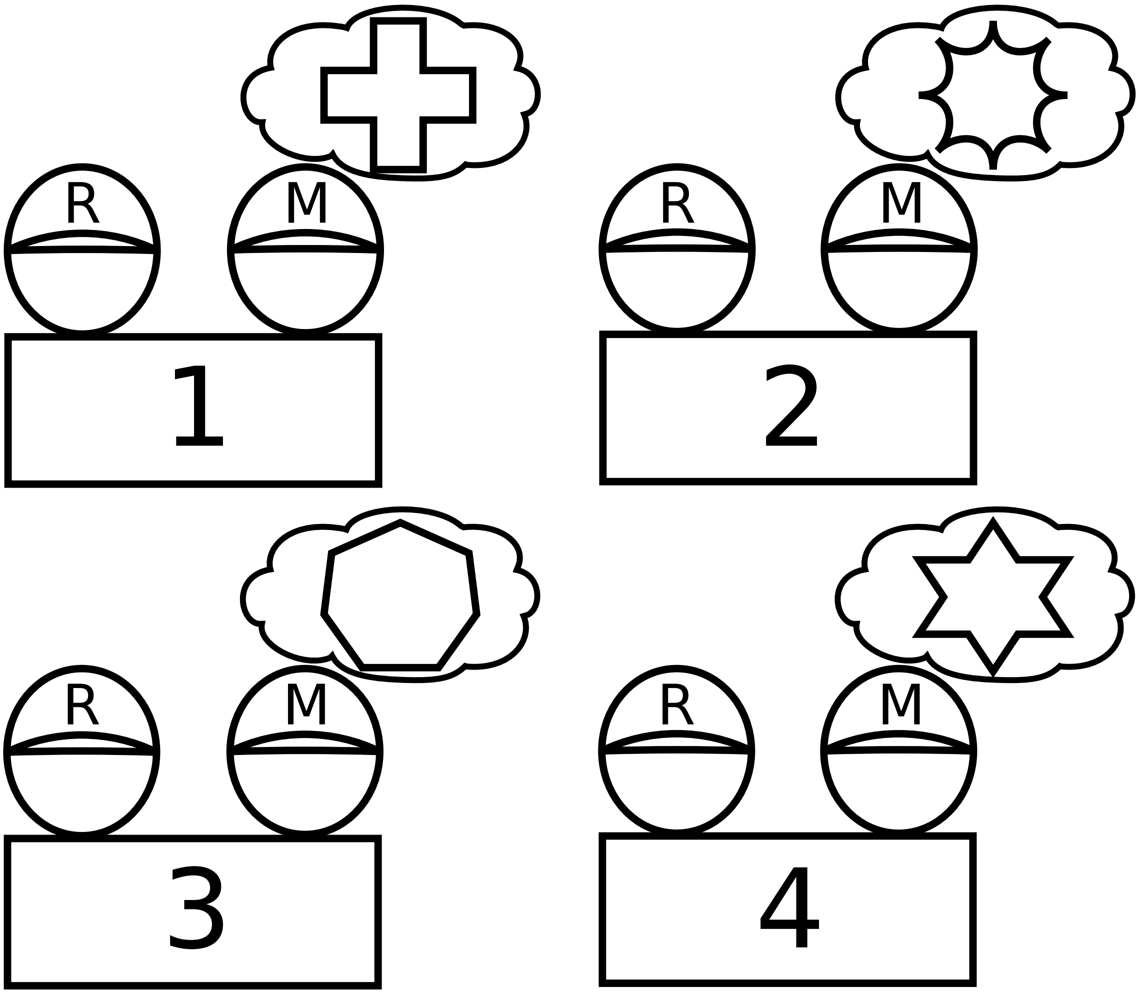


Figure 3.3: Protocol - step 1

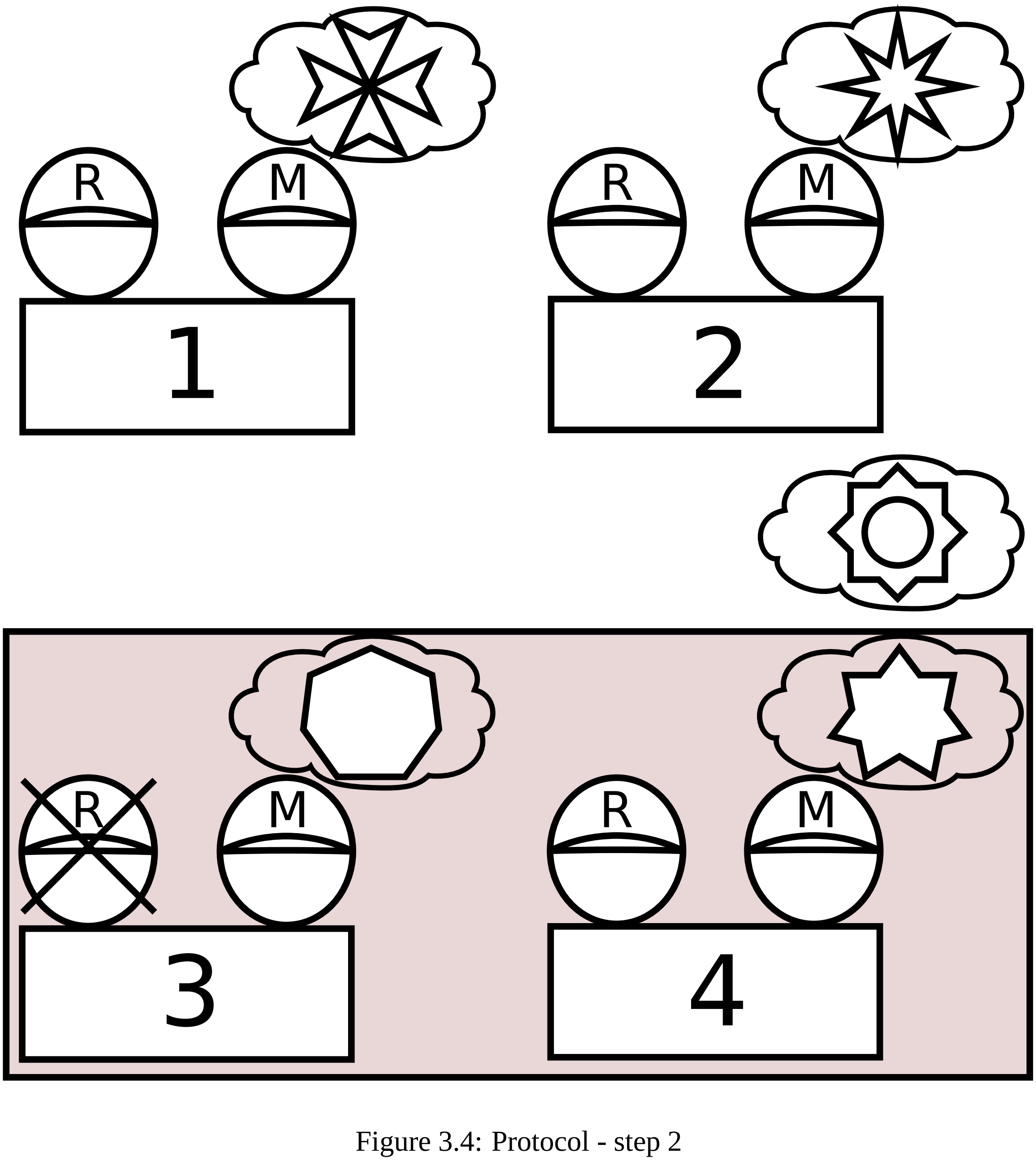


Figure 3.4: Protocol - step 2

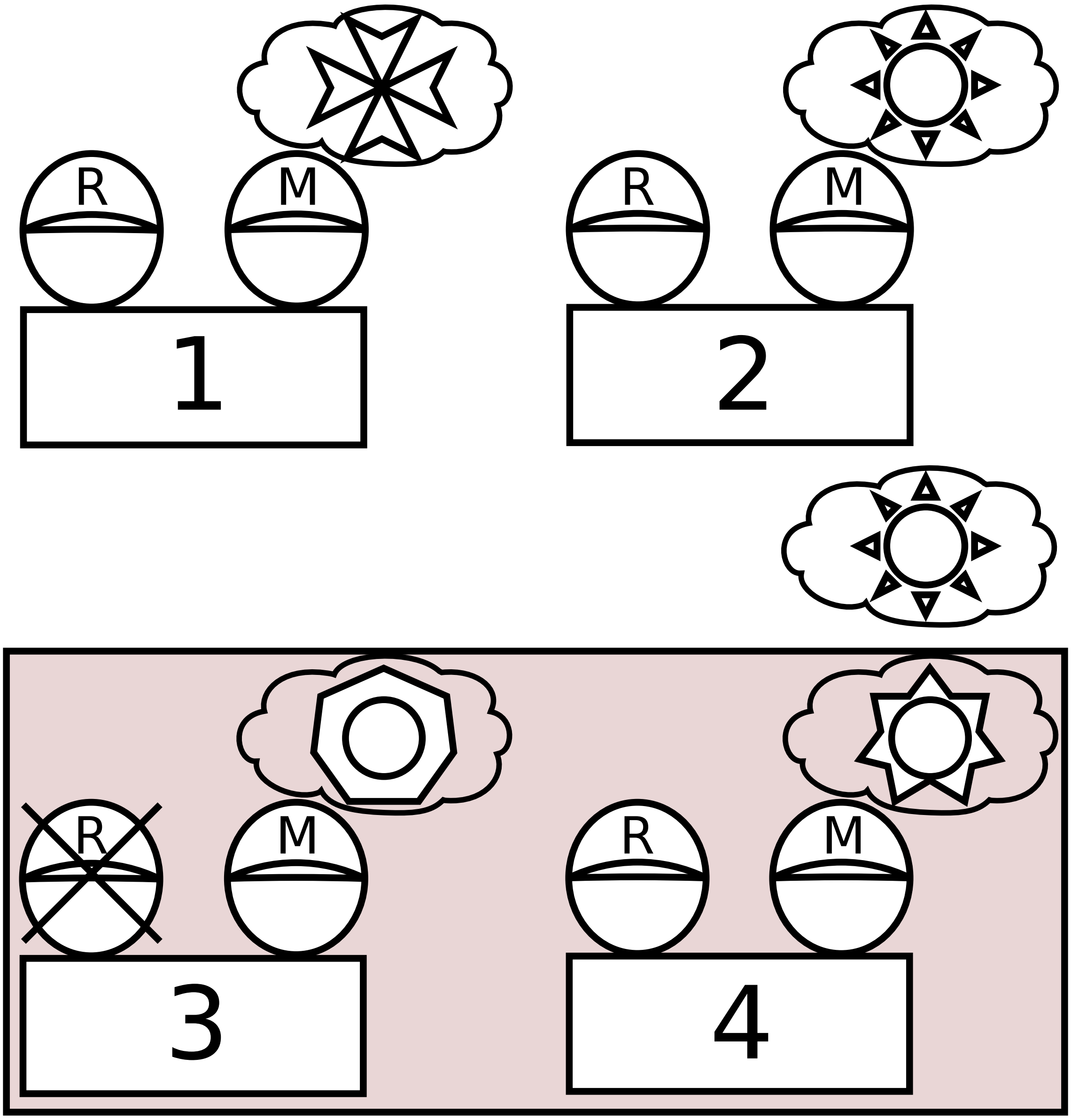


Figure 3.5: Protocol - step 3

Table 3.4: Coalition Representative ($CR$) : Main

{$CR$ receives the positions from the members of his coalition}
{$CR$ computes the coalition's position} {using aggregation operator $\Pi$}
**while** a consensus is not reached **do** {$\not\bowtie$}
  {all representatives $CR$ exchange their coalition's position using the parallel diffusion}
  **if** a cycle is detected **then** {$\Theta$}
    `call(switch-over mode)`
  **end if**
**end while**

Table 3.5: Coaltion Representative ($CR$): Switch-over mode

process of proposition/acceptance of coalition merging
**if** no coalition is formed **then**
  the two nearest coalitions merge {chosen using *nearest coalitions chooser*}
**end if**

### 3.3.4 Role of a coalition's representative (CR)

The representative's algorithm plays a leading role. Each representative has a list of interlocutor's $RepresentativeList \subset \mathcal{A}$ initialized by a set with one element, himself. The following algorithm is carried out by each representative in a distributed way.

#### Simplified view

#### Detailed view

The main algorithm 3.6 on the next page describes an exchange of position among agents (other representatives and coalition' members). Each turn, the representative also check if a switch-over is necessary (see algorithm 3.7 on the facing page). If it is, a coalition merging must occur (see algorithm 3.8 on page 58).

In switch-over mode, representatives decide which coalitions are going to merge (using $CMPC$ and $CMAC$); if no coalition desires to merge, the system chooses them.

## 3.4 Validation and complexity of the algorithm

### 3.4.1 Termination

Without any assumption on the criteria of switch-over mode releasing, we are not able to guarantee that the process terminates. However, if we assume that a criterion checks the existence of a loop, we can prove that it ends. In fact, if the same situation occurs twice, then a coalition is necessarily formed. If we assume that the number of situations is finite, then necessarily this case will happen. In the worst case (in number of turns), there

Table 3.6: Representative $a_i$: Main

**receive**(“$RepresentativeList$”,Organizer)
$MembersList = \{a_i\}$
**receive**(“$\{\omega_j, a_j \in MemberList\}$”,$MembersList$) {receives the positions from the members of his coalition}
$\omega_c = \amalg(\{\omega_j, a_j \in MemberList)\}$ {computes the position of the coalition using the aggregation operator}
$\{\omega_k, k \in A\} \hookleftarrow ParallelDiff(\omega_c, RepresentativeList)$
**send**(“$\{\omega_k, k \in A\}$”,$MemberList$)
**while** $\not\triangleright(\{\omega_k, k \in A\})$ **do** {no consensus is reached}
    `call(switch-over)`
    **receive**(“$DepPref$”,$MemberList$){he receives members’ positions}
    $h \hookleftarrow ParallelDiff(DepPref, RepresentativeList)$ {broadcasts the position of the coalition and receives positions from other coalitions}
    **send**(“$DepPref$”,$MemberList$) {sends the positions of other coalitions to his coalition}
**end while**

Table 3.7: Representative $a_i$: switch-over

**if** $\Theta = True$ **then** {a cycle is detected}
    **send**(“proposition to switch-over”,$RepresentativeList$)
    **if receive**(“acceptance to switch-over”,$RepresentativeList$) **then** {receive acceptance from enough representatives}
        `call(coalition merging)` {see algorithm 3.8 on the following page}
    **end if**
**end if**
**if receive**(“proposition to switch-over mode”,$RepresentativeList$) **then** {a proposition is received}
    **if** $RSAC = True$ **then** {agreement to switch-over}
        **send**(“acceptance to switch-over”,$RepresentativeList$)
    **end if**
    **if receive**(“acceptance to switch-over”,$RepresentativeList$) **then** {receive acceptance from enough representatives}
        `call(coalition-merging)` {see algorithm 3.8 on the next page}
    **end if**
**end if**
{end of the decision to switch-over}

Table 3.8: Representative: coalition-merging

**send**("proposition to merge coalitions",$\{a \in A/CMPC(h,a) = True\}$) {sends a proposition of merging}
$merging \leftarrow False$
**if receive**("acceptance to merge coalitions",$rep_j$) **then** {receives an acceptance to merge}
  `mergeWith(`$rep_j$`)`
  **send**("a coalition-merging occurs",$RepresentativeList$)
  $merging \leftarrow True$
**end if**
**if receive**("proposition to merge coalitions",$rep_j$) **then** {receives a proposition to merge}
  **if** $CMAC(rep_j)$ **then** {}
sends an acceptance to merge **send**("acceptance to merge coalitions",$rep_j$)
    `mergeWith(`$rep_j$`)`
    **send**("a coalition-merging occurs",$RepresentativeList$)
    $merging \leftarrow True$
  **end if**
**end if**
**if receive**("a coalition-merging occurs",$rep_j$) **then** {}
  $merging \leftarrow True$
**end if**
**if** ~~m~~erging **then**
  if I am concerned, I merge with the closer coalition
**end if**

will only be formations of forced coalitions, what will lead to a great coalition. In fact, the number of situations is not finite because preferences use real numbers. To escape this problem, we consider that two sights are equal if all their preferences are rather close w.r.t. the given distances as introduced.

**Definition 3 (Pseudo-equality between preferences).** *Let $\varepsilon$ be a small real number.*
*Let $\delta$ and $\delta'$ be two preferences.*
*We shall say that $\delta$ and $\delta'$ are pseudo-equal ($\delta \simeq \delta'$) if:*
$\forall \sigma \in \Sigma$, $|delta(\sigma) - delta'(\sigma)| < \varepsilon$.

**Definition 4 (Pseudo-equality between sights).** *Let $\varepsilon$ be a small real.*
*Let $v_t$ and $v_{t'}$ be two sights.*
*We shall say that $v_t$ and $v_{t'}$ are pseudo-equal ($v_t \simeq v_{t'}$) if:*
$\forall a \in \mathcal{A}$, $|v_t(a) \simeq v_{t'}(a)| < \varepsilon$.

**Definition 5 (A pseudo-cycle in a history).** *We say that a history $h = (v_t)_{1 \le t \le T}$ contains a pseudo-cycle if:*
$\exists(\tau_1, \tau_2) \in \{1, \ldots, T\}^2$, $\tau_1 \neq \tau_2$ *such that* $v_{\tau_1} \simeq v_{\tau_2}$.

**Definition 6 (Consensus Reaching Problem (CRP)).** *A $CRP$ is defined as a tuple $\langle A, \mathcal{S} \rangle$, where:*

- *$A$: the set of agents that have to chose a solution;*
- *$\mathcal{S}$: the set of choices;*

In a $CRP$ $\langle A, \mathcal{S} \rangle$, cycles-like are detected if:

**Definition 7 (A $CRP$ detects pseudo-cycle).** *A $CRP$ $\langle A, \mathcal{S} \rangle$ detects pseudo-cycle if: $h$ contains a pseudo-cycle $\Rightarrow$ ($\exists a_0 \in A$ such that $a_0.RSPC(h) = True$) $\wedge$ ($\forall a \in A$, $a.RSAC(h) = True$) ].*

In other words, a $CRP$ detects cycles-like if at least one agent detects it and all then accept to change mode.

**Lemma 1.** *If a $CRP$ detects cycles-like and there is a pseudo-cycle, then a coalition will be formed.*

*Proof.* If a $CRP$ detects cycles-like and there is a pseudo-cycle, then at least one agent will propose to change mode and all other will accept. Then, necessarily, two agents or coalitions will form a coalition: either they choose to do it, or they have been compelled to do so. ∎

**Theorem 1.** *If a $CRP$ detects pseudo-cycle, then the program terminates.*

*Proof.* Let $\varepsilon$ be the threshold of pseudo-equality among sights (two sights are pseudo-equal if all their real values are equal to within $\varepsilon$).
Let $n$ be the number of agents.
Let $k$ be the number of solutions.
A degree of preference between two solutions may have $\frac{2}{\varepsilon}$ different values; so, an opinion (an anti-symmetrical matrix) may have $(\frac{2}{\varepsilon})^{k(k-1)/2}$ different values. The number of sights is thus $n(\frac{2}{\varepsilon})^{k(k-1)/2}$.
During an enough large number of turns, the history contains necessarily two identical sights. In this case, a pseudo-cycle exists and then a coalition will be formed (see Lemma lemma 1). The history of coalition formation may be represented by trees: nodes are coalitions and edges link old isolated coalitions to the new formed coalition. If there is only one tree, that means that a great coalition has been formed (*i.e.* a coalition that contains all agents). Since the number of internal nodes is $n-1$, the coalition formation can occur at worst $n-1$ times. Finally, after $n(n-1)(\frac{2}{\varepsilon})^{k(k-1)/2}$ turns at worst, there is consensus. ∎

### 3.4.2 Complexity

We assume that the computation time is small regards to the communication time.

To estimate the maximal number of exchanged messages, we assume that the worst situation always happens.
Each turn, $n$ agents as member send to $n$ agents as representative:

- a message that contains its position;

Each turn, $n$ agents as representative send to $n$ agents as member:

- a message that contains the positions of other coalitions;

Each turn, $n$ agents as representative send to $n-1$ agents as representative:

- a message that contains the position of its coalition;
- a message that proposes or not to switch to coalition formation mode;
- optionally, a message that accepts or not to switch to coalition formation mode;
- optionally, a message that proposes to form a coalition;
- optionally, a message that transmits acceptance to form a coalition;
- optionally, a message that transmits a vote for the two closest coalitions;

As seen in proof 3.4.1 on the page before, the number of turns is bounded by $n(n-1)(\frac{2}{\varepsilon})^{k(k-1)/2}$. The maximum number of exchanged messages by turn is equivalent to $\alpha n^2$. Thus, the total maximum number of exchanged messages is equivalent to $\alpha n^4(\frac{2}{\varepsilon})^{k(k-1)/2}$.

The maximum complexity is very high, but we assumed that the number of agents is small and that we have enough time to reach a consensus. Freedom against speed: as the process is less controlled, a desired state is more long to be reached.

## 3.5 Conclusion

Agents that interact with each others may need to make a common decision. In our context, it means that a consensus must be reached.
Based on the rules defined in the previous chapter, we propose a protocol that allows agents to choose a solution (among a set of solutions). It involves a parallel exchange of positions between agents in order to modify their opinions. If they don't, the formation of coalitions (which may be imposed if necessary) avoids to break the deadlock. Supervisors can check that agents obey the rules because the protocol is based on observable data.
The decision may take a long time to be reached, but it is not the most desired property. Moreover, chapter 7 on page 119 shows that, in practice, it is not so long.
Though our protocol is not logic based, it respects the desiderata for agent argumentation protocol proposed in [MPW02]:

1. Stated Dialogue Purpose: the set of choices is known in advance;
2. Diversity of Individual Purposes: agents' purposes are part of the set of choices;
3. Inclusiveness: their is no reason to preclude agents;
4. Transparency: rules and structure of the dialogue are clear and entirely known by agents;
5. Fairness: agents may play several roles, but even if representative agents have more rights and responsibilities, they do not have more power (all agents are equal);
6. Clarity of Argument Theory: agents that will use another dialectic system (it is allows by the protocol, but no help is provided to do so);

7. Separation of Syntax and Semantic: first, the separation concerns only the choice set (the other concepts are domain-independent), and it is applied; secondly, the conformity of protocol syntax is included in the protocol and agents cannot simulate insincerely any internal state, because no internal state is required;

8. Rule-Consistency: deadlocks and infinite cycles are detected by the cycle-detection;

9. Encouragement of Resolution: if an agent refuse to try to resolve the problem (*i.e.* to find a consensus), he will be ejected and then loss money;

10. Discouragement of Disruption: as noted in [Kra01], agents are allowed to have disruptive behaviors (*e.g.* uttering the same position repeatedly), but experiments (see chapter 7 on page 119) show that such behaviors leads to lesser incomes;

11. Enablement of Self-Transformation: with totally autonomous agents, it is obvious that they are authorized to change their opinion and position;

12. System Simplicity: our goal is to propose a protocol as simple as possible, consistent with other criteria;

13. Computational Simplicity: heterogeneity of agents oblige to assume that agents may have low computational capabilities.

This protocol requires a formalism of opinions. The next chapter 4 on page 65 produces our formalism that has a certain advantage over some existing formalisms.

# Bibliography


[Akn00] Samir Aknine. *Multi-agent systems coordinations models and method*. PhD thesis, Universite Paris IX-Dauphine, 2000.

[Arr63] Kenneth Arrow. *Social Choice and Individual Values*. Wiley, New York, 2nd edition, 1963.

[Arr91] K. Arrow. *The Origins of the Impossibility Theorem*, chapter 1, pages 1–4. Elsevier Science Publishers B. V., Amsterdam, 1991.

[caCP96] Jorge Anacleto Louçã, Helder Coelho, and Suzanne Pinson. Forming coalition in task oriented multi-agent systems. In *Proceedings Of IBERAMIA 1996*. IBERAMIA, 1996.

[CAP02] Philippe Caillou, Samir Aknine, and Suzanne Pinson. A multi-agent method for forming and dynamic restructuring of pareto optimal coalitions. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1074–1081. ACM Press, 2002.

[DDKV01] Frank Dignum, Barbara Dunin-Kęplicz, and Rineke Verbrugge. Agent theory for team formation by dialogue. *Lecture Notes in Computer Science*, 1986:150–??, 2001.

[DJJT01] M. Dastani, N. Jacobs, C.M. Jonker, and J. Treur. Modeling user preferences and mediating agents in electronic commerce. In C. Sierra F. Dignum, editor, *Agent-Mediated Electronic Commerce*, volume 1991 of *LNAI*, pages 164–196. Springer verlag edition, 2001.

[Dun95] P.M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programmation and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[Kra01] Erick C.W Krabbe. The problem of retraction in critical discussion. *Synthese: An International Journal for Epistemology, Methodology and Philosophy of Science*, 127:141–159, 2001.

[LS89] F. Lin and Y. Shoham. Argument systems - a uniform basis for non monotonic reasoning. In *Proceedings of the first International Conference on Principles of knowledge representation and reasoning, KR'1989*, pages 245–255, 1989.

[MPW02] P. McBurney, S. Parsons, and M. Wooldridge. Desiderata for agent argumentation protocols. In C. Castelfranchi and W. L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2002.

[Nas50] J. Nash. Equilibrium points in n-person games. In *Proc. National Academy of Sciences*, volume 36, pages 48–49, 1950.

[PL92] G. Pinkas and R.P. Loui. Reasoning from inconsistency. In *Proceedings of the 3rd international conference on principles of knowledge representation and reasoning, KR'1992*, pages 709–719, 1992.

[Pol92] J.L. Pollock. How to reason defeasibly. *Artificail intelligence*, 57:1–42, 1992.

[RcB01] Juliette Rouchier and François Bousquet. The possibility to use gift as a paradigm for exchanges among agents. In *Proceedings of the 10th European Workshop on Multi-Agent Systems*, Annecy, France, 2001.

[RZ94] J. R. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.

[Sha53] L. S. Shapley. *Contributions to the Theory of Games*, chapter A Value for N-Person Game. Princeton University Press, 1953.

[SK95] O. Shehory and S. Kraus. Task allocation via coalition formation among autonomous agents. In *Proc. of IJCAI-95*, pages 655–661, Montreal, August 1995.

[VE00] G. Vauvert and A. El Fallah - Seghrouchni. Coalition formation among egoistic agents. In *Proc. of MAMA'2000*, Wollongong, Australia, December 11-13 2000. International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce. Awarded "One of the three best papers" (among 92 accepted papers).

[VE01a] G. Vauvert and A. El Fallah - Seghrouchni. Coalition formation among strongly autonomous and weakly rational agents. In *Proc. of MAA-MAW'2001*, Annecy, France, May, 2-4 2001. Modelling Autonomous Agents in a Multi-Agent World. 10th European Workshop on Multi-Agent Systems. Acceptation rate of papers : 13/73=18%.

[VE01b] Guillaume Vauvert and Amal El Fallah - Seghrouchni. A distributed algorithm for coalition formation among strongly autonomous and weakly rational agents. In *Proc. of IAT'2001*, Maebashi City, Japan, October 23-26 2001.

[VJ98] N. Vulkan and N. R. Jennings. Efficient mechanisms for the supply of services in multi-agent environments. In *Proc. of the First International Conference on Information and Computation Economics*, South Carolina, 1998.

[ZR94] G. Zlotkin and J. S. Rosenschein. Coalition, cryptography, and stability: Mechanisms for coalition formation in task oriented domains. In *Proc. of AAAI94*, pages 432–437, Washington, 1994.

# Chapter 4

# Opinion

## Contents

## Figures



## Tables

## 4.1 Introduction

In chapter 3 on page 41, a protocol to reach a consensus based on an iterative exchange of opinions is proposed. It requires formalisms and operators [VE02].
Usually, the word *preference* refers to the comparison among all choices, while we think that it must refer to the comparison among two choices, without pay attention to other comparisons. So, in this thesis, the word *preference* refers to a level of preference between two choices, while the term *opinions* points out the set of preferences. We hope that this unusual use of words keeps the thesis clear.

### 4.1.1 Opinion representation formalism

First, we need formalisms that represents:

1. Agents' preferences: each agent prefers certain choices and dislikes others. The preference represents the intimate advice of agents, what they really prefer, independently from others' influences.

2. Agents' positions: during a negotiation, agents don't change their proper preferences, but they may change their position, in order to enable the group (or the system) to reach a consensus.

3. Agents' arguments: the protocol allows agents to influence each others by sending a particular kind of informations. The chosen argument is the agent's current position.

Three different formalisms could be consumed to represent these three concepts; but the follow of a single representation is possible. Moreover, it is interesting because:

1. The manipulation of one formalism requires less computational resources than of three because operators are stored only one time.

2. Operators that manipulate two concepts are easier to develop.

3. Since less resources are used, smaller agents may be used and so a larger variety of agents are able to use the protocol (agents' heterogeneity).

### 4.1.2 Required operators

Several operators are also required by the protocol:

1. *chooser operator*: given an opinion, this operator computes the preferred choice;

2. *aggregation operator*: given a set of opinions (the opinions of group's members), this operator computes the aggregated opinion (the opinion of the group);

3. *consensus detector*: given a set of opinions, this operator return $True$ if a consensus is reached;

4. *nearest opinion chooser operator*: given a set of opinions, this operator computes the couple of nearest opinions, *i.e.* the two opinions that are the most similar;

5. *opinion cycle detector*: given a history (the sequence of agents' positions during the decision process), return $True$ if a cycle occurs, *i.e.* if a situation happens twice.

## 4.2 Approaches

In [Arr91], Kenneth Arrow tackles the problem of finding a function of aggregation of preferences that respects the intuitive idea of such aggregation. He proposes several axioms that must be verified. Then, he formalizes the concept of preference as a binary relation $<$ that is antisymmetric, transitive and not reflexive. Then, he proves a theorem:

**Theorem 2 (Arrow's Impossibility Theorem [Arr63]).** *No universally applicable non-dictatorial method of aggregating individual preferences into group preferences can satisfy both the Pareto Preference condition (Unanimous individual preferences are group preferences) and the condition of Independence of Irrelevant Alternatives (Group preference between two prospects depends only on individual preferences between those same prospects).*

So that the only solution is a totalitarian system where only one agent may decide: this is what is called the Impossibility Theorem.
This result seems to keep to aggregate preferences. That leads many works to relax some axioms in order to solve the problem. In fact, the impossibility results from the poorness of the model: it is not rich enough to be able to represent some aspects of preferences. As shown in [Sen70], many representations of preferences have been proposed in order to solve the impossibility problem of K. Arrow:

1. structural approach: as a preference ordering of the choices, ordered from the worst to the best, the result is a total order;

2. numerical approach: as a utility function, the result is a total order, but with a measure of the difference of preference between two choices (what is richer than preferred/not preferred);

3. hybrid approach: as a preferred relation with some degree of preference of any alternative over another, where the degree is interpreted as a degree of credibility.

### 4.2.1 Structural approaches

The order relation $\succeq_{\mathcal{P}}$ defined by: $\forall (c_1, c_2) \in \mathcal{S}^2$, $c_1 \succeq_{\mathcal{P}} c_2$ is interpreted as "$c_1$ is preferred to $c_2$". An order relation is reflexive, anti-symmetrical and transitive.
This model is too poor, because:

1. the indifference (no preference) is not modeled: one must always choose between two choices;

2. there is no level of preference (no intermediate degree): an agent prefers one over another, but not more or less;

3. a rational preference relation may be non-transitive (see section 7.2.3 on page 120).

The two first points are addressed by the numerical approach (see below).
Let us discuss the third point more deeply.

#### Transitivity

Most of the approaches consider that transitivity is a key point to represent preferences. The main reason is that for a non transitive relation, an agent may prefer $c_1$ over $c_2$, and $c_2$ over $c_3$, and $c_3$ over $c_1$, which seems to be irrational.
Let see with an example. An agent wants to buy a car: he has to make a choice among three cars. Two criteria allow him to decide:

- the price, that he will try to minimize;

- the luxury level, that he will try to maximize.

The three cars have the characterized by:

1. $c_1 = (price = \$10K, luxury = 4)$;
2. $c_2 = (price = \$12K, luxury = 6)$;
3. $c_3 = (price = \$14K, luxury = 8)$.

Our agent must have a function $choose : Car \times Car \leftrightarrow Car$ that allows him to choose a car among two proposed cars $ci$ and $cj$:

1. if $|ci.price{-}cj.price| < \$3K$, then $f(ci, cj) = c_k \in ci, cj/c_k.luxury = max(ci.luxury, cj.luxury)$;
2. else $f(ci, cj) = c_k \in ci, cj/c_k.price = min(ci.price, cj.price)$.

If the two prices are approximately the same, he will choose the most luxury car; else, he will choose the cheapest one.
We obtain that:

- comparing $c_1$ and $c_2$: $c_2 \preceq c_1$ using the rule $1$;
- comparing $c_2$ and $c_3$: $c_3 \preceq c_2$ using the rule $1$;
- comparing $c_1$ and $c_1$: $c_3 \preceq c_1$ using the rule $2$.

So the relation $\preceq$ is not transitive. The result is surprising, but it is caused by a mistake. The example is correct, the rules are rational. So what is wrong ?
These rules provide a relative comparison between two choices. It's what a preference means: I prefer this choice over this another one, I don't think about consequences at global level (like the not-transitivity). They don't provide an absolute judgment. When somebody may choose among a large amount of choices, he cannot guarantee that there is no cycle in his graph of preference, especially because of his bound rationality. I think that this is the main default of the usual transitive model of preferences: in fact, it is the model of an absolute judgment, which is not always possible for a bounded-rational agent; it is not the model of a preference, which is a set of binary choices. The transitivity is not an inherent property of preferences in a rational context.
Given the above opinion, a problem arises. Let assume that an agent owns the car $c_1$. I propose him the car $c_2$; as he prefers this car over $c_1$, he will accept to give me some money to exchange $c_1$ and $c_2$. Now, I propose him the car $c_3$; as he prefers this car over $c_2$, he will accept to give me money to exchange $c_2$ and $c_3$. Now, I propose him the car $c_1$; as he prefers this car over $c_3$, he will accept to give me money to exchange $c_3$ and $c_1$. So we can get an infinite amount of money from the agent: he is called a "money pump".
What to answer to this paradox ? First, the agent may buy the three cars with all the money he spend to exchange his car. Secondly, the agent doesn't have an infinite amount of money; so, he will stop one day. But the true answer is that an agent must not be blindly-opinion-driven. His opinion allows him to represent his preferences, but he may reason at two levels: he may use his opinion to make a choice, but he may also observe himself when he is acting. Reasoning at opinion level doesn't prevent him to meta-reason. He will then realize that his behavior is not rational, even if his opinion is rational. The not-rational behavior is not a consequence of the opinion's representation, but of the use of the knowledge.

### 4.2.2 Numerical approaches

A utility function (also called a valuation function) $u : \mathcal{S} \mapsto \mathbb{R}$; $\forall(c_1, c_2) \in \mathcal{S}^2$, $u(c_1) > u(c_2)$ is interpreted as "$c_1$ is preferred to $c_2$".
This formalism represents more sharply the degree of preference. With structural approaches, if $c_1 \succeq_{\mathcal{P}} c_2$ and $c_3 \succeq_{\mathcal{P}} c_4$, the degrees of preference between $c_1$ and $c_2$ and between $c_3$ and $c_4$ are the same. With a valuation function, $u(c_1) - u(c_2) < u(c_3) - u(c_4)$ means that $c_1$ is less preferred over $c_2$ than $c_3$ is preferred over $c_4$.

1. $u(c_1) = u(c_2)$ is interpreted by "the agent has no preference between $a$ and $b$";

2. $u(c_1) - u(c_2)$ expressed the intensity of the preference.

Since the preference is a comparison between two choices, the absolute value of the utility is not significant. What is significant is the difference of utility in comparison with other differences. The main default of utility functions is that the differences of utilities (that contain really the concept of preference) are too much constrained: $\forall(c_1, c_2, c_3) \in \mathcal{S}^3$, $u(c_1) - u(c_2) = (u(c_1) - u(c_3)) + (u(c_3) - u(c_2))$.
Let remember the example above: three cars $c_1$, $c_2$ and $c_3$, with characteristics $c_1 = (price = \$10K, lux. = 4)$, $c_2 = (price = \$12K, lux. = 6)$ and $c_3 = (price = \$14K, lux. = 8)$. To decide, he chooses the following rules (trying to maximize the utility):

1. if prices are close (less than $\$3K$), then the difference of utility is the difference of luxuries divided by the maximum;

2. else the difference of utility is the difference of prices divided by the maximum.

Finally, we obtain that: $u(c_2) - u(c_1) = 2/8 = 0.25$ and $u(c_3) - u(c_2) = 2/8 = 0.25$ using the rule 1, and $u(c_3) - u(c_1) = 4000/14000 = 0.2857\ldots$ using the rule 2. We should obtain: $u(c_1) - u(c_2) = (u(c_1) - u(c_3)) + (u(c_3) - u(c_2))$. But $u(c_1) - u(c_2) = -0.25$, $(u(c_1) - u(c_3)) + (u(c_3) - u(c_2)) = -0.2857 + 0.25 = -0.0357$. So the preference cannot be represented sharply as a utility function.

### 4.2.3 Hybrid approaches

In [DJJT01], the model of users' preferences is based on several kinds of transitivities, because the classic transitivity doesn't respect the intuitive concept of preference. The model is based on binary relation $\mathcal{R}$ that represent strict preferences, to which we add two symbols representing indifference and incomparability. Interval-valued preference structures use some degrees between $0$ and $1$. In this model, a binary relation $\mathcal{R}$ is mixed with a valuation: $c_1 \mathcal{R} c_2$ means that $c_1$ is preferred to $c_2$, but this relation $c_1 \mathcal{R} c_2$ has a degree of credibility. So, relations are comparable.
Several kind of transitivities are defined:

- The *min-transitivity* [Bil98] is defined by: $\forall(c_1, c_2, c_3) \in \mathcal{S}$, $min(c_1 \mathcal{R} c_2, c_2 \mathcal{R} c_3) \leq c_1 \mathcal{R} c_3$.

- The *weak-transitivity* [Bil98] is defined by: $\forall(c_1, c_2, c_3) \in \mathcal{S}$, if $c_1\mathcal{R}c_2 > c_2\mathcal{R}c_1$ and $c_2\mathcal{R}c_3 > c_3\mathcal{R}c_2$ then $c_1\mathcal{R}c_3 > c_3\mathcal{R}c_1$. Since $c_1\mathcal{R}c_2$ and $c_2\mathcal{R}c_3$ are more credible than $c_2\mathcal{R}c_1$ and $c_1\mathcal{R}c_2$ respectively, $c_2\mathcal{R}c_3$ must be more credible than $c_3\mathcal{R}c_2$.

- The **Stochastic transitivity** [Mon88]

  **strongly stochastic** transitivity

  **moderately stochastic** transitivity

  **weakly stochastic** transitivity

- The $T$-transitivity [FR94], with $T$ a t-norm. The three main continuous t-norms are:

  **minimum operator** $M$ []

  **algebraic product** $P$

  **Lukasiewicz t-norm** $W$

- The **FG-transitivity** [Swi01] generalizes stochastic transitivity

- The **U-transitivity** [MB]

This approach is based on several levels of relation credibility. But the origin of the credibility is unknown. We are going to precise why there is more or less credibility in our preferences and thus how to aggregate them.

### 4.2.4 Our approach

Our goal is to allow to represent finely an opinion while not being constrained by the degree of preference too much (like with the numerical approach).
Informally, the preference for a choice $c_1$ over a choice $c_2$ is the expression of the intensity of its author's will to have $c_1$ chosen instead of $c_2$. To represent preferences, we propose to use *degrees* that range from $-1$ to $1$: the closer to $1$ a degree is, the more the first choice is preferred to the second (and reciprocally).

**Example 3.** *Let $\mathcal{S} = \{black, white\}$ be the set of choices.*
*If I strongly prefer $black$ over $white$, then $\delta_{black,white} = 1$.*
*If I have a little preference of $white$ over $black$, then $\delta_{black,white} = -0,5$.*
*If I have no preference of $white$ over $black$, then $\delta_{black,white} = 0$.*

Then, an opinion is the set of preferences comparing every choices to every other ones. Our formalism may be viewed as a generalization of several others. If :

i we limit values of degrees to $\{-1, 0, 1\}$,

ii $c_1\mathcal{R}c_2 \iff \delta_{c_1,c_2} = 1$

iii $\delta_{c_1,c_2} = 0 \iff c_1 = c_2$

iv $c_1 \mathcal{R} c_2 \wedge c_2 \mathcal{R} c_3 \Rightarrow c_1 \mathcal{R} c_3$

v we don't take the level of conflict into account,

then our formalism is equivalent to a total order.
Incomparability when added leads to a partial order; in our approach, the semantics of the incomparability is a high level of conflict. To represent a valuation, we have to impose the constraint: $\delta_{i,j} \geq 0 \wedge \delta_{j,k} \geq 0 \Rightarrow \delta_{i,k} = \delta_{i,j} + \delta_{j,k}$.
Several approaches allow agents to use infinite value for degrees of preference, what allow agents to represent the refute or to impose a choice, as a veto. In our approach, a veto reduces the set of choices, before the consensus process begin.
What happens at the group level ? The first idea is to compute the mean of the degrees

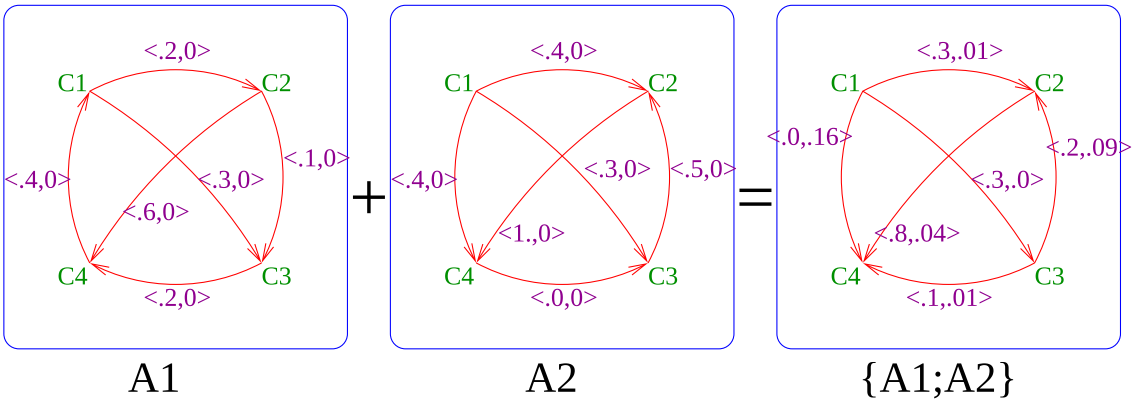


Figure 4.1: Opinion

$(\omega(c, c') = (\omega(c, c') + \omega(c, c'))/2)$, but this formula leads to strange results: a preference of two agents with opposite degrees of preferences is close to $0$, while incompatible preference could be find. The reason is that to compute the mean is too simple for a group: we need to keep more information. We want to keep the dispersion of preferences to save and to manage the level of conflict. Naturally, we think of standard deviation that summarizes the dispersion of values in a statistic set.

## 4.3 Our formalism of an opinion

### 4.3.1 Notation

Notation:

- let $A$ be the set of agents, and let lower-case letters $a, b, \ldots$ denote agents;
- let $\mathcal{S} = \{c_1, \ldots, c_k\}$ be the set of choices among which agents have to choose;
- let $\Delta$ be the set of degrees of preference (degree of preference will be defined later);
- let $\varsigma$ be the set of level of conflicts (conflict level will be defined later);

- let $\Omega = \{\omega_1, \ldots, \omega_k\}$ be the set of opinions;
- let $V$ be the set of views; a view is the set of all agents' opinions;
- let $H$ be the sequence of histories from turn $t = 1$ to $t = T$.

We call:

- "preference" the preference for a choice over another one: it is a comparison between two choices;
- "opinion" the set of all preferences over all others: it is the set of all comparisons between two pairs of choices.

### 4.3.2 A preference

A preference between two choices $ci$ and $cj$ is defined by a degree of preference $\delta_{ci,cj}$ and a level of conflict $\sigma_{ci,cj}$ (standard deviation).

**Definition 8 (Opinion sets).** *The set of choices is $\mathcal{S}$, the set of degrees is $\Delta = [-1, 1]$ and the set of levels of conflict is $\varsigma = [0, 1]$.*

**Property 1 (Degree set).** *The set of degrees is:*

1. *stable when computing the opposite;*
2. *continuous;*
3. *contains a unique element $0$ that represents the indifference.*

Interpretation:
A degree $\delta_{ci,cj}$ between $ci$ and $cj$ is interpreted as follows:

- $0 < \delta_{ci,cj} \leq 1 \iff$ "I prefer $ci$ to $cj$ with a degree $\delta_{ci,cj}$";
- $-1 \leq \delta_{ci,cj} < 0 \iff$ "I prefer $cj$ to $ci$ with a degree $-\delta_{ci,cj}$";
- $\delta_{ci,cj} = 0 \iff$ "I have no preference between $ci$ and $cj$".

A level of conflict $\sigma_{ci,cj}$ between $ci$ and $cj$ is interpreted as follows:

- $\sigma_{ci,cj} = 0 \iff$ "everybody agrees the degree of preference" (low level of conflict);
- $\sigma_{ci,cj} = 1 \iff$ "the maximum level of conflict is reached";
- $\sigma_{ci,cj} < \sigma'_{ci,cj} \iff$ "preference with level of conflict $\sigma_{ci,cj}$ is less conflicting than preference with level $\sigma'_{ci,cj}$".

### 4.3.3 An opinion

**Definition 9 (Opinion).** *An* **opinion** $\omega \in \Omega$ *is an application* $\omega : \mathcal{S} \times \mathcal{S} \to \Delta \times \varsigma$ *with the following properties:*

1. $\forall ci \in \mathcal{S}$, $\omega^a_{ci,ci} = \langle 0, 0\rangle$*:* $a$ *is indifferent to* $ci$ *and* $ci$*;*
2. $\forall (ci, cj) \in \mathcal{S}^2$, $\omega^a_{ci,cj} = \langle \delta, \sigma\rangle \Rightarrow \omega^a_{cj,ci} = \langle -\delta, \sigma\rangle$*: the degree of preference is anti-symmetric.*

**Remark 3.**
1. *the degree of preference may depend on the degrees between other choices, but it is not always the case (forced when using an order); in particular, cycles are allowed;*
2. *the level of conflict may be not equals to* $0$ *for one agent, because an agent may not be sure of his degree of preference if he has some reasons to prefer* $a$ *to* $b$ *and some reasons to prefer* $b$ *to* $a$*.*

Subsequently, we will assume that the support is ordered and that we may use the index of a choice at the place of the choice: $\langle \delta_{i,j}, \sigma_{i,j}\rangle \eta_{i,j} = \omega(ci, cj)$.

**Example 4.** *With four choices, we may have the opinion below:*

$$\omega_1 = \begin{pmatrix} 0 & \langle +0.2, 0.00\rangle & \langle +0.3, 0.00\rangle & \langle -0.4, 0.00\rangle \\ & 0 & \langle +0.1, 0.00\rangle & \langle +0.6, 0.00\rangle \\ & & 0 & \langle +0.2, 0.00\rangle \\ & & & 0 \end{pmatrix}$$

*The choice* $1$ *is a few prefered to choice* $3$ *(preference=*$+0.3$*).*

### 4.3.4 Some projections

Now, we are going to give two projections: a weighted oriented graph of preference, and a preference relation. Hence cycles and transitivity will be expressed more easily.

#### Associated graph of preference

Degrees of preferences may be represented by a total, oriented, weighted and anti-reflexive graph.

**Definition 10 (Associated graph of preference).** *Let* $\mathcal{S}$ *be a support and let* $\omega$ *be an opinion with support* $\mathcal{S}$*.*
*We call* **preference graph** *associated to opinion* $\omega$ *a total, weighted and oriented graph* $G_\omega = (V, E, W)$*, where* $V$ *is the set of vertices,* $E \subset V \times V$ *is the set of edges, and* $W : E \to [0, 1]$ *a weighting function, with following properties:*

1. $\forall c \in \mathcal{S}, W(c, c) = 0$*: null weights for reflexive edges;*
2. $\forall (c_1, c_2) \in \mathcal{S}^2$, $c_1 \neq c_2$, $(\delta_{c_1,c_2} > 0 \Rightarrow (c_1, c_2) \in E \wedge W(c_1, c_2) = \delta_{c_1,c_2}) \wedge (\delta_{c_1,c_2} = 0 \Rightarrow ((c_1, c_2) \in E \vee (c_2, c_1) \in E \wedge W(c_2, c_1) = 0) \wedge ((c_1, c_2) \in E \Rightarrow \delta_{c_1,c_2} = W(c_1, c_2))$*: direction of edges depend on the value of the level of preference*

**Proposition 1.** *There exists a bijection between the degrees of an opinion and the associated graph of preference (modulo the direction of the edges with null weights).*

*Proof.* Let $G_\omega = (V, E, W)$ be a preference graph.
Let $\overline{G}$ be the congruence class of $G$ defined by:
$\overline{G} = \{H = (V', E', W')/mboxs.t.V = V', \forall(x, y) \in V/mboxs.t.W(x, y) \neq 0, (x, y) \in V', \forall(x, y) \in V/mboxs.t.W(x, y) = 0, ((x, y) \in V' \wedge W'(x, y) = 0) \vee ((y, x) \in V' \wedge W'(y, x) = 0)\}$.
It's easy to prove that $\overline{G}$ is a congruence class ($\forall H \in \overline{G}, \overline{H} = \overline{G}$). ∎

#### Relation

From an opinion, we can extract a relation of preference.

**Definition 11 (Preference relation $\succeq_\mathcal{P}$).** *Let $\omega$ be an opinion and $G = (V, E, W)$ an associated graph.*
*We define the* **preference relation** $\succeq_\mathcal{P}$ *by one of these equivalence below:*

1. $x \succeq_\mathcal{P} y \iff \delta_{x,y} > 0$;
2. $x \succeq_\mathcal{P} y \iff (x, y) \in E \wedge W(x, y) \neq 0$.

**Proposition 2.** $\succeq_\mathcal{P}$ *is an anti-reflexive and antisymmetric relation.*

*Proof.* 1. anti-reflexive: $\forall c \in \mathcal{S}$, $\delta_{c,c} = 0$. So $\not\exists(\delta_{c,c} > 0)$, and then $\not\exists(x \succeq_\mathcal{P} x)$;

2. antisymmetric: let assume that $x \succeq_\mathcal{P} y$. $x \succeq_\mathcal{P} y \Rightarrow \delta_{x,y} > 0 \Rightarrow \omega_{y,x} < 0 \Rightarrow y \not\succeq_\mathcal{P} x$. ∎

### 4.3.5 Our opinion formalism and the fuzzy preferences

Fuzzy preferences are very useful to model preferences of a rational decision maker [FR94]. It is a reciprocal fuzzy relation, with transitivity imposed as a condition of rationality.

**Definition 12 (Reciprocal fuzzy relation).** *Let $\mathcal{S} = \{c_1, \ldots, c_k\}$ be the set of choices.*
*A* **reciprocal fuzzy relation** *$Q$ on $\mathcal{S}$ is an application $Q : \mathcal{S}^2 \to [0, 1]$ with the condition ($q_{ij} = Q(x_i, x_j)$):*
$q_{ij} + qji = 1$ *for all $i, j \in \{1, 2, \ldots, k\}$ (reciprocity condition)*

**Proposition 3.** *The reciprocal fuzzy relation is equivalent to our degree of preference.*

*Proof.* Let $t : \Lambda \to [0, 1]$ ($\Lambda = [-1]$) defined by:
$\forall\delta \in \Lambda, t(\delta) = \frac{\delta+1}{2}$. $\forall\delta \in \Lambda, t(\delta_{i,j}) = q_{i,j}$.

i $t(\Lambda) = [0, 1]$;

ii $\delta_{i,j} = \delta_{j,i} \iff t(\delta_{i,j}) = t(-\delta j, i) \iff \frac{\delta_{i,j}+1}{2} = \frac{-\delta j,i-1}{2} \iff q_{i,j} + q_{j,i} = 1$

condition $q_{ij} + qji = 0$ of the reciprocal fuzzy relation is equivalent to $\delta_{ij} + \delta_{ji} = 0$ ∎

The differences between the two formalisms are:

1. Our formalism doesn't constrain transitivity;

2. The reciprocal fuzzy relation doesn't take conflict level into account.

## 4.4 Aggregation operator

The aggregation of preferences has been the subject of a growing body of litterature in the fields of law, economics and philosophy. The search for ways to escape from Arrow's theorem (see theorem 2 on page 66) has been following two tracks:

- to assume a collective rationality weaker than transitivity, already studied in subsubsection 4.2.1 on page 67;

- since [Saa98], the independence of irrelevant alternatives (IIA) is an inappropriate requirement when preferences are transitive (see [Ale99]).

We argue that IIA is appropriate in preference, for the same reason that we argue that transitvity is not appropriate: a preference is a local comparison among two alternatives.
The main interest of our opinion model is its ability to compute naturally the opinions of a group contrary to other approaches. In fact, using a total order to model individual preferences prevents from computing groups' preferences with the same formalism. For example, if $a$'s preference is $c_1 > c_2$, and $b$'s preference is $c_2 > c_1$, what is the preference of $a, b$ ?
In our framework and in MAS in general, opinions' aggregation is useful to:

i estimate the opinion of a group, which may be used to choose which actions to be performed;

ii compute the new position of an agent (others' opinions are informations so that an agent should take into account in order to evolve his private opinion).

A way to do that is to aggregate the opinions of others with small weights (using a weighted aggregation, as defined in section 15).
Our formalism is able to represent both agents' and groups' opinions. Values are computed using the aggregation operator (given section 4.4).
The degrees of preference model the global opinion of the group, since the levels of conflict allow to know if the preferences reach a consensus.

### 4.4.1 Characterization

According to the rationality of the aggregation, we propose axioms necessarily respected by the aggregation operator.

**Definition 13 (Aggregation operator).** *Let* $n \in \mathbb{N}^*$.
*An* **aggregation operator** $\amalg_n$ *is an application* $\Omega^p \rightarrow \Omega$ *with the following properties:*

i *Independence:* $\amalg_n(\omega^1_{ci,cj},\dots,\omega^n_{ci,cj}) = f(\omega^1_{ci,cj},\dots,\omega^n_{ci,cj})$*: the aggregation of two opinions on two choices doesn't depend on opinions on other choices;*

ii *Defined everywhere:* $\forall(\omega^1,\dots,\omega^n) \in \Omega^n$, $\forall(ci,cj) \in \mathcal{S}$, $\amalg_n(\omega^1_{ci,cj},\dots,\omega^n_{ci,cj})$ *is defined: all opinions could be aggregated;*

iii *Keep equality:* $\amalg_2(\langle\delta,\sigma\rangle,\langle\delta,\sigma'\rangle) = \langle\delta,\sigma''\rangle$*: the aggregation of two same degrees equals the degrees;*

iv *Equity:* $\forall\tau$ *a permutation on* $[1,n]$, $\amalg_n(\langle\delta_1,\sigma_1\rangle,\dots,\langle\delta_n,\sigma_n\rangle) = \amalg(\langle\delta_{\tau(1)},\sigma_{\tau(1)}\rangle,\dots,\langle\delta_{\tau(n)},\sigma_{\tau(n)}\rangle)$*: the result of the aggregation is equitable, it doesn't depend on the order of opinions;*

v *Opposition:* $\amalg_2(\langle\delta,\sigma\rangle,\langle-\delta,\sigma'\rangle) = \langle 0,\sigma''\rangle$*: if two agents have opposite choices, then the result of aggregation is that the degree of preference is null (but not the level of conflict);*

vi *Associativity:* $\amalg_2(\amalg_2(\omega,\omega'),\omega'') = \amalg_2(\omega,\amalg_2(\omega',\omega''))$*: the opinion of an aggregated opinion must not depend on how the group has been formed (e.g. when agents join the group).*

### 4.4.2 Example of our aggregation operator

**Definition 14 (Aggregation of groups' opinions).** *Let* $(\omega_i)_{1\le i\le n}$ *be a sequence of opinions:* $\forall i, \omega_i = \langle\delta_i,\sigma_i\rangle$*.*
*The quadratic mean is defined by:* $\forall i, \overline{m}_i = \sigma_i^2 - \delta_i^2$*.*
*We define* $\breve{\amalg}((\omega_i)_{1\le i\le n}) = \langle\delta,\sigma\rangle$ *where:* $\delta = \frac{1}{n}\sum_{i=1}^n \delta_i$*,* $\overline{m} = \frac{1}{n}\sum_{i=1}^n \overline{m}_i$ *and* $\sigma = \sqrt{\overline{m}-\delta^2}$

**Remark 4.** *Formulae are not randomly chosen, but are consequences of formulas used in statistic. Given a standard deviation* $\sigma$*,* $m$ *a mean and* $\overline{m}$ *a quadratic mean, from the Huygens/König formula, we deduce:* $\sigma = \sqrt{\overline{m}-m^2}$*. In this paper,* $m = \delta$*, so* $\sigma = \sqrt{\overline{m}-\delta^2}$*. The same formula are used to compute* $\overline{m}_i = \sigma_i^2 - \delta_i^2$*.*

**Proposition 4.** $\breve{\amalg}$ *is an aggregation operator.*

*Proof.* $\breve{\amalg}$ is an aggregation operator

i Independence: obvious;

ii Everywhere defined: obvious;

iii Keeps equality: since $\delta = \frac{1}{n}\sum_{i=1}^n \times\delta_i$, $\exists\delta'\forall i\delta_i = \delta' \Rightarrow \delta = \delta'$;

iv Equity: since $\delta$ and $\overline{m}$ only depend on sums of $\delta_i$ and $\overline{m}_i$, and since the operator $\sum$ is commutative, then the result doesn't depend on the order of opinions;

v Opposition: $\breve{\amalg}(\langle\delta,\sigma\rangle,\langle-\delta,\sigma'\rangle) = \langle 0, \sqrt{\frac{1}{2}(\sigma^2+\sigma'^2-2\delta^2)}\rangle$;

vi Associativity:

let $\langle M_1, \sqrt{\overline{M}_1 - M_1^2}\rangle$ $=\amalg(\omega, \omega')$, where $M_1$ $=(\delta + \delta')/2$, and $\overline{M}_1$ $=(\overline{m} + \overline{m}')/2$;

let $\langle M_2, \sqrt{\overline{M}_2 - M_2^2}\rangle$ $=\amalg(\omega', \omega'')$, where $M_2$ $=(\delta' + \delta'')/2$, and $\overline{M}_2$ $=(\overline{m}' + \overline{m}'')/2$.

Then let $\langle M_3, \sqrt{\overline{M}_3 - M_3^2}\rangle$ $=\amalg(\amalg(\omega, \omega'), \omega'')$, where $M_3 = (M_1 + \delta'')/2$, and $\overline{M}_3$ $=(\overline{M}_1 + \overline{m}'')/2$;

and let $\langle M_4, \sqrt{\overline{M}_4 - M_4^2}\rangle$ $=\amalg(\omega, \amalg(\omega', \omega''))$, where $M_4$ $=(\delta + M_2)/2$, and $\overline{M}_4$ $=(\overline{m} + \overline{M}_2)/2$.

Now, we have just to prove that $M_3 = M_4$ and $\overline{M}_3 = \overline{M}_4$:

$M_3$ $=(M_1 + \delta'')/2$ $=(\delta + \delta' + \delta'')/3$ $=(\delta + M_2)/2$ $=M_4$

$\overline{M}_3$ $=(\overline{M}_1 + \overline{m}'')/2$ $=(\overline{m} + \overline{m}' + \overline{m}'')/3$ $=(\overline{m} + \overline{M}_2)/2$ $=\overline{M}_4$

∎

An example of aggregation is given in figure 4.1. The opinions of the two agents at the left are aggregated into one opinion (the right one). Let us remark that the levels of conflict that vary from $0$ to $0.16$, depend on the closeness of degrees of preferences.

**Definition 15 (Weighted aggregation).** *Let $p \in \mathbb{N}^*$.*
*A* **weighted aggregation operator** $\tilde{\amalg}$ *is an application $(\Omega \times \mathbb{R}^+)^p \to \Omega$ defined by:* $\tilde{\amalg}((\omega_1, w_1), \ldots, (\omega_p, w_p))$ *aggregates all opinions, replacing the degrees $\delta_i$ by $w_i \times \delta_i$ and the level of conflict $\sigma_i$ by $w_i \times \sigma_i$.*

**Example 5.** *With four choices, we may have the two opinions below:*

$$\omega_1 = \begin{pmatrix} 0 & \langle +0.2, 0.00\rangle & \langle +0.3, 0.00\rangle & \langle -0.4, 0.00\rangle \\ & 0 & \langle +0.1, 0.00\rangle & \langle +0.6, 0.00\rangle \\ & & 0 & \langle +0.2, 0.00\rangle \\ & & & 0 \end{pmatrix}$$

$$\omega_2 = \begin{pmatrix} 0 & \langle +0.4, 0.00\rangle & \langle +0.3, 0.00\rangle & \langle +0.4, 0.00\rangle \\ & 0 & \langle -0.5, 0.00\rangle & \langle +1.0, 0.00\rangle \\ & & 0 & \langle +0.0, 0.00\rangle \\ & & & 0 \end{pmatrix}$$

*The resulting aggregation is then:*

$$\omega_3 = \begin{pmatrix} 0 & \langle +0.3, 0.01\rangle & \langle +0.3, 0.00\rangle & \langle +0.0, 0.16\rangle \\ & 0 & \langle -0.2, 0.09\rangle & \langle +0.8, 0.04\rangle \\ & & 0 & \langle +0.1, 0.01\rangle \\ & & & 0 \end{pmatrix}$$

*A degree of preference of the group (between two choices) is the average of the corresponding individual degrees; the level of dispersion reflects the difference of preferences.*

## 4.5 Cycle detector

In order to be sure that the process terminates, we have to detect when a situation happens twice.

### 4.5.1 Characterization

The idea is to log the exchanged opinions (the history) and to detect similar situations called "views" (notation: $u \approx_v v$) thanks to the operator called "cycle detector" as follows.

**Definition 16 (View).** *A* **view** $v$ *is an application* $A \to \Omega$*.*

**Definition 17 (History).** *An* **history** $h \in H$ *is a sequence* $(v_t)_{1 \leq t \leq T}$ *of views, where* $T$ *is the length of the history.*

**Definition 18 (Partial order on opinions).** *A* **partial order on opinions** $\succ_o$ *is defined by:*
$\forall(\omega^a, \omega^b) \in \Omega^2$*,* $\omega^a \succ_o \omega^b \Longleftrightarrow \forall(ci, cj) \in \mathcal{S}^2$*,*$\delta^a_{ci,cj} \geq \delta^b_{ci,cj} \wedge \sigma^a_{ci,cj} \leq \sigma^b_{ci,cj}$*.*

**Definition 19 (Partial order on views).** *Let* $(\omega^a)_{a \in A}$ *the agents' opinions.*
*A* **partial order on views** $\succ_v$ *is defined by:*
$\forall(v, v') \in V^2$*,* $v \succ_v v' \Longleftrightarrow \forall(a, b) \in A^2, \omega^a \succ_o \omega^b$*.*

**Definition 20 (Cycle detector).** *A* **cycle detector** $\Theta$ *is an application* $H \times \mathbb{R}^* \times \mathbb{R}^* \to \{False, True\}$ *characterized as:*

1. $\forall h \in H, h = (v_t)_{1 \leq t \leq T}, \exists t \in [1, T[, \forall a \in A, v_t(a) = v_T(a) \Rightarrow \Theta(h) = True$*: detects true cycles (i.e. when a situation happens twice);*

2. $\forall(u, v) \in V^2$*,* $u \approx_v v \Rightarrow \forall(u', v') \in V^2, u \succ_v u' \succ_v v' \succ_v v, u' \approx_v v'$*: if* $u$ *and* $v$ *correspond to a cycle, then all the couples of views* $(u', v')$ *situated between* $u$ *and* $v$ *must be detected as cycles too.*

### 4.5.2 Example of our distance cycle detector

**Definition 21 (Opinion preference distance).** *An* **opinion preference distance** $|.,.|^p_o$ *is an application* $\Omega \times \Omega \to \mathbb{R}$ *defined by:*
$\forall(\omega, \omega') \in \Omega^2$*,*$|\omega, \omega'|^p_o = \max_{i,j} |\delta_{i,j} - \delta'_{i,j}|$*.*

**Definition 22 (Opinion conflict distance).** *An* **opinion conflict distance** $|.,.|^c_o$ *is an application* $\Omega \times \Omega \to \mathbb{R}$ *defined by:*
$\forall(\omega, \omega') \in \Omega^2$*,*$|\omega, \omega'|^c_o = \max_{i,j} |\sigma_{i,j} - \sigma'_{i,j}|$*.*

**Definition 23 (View preference distance).** *A* **view preference distance** $|.,.|^p_v$ *is an application* $V \times V \to \mathbb{R}$ *defined by:*
$\forall(v, v') \in V^2$*,*$|v, v'|^p_v = \max_{a \in A} |\omega^a_v, \omega^a_{v'}|^p_o$*.*

**Definition 24 (View conflict distance).** *A* **view conflict distance** $|.,.|^c_v$ *is an application* $V \times V \to \mathbb{R}$ *defined by:*
$\forall(v, v') \in V^2$*,*$|v, v'|^c_v = \max_{a \in A} |\omega^a_v, \omega^a_{v'}|^c_o$*.*

**Definition 25 (Distance cycle detector).** *Let* $(\epsilon_p, \epsilon_c) \in \mathbb{R}^{*2}$ *be two thresholds.*
$\breve{\Theta}$ *is an application* $H \times \mathbb{R}^* \times \mathbb{R}^* \to \{False, True\}$ *defined by:*
$\forall h \in H$, $h = (v_t)_{1 \le t \le T}$, $\breve{\Theta}(h)=True \iff \exists t \in [1, T-1]$, $|v_t, v_T|_v^p \le \epsilon_p \wedge |v_t, v_T|_v^c \le \epsilon_c$.

$\breve{\Theta}$ returns true if the two views are close enough considering both view preference distance ($\epsilon_p$) and view conflict distance ($\epsilon_c$).

**Proposition 5.** $\breve{\Theta}$ *is a cycle detector.*

*Proof.* i $v_t = v_T \Rightarrow |v_t(a), v_T(a)| = 0 \Rightarrow \breve{\Theta} = True$

ii we can prove that $u \approx_v v \wedge u \succ_v u' \succ_v v' \succ_v v \Rightarrow |u', v'|_v^p \le |u, v|_v^p \wedge |u', v'|_v^c \le |u, v|_v^c \Rightarrow u' \approx_v v'$.

∎

# 4.6 Chooser operator

## 4.6.1 Characterization

The necessary axiom of a chooser operator is that if a best choice exists, then it will be elected.

**Definition 26 (Chooser operator).** *Let* $E_{max} = \{ci \in \mathcal{S} / [\forall cj \in \mathcal{S}, \delta_{ci,cj} \ge 0] \wedge [\forall (ck, cl) \in \mathcal{S}^2, (\delta_{ci,cj} \ge \delta_{ck,cl}) \wedge (\sigma_{ci,cj} \le \sigma_{ck,cl})\}$.
*A* **chooser operator** $\bigcirc$ *is an application* $\omega \to \mathcal{S}$ *with the property: if* $E_{max} \ne \emptyset$*, then* $\bigcirc(\omega) \in E_{max}$

**Remark 5.** *Generally,* $E_{max}$ *is empty; so we defined several heuristics to make this choice. In the following, we present one of them called "degrees first, conflicts next"*

## 4.6.2 Example of our chooser

**Definition 27 (Weight of a choice).** *We call* **weight of a choice** $ci \in \mathcal{S}$ *for the opinion* $\omega$ *the value:*
$w_\omega(ci) = \frac{1}{|S|-1} \sum_{cj \in \mathcal{S} \setminus \{ci\}} \delta_{ci,cj}$.

**Definition 28 (Efficient opinion).** *An opinion* $\langle \delta, \sigma \rangle$ *is efficient if:*
$\nexists \langle \delta', \sigma' \rangle, \delta \ge \delta' \wedge \sigma \le \sigma' \wedge (\delta > \delta' \vee \sigma < \sigma')$.

It's difficult to take into account the degree of preference and the conflict level in the same time, because we don't know which criteria must be used before the other; in this heuristics, we favor degrees.

**Definition 29 (Degrees first, conflicts next).** *Step 1: Build the set of the best choices* ($I$) *as follows:*

*i let* $(w_i)_{i \in \mathcal{S}}$ *be the sequence of weights of choices of* $\mathcal{S}$ *computed using definition 27.*

*ii let* $w_{max} = \max_i w_i$.

iii *let* $\epsilon \in \mathbb{R}^*$ *be a threshold.*

iv *let* $I = \{i \in [1,n]/w_i \geq w_{max} - \epsilon\}$ *be the set of choices that are close to the maximum.*

*Step 2: K is a restriction of I such that K is a total order.*

i *let* $\succeq_{\mathcal{P}}$ *be the preference relation defined by* $\delta_{ci,cj} \geq 0 \Longleftrightarrow i \succeq_{\mathcal{P}} j$.

ii *let* $Q$ *be the set of relations between choices of* $I$ *ordered by* $\sigma_i$.

iii *let apply the process:*

*let* $K$ *be an empty partial order.*

*while* $Q \neq \emptyset$ *do*

*let* $(i \succeq_{\mathcal{P}} j) = min(Q)$; $Q \leftarrow Q \backslash \{i \succeq_{\mathcal{P}} j\}$. */* less conflict*/*

*if* $K \cup (i \succeq_{\mathcal{P}} j)$ *doesn't contain a cycle, then add the relation to* $K$.

*endwhile*

*We call "degrees first, conflicts next" the application* $\omega_{\mathcal{S}} \mapsto max(K)$.

**Proposition 6.** *The application "degrees first, conflicts next" is an opinion chooser.*

*Proof.* In order to compute $max(K)$, we have to prove that $K$ is a total order. Then we have to prove that if $Emax$ is not empty, then an element $c^\star$ of $Emax$ will be chosen. The leading idea behind the proof: i) $c^\star \in I$; ii) the relations that contain $c^\star$ are added before the others; iii) $max(K) \in E_{max}$.
First, $K$ is a partial order, because it is an empty order in which we only add relations that don't add cycles. Why is it a total order ? For all $(ci, cj) \in \mathcal{S}^2$, we try to add $ci \succeq_{\mathcal{P}} cj$ or $cj \succeq_{\mathcal{P}} ci$. If the relation is refused, it means that there exists a sequence of choices between $ci$ and $cj$, ordered in the opposite order than the refused relation. So $ci$ and $cj$ are already comparable. Since we try to add all relations ($ci \succeq_{\mathcal{P}} cj$ or $cj \succeq_{\mathcal{P}} ci$), the order transitively closed is total.
Secondly, let us proof that $max(K) \in E_{max}$. Let $c^\star \in E_{max}$:

i $w_ci = \sum_j \delta_{ci,cj}$ and $\forall(ci,cj,ck) \in \mathcal{S}^3$,$\delta_{c^\star,cj} \geq \delta_{ck,cl}$, so $w_c^\star$ is maximal, so $E_{max} \subset I$ (all elements of $E_{max}$ have the same weight);

ii $\forall(ci,cj,ck) \in \mathcal{S}^3$,$\sigma_{c^\star,cj} \leq \sigma_{ck,cl}$, so the relations that contain an element $c^\star$ of $E_{max}$ are added before the others;

iii $K$ contains all relations that contain at least one element $c^\star$ of $E_{max}$; since $\forall cj \in \mathcal{S}, c^\star \succeq_{\mathcal{P}} cj, max(K) \in E_{max}$.

∎

**Example 6.** *Let us apply this operator to the aggregated opinion of the figure 4.1.*
*Step 1: let us compute the sequence of weights:*

- $w_1 = (-.3 + .3 + .3)/3 = .1$

- $w_2 = (.3 - .2 + .2)/3 = .1$
- $w_3 = (-.3 + .2 + .4)/3 = .1$
- $w_4 = (-.3 - .2 - .4)/3 = -.3$*; so* $I = \{c_1, c_2, c_3\}$.

*Step 2:* $Q = \{c_1 \succeq_{\mathcal{P}} c_3; c_2 \succeq_{\mathcal{P}} c_1; c_3 \succeq_{\mathcal{P}} c_2\}$. *We may remark that the most important relations form a cycle.*

- $K_0 = \emptyset$
- $K_1 = \{c_1 \succeq_{\mathcal{P}} c_3\}$
- $K_2 = \{c_1 \succeq_{\mathcal{P}} c_3; c_2 \succeq_{\mathcal{P}} c_1\}$
- $K_3 = \{c_1 \succeq_{\mathcal{P}} c_3; c_2 \succeq_{\mathcal{P}} c_1\} = K_3$

*Finally,* $c_2$ *is the preferred car.*

## 4.7 Consensus detector

A consensus operator has to answer the question: do all agents agree ? The vote is often used: first, each agent chooses one choice and the one that has the maximum of votes is elected. In some vote systems, agents may choose more than one choice (often two), but all choices have the same weight. We propose to extend this system by aggregating all opinions into a lone one using an aggregation operator, and then by choosing by applying a chooser operator on the aggregated opinion.
Two parameters are taken into account: the degree of preference and the conflict level.

### 4.7.1 Characterization

**Definition 30 (Consensus detector).** *A* **consensus detector** $\bowtie$ *is an application* $\Omega \to \{False, True\}$ *defined by:* $\bowtie(\omega) = True \iff \forall(ci, cj) \in \mathcal{S}^2, \sigma_{ci,cj} = 0$

It seems rational to impose that if one choice is preferred by all agents, then this choice will be elected.

### 4.7.2 Example of consensus detector

**Definition 31 (Epsilon consensus detector).** *Let* $\epsilon \in \mathbb{R}^*$.
*An* **epsilon consensus detector** $\breve{\triangleleft}_\epsilon$ *is an application* $\Omega \to \{False, True\}$ *defined by:* $\forall(ci, cj) \in \mathcal{S}^2, \sigma_{ci,cj} \leq \epsilon \Rightarrow \bowtie(\omega) = True$

**Proposition 7.** *For all* $\epsilon \in \mathbb{R}^*$, $\breve{\triangleleft}_\epsilon$ *is a consensus detector.*

*Proof.* The proof is obvious: $\sigma_{ci,cj} = 0 \Rightarrow \sigma_{ci,cj} \leq \epsilon \Rightarrow \bowtie(\omega) = True$. ∎

# 4.8 Nearest opinion chooser

## 4.8.1 Characterization

We are not always able to determine the two nearest opinions, because it is difficult to define a distance on opinions. To characterize a nearest opinion chooser, we need to represent the fact that two opinions that are more near for all couple of choices (for their degrees of preference and for their level of conflict) than two others will be preferred (but not necessarily chosen). So we will first compute vectorial deviations between two opinions, and then order partially these couples to find the best ones.

**Definition 32 (Efficient Vectors).** *Let $p \in \mathbb{N}^*$ and let $E^p$ be a vectorial space. Let $\preceq$ a partial order on $E^p$ defined by $\forall(u,v) \in E^p$, $u \preceq v \iff \forall i \in [1,p]$,$u_i \leq v_i$. The* **efficient vectors** *of $E^p$ are the maximal elements of $(E^p, \preceq)$.*

**Definition 33 (Vectorial Total Deviation).** *Let $\mathcal{S}$ be a support.*
*Let $\omega$ and $\omega'$ be two opinions with support $\mathcal{S}$.*
*Let $n = |\mathcal{S}|$.*
*A* **Vectorial Total Deviation** *$\langle .,. \rangle$ is an application $\omega \times \omega' \rightarrow \Delta^{n(n-1)/2} \times \varsigma^{n(n-1)/2}$ defined by:*
$\langle \omega, \omega' \rangle = (|\delta_{1,2} - \delta'_{1,2}|, \ldots, |\delta_{1,n} - \delta'_{1,n}|, |\delta_{2,3} - \delta'_{2,3}|, \ldots, |\delta_{2,n} - \delta'_{2,n}|, \ldots, |\delta_{n-1,n} - \delta'_{n-1,n}|, |\sigma_{1,2} - \sigma'_{1,2}|, \ldots, |\sigma_{1,n} - \sigma'_{1,n}|, |\sigma_{2,3} - \sigma'_{2,3}|, \ldots, |\sigma_{2,n} - \sigma'_{2,n}|, \ldots, |\sigma_{n-1,n} - \sigma'_{n-1,n}|)$.

**Definition 34 (Nearest opinions chooser).** *A* **nearest opinions chooser** *is an application $\mathcal{B} \rightarrow \Omega^2$ with the constrain:*
*given the set of vectorial total deviation $VTD$ computed on $\mathcal{B}\times\mathcal{B}$ ($VTD = \{\langle \omega, \omega \rangle / (\omega, \omega) \in \mathcal{B}^2\}$),*
*given the set of efficient opinions of $VTD$ namely $VTD_{eff}$, $\Psi(B) \in VTD_{eff}$, we must not be able to find two nearest opinions closer than the two chosen opinions.*

**Remark 6.** $\exists(\omega, \omega') \in \Omega^2, \omega = \omega' \Rightarrow (\Psi = (\omega'', \omega''') \Rightarrow \omega'' = \omega''')$*: if there exists two equal opinions, then the chosen opinions will be equal too.*

## 4.8.2 Example of our norm nearest opinion chooser

**Definition 35 (Vectorial Preference Deviation).** *Let $\mathcal{S}$ be a support.*
*Let $\omega$ and $\omega'$ be two opinions with support $\mathcal{S}$.*
*Let $n = |\mathcal{S}|$.*
*A* **vectorial preference deviation** *$\langle .,. \rangle_p$ is an application $\Omega \times \Omega \rightarrow \Delta^{n(n-1)/2} \times \varsigma^{n(n-1)/2}$ defined by:*
$\langle \omega, \omega' \rangle_p = (|\delta_{1,2} - \delta'_{1,2}|, \ldots, |\delta_{1,n} - \delta'_{1,n}|, |\delta_{2,3} - \delta'_{2,3}|, \ldots, |\delta_{2,n} - \delta'_{2,n}|, \ldots, |\delta_{n-1,n} - \delta'_{n-1,n}|)$.

**Definition 36 (Vectorial Conflict Deviation).** *Let $\mathcal{S}$ be a support.*
*Let $\omega$ and $\omega'$ be two opinions with support $\mathcal{S}$.*
*Let $n = |\mathcal{S}|$.*
*A* **vectorial conflict deviation** *$\langle .,. \rangle_c$ is an application $\Omega \times \Omega \rightarrow \Delta^{n(n-1)/2} \times \varsigma^{n(n-1)/2}$ defined by:*
$\langle \omega, \omega' \rangle_c = (|\sigma_{i,j} - \sigma'_{i,j}|)_{1 \leq i < j \leq n}$.

**Definition 37 (Square Norm).** *Let* $p \in \mathbb{N}^*$.
*Let* $E^p$ *be a vectorial space.*
*The usual* **square norm** $\|.\|$ *is defined by:*
$\forall v \in E^p, \|v\| = \sqrt{\sum_{i=1}^{p} v_i^2}$

**Definition 38 (Norm Nearest Opinions Chooser).** *Let* $p \in \mathbb{N}^*$.
*A* **square norm nearest opinions chooser** $\ddot{\Psi}$ *is an application* $\mathcal{B} \rightarrow \Omega^2$ *defined by:*

*i* *let* $E_{min}$ *be the set of couples of opinions* $(\omega, \omega')$ *such that* $\|\langle \omega, \omega' \rangle_p\|$ *is minimal;*

*ii* *if* $E_{min}$ *contains more than one couple, then let remove couples of opinions* $(\omega, \omega')$ *such that* $\|\langle \omega, \omega' \rangle_c\|$ *is minimal;*

*iii* *if* $E_{min}$ *contains more than one couple, then let choose a couple randomly.*

**Proposition 8.** *The application* $\ddot{\Psi}$ *is a nearest opinion chooser.*

*Proof.* if two opinions are nearer, then it means that all coordinates are less or equal, what leads necessarily to a smaller norm. ∎

### 4.8.3 Consensus operator based on the multi-set of opinions

#### Characterization

**Definition 39 (strict chooser operator).** *A* **strict chooser operator** $\odot$ *is a chooser operator defined by:*
$\odot(\omega) = \{c^\star\} \iff \exists! c^\star \in \mathcal{S}, \forall c \in \mathcal{S} \backslash \{c^\star\}, \omega(c^\star) > \omega(c)$*; in other cases,* $\odot(\omega) = \emptyset$

**Definition 40 (Consensus operators based on aggregated opinions).** *A* **consensus operators based on aggregated opinions** $\odot$ *is an application* $\Omega \rightarrow \{False, True\}$ *that obeys to the postulates:*
$\exists c^\star \in \mathcal{S}, \odot(\omega) = \{c^\star\} \Rightarrow \triangleleft(\omega) = True$ *(relation of preference).*
$\triangleleft(\omega) = False$ *is interpreted as "consensus is not reached".*

**Definition 41 (Consensus operator based on the multi-set of opinions).** *Let* $B \in \mathcal{B}$.
*A* **consensus operator based on the multi-set of opinions** $\triangleright$ *is an application* $\mathcal{B} \rightarrow \{False, True\}$ *that obeys to the following postulates:*

*i* $\exists c^\star \in \mathcal{S}, \forall \omega \in B, \odot(\omega) = \{c^\star\} \Rightarrow \triangleright(B) = True$*;*

*ii* $\forall (\omega, \omega') \in B^2, \omega = \omega' \Rightarrow \triangleright(B) = True$*.*

$\triangleright(B) = False$ *is interpreted as "consensus is not reached".*

#### Examples

**Definition 42 (Strict consensus operator).** $\triangleright(B) = \{c^\star\} \iff \exists c^\star \in \mathcal{S}, \forall \omega \in B, \odot(\omega) = \{c^\star\}$

**Definition 43 (Strict majority consensus operator).** *Let $Q_{(c \in \mathcal{S})}$ be a sequence of set of agents defined by:*
$Q_c = \{a \in A / \odot(\omega(a)) = c\}$.
$\triangleright(B) = \{choice^\star\} \iff \exists |Q_{c\star}| \geq |B| mod 2 + 1$.

**Definition 44 (Weighted majority consensus operator).** *Let $Q_{(c \in \mathcal{S})}$ be a sequence of set of agents defined by:*
$Q_c = \{a \in A / \odot(\omega(a)) = c\}$.
$\triangleright(B) = \{choice^\star\} \iff \exists |Q_{c\star}| \geq |B| mod 2 + 1$.

**Definition 45 (Elimination sequence by the lower part).** *Let $P_S$ an opinion with support $S$.*
*We call* **elimination sequence by the lower part** *a sequence $(S_n)_{n \geq 0}$ of supports defined by:*

$$\begin{cases} S_0 &= S \\ S_{n+1} &= S_n \backslash \{x \in S_n / \forall y \in S_n, poids_P(x) \leq poids_P(y)\} \end{cases}$$

**Proposition 9.** *The sequence of elimination by the low part converges to $\emptyset$ in a finite number of steps.*

*Proof.* The sequence is strictly decreasing (while $S_m \neq \emptyset$): each step, at least one element is removed. Since the set is finite, $\exists m / S_{m+1} = \S_m$.
∎

**Definition 46 (Chooser by elimination by the low part).** *Let $\mathcal{S}$ be a support and $\omega$ an opinion.*
*Let $(\mathcal{S}_n)_{n \geq 0}$ be the sequence of elimination by the low part.*
*Let $m = min\{n / \mathcal{S}_n = \emptyset\} - 1 = n / \mathcal{S}_{n+1} = \emptyset$.*
*We call* **chooser by elimination by the low part** *the function $\omega \mapsto \mathcal{S}_m$.*

## 4.9 Experiments

### 4.9.1 Experimental protocol

The goal of these experiments is to compare the results of a vote using an aggregation of opinions (our formalism) and a usual vote systems (agents vote for one candidate).
To compare the two systems requires to compare the same initial data, the opinions. We could express them using one of the two formalisms before translating into the other, but in this case, the first would be advantaged, because the translation neglects some pieces of information. We decide to use a third formalism (called the main formalism), used in data analysis.
The second difficulty is to compare the results. How to estimate that a decision procedure is better than another ? We propose to compute the satisfaction of each voter by calculating a distance based on opinion between the elected candidate and his preference.

### The representation of advices

We assume that voters judge candidates by several themes. Each candidate and each voter has a position about each theme, modeled by a real value between $0$ and $1$.
To compute his degree of preference between two candidates, each voter compute the distance between his position and the position of the two candidates and makes the subtraction.
To vote, the voter ranks candidates according to the first theme only. Using an another theme or a linear combination doesn't change results.

### The election

In the aggregation of opinions procedure, the meaning of the degrees of preferences is computed and the candidate that is preferred according to these means is elected.
The winner of the vote system is the candidate that has the more votes.

### The satisfaction

In both cases, the satisfaction is computed using a distance between the voters' position and the position of the elected candidate. The distances are not the same (a $n$ dimension space in the first case, a $1$ dimension space in the second). So, results are normalized between $0$ and $1$.

**Definition 47 (Candidate).** *Let $T$ be a set of $n$ themes.*
*A* **candidate** *is a choice $c \in \mathcal{S}$ with a position on each theme that may be modeled by:* $pos_c^{cand} : T \to [0,1]^n$.

**Definition 48 (Voter).** *Let $T$ be a set of $n$ themes.*
*Let $\mathcal{S}$ be a set of candidates (the choices).*
*The position of a* **voter** *is modeled by:* $pos_a^{voter} : T \to [0,1]^n$.

**Definition 49 (Voter's Opinion Computation).** *Let $T$ be a set of $n$ themes.*
*Let $\mathcal{S}$ be a set of candidates (the choices).*
*Let $a \in A$ be an agent.*
*The computation of the degree of preference is given by the formula:*
$\forall (c_1, c_2) \in \mathcal{S}^2$, $\delta^a_{c_1,c_2} = |\sqrt{\sum_{t \in T} (pos_a^{voter}(t) - pos_{c_1}^{cand}(t))^2} - \sqrt{\sum_{t \in T} (pos_a^{voter}(t) - pos_{c_2}^{cand}(t))^2}|$.

**Definition 50 (Aggregation of Votes).** $\forall (c_1, c_2) \in \mathcal{S}^2$, $\delta^{aggreg}_{c_1,c_2} = \frac{1}{|A|} \sum_{a \in A} \delta^a_{c_1,c_2}$.

**Definition 51 (Voter's Valuation Computation).** *Let $T$ be a set of $n$ themes.*
*Let $\mathcal{S}$ be a set of candidates (the choices).*
*Let $A$ be a set of voters (the agents).*
$val_c = |pos_a^{voter}(1) - pos_c^{cand}(1)|$.

$vote(a) = c/val_c$ *minimal.*
$c^\star = \{c \in \mathcal{S}/pos_c^{cand}(t_{i_1})maximal\}$

**Random votes**

To begin with, a number of choices $n$ (the candidates) is randomly chosen between $3$ and $15$; this number is also the number of themes. Then, $n$ candidares are randomly generated (*i.e.* their positions are randomly computed between $0$ and $1$).
Then, a population of voters is randomly computed: they generated a random ordered list of themes.
Two decision procedures are compared: the traditional vote and the agregation of opinions.

### 4.9.2 Results

Some parameters have been chosen:

- number of elections : 20
- number of candidates : between $3$ and $18$ (randomly chosen)
- number of voters : 100
- demandings : 1, 5 and 10

The two following figures (figure 4.2 on the next page and figure 4.3 on the facing page) show the distances between the voters' positions and the elected candidate's position. A voter is more satisfied if the value is close to $0$.
On the first one (figure 4.2 on the next page), we see that :

- for the classic vote, the frequency decreases slowly when the distance increases;
- for the aggregated vote, the frequency looks like a Gauss bell.

If we interpret the distance as a measure of unsatisfaction (small distances = great satisfaction), we may conclude that the first vote system gives broadly as many satisfied voters as voters who are not satisfied and that, on the contrary, the second shows that most voters are fairly satisfied. These results are shown figure 4.2 on the facing page.
The cumulate frequency allows to know the frequency of voters which level of satisfaction is at least $x$; for instance, figure 4.3 on the next page shows that $70\%$ of the voters have level of unsatisfaction less or equal than $10$. In fact, what ever the threshold of satifaction, the vote is always better than the aggregation (at the right side, they are similar).

In fact, these results are provided by an interpretation of the distances. We may chose to attach much importance to high distances, *i.e.* to interpret high distances (more than $15$) as a higher unsatisfaction than previously, *i.e.* to increase the demanding. To do that, we chose to raise to a $n$th power, with $n = 5$ and $n = 10$. The figure 4.5 on page 88

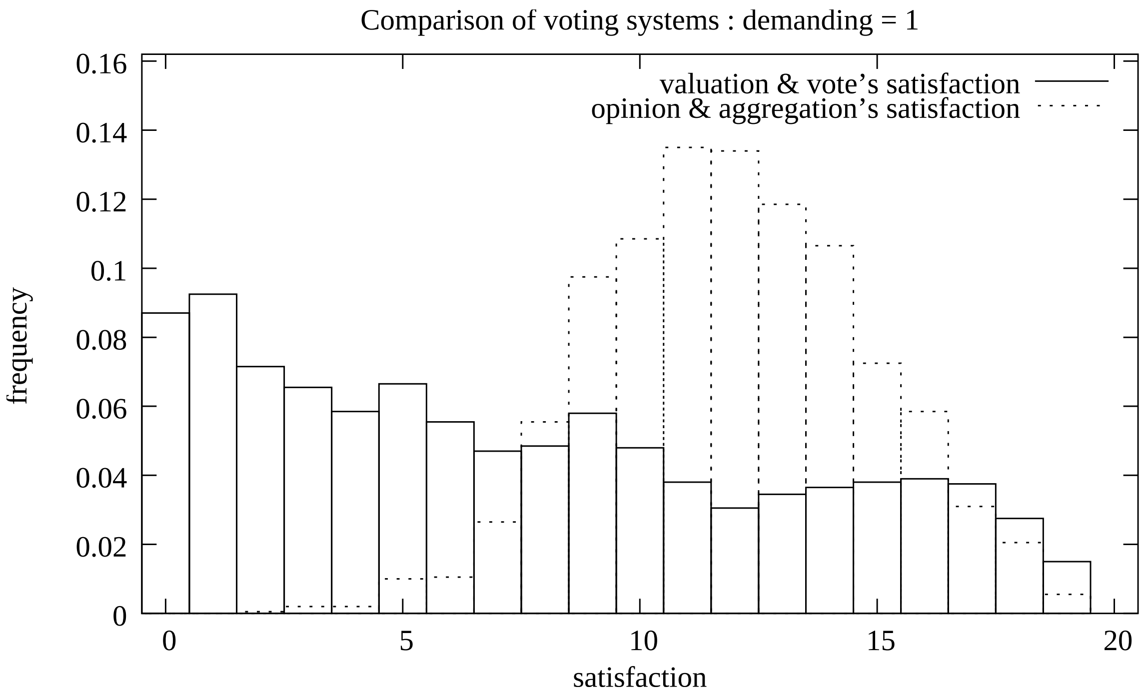


Figure 4.2: Frequency of satisfaction : demanding = 1

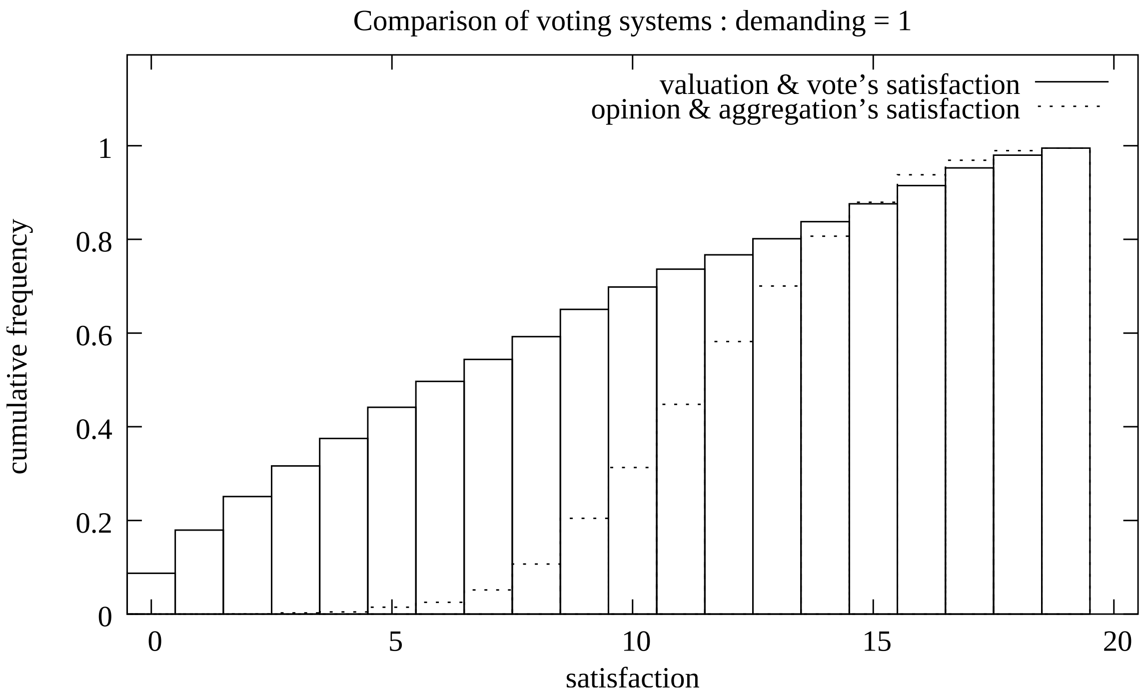


Figure 4.3: Cumulate frequency of satisfaction : demanding = 1

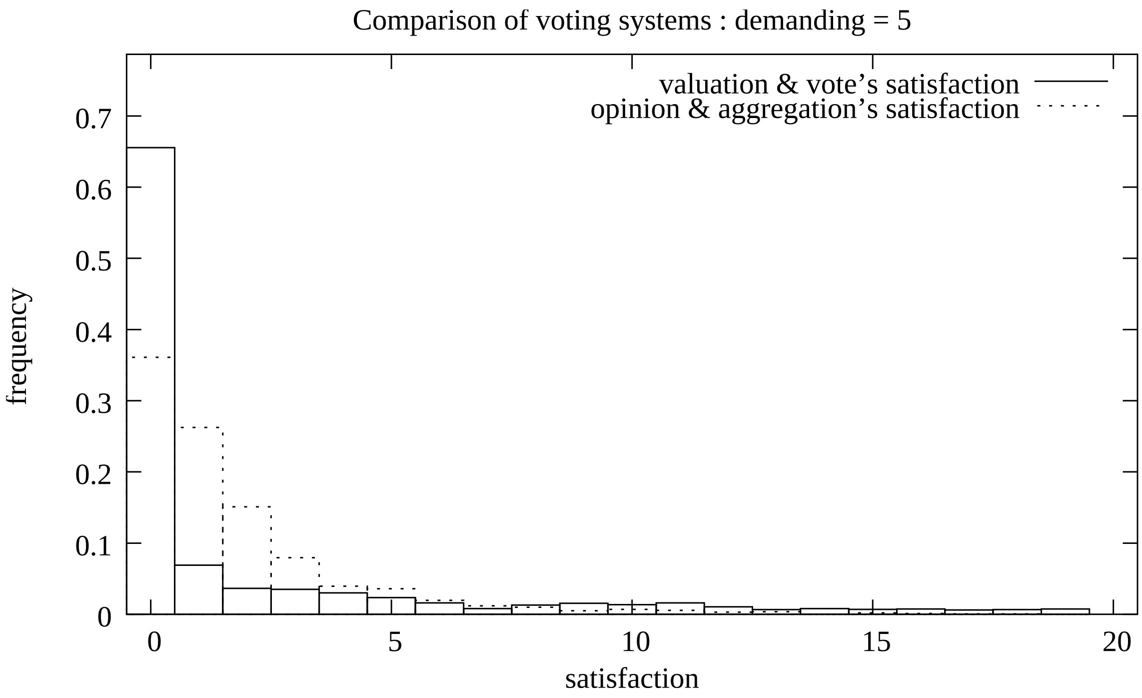


Figure 4.4: Frequency of satisfaction : demanding = 5

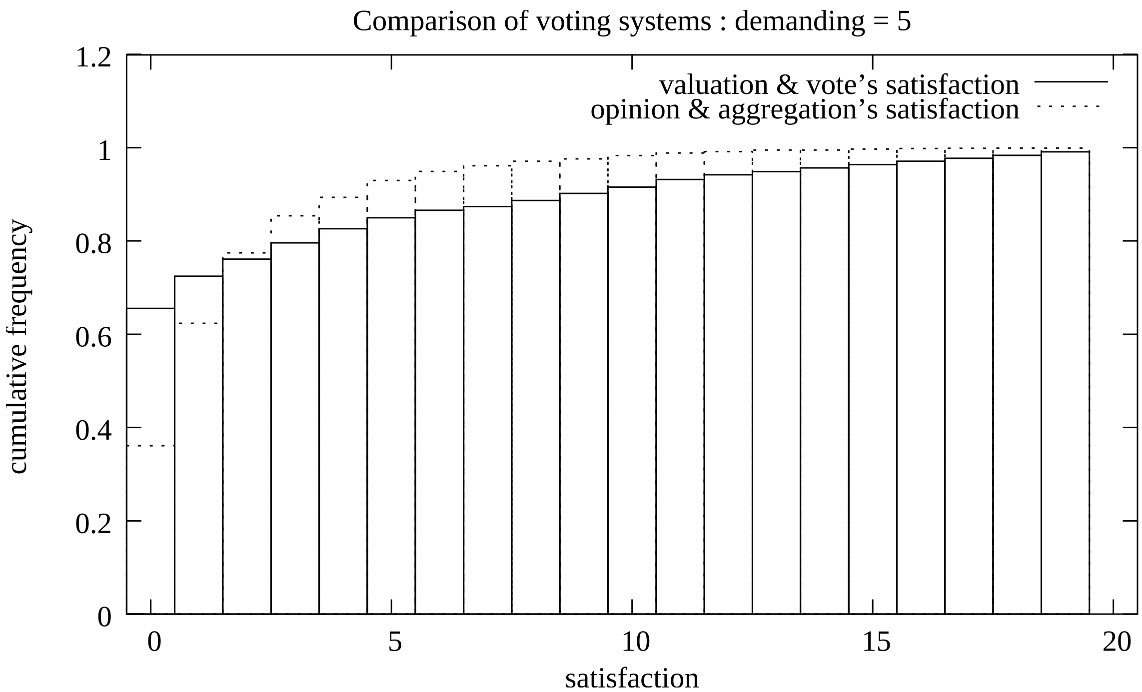


Figure 4.5: Cumulate frequency of satisfaction : demanding = 5

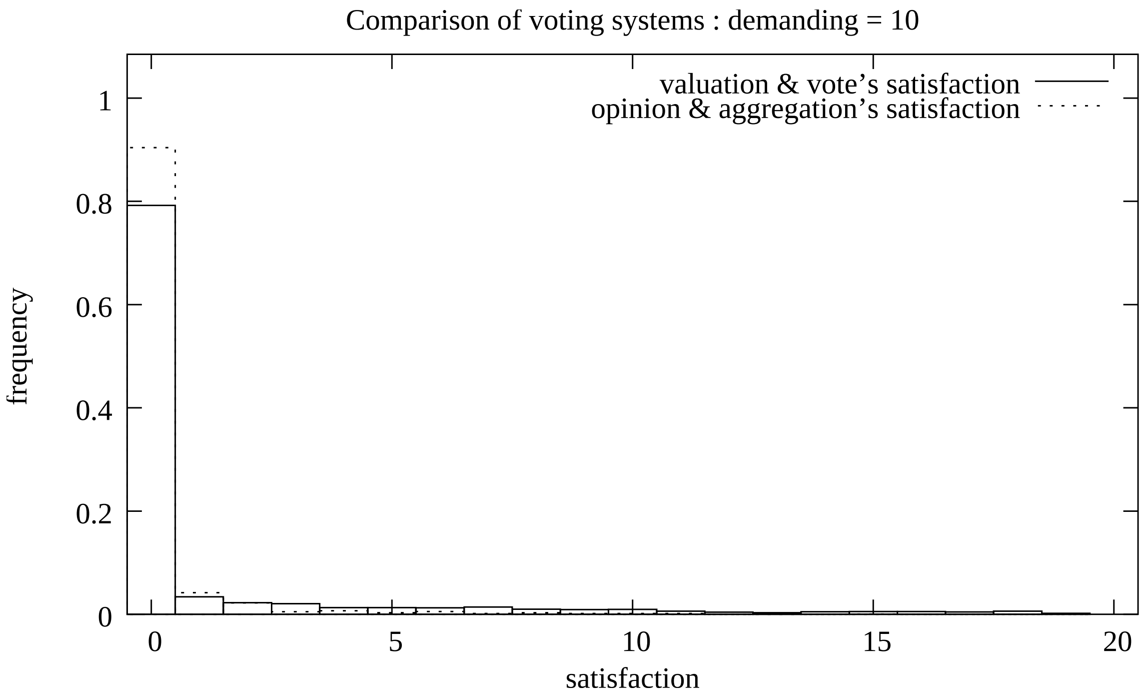


Figure 4.6: Frequency of satisfaction : demanding = 10

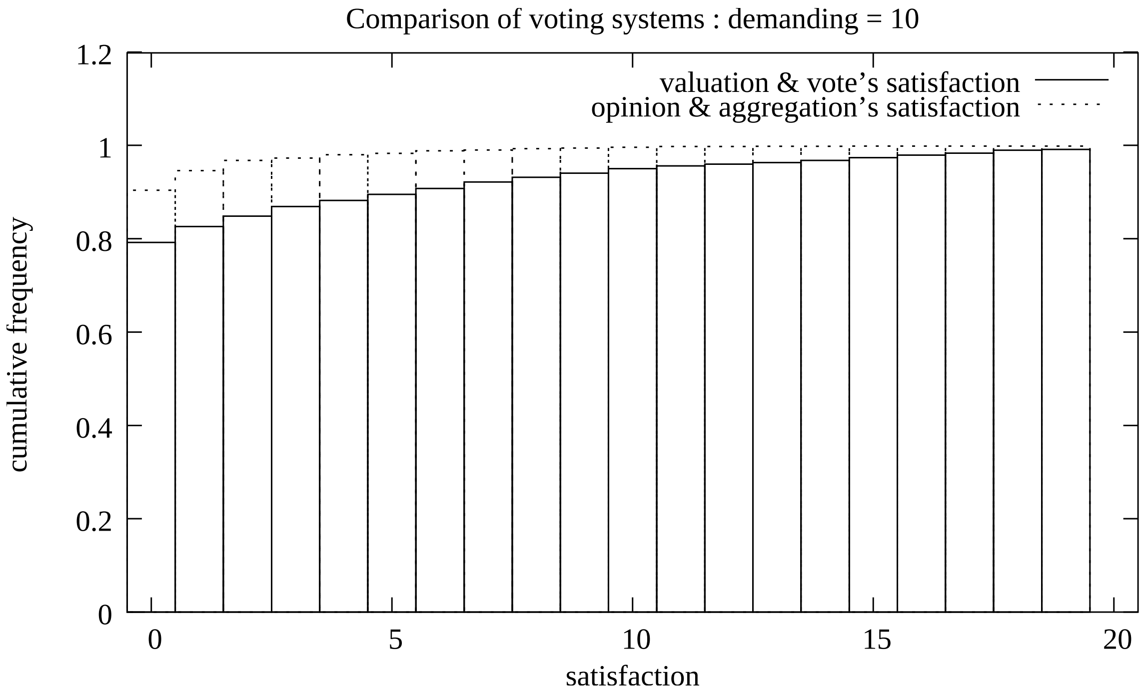


Figure 4.7: Cumulate frequency of satisfaction : demanding = 10

and figure 4.7 on the page before show that when the demanding increases, the cumulate frequencies of aggregation vote becomes higher than classic vote.

To conclude, the elected candidate using a classic vote system satisfied highly a small part of the voters, since using an aggregated vote system, voters are fairly satified. The choice of a vote system depends on the level of unsatisfaction that may be considered as acceptable.

## 4.10 Conclusion

In the previous chapter 3 on page 41, the protocol required a formalism of opinions and several operators.
Several formalisms of preference representation have already been presented by other researchers: numerical, structural or hybrid approaches.
The last approach (*e.g.* fuzzy preferences), is the more expressive, but a preference is regarded as a global comparison among choices, what leads to the problem of transitivity; on a contrary, we consider a preference as a local comparison.
Hence, we can easily compute an aggregated opinion. Other operators (chooser, consensus detector, cycle detector and nearest opinion chooser) are characterized and some example of such operators are given. These examples are used in the chapter 7 on page 119 that test our protocol.
Experiments about the choice of a solution gives a comparison between the disappointment of our operators and traditional voting systems. They show that our system produces concentrate satisfactions, since in the other case, satisfaction is more diffused.

# Bibliography


[Ale99] F. Aleskerov. *Arrovian aggregation models*. Kluwer Academic Publisher, 1999.

[Arr63] Kenneth Arrow. *Social Choice and Individual Values*. Wiley, New York, 2nd edition, 1963.

[Arr91] K. Arrow. *The Origins of the Impossibility Theorem*, chapter 1, pages 1–4. Elsevier Science Publishers B. V., Amsterdam, 1991.

[Bil98] Taner Bilgiç. Interval-valued preference structures. *European Journal of Operational Research*, (105(1)):162–183, 1998.

[DJJT01] M. Dastani, N. Jacobs, C.M. Jonker, and J. Treur. Modeling user preferences and mediating agents in electronic commerce. In C. Sierra F. Dignum, editor, *Agent-Mediated Electronic Commerce*, volume 1991 of *LNAI*, pages 164–196. Springer verlag edition, 2001.

[FR94] J. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publisher, Dordrecht, The Nederlands, 1994.

[MB] Hans De Meyer and Bernard De Baets. New transitivity conditions for reciprocal fuzzy relations.

[Mon88] B. Monjardet. A generalisation of probabilistic consistency: linearity conditions for valued preference relations. In Springer-Verlag, editor, *Non-Conventional Preference Relations in Decision Making*, volume 301 of *Lecture Notes in Economics and Mathematical Sciences*. Eds. J. Kacprzyk and M. Roubens, 1988.

[Saa98] D. Saari. Connecting and resolving sen's and arrow's theorems. *Social Choice and Welfare*, 15:239–261, 1998.

[Sen70] A.K. Sen. *Collective Choice and Social Wellfare*. Holden-Day, 1970.

[Swi01] Z. Switalski. General transitivity conditions for fuzzy reciprocal preferences matrices. In *Proceedings of EUROFUSE Workshop on Preference Modelling and Applications*, pages 99–104, Granada, Spain, 2001.

[VE02] Guillaume Vauvert and Amal El Fallah - Seghrouchni. Formal specification of opinions applied to the consensus problem. In *Proc. of the VIIIth IBERAMIA'2002*, Seville, Spain, 2002. To appear.

# Chapter 5

# Alliances among Totally Autonomous Agents

## Contents



## Figures

## Tables

Usually, in MAS, the term *coalition* is used instead of the term *alliance*. Our reaching consensus protocol requires another kind of group, which looks better like a coalition: agents rally *against* other groups. Moreover, in our context of totally autonomous agents, the concept of usually called *coalition* is closer to the notion of *alliance*: agents regroup to take of advantage of the synergy of skills. Possibly, they may ally against some other teams, but it is not always the case. We hope that this exchange of words keeps the thesis clear.

## 5.1 Introduction

Main interest of MAS results from collaboration of agents. Several approaches has been studied to support this collaboration: distributed problems solving, contractual networks, organization based approaches, protocols of negotiation, alliance formation, *etc.*
Alliances [VE01] allow agents to satisfy needs requiring synergy for competences of several agents as, for example, within the framework of the resolution of complex tasks which agents acting alone would be unable to compute or much less effective.
An alliance can be defined as a short-term organization based on specific and contextual engagements thus making possible for agents to benefit from their respective competences (in economic context for example, several companies gather "virtually" to answer to bids requiring various competences). Alliance formation offers several advantages:

1. the concept of punctual engagement allows agents to react in an opportunistic way and to dynamically revise their interests and consequently their objectives;

2. the alliances formation and dissolution are context-dependent, thus they allow agents to dynamically adapt their dealings;

3. contrary to a static organization (*i.e.* preset), alliances formation allows to apprehend in a more flexible way open and dynamic environments.

From the point of view of MAS, researches in alliances formation try to propose protocols of automation which are more "realistic". Indeed, the search for optimal solutions (generally measured in term of maximum total profit) require NP-complete algorithms and because of this complexity, several simplifying assumptions are introduced in order to generate solutions computable in moderate time. One of the strongest assumptions consists in supposing that agents are co-operative and even altruistic. Within our framework of totally autonomous agents, contrary to existing approaches (see section 5.3 on page 96) and according to with a more general sense, we make no assumption on predispositions of agents to collaborate. We are satisfied to motivate them to do it (*e.g.* agents will be brought to cooperate if they find a certain interest there). Thus, each agent is free to behave and is likely to have selfish objectives (*e.g.* within the framework of electronic commerce). However a set of egoistic agents generally leads to deadlock, which finally satisfies nobody. To avoid this type of situations, we propose a protocol for the alliances formation which respects the freedom of agents while guaranteeing to reach a consensus. This protocol is based on the reaching of a consensus (see chapter 3 on page 41).

## 5.2 Related work

### 5.2.1 The need of cooperation

Agents may work jointly in order to increase the efficiency of task achievement at the system level (more tasks can be satisfied) and at the agent level (increase agent's ability to satisfy their goals and maximize their own personal payoff [SK96a]) [ZR94]. A coalition is thus a group of enterprises who have decided to cooperate in order to carry out a common task [SSJ97]. Agents may be members of more than one coalition [SSJ97] in order

to increase the set of achievable tasks.
In economic context, rational agents are lead to cooperate in several situations [SSJ97]:

1. agents cannot perform tasks by themselves;

2. other agents are more efficient in performing tasks;

3. working on the task collaboratively will increase benefits or reduce its costs.

To increase the efficiency of task achievement, agents may work jointly [ZR94, SL95, SK96b, Kep94] and may form alliances viewed as groups of agents whom have decided to cooperate in order to carry out a common task [SSJ97].
Suitable to dynamic environments, alliance formation is usually studied from three perspectives, considered independently even if they are not:

1. alliance structure generation (partitioning or covering the set of agents);

2. solving the optimization problem (solving their joint problem and receiving eventually a benefit depending on used resources and spent time);

3. dividing the value of the solution among agents (decided by agents themselves or imposed before beginning the process, addressed by game theory [Rap70]).

Multi-Agent Systems are becoming more and more important for three main reasons pointed out in [Kra97, SLA$^+$99]:

- a growing communication infrastructure over which separately designed agents can interact (intranets and internets);

- applications for computer support for negotiation at the operative decision making level;

- industrial trend towards virtual enterprises (cooperation *via* electronic interactions).

### 5.2.2 What is an alliance?

An alliance may be defined a group of agents whom have decided to cooperate in order to carry out a common task [SSJ97, Akn01, APS02]. An alliance can work on a single task at each time [SSJ97, SK95b], but sometimes agents may be members of more than one alliance [SSJ97].
Each real-world organization has his own structure which is influenced by many factors (efficiency of the conception, size and age, technical system, environment, power [Min81]). All organizational theories agree that no optimal structure exists, thus organizations have to modify dynamically their structure to follow environment changes. Alliance formation takes into account requirements and constraints arising from the dynamic nature of the environment [SSJ97], even if new pieces of information are generally not integrated during the whole process, but at some given times. After [Tay11], most of theories criticize hierarchical and centralized structures [Fol24, Dub38], and a new form of organization without centralized authority based on decentralized responsibilities has been propounded in [Dub38]. Each team is technically and economically autonomous and

no power relation exists between teams. This structure looks like an alliance. In MAS, alliance formation allows to coordinate agents when information is distributed and changes dynamically [SSJ97].

### 5.2.3 Alliance formation

Alliance formation addresses three problems [SLA$^+$99], which are usually considered independent even if they are not:

1. alliance structure generation, else partitioning or covering the agents;
2. solving the optimization problem (solving their joint problem and receiving eventually a benefit depending on used resources and spent time);
3. dividing the value of the solution among agents (decided by agents themselves or imposed before beginning the process, addressed by game theory [Rap70]).

The problem of distributed task allocation has been tackled in the Distributed Problem Solving (DPS) context [Smi80]: an agent that attempts to satisfy a task may divide it into several sub-tasks using sub-contracts. In this case, a task is allocated to a single agent which is responsible of its performance (a distribution is necessary). Efficiency is evaluated through simulation.
Many works in game theory [Rap70, NM47, KW91, LR57] address the problem of alliance formation, but often concentrate on the distribution of the benefits, the stability and the fairness. Algorithms are usually centralized, with exponential complexity and not limited in communication and time and alliance are statically evaluated. Stability was widely studied in [KR84].
Set Partitioning Problem and Set Covering Problem have been studied in operational research, combinatorial algorithms and graph theory [GN, BP72, CK75]. They have been shown to be NP-hard problems [GJ79]. As argued in [SK95b], these approaches provide no appropriate solution to the problem of alliance formation among agents, due to three main deficiencies: exact and optimal solutions have exponential complexity to be found; approximated solutions have polynomial complexity but sub-groups studied are artificially limited in size ([SSJ97] uses this technique to reduce complexity); and solutions are centralized. Furthermore, those works don't take into account agents' autonomy (this aspect will be developed in chapter 2 on page 19).
Distributed Artificial Intelligence (DAI) uses concepts of theory game, but solutions are distributed. In this case, complexity, task allocation and communication are efficient. However, some underlying assumptions such as super-additive environment ([SK95a, ZR94, Kep94]) do not hold in real-world MAS.
By the contrary, MAS deals with interactions among self-motivated, rational and autonomous agents.

## 5.3 Alliances formation approaches

Many alliance formation approaches exist, but address different problems in different domains. Sarit Kraus has proposed a classification of works in alliance formation [Kra97].

We will extend this classification to emphasize our criteria.

### 5.3.1 Domain

The first set of criteria is about the domain. Distributed authority, communication and negotiation are always considered:

- Each agent has an individual goal [SK95a] *vs* A common goal has to be reached (social welfare maximizing) [SSJ97, SK95b, SK95b].
- Agents are self-interested (they act to satisfy them-selves) [SK95a, SK96a] *vs* Altruistic (they act to satisfy the system).
- Only pure software agents *vs* At least one human agent (the cognitive part is human since the communicative part is software in order to allow interaction with software agents).
- The rationality is known (it may be a group rationality [SSJ97], a personal rationality [SK95a, SK96a], or a coalitional rationality [SK95b]) *vs* no rationality is assumed.
- Bounded rationality [SL97, SL95] (in [SL97], bounded rational value of an alliance is determined by three parameters: the domain problem – task and resources, the execution architecture – limited and costly computation, and especially the possibility for agents to design their protocols) *vs* Unbounded rationality.
- Effects of computational limitations on alliance structure formation and stability have been studied in[SL95, SL97].
- Externalities occur where the actions of agents have an effect on agents other than themselves. They may be positive (they are benefit on other agents) [SK95a] *vs* negative (they are cost on other agents).
- Number of agents: a dozen [SSJ97], a hundred [Wel93], thousands.
- Size of solution space (number of possible alliances): too many to be enumerated and evaluated (in case of costly and/or limited time) [SK96a, SLA$^+$99] *vs* Small space.
- Defined protocols agreed (regulations should be agreed in advance and are incorporated into all of the agents [SK96a], since each agent chooses its own strategy [SK96a]) *vs* non pre-defined protocols.
- Static *vs* evolutionary evaluation of incomes: the evaluation of incomes may:

  never change, then the alliance struture is definitive.

  change only after an alliance structure has been found; after a change, the new alliance structure research may take the previous structure into account;

  change during the process of alliances formation; it is then difficult to reach a final state.

- The evaluation of incomes may be a common knowledge *vs* an individual knowledge.
- The computation time may be costly [SL95] (but limited) *vs* costless; it is linked to bounded rationality.
- The tasks may be independent (the income of two tasks may differ if it is carried by one or two agents) [SK95b] *vs* independent.
- The goal may be to satisfy as many tasks as possible [SK95b] *vs* to satisfy all tasks.
- Agents may have enough competences to carry out all tasks *vs* that requires to call some other agents.
- The dynamics of the system may be complete (agents appear/disappear and task may arrive constantly) *vs* partial (it is not modified during alliance formation process) [SSJ97, SK96b].
- The resources may be transferable between agents (agents may buy services to others) – what allow more beneficial alliance [SK95b, SK96a] *vs* not transferable.
- A monetary system for side-payment is assumed [SK96a] (it is a particular case of transferable resources).
- An agent may be a member of a lone alliance (set partitioning) [SK96a] *vs* or of several alliances (set covering) [SK98].
- CFG (the value of each alliance is independent of non-members' actions) [SK96a, SLA$^+$99, Kep94, SL97, ZR94] *vs* non-CFG.
- The value of an alliance may be superior to the sum of the value of its member (super-additivity) [Kep94, SK95a, ZR94], or it may be inferior (sub-additivity) [SL97, ZR94], or no assumption may be made on additivity (most of cases).
- The communication may have a bounded range of action [Leg03]. In this case, the alliances are also based on spatial positions, that may evolve quickly. Thus, the information about the organisational structure may be often partially false, what require dynamic reorganization.

Self-interested agents can deal with selfish goals (e-commerce) and altruistic agents can deal with common goals (problem solving in DPS), but there are other possibilities. Agents are altruistic if they are designed to collaborate, since a common goal is the aim of the system. Self-interest and altruism affect the design of agents, since individual and common goals concern the type of problem. The type of goal is given by the problem, since the type of agent is defined by environment (DPS, e-commerce), or resolution choice (DAI, MAS, ...). For example, a task allocation in DPS can be solved using altruistic agents since goals are individual. An another example: to solve a problem in MAS, agents have to try to satisfy a common goal even if they are self-interested. Generally, if the system's goal is to reach a common goal, benefits are measured from the system viewpoint.
Strong rationalities have been used to enable efficient protocols:

- Personal rationality [Har77, LR57, Rap70]: an agent will join an alliance only if the payoff he will receive is greater than what he can obtain by staying outside (eventually getting a payoff that compensates him for the loss of resources or non fulfillment of some of its tasks).
- Coalitionally rational [SK95a]: each alliance will add new members only when its new value is greater than the value of the original alliance.
- Group rationality [Har77, Rap70]: agents are group rational if forming an alliance always increases the global benefit.

In super-additive environment, grand alliance is optimal; thus the only problem that still remains is how the payoff should be distributed among its members.
In [SL97], bounded rational value of an alliance is determined by three parameters: 1) as usual, the domain problem (task and resources); 2) the execution architecture (limited and expensive computation) and especially 3) the possibility for agents to design their protocols. Effects of computational limitations on alliance structure formation and stability have been studied in [SL95] and [SL97]. The third parameter is ambitious but we think that it is a necessary condition to design autonomous systems (section 2.4 on page 32).

### 5.3.2 Quality of solutions

Solutions provided by these protocols are different since each protocol have its own properties. The following properties are important to choose the adapted protocol:

- Quality of solution (optimality) if a measure is available.
- Complexity in time and space to reach final state.
- Anytime, design-to-time [SL97].
- Certainty to reach a final state.
- Stability (studied in [Rap70]).
- Limited number of agents per alliance [SSJ97, SK98].

### 5.3.3 Comparison with our work

Several works in MAS were interested in dynamic organizations and in particular in alliances formation. The majority of them suppose *a priori* that all agents have the same rationality, which makes it possible to reduce the space of search considerably. The strongest assumption consists in supposing that agents are altruistic, which leads, as in [SK95b], to calculate in a distributed way the solution which maximizes the common utility.
Works with which we compare this paper relate to only egoistic agents, because considering altruistic agents radically changes the problem and thus the solutions. In [SK96b], the problem is simplified to compute in a distributed way the best solution for the system

as a whole. It's a difficult problem, because it is not easy to have a total sight of the system for any one of its components. For us, it is a question of managing to reconcile the inevitably conflict individual interests (if not it would not have there problem) to arrive, despite everything, with a solution which is accepted by all, but within the meaning of the legitimacy of its construction.
The problem of alliance formation consists in seeking a solution which is most satisfactory for the set of the agents. It is typically the case when one seeks to calculate the Shapley value (which corresponds roughly to the expectancy of the utility): one seeks to find the configuration which satisfies overall more the agents. Admittedly, the total interest of MAS is obvious but the imposed method keeps the personal freedom of the agents in check. Indeed, either the configuration is calculated in an external way, or it is calculated by the agents themselves, but in giving them a definite role which they must follow. It is the case for example in [ZR94] which proposes a distributed computation process of the Shapley value. The authors impose on the agents that the best solution is the one which maximizes the Shapley value. Admittedly, to guarantee a relative satisfaction of the agent, they take care not to force them to form an alliance which does not bring to them more than what they would have had before. But this deterministic process prevent the agents from developing their own strategies: if an agent agrees to take the risk to create an alliance with poor yield in the hope to form a very productive alliance later, that can be very beneficial for him.
In [Kep94], Steven Kepchel gave a little more freedom to the agents while allowing them to make different estimations of the income of an alliance. The main problem is the distribution of the benefits which are not known with precision at the instant of the formation. The proposed algorithm consists in a succession of aggregations of agents and alliances. What is interesting here is that the agents can have different hopes from the incomes and manage an agreement all the same. But, like much of others, this algorithm supposes that all is convertible into currency, that all can be defined by a real value, which has consequences on the algorithm itself, thus returning it very dependent on this assumption. This it the reason of the choice of the preferences exchanges which are independent of the criteria which produced them, allowing the agents to work out their own strategies in full freedom .
A similar problem has been studied by Kenneth Arrow in [Arr91]: the combination of individual preferences in a collective one. He shows the impossibility to find a vote procedure which respects his five intuitive axioms and which is not dictatorial. The algorithm we present here doesn't respect these axioms, but the procedure try to be the least dictatorial.
In [Akn01, APS02], the authors propound to use the integral of Choquet to aggregate the preferences. This operator has several advantages against previously ones. However, since no operator is perfect [Akn01, APS02], and none is absolutely legitimate. Moreover, their first method doesn't guaranted to convergence, but it is appliable to both collaborative and competitive agents (but their definition of competivity is a definition of weak competitivity). Their first method always converge, but is appliable only on cooperative agents.
The second criticism relates to the dependences between the alliances such as defined in [Kep94]: in an iterative formation, an agent cannot take into account the possible future participation of other agents ; indeed, the interest for an alliance depends on the set of

its members and not only on the increase in its value at the stage in progress. In our approach, all solutions being estimated overall and in parallel, the agents can fully estimate their interest for each solution.
Agents are free, which enables us to take into account naturally what often seems constraints: dependence of tasks and then of alliances. The protocol is based only on preferences, it does not take into account of the reasoning which generated them and consequently, each agent is free to have its own rationality. Indeed, a good rationality from a global point of view (of alliance, altruism) is not necessarily good individually and can discourage agents (all or some) to apply it. Thus, it is no more possible to use these presuppositions to calculate the best solution in a context of egoistic agents. In the same way, an individual strategy cannot be always optimal because it depends on the others ones.

## 5.4 Alliance formation context

The problem of task allocation binds agents to cooperate in order to fulfill tasks (each agent is able to fulfill sub-tasks). We assume that all tasks can and must be fulfilled. A task might be dependent on another (precedence order, income decrease, same/different agent for some sub-tasks) and the value of an alliance may depend on non-member actions: this may be taken into account by a modification of solution space and of sub-tasks incomes (but no experimentation have been made upon). Resources may be not transferable, but if they are, agents may exchange resources outside the protocol without modifying it.
A monetary system is used for experimentations to simplify computation, but since the protocol is only based on preferences exchange, it is not necessary (agents need only criteria to compute their preferences).
The solution space might not be too large; but if it is, each agent might use heuristics to quickly evaluate the best solutions.
The number of agents may be large (around 25), and experimentations show that the number of turns decreases when the number of agents increases (time however increases because each turn take more time than the one before).
Evaluations of incomes are individual and may evolve during the process. Computation and communication time might be taken into account, by decreasing sub-task income as time elapses, but strategies and experimentations don't take that into consideration.
Experimentations assume that agent may fulfill different sub-tasks in different alliances, but the protocol run with a partition of agents: solution space has simply to be reduced.
Considering the optimality of the chosen task allocation has no meaning here: it depends on agents' strategies. However, chosen solution is legitimate, because no agent is favored.
In this defined context, we propose a protocol that takes into account strong autonomy and weak rationality ([VE01] and see subsubsection 2.2.3 on page 26) to reach a consensus about a sub-task allocation.

## 5.5 Alliance formation protocol

Each agent likes some solutions and dislikes others. To reach a consensus, agents have to exchange information to possibly make their preferences evolve. Argumentation should be used, but it needs a complex process, it binds agents to have a common communication language and to know the rationality of others. Heterogeneous agents should prefer to exchange basic information that doesn't need such a formal process. Thus, at each turn, agents send their preferences to others and consider other's preferences to compute their next preferences. Because agents who don't make concessions are more likely to be ejected from the final solution (see chapter 7 on page 119), agents may be flexible. If they aren't, they may form coalitions; if no coalition is formed, agents choose two agents whom are obliged to ally. Finally, coalition formation leads to facilitate a consensus to be reached. This algorithm is more broadly borne out and described in [VE00, Vau00].

Alliance formation addresses two problems:

i alliance structure generation (partitioning or covering the agents);

ii dividing the value of the solution among agents.

We assume that the income is divided before the alliance structure generation.
Thus, how to form alliances among enterprises ? The problem is to reach a consensus among the allocation of tasks between them.
We chose the framework of tasks resolution by agents: a system receives a set of tasks divided in sub-tasks and proposes these sub-tasks to registered agents. They try to distribute the sub-tasks among themselves, each one being able to carry out only one part of it. A task is carried out when all its sub-tasks are carried out. To simplify, the decomposition is fixed, because of consequences on the incomes of the agents. We suppose moreover that there is no constraint on the sub-tasks order.
The objective of the proposed algorithm is to allow free agents having potentially incompatible interests to find a consensus on the distribution of the sub-tasks, but from their point of view.
Agent's motivations to carry out the sub-tasks are the associated profits, but the proposed protocol takes only opinions into consideration.

### 5.5.1 Case study

Let us now present the concepts of the alliance formation problem and highlight their meaning within an application: airlines choose to cooperate in order to provide their passengers with a unified reservation system. The problem is that for each travel, several airlines are in competition on some stages.

### 5.5.2 Formalization

**Definition 52 (Alliance Formation Problem (AFP)).** *A $AFP$ is defined as a tuple $\langle A, \mathcal{T}, \mathcal{S}, \mathcal{C}, \mathcal{P}\rangle$, where:*

- $A$*: the set of agents candidate to the execution of sub-tasks;*

- $\mathcal{T}$*: the set of tasks to be accomplished;*

- $\mathcal{S}$: *the set of sub-tasks to be carried out;*

- $\mathcal{C}$: *the set of competences necessary to perform the sub-tasks;*

- $\mathcal{P}$: *the set of incomes.*

*Where:*

- *An* **agent** $a \in A$ *is defined by:* $a = \langle C, strategy \rangle$, $C \subset \mathcal{C}$, *and strategies contain preferences computation and some criteria used to form alliances (see subsection 5.5.3 on the following page).*

- *A* **task** $t \in \mathcal{T}$ *is defined by the set of sub-tasks it contains:* $t = \langle S \rangle, S \subset \mathcal{S}$.

- *A* **sub-task** $s \in \mathcal{S}$ *is defined by* $s = \langle C, p \rangle, C \subset \mathcal{C}, p \in \mathcal{P}$, *where* $p$ *is the set of competences which an agent must have to be able to carry out the sub-task, and* $p$ *the associated profit. This profit will be used by agents to compute their preferences.*

- *A* **competence** $c \in \mathcal{C}$ *is a single item which represents what is required for a sub-task to be carried out by an agent. A sub-task may require one or more competence.*

- *A* **profit** $p \in \mathcal{P}$ *is used as an income, but only to simplify the internal computations of agents:* $\mathcal{P} = [0, MaxProfit]$, $MaxProfit \in \mathbb{R}$. *However, several types of decision making should be used to compute preferences.*

**Example 7.** *In our example, an agent is an airline.*
*Let* $A = \{EUropeanAirlines, AMericanAirlines, WOrldAirlines, \ldots\}$.
*A task is a flight between two cities which puts into others:*
$\mathcal{T} = \{$*New York-MAdrid (via PAris and LYon), Los Angeles-MOscow (via New York and PAris) and BErlin-JOhannesburg (via PAris)*$\}$.
*The set of sub-task is:*
$\mathcal{S} = \{$*New_York*$\rightarrow$ *PAris, LYon*$\rightarrow$ *MAdrid, PAris*$\rightarrow$ *MOscow,* $\ldots\}$. *We can now define the task* $NY - MA$ *by* $NY - MA = \langle \{$*NY*$\rightarrow$ *PA, PA*$\rightarrow$ *LY, LY*$\rightarrow$ *MA*$\}, \ldots \rangle$.
*A flight:*

- *requires authorization to do a national stage* $autXY$, *where* $XY$ *is the symbol of a state;*

- *has a passengers capacity:* $\{WeigthCapacity, MiddleCapacity, LightCapacity\}$;

- *has a range of action:* $\{VerySmallRange, SmallRange, MiddleRange, LongRange\}$;

- *provides incomes:* $\mathcal{P}$=$[0, 10000]$ *and* $NY - M$=$\langle\{$*NY*$\rightarrow$ *P,P*$\rightarrow$ *L,L*$\rightarrow$ *M*$\}$*,8000*$\rangle$.

*The set of competences is thus:*
$\mathcal{C} = \{autXY, WC, MC, LC, VSR, SR, MR, LR\}$.
*Hence, the agent* $EUA$ *is defined by:*
$EUA = \langle \{autFR, autEU, autRU, WC, MC, SR, MR\} \rangle$.
*We choose* $\mathcal{P} = [0, 10000]$.
*For example, the sub-task* $NY - MA$ *is defined by:*
$NY - MA = \langle \{NY \rightarrow$ *PA, PA*$\rightarrow$ *LY, LY*$\rightarrow$ *MA*$\}, 8000 \rangle$.

To solve this problem, agents exchange their preferences about possible solutions. If no consensus is reached, they may form a coalition. So, agents need to represent solutions, preferences, alliances and coalitions.
A solution is an assignment of each sub-task to an agent which is able to perform it.

**Definition 53.** *We solve an $AFP$ as a $CRP = \langle A, \mathcal{S} \rangle$ (Consensus Reaching Problem):*

- $CRP.A = AFP.A$;
- $CRP.\mathcal{S} = \{AFP.A\}$;

**Example 8 (opinion).** *The solution $\sigma_{15} = [$NY$\rightarrow$ PA$_2 \hookrightarrow$WOA, LY$\rightarrow$ MA$\hookrightarrow$BUA, PA$\rightarrow$ MO$\hookrightarrow$EUA,$\ldots]$.*
*Let $\overline{\Sigma}_1 = \{\sigma_0, \sigma_2, \sigma_4\}$ the set of solutions which provide outcomes and $\overline{\Sigma}_2 = \{\sigma_1, \sigma_3, \sigma_5\}$ the set of solutions which provide none. $\delta(\sigma, \sigma') = 0$ if $\sigma$ and $\sigma'$ are in the same set, and $\delta(\sigma, \sigma') = 1$ otherwise.*

**Definition 54 (Alliance).** *An alliance $\Omega(\sigma, t) \subset A$ associated to the task $t \in \mathcal{T}$ in the solution $\sigma \in \Sigma$ is defined by: $\Omega(\sigma, t) = \{a \in A / \exists s \in \mathcal{S}, s \in t.S, \sigma(s) \ni a\} = \bigcup_{s \in t.S} \sigma(s)$. An alliance contains all the agents which take part in a task.*

**Definition 55 (Initial Preference Computation).** *An $IPC$ is an element $\delta \in \Delta$, that should be computed using incomes or another criterion.*

**Definition 56 (Dependent Preference Computation).** *A $DPC$ is a function $H \rightarrow \Delta$, $h \mapsto \delta$.*

**Example 9.** *Let $\delta = IPC$, $\forall(\sigma_1, \sigma_2) \in \Sigma^2)$, $\delta(\sigma_1, \sigma_2) = profit(\sigma_2) - profit(\sigma_1)$. $\delta$ is an antisymmetrical application.*
*Let $\delta = DPC(h)$, $h = (v_t)_{t \in \mathbb{N}}$. $\forall(\sigma_1, \sigma_2) \in \Sigma^2$, $\delta(\sigma_1, \sigma_2) = [\sum_{a \in A}(v_T(a))(\sigma_1, \sigma_2)]/|A|$. $\delta$ is an antisymmetrical application.*

### 5.5.3 Agent's Strategies

#### Member's strategy

The strategy depends on Independent Positions Computation $IPC$ and Dependent Positions Computation $DPC$. Several strategies will be experimented in chapter 7 on page 119.

#### Representative's strategy

Representative agent has particular procedures that define the coalition's strategy (see definition 2 on page 51 for more details):

- Releasing Switch-over Proposal Criterion $RSPC$.
- Releasing Switch-over Acceptance Criterion $RSAC$.
- Coalition Merging Proposal Criterion $CMPC$.

- Coalition Merging Acceptance Criterion $CMAC$.

Example of criteria:

**Example 10.** *Let* $h = (v_t)_{1 \le t \le T}$.
*Let us define* $RSPC$*:*

- $RSPC(h) = False$ *if* $T \le 2$*;*
- $RSPC(h) = (v_T = v_{T-1}) \vee (v_{T-1} = v_{T-2}) \vee (v_T = v_{T-2})$ *otherwise.*

*To reduce computation complexity, only loops of length* $3$ *or less are detected and to simplify computations,* $RSAC = RSPC$.
*Let* $d : \Delta \times \Delta \to \mathbb{R}$ *a distance between agents preferences, for example:* $\forall(\delta_1, \delta_2) \in \Delta^2$, $d(\delta_1, \delta_2) = \sum_{(\sigma_1,\sigma_2)\in\Sigma^2} |\delta_1(\sigma_1, \sigma_2) - \delta_2(\sigma_1, \sigma_2)|$*. For an agent* $a$, $CMPC(h)$ *is the set of agents which preferences are similar enough to him using a threshold.*
*We can use the same application to compute* $CMAC$ *but using a greater threshold.*

## 5.6 Conclusion

Usually, alliance formation allows agents to dynamically make their organizational structure evolve according to environment changes. Many researchers have already proposed protocols, which criteria depends on the context.
However, nobody have already propose a protocol that deal with totally autonomous agents as defined in chapter 2 on page 19, what add hard contrains on the design of them. In this context, a protocol of alliance formation is based on a protocol of consensus reaching, what allows agents to autonomously choose their partners of interaction.
As in [caCP96], it would be interesting to make it possible for the agents to keep a track of alliances which were beneficial, in order to accelerate convergence towards a consensus. Building models of the others made possible agents to be in confidence with others and then to limit suspicion. However, trust may be used only in certain contexts: no anonymity, long-time interactions.
Alliance formation is regarded as a new step of autonomy, allowing agents to choose their organizational structure. A plateform (chapter 6 on page 111) has been developped in order to make experiments (chapter 7 on page 119) on this protocol.

# Bibliography


[Akn01] *Systèmes Multi-Agents, Ingénierie, Technologie, Expérimentations*, chapter Méthodes de Coordination multi-agent par formation de coalitions, pages 149–161. Hermès, Paris, 2001.

[APS02] Samir Aknine, Suzanne Pinson, and Melvin Shakun. Modèles de formation de coalitions multi-agents. In *Reconnaissance des Formes et Intelligence Artificielle*, Angers, January 2002. RFIA'2001.

[Arr91] Kenneth Arrow. *The Origins of the Impossibility Theorem*, chapter 1, pages 1–4. Elsevier Science Publishers B. V., Amsterdam, 1991.

[BP72] E. Bala and M. Padberg. On the set covering problem : an algorithm for set partitioning. In *Operation Research*, volume 20, pages 1152–1161. 1972.

[caCP96] Jorge Anacleto Louçã, Helder Coelho, and Suzanne Pinson. Forming coalition in task oriented multi-agent systems. In *Proceedings Of IBERAMIA 1996*. IBERAMIA, 1996.

[CK75] N. Christofides and S. Korman. A computational survey of methods for the set covering problem. In *Mathematics of Operations Research*, volume 21 (5), pages 591–599. 1975.

[Dub38] H. Dubreuil. *La Fin des Monstres, Idée d'une Organisation Contraire à la Centralisation et à l'Étatisme*. Grasset, 1938.

[Fol24] M .P. Follet. *Dynamic Administration*. Pitman, 1924.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability : a Guide to the Theory of NP-Completeness*. W. H. Freedman and Company, New York, 1979.

[GN] R. S. Garfinkel and G. L. Nemhouser. The set-partitioning problem : Set covering with equality constraints. In *Operation Research*, volume 17, pages 848–856.

[Har77] J. C. Harsanyi. *Rational Behavior and Bargaining Equilibrium in Games and Social Situations*. Cambridge University Press, 1977.

[Kep94] S. P. Kepchel. Forming coalitions in the face of uncertain rewards. In *Proc. of AAAI94*, pages 414–419, Seattle, Washington, 1994.

[KR84] J. P. Kahan and A. Rapoport. *Theories of Coalition Formation*. Lawrence Erlbaum Associates Publishers, Hillsdale, NJ, 1984, 1984.

[Kra97] S. Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94(1-2):79–98, 1997.

[KW91] S. Kraus and J. Wilkenfeld. Negotiations over time in a multi agent environment. In *Proc. of IJCAI-91*, pages 56–61, Australia, 1991.

[Leg03] François Legras. *Organisation dynamique d'équipes d'engins autonomes par écoute flottante*. PhD thesis, SUPAERO, Toulouse, November 2003.

[LR57] R. D. Luce and H. Raiffa. *Games and Decision*. John Wiley and Sons, Inc, 1957.

[Min81] H. Mintzberg. *The structuring of organizations : a synthesis of the research*. 1981.

[NM47] J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, N. J., 1947.

[Rap70] A. Rapoport. N-person game theory. Technical report, University of Michigan, 1970.

[SK95a] O. Shehory and S. Kraus. Coalition formation among autonomous agents: Strategies and complexity. *Lecture Notes in Artificial Intelligence*, From Reaction to Cognition, C. Castelfranchi and J. P. Muller (Eds.)(957), 1995.

[SK95b] O. Shehory and S. Kraus. Task allocation via coalition formation among autonomous agents. In *Proc. of IJCAI-95*, pages 655–661, Montreal, August 1995.

[SK96a] O. Shehory and S. Kraus. A kernel-oriented model for autonomous-agent coalition-formation in general environments : Implementation and results. In *Proc. of AAAI-96*, pages 134–140, Portland, Oregon, August 1996.

[SK96b] O. Shehory and S. Krauss. Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. In *Proc. of ICMAS-96*, pages 330–337, Kyoto, Japan, 1996.

[SK98] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, 1998.

[SL95] T. W. Sandholm and V. R. Lesser. Coalition formation among bounded rational agents. In *Proc. of IJCAI-95*, pages 662–669, Montréal, 1995.

[SL97] T. Sandholm and V. Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, Special issue on Economic Principles of Multi-agent Systems(94(1)):99–137, 1997.

[SLA+99] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohme. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 1999.

[Smi80] R. G. Smith. The contract net protocol : High-level communication and control in a distributed problem solver. In *IEEE Transaction on Computers*, volume 29 (12), pages 1104–1113, 1980.

[SSJ97] O. M. Shehory, K. Sycara, and S. Jha. Multi-agent coordination through coalition formation. In M. Singh A. Rao and M. Wooldridge, editors, *Lecture Notes in Artificial Intelligence*, volume Intelligent Agents IV - 1365, pages 143–154. Springer, 1997.

[Tay11] F. W. Taylor. *Principles of scientific management*. 1911.

[Vau00] Guillaume Vauvert. Formation de coalitions pour agents rationnels. In *Proc. de JLIPN'2000*, Villetaneuse, France, September 11-12 2000. VIIIèmes Journées du L.I.P.N. - Systèmes Multi-Agents & Spécifications Formelles et Technologies Logicielles.

[VE00] G. Vauvert and A. El Fallah - Seghrouchni. Coalition formation among egoistic agents. In *Proc. of MAMA'2000*, Wollongong, Australia, December 11-13 2000. International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce. Awarded "One of the three best papers" (among 92 accepted papers).

[VE01] G. Vauvert and A. El Fallah - Seghrouchni. Coalition formation among strongly autonomous and weakly rational agents. In *Proc. of MAAMAW'2001*, Annecy, France, May, 2-4 2001. Modelling Autonomous Agents in a Multi-Agent World. 10th European Workshop on Multi-Agent Systems. Acceptation rate of papers : 13/73=18%.

[Wel93] M. P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1:1–23, 1993.

[ZR94] G. Zlotkin and J. S. Rosenschein. Coalition, cryptography, and stability: Mechanisms for coalition formation in task oriented domains. In *Proc. of AAAI94*, pages 432–437, Washington, 1994.

# Chapter 6

# Implementation

## 6.1 Introduction

### 6.1.1 Platform specification

We need a platform to do experiments on protocols [VE00]. From our point of view, a protocol is the specification of a structured exchange of messages. A message may be allowed (other agents are waiting it) or mandatory (the agent is authorized to send it). It may depend on the previously exchanged messages (many states are reachable) and on the sender's roles (each role own its proper messages).
We need a platform with the following properties:

- protocols may be quickly implemented, modified and tested;
- agent's identities and roles may be taken into account to send and receive messages;
- agents may reason and act (send messages) concurrently;
- messages may be exchanged asynchronously.

### 6.1.2 Platform global view

#### Petri Nets

Petri Nets [Pet62] has been introduced as a mathematical tool for modeling distributed systems and notions of concurrency, non-determinism, communication and synchronization. A simple (black and white) Petri Net is a digraph with nodes that are places (circles) or transitions (rectangles). Nodes of different kind are connected together by means of arcs. Arcs are of two kind:

- input arcs that connect one place to one transition;
- output arcs that connect one transition to one place.

A Petri Net can be initialized by indicating the tokens which are contained in each place at starting time. At any time the distribution of tokens among places defines the current state of the modeled system. A Petri Net can be executed by: A transition is said to be

Table 6.1: Execution of a Petri Net

establishing an initial marking
**repeat**
  choosing a set of eligible transitions
  firing a transition among the set of eligible ones
**until** no more transition is eligible

eligible if all its input places contain (at least) one token. Then if it does fire, one token is removed from each of its input places and one token is added to each of its output places.

**Description**

In our case, a protocol is set an interaction among several roles; each of them may be is described by a set of expected messages sending and receptions. Petri Nets are useful to modelize expected behaviors, because there is a separation between the actions (firings of transitions), the internal (transition firings) and external events (messages) that allow to fire new transitions (places), and the structure of the succession of actions (connections by means of arcs).
In our tests, we assume that agents abide by the protocol; so their behavior is the expected one. A role may then be modeled as a Petri Net. Transitions model the actions (reasoning and message sendings); arcs model the succession of actions; places model either the fact that transitions have already been fired (internal event), either the reception of a message (external event). The agent is responsible of the message sending during a transition firing.
The figure 6.1 on the facing page shows the global architecture of the platform. All concepts are explained in section 6.2.

## 6.2 Concepts of the system

### 6.2.1 The envelope

Heterogeneous agents are not able to understand languages. In fact, a message contains a unique identifier (it can be a number at system level or a string to facilitate human comprehension), and eventually a data.

```
message
  - header
  - data (optional)
```

In order to be correctly transmitted, a message must be linked with an address, that is to said, the agent's identifier (his name, a number) and a role identifier.

```
address
  - agent
```

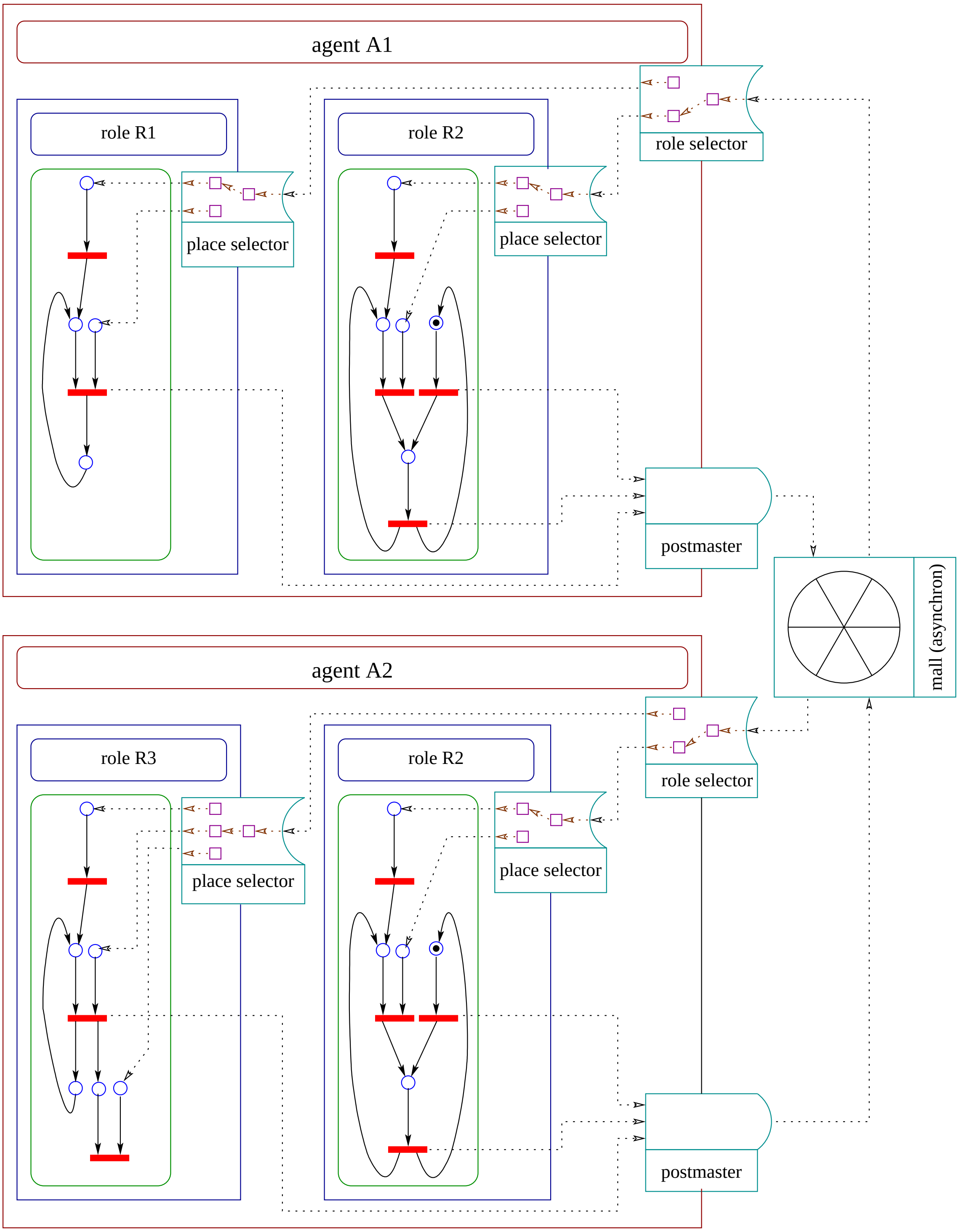


Figure 6.1: Global architecture

- `role`

The entity that is transmitted is called an envelope. It is composed by an address and a message.

**`envelope`**

- `address`
- `message`

### 6.2.2 The agent

The dynamics of each role is modelled by a Petri Net. A Petri Net is a oriented digraph; it contains:

- places that may contain a token;
- transitions that may fire;
- arcs that rely places to transitions or transitions to places.

The rules of activations are the same as usual:

- a place is active if:

  it contains an item;

  all transitions that precede the place have fired.

- a transition fires when

  all places that precede the transition are active.

A new rule has been added: a place may be activated by a message. This kind of place is characterized by a set of agents, a role identifier and a header. The place is active if all the agents of the set with the expected role have sent a message with the expected header. Data (optional) contained in messages are transmitted to the procedure launched when the transition fires.

**`message place`**

- `set of expected agents`
- `set of received agents`
- `role`
- `header`

**`Petri Net`**

- `set of places`

  `standard place`

  `message place`

- `set of transitions`

A component called `place selector` updates the set of received agents of all `message places`.

**`role`**

- `Petri Net`
- `place selector`

An agent may have several roles; so, he needs a component that switches envelopes depending on the role of the receiver. This component is called the `role selector`: it dispatches the envelope to the right `place selector`. Moreover, a component called `postman` sends envelopes to other agents.

**`agent`**

- `set of roles`
- `role selector`
- `postman`

### 6.2.3 System

Message go into agents by the role selector, and go outside by the postman. Outside agents, messages are send to and received from the `central mall`:

- it dispatches envelopes to the right recipient (the role selector of the recipient);
- it mixes envelopes in order to simulate the asynchronicity of real systems.

### 6.2.4 Travel of a message

When a transition fires, it calls a procedure. This procedure may contain an envelope sending.

1. the envelope is sent to the `postman`;
2. it transmits it to the `mall`;
3. the `mall` reads the recipient's name and send it to the right `role selector`;
4. this last one reads the role's name and send the envelope to the right `place selector`;
5. the `place selector` knows which `message place` is waiting which header, sender's name and sender's role: if the `message place`'s characteristic matches with the envelope, the agent's name is added to the received recipient and possibly the data;
6. if the `set of awaited agents` equals the `set of received agents`, then the place becomes active;

Eventually, a transition may fire and the process continue until no more message is traveling and no more transition is firing.

## 6.3 Software description

This platform requires a language that make the concurrency useful. We have chosen the Java language because it allows to simulate it simply by the means of threads. This platform has to evolve in order to execute agents on several computers wired by an ethernet. Thus, we decide to:

- not use shared variables: all exchange of information between two agents uses the exchange of message;

- all the communication is encapsulated in an extensible way: we can easily use protocols of communication (TCP/IP, RPC) instead of the `mall` local sending messages.

In this way, we will be able to make experiments on a real network.

Features of the Java object oriented language are very useful: heritage to write code one time, Finally, $54$ classes have been written (145Ko of source code, 5400 words, 13000 lines) for agents and opinions, plus $28$ classes for Petri Nets (32Ko of source code, 3300 words and 1500 lines).

## 6.4 Conclusion

The platform we have developed allows to quickly implement and test protocols. An agents may play several roles that are expected sequences of messages sendings and receptions. A role is modelled and implemented by a Petri Net that interacts with others using `message places` and with agent decision making process using transitions firings.
This platform has been designed in order to be easily extended to support network protocols.
It has been used to do experiments of the next chapter.

# Bibliography


[Pet62] Carl Adam Petri. *Communication with Automata*. PhD thesis, University of Bonn, 1962.

[VE00] Guillaume Vauvert and Amal El Fallah - Seghrouchni. Démonstration : Formation de coalitions pour des agents libres ; application à la gestion de transport aérien. In *Proc. of JFIAD-SMA'2000*, Saint-Jean-La-Vêtre, France, October 2-4 2000. 8èmes Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents.

# Chapter 7

# Experiments

## 7.1 Introduction

We have proposed a protocol of alliance formation among totally autonomous agents. Results depend mainly on the agents' strategies. It does not mean to measure the quality of the result.
However, we have seen (section 3.5 on page 60) that a protocol must discourage disruption. In our case, if agents are extremely rigid (*i.e.* if they refuse to change their opinion), the system forces agents to form coalitions, what leads to a non-legitimate solution. In order to get avoid such solutions, these strategies must be discouraged.
We are going to make experiments on several strategies to know if rigid behaviors are interesting. More generally, we have to find efficient strategies.
In order to test our formalism and operators, we have built a family of strategies and organized a tournament between these strategies.

## 7.2 Protocol of experiment

### 7.2.1 Description of the problem

The problem chosen to test our strategies is an allocation of tasks in an e-commerce context (see [VE01a, VE01b]). Some sub-tasks have to be allocated to several agents who are not able to fulfill all tasks because they have limited skills (no agent meets all the requirements of a task).

### 7.2.2 Steps allocation

In our study, several airlines have to allocate steps of flights among themselves. Each step is identified by the cities between which it takes place, and requires capacities to be carried out. Each airline owns several capacities that allow him to carry some tasks.
Abbreviations of cities and capacities are given table 7.1 on the next page. Flights (the tasks) and steps (sub-tasks) are described table 7.2 on page 121. Capabilities of airlines

(the agents) are described table 7.3 on the next page.
The $7$ agents have to allocate $8$ sub-tasks among themselves. They are able to carry only between $1$ and $3$ tasks (see table 7.4 on the facing page), what leads to $32$ available solutions.

| Abr. | Capacity |
|---|---|
| VShtRg | Very Short Range |
| ShtRg | Short Range |
| MidRg | Middle Range |
| LgRg | Long Range |
| LghCap | Light Capacity |
| MidCap | Middle Capacity |
| LrgCapa | Large Capacity |
| AutFR | Auth. France |
| AutUS | Auth. USA |
| AutEU | Auth. Europa |
| AutAF | Auth. Africa |
| AutRU | Auth. Russia |
| AutIC | Auth. Intercontinental |

| Abr. | City |
|---|---|
| NY | New York |
| MA | Madrid |
| PA | Paris |
| LY | Lyon |
| LA | Los Angeles |
| MO | Moscow |
| JO | Johannesburg |
| BE | Berlin |

Table 7.1: Symbols means

$CapacitySet = \{VShtRg, ShtRg, MidRg, LgRg, LghCap, MidCap, LrgCapa, AutFR, AutUS, Aut$
The table 7.5 on page 122 gives an example of a reached solution.

### 7.2.3 Computation of opinions

Each agent chooses to take the others' opinions into account with a more or less great weight. Previous experiments show that, at the beginning, it is in their interest to be rigid (*i.e.* do not take others' opinions into account), because that may influence the opinions of the others on a long term. On contrary, at the end, they should better be flexible in order to have chance to be assigned a task.
Then, at what speed do agents decrease their rigidity ? We define a strategy as a speed of decreasing. Formally, the rigidity $r$ is defined by: $\forall a \in A$, $\forall \alpha \in [0,1]$, $\forall t \in [1,T]$,$r(t) = \exp^{-\alpha t}$. The agent computes his new opinion as follows:

$$
\begin{array}{l}
NY \rightsquigarrow MA \quad = \langle NY \to 1PA, PA \to LY, LY \to MA \rangle \\
\left|\begin{array}{rl}
NY \to 1PA = \langle & income = 585, \\
 & capa. = \{AutIC, MidRg, LrgCapa\}\rangle \\
PA \to LY = \langle & income = 39, \\
 & capa. = \{AutFR, VShtRg, MidCap\}\rangle \\
LY \to MA = \langle & income = 91, \\
 & capa. = \{AutEU, ShtRg, LghCap\}\rangle
\end{array}\right. \\
LA \rightsquigarrow MO \quad = \langle LA \to NY, NY \to 2PA, PA \to MO \rangle \\
\left|\begin{array}{rl}
LA \to NY = \langle & income = 396, \\
 & capa. = \{AutUS, ShtRg, LrgCapa\}\rangle \\
NY \to 2PA = \langle & income = 585, \\
 & capa. = \{AutIC, MidRg, LrgCapa\}\rangle \\
PA \to MO = \langle & income = 248, \\
 & capa. = \{AutEU, AutRU, ShtRg, MidCap\}\rangle
\end{array}\right. \\
BE \rightsquigarrow JO \quad = \langle BE \to PA, PA \to JO \rangle \\
\left|\begin{array}{rl}
BE \to PA = \langle & income = 87, \\
 & capa. = \{AutEU, MidCap, ShtRg\}\rangle \\
PA \to JO = \langle & income = 868, \\
 & capa. = \{AutAF, MidCap, LgRg\}\rangle
\end{array}\right.
\end{array}
$$

Table 7.2: Flights Tasks

$$
\begin{array}{rl}
EuropeanAirlines = & \{AutFR, AutEU, AutRU, MidCap, LrgCapa, ShtRg, MidRg\} \\
InternationalAirlines = & \{AutIC, AutAF, MidRg, LgRg, LrgCapa, MidCap\} \\
AmericanAirlines = & \{AutUS, AutIC, AutEU, ShtRg, MidRg, LgRg, LrgCapa\} \\
AfricaAirlines = & \{AutAF, ShtRg, MidRg, LgRg, MidCap, LrgCapa\} \\
USAirlines = & \{AutUS, VShtRg, ShtRg, LghCap, MidCap, LrgCapa\} \\
FranceAirlines = & \{AutFR, AutEU, VShtRg, ShtRg, MidCap\} \\
BusinessAirlines = & \{AutEU, AutAF, VShtRg, ShtRg, LghCap\}
\end{array}
$$

Table 7.3: Airlines agents

$$
\begin{array}{rl}
NY \to 1PA \mapsto & \{InternationalAirlines, AmericanAirlines\} \\
NY \to 2PA \mapsto & \{InternationalAirlines, AmericanAirlines\} \\
PA \to JO \mapsto & \{AfricaAirlines, InternationalAirlines\} \\
LY \to MA \mapsto & \{BusinessAirlines\} \\
PA \to MO \mapsto & \{EuropeanAirlines\} \\
BE \to PA \mapsto & \{EuropeanAirlines, FranceAirlines\} \\
LA \to NY \mapsto & \{AmericanAirlines, USAirlines\} \\
PA \to LY \mapsto & \{FranceAirlines\}
\end{array}
$$

Table 7.4: Possible assignements

$$
\begin{array}{rl}
NY \rightarrow 1PA \mapsto & \{InternationalAirlines\} \\
NY \rightarrow 2PA \mapsto & \{AmericanAirlines\} \\
PA \rightarrow JO \mapsto & \{AfricaAirlines\} \\
LY \rightarrow MA \mapsto & \{BusinessAirlines\} \\
PA \rightarrow MO \mapsto & \{EuropeanAirlines\} \\
BE \rightarrow PA \mapsto & \{EuropeanAirlines\} \\
LA \rightarrow NY \mapsto & \{USAirlines\} \\
PA \rightarrow LY \mapsto & \{FranceAirlines\}
\end{array}
$$

Table 7.5: Reached solution

1. first, he aggregates the opinions of other agents: $\omega_m = \amalg(\{\omega'_b / b \in A \backslash \{a\}\})$;

2. then he applies a weighted aggregation to aggregate his preferences weighted by $r$ and other agents' preferences weighted by $1 - r$; as result, the strategy is defined by:
$s^a_\alpha(t/10) = \tilde{\amalg}(< \omega_a, r >, < \omega_m, 1 - r >)$.

### 7.2.4 Tournament

In our tournament, one agent has a strategy $\alpha$ since all others have a strategy $\beta$. For each fight, we measure the ratio of income (in comparison with the agent's maximal income) for the agent using the strategy $\alpha$. All agents play the role of the lone agent with strategy $\alpha$. Then, we compute the average of income.

## 7.3 Results

Many parameters influence the process, but three of them more: agents' strategies, skills repartition (more or less competition) and the number of agents.
On the two first figures (figure 7.1 on the facing page and figure 7.2 on page 124), the results are presented as follows : the strategy $\alpha$ takes place on the X-axis, and the mean of percentages of income (for all agents that used the strategy $\alpha$) on Y-axis. Each curve represents the set of results for a fixed value of $\beta$ ($\beta \in [0.0, 0.2, \ldots, 1.0]$ represents strategies of other agents).

To measure the influence of the first parameter, the number of agents is fixed (7). The goal of this experiments is to find the best average strategy according to other strategies. In figure 7.1 on the facing page, only the strategy of a particular agent varies : $\alpha$ varies from $0.0$ (flexible strategy) to $1.0$ (rigid strategy) by step $0.1$, since all other agents use the same value $\alpha = 1.0$. Results are the average of a large amount of experimentations ($350$). As expected, agent's income begin to increase when agent's strategy become more and more rigid. But, around $0.7$, agent's income decreases: to be too rigid should lead an agent to be excluded from chosen solution, he will so earn less income. The figure 7.2 on page 124 exhibits that this result is true for all other agents' strategies, since optimal value

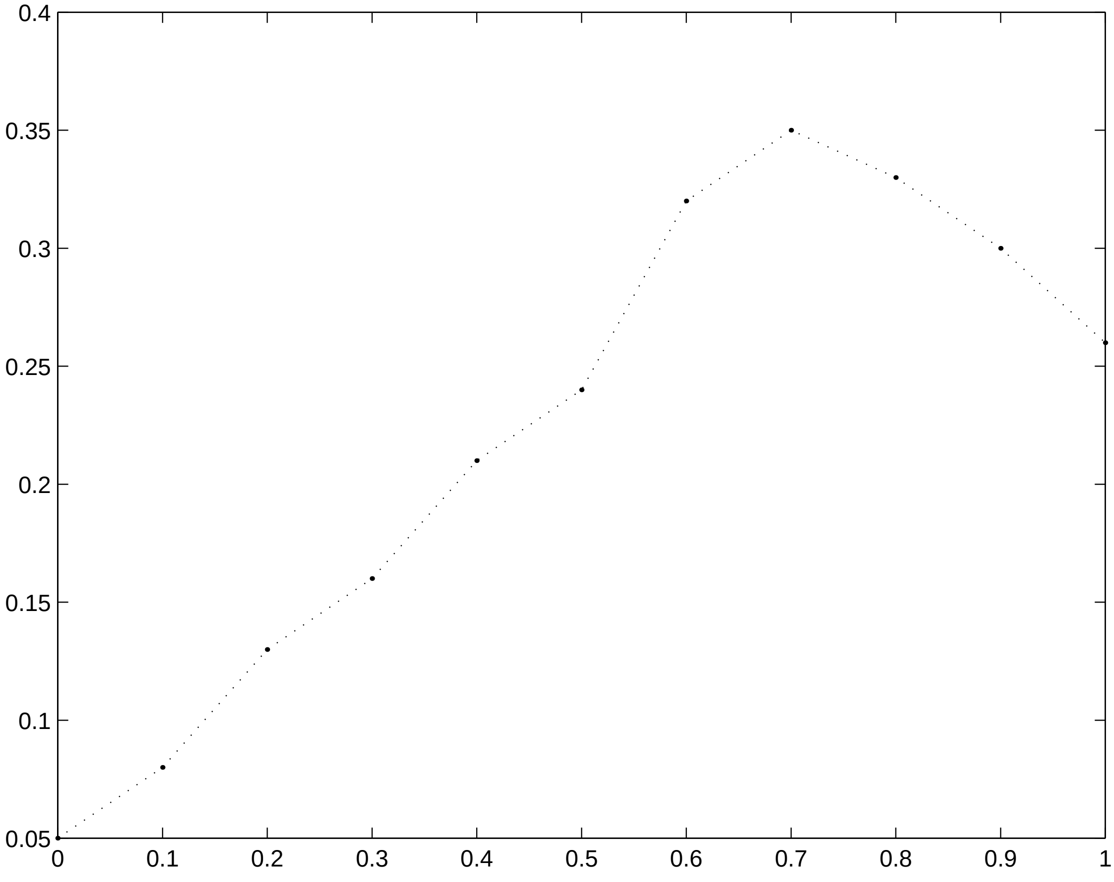


Figure 7.1: Income function of strategy for one global strategy

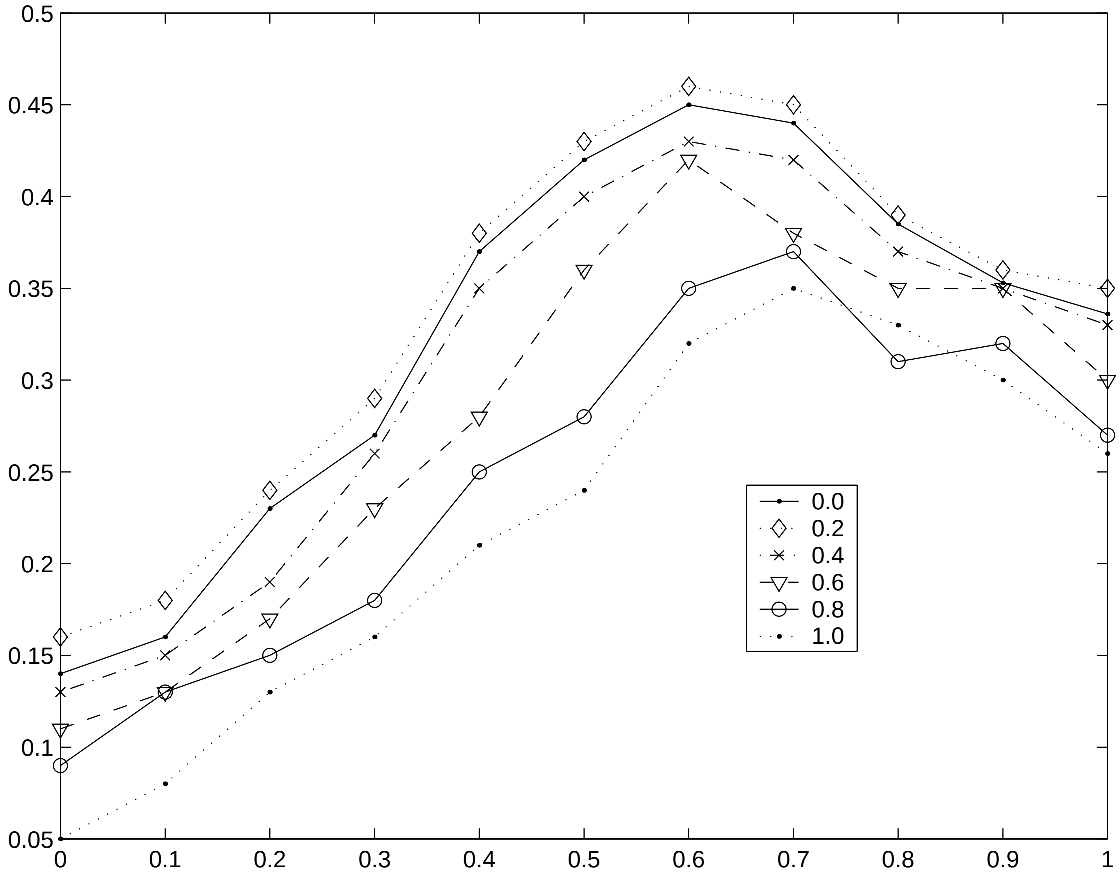


Figure 7.2: Income function of strategy for $6$ strategies

is around $0.6$. That should lead agent to choose flexible strategies.
The figure 7.3 shows that when more agents are rigid, consensus is hardly reached. If

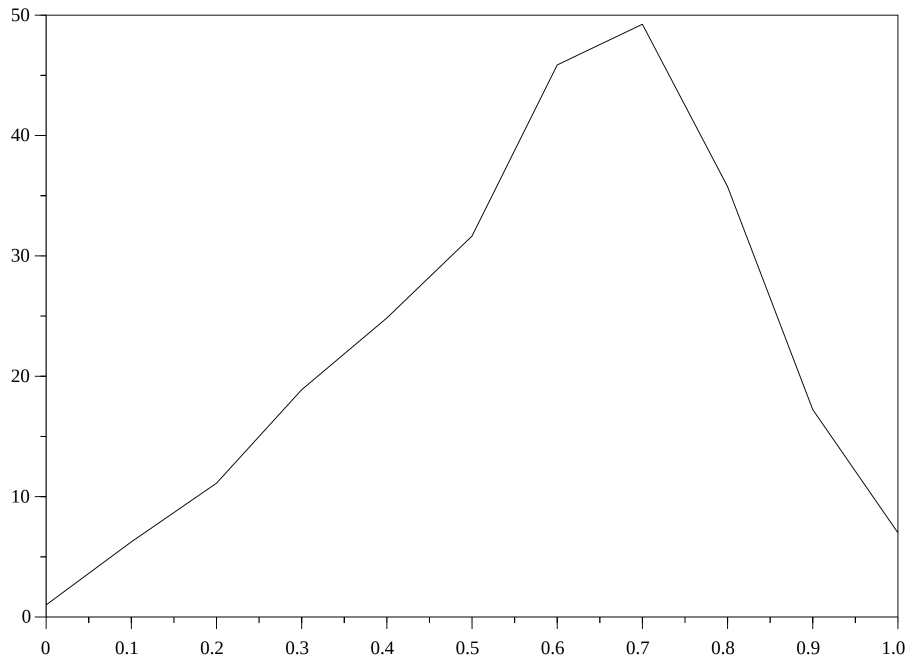


Figure 7.3: Number of turn in function of strategy

agents are too rigid, jamming detection leads to form a coalition and consequently to reach a consensus more quickly, even if the one is not desired.

The more the agents have competences, the more they have to compete with others. We studied the influence of the number of agents per sub-task (competition level) on the incomes (figure 7.4 on the next page) and on the number of turns (figure 7.5 on the following page). As expected, when competition increases, incomes decrease and consensus become more difficult to reach.
As the number of agents increases (figure 7.6 on page 127), there are more and more agents able to fulfill sub-tasks and competition increases. But if the number of agents is greater than $25$ (this value depends on other parameters), then reaching a consensus is easier, because the formed alliances contains enough agents to fulfill all the tasks: usually, a single alliance fulfill all tasks.

## 7.4 Conclusion

From the point of view of deployment of MAS in an economic context, it is necessary to consider weakly rational, strongly autonomous and heterogeneous agents. To manage to form alliances within this framework, we propounded an open, distributed and egalitarian

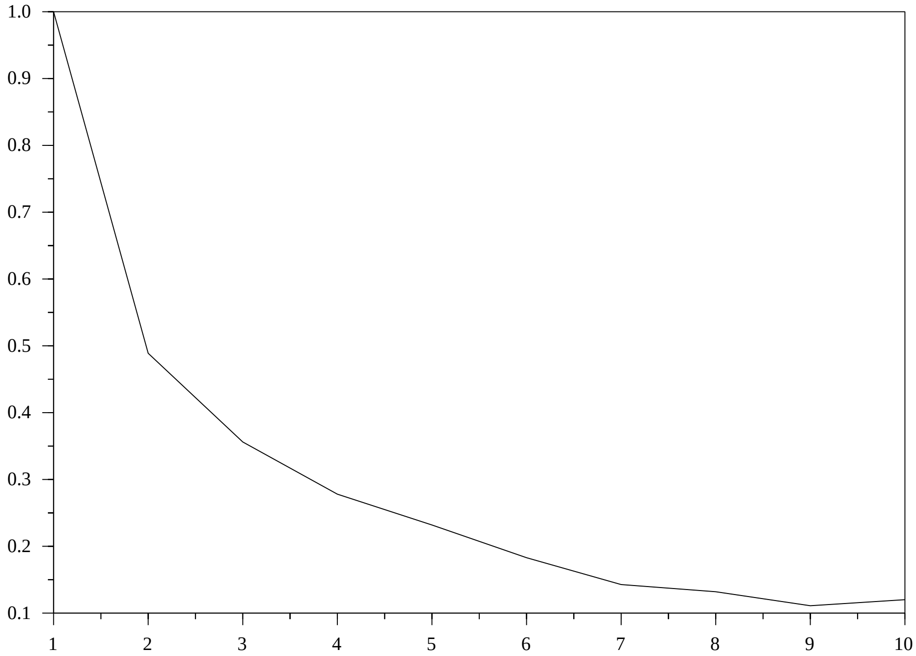


Figure 7.4: Income in function of number of agents per sub-task

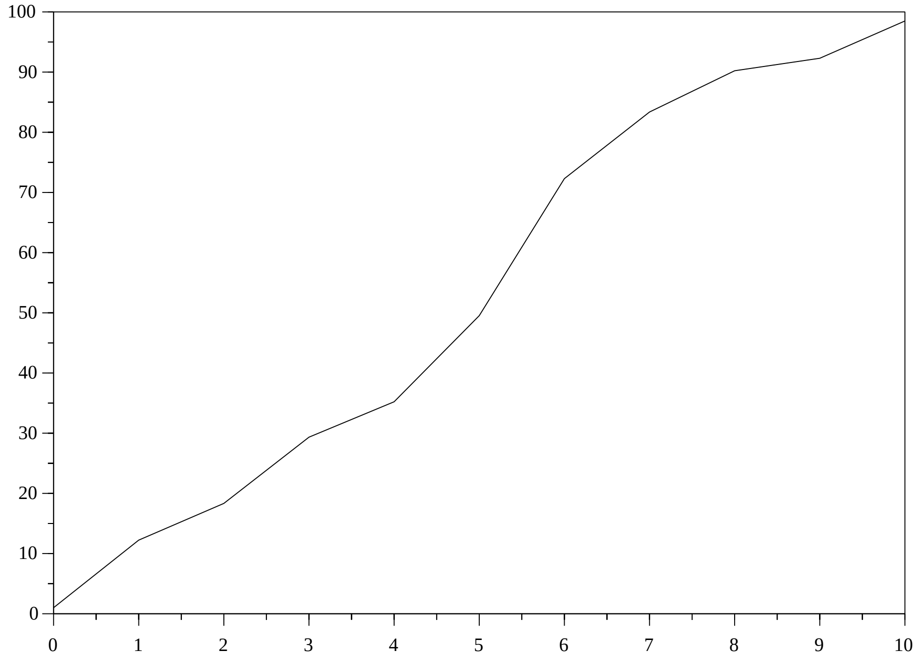


Figure 7.5: Number of turns in function of number of agents per sub-task

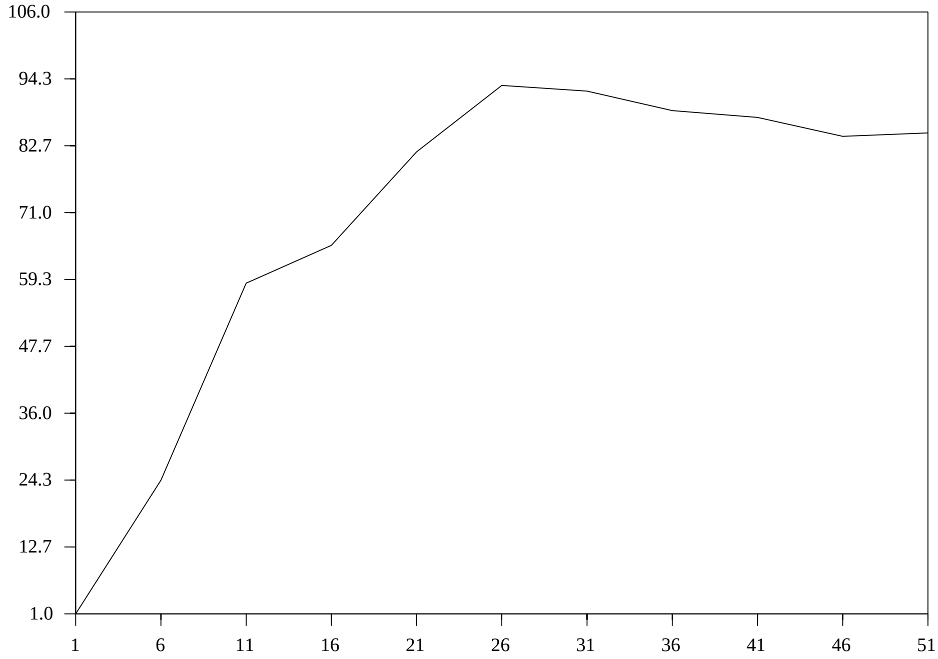


Figure 7.6: Number of turns in function of number of agents

protocol based on an exchange of preferences computed with quantitative and qualitative criteria dependent on the specific strategies of the agents. Moreover, we set up releasing procedures thanks to the flexible concept of coalitions to avoid the system paralysis. We have shown that, with this protocol, to be extremely rigid is not optimal and high competition leads to a faster consensus.

# Bibliography

[VE01a] G. Vauvert and A. El Fallah - Seghrouchni. Coalition formation among strongly autonomous and weakly rational agents. In *Proc. of MAAMAW'2001*, Annecy, France, May, 2-4 2001. Modelling Autonomous Agents in a Multi-Agent World. 10th European Workshop on Multi-Agent Systems. Acceptation rate of papers : 13/73=18%.

[VE01b] Guillaume Vauvert and Amal El Fallah - Seghrouchni. A distributed algorithm for coalition formation among strongly autonomous and weakly rational agents. In *Proc. of IAT'2001*, Maebashi City, Japan, October 23-26 2001.

# Chapter 8

# Conclusion

From the point of view of deployment of MAS in an economic context, it is necessary to consider that agents are totally autonomous, rational and heterogeneous: so, we define a subset of a Multi-Agent System, a System of Autonomous and Rational Agents with Heterogeneity (SARAH).
To arrive, within this framework, to form alliances, we proposed a protocol based on an exchange of preferences to evolve preferences and on coalition formation to avoid deadlocks.

## 8.1 Synthesis

In a SARAH, agents may be natural or artificial. Thus, they must be regarded as totally autonomous by the designer of a protocol, *i.e.* they make any decision.
Moreover, they are heterogeneous because the design is not centralized, and may make infringements (intentional or not).
In this context, a protocol must have several properties:

- rules are based on observable data (speech acts) in order to be supervised;
- no assumptions on agents' behaviors (autonomy);
- no complex capability required (heterogeneity);
- agents are motivate to abide by it using sanctions according their rationality;
- it must be distributed, egalitarian and universal;

Agents are autonomous at interaction level too; thus, they must be free to chose their partners, *i.e.* which alliances to form (an alliance is an organization that may dynamically created and dissolved according to needs).
In this context, we need a protocol that allows agents to reach a consensus on alliances to form and that has the properties above.
As agents are totally autonomous and have their own preferences, they have to use arguments to evolve others' positions. Due to heterogeneity constrains, arguments must be as simple as possible: the current positions. Thus, our protocol is based on an exchange of current positions in order to allow agents to evolve others' positions. However, agents

are not obliged to change their advice, what could lead to deadlocks. To avoid that, we propose to allow them to form coalitions: a coalition is a group of agents that act together in order to influence more strongly other agents. If nobody accept to form a coalition, we assume that the two agents with nearest positions are obliged to make a coalition.
Experiments done on the platform we developed show that strategies that leads to not desirable solutions are not profitable, what encourage agents to make compromises and hence reach a consensus.

## 8.2 Future works

### 8.2.1 Application to software engineering

Our thesis focuses on the context of electronic context. However, a SARAH is also suitable in software engineering, because same constrains and properties arise:

- totally autonomous components are regarded as very interesting, because they allow more efficient analysis, design, and implementation;
- in large scale systems, components are often implemented by several programmers, at different times, and on hardware that may be light or strong : components are heterogeneous;
- components may infringement their specification, either due to implementation mistake, either due to hardware failures;
- autonomous agents must have a motivation to act, what is not explicitly given;
- in large scale systems, protocols must be distributed and universal.

However, all these constrains are computationally very costly and seems to be hardly useful in this context. In fact, it is not the case, because there is no intentional infringement: so, agents may be *a priori* in confidence with others. So, a protocol may begin to make few supervision, and if an infringement is detected, he may increase the number of supervisors.
As in [caCP96], it would be also interesting to make it possible the agents to keep a track of alliances which were beneficial, in order to accelerate convergence towards a consensus. Building a model of the others will make possible to integrate the concept of confidence and will limit suspicion.

### 8.2.2 Autonomy at protocol-level

Finally, a SARAH may be regarded as a set of heterogeneous autonomous agents that interact within a protocol that check that it is abided by. As they are totally autonomous, they are free to reason without direct intervention and to ally with anybody.
However, their autonomy is not complete: the protocol is imposed by the designer of the system, since it could be chosen by agents themselves. In this case, we could said that

they are totally autonomous at system level.
As in case of autonomy at agent-level, several levels of system-autonomy exist:

1. The protocol is imposed to agents designers and thus to agents.

2. Agents may chose the protocol in a library. Each protocol is formally described: it requires some pre-conditions, it provides services, it has some properties (egalitarian, not centralized, . . . ), it provides some post-conditions,. . . The protocol is chosen using another imposed protocol (a consensus reaching protocol). The autonomy is bound, because the library of protocols contains a finite number of protocols and because the consensus-protocol (that allows them to choose the protocol to abide by) is imposed.

3. Agents can design themselves their protocols. It require more capabilities and today only few works exist [Kon02]. The reaching-protocol is imposed, but agents may dynamically design and use their own protocols. They must be able to dynamically design, modify, criticize and use protocols.

4. Agents have the same capabilities than above, but moreover, they are able to choose their consensus-protocol. It require stronger skills, and particularly, the consensus-protocol may contain the way to modify it-self.

Before choosing a protocol, agents must decide the property he must own. Then, in this framework, a protocol must be chosen or built.
Really, reaching-protocol may be called the constitution, protocol-properties the law, and other protocols the decrees. If the MAS is not too large, all agents can decide; else, only a reasonable part of the agents (chosen by a vote system) will participate to the decision making process.
The total autonomy leads to regard agents as close of human beings. A large part of works made on people interactions (law, human rights, politic, rules,*etc.*) may be used to build protocol among agents; however, they must be well formalized and make computationally useful.

# Bibliography

[caCP96] Jorge Anacleto Louçã, Helder Coelho, and Suzanne Pinson. Forming coalition in task oriented multi-agent systems. In *Proceedings Of IBERAMIA 1996*. IBERAMIA, 1996.

[Kon02] Jean-Luc Koning. Dynamic choice of agent interaction protocols. In *Intelligent Information Processing*, pages 167–176, 2002.

# List of Figures

# List of Algorithms

# Index

**RÉSUMÉ:**Dans les systèmes multi agents, l'agent est défini notamment comme une entité autonome, mais souvent au sens faible. En commerce électronique, nous sommes amenés à considérer une autonomie totale, au niveau agent et au niveau interaction. Le niveau agent a de nombreuses conséquences sur la conception de protocoles: prise en compte de l'hétérogénéité, protocole non respecté (intentionnellement ou non), contraintes uniquement sur les données observable, etc. Dans ce cadre, prendre une décision commune est très difficile. Nous proposons un protocole d'atteinte de consensus basé sur l'échange d'opinions pour faire évoluer les positions, et sur la formation de coalitions pour éviter les blocages (lorsque les positions n'évoluent plus). Toutes les règles sont vérifiables par les agents eux-mêmes, ce qui garantit le respect du protocole sans restreindre l'autonomie. Le formalisme d'opinion proposé permet de représenter finement les préférences des agents et d'agréger les opinions d'un groupe. Au niveau des interactions, l'autonomie totale implique que c'est aux agents de décider de la formation d'alliances, ce qui, dans notre contexte, ne peut être vu que comme un problème de consensus. Les expérimentations réalisées sur la plateforme que nous avons développé montrent que les strategies risquant de mener à des solutions moins légitimes ne sont pas profitables, ce qui incite les agents à faire des concessions.

**TITLE:**Consensus reaching based on an exchange of preferences among rational and autonomous agents : application to alliances formations

**ABSTRACT:**In multi-agent systems, the concept of agent is defined notably as an autonomous entity, but often in a weak means. In electronic commerce, we are induced to consider agents as totally autonomous, at two levels: at agent level and at interaction level. The agent level has many consequences on the design of protocols: to take heterogeneity and infringements (intentional or not) into account, rules only on observable data, etc. In this framework, to make a common decision is very difficult. We propose a protocol of consensus reaching based on an exchange of opinions to evolve others' positions, and on the formation of coalitions to avoid deadlocks (when positions don't evolve anymore). All rules are checkable by agents them-selves, what guarantees that agents abide by the protocol, without making no restriction on agents' autonomy. The formalism of opinion that we proposed allows to model finely agent's preferences and to aggregate opinions of group's members. At interaction level, the total autonomy implies that agents must decide by themselves which alliance to form, what, in our context, must be regarded as a consensus problem. Experiments made on the platform that we have developed show that strategies that have a good chance to lead to not very legitimate solutions are not profitable, what incites agents to make concessions.

**DISCIPLINE : Informatique**

**MOTS-CLÉS:**systèmes multi-agents, autonomie, commerce électronique, preference, protocole, formation de coalitions, formation d'alliances, atteinte de consensus

**Laboratoire d'Informatique de Paris Nord**
**UMR 7030 - Institut Galileé - Université Paris 13**
**99, avenue Jean-Baptiste Clément - 93430 Villetaneuse**