FROM INTERACTIONS TO COLLECTIVE BEHAVIOUR IN AGENT-BASED SYSTEMS

Yves Demazeau, PhD., CNRS Research Fellow

Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle, Institut IMAG
46 avenue Félix Viallet, 38041 Grenoble Cedex, France
e-mail: Yves.Demazeau@imag.fr

Abstract. In this paper, we present the MAGMA approach to multi-agent systems, including a proposal of an integrative environment based upon the decomposition between agents, environments, interactions, and organisations. We then focus on the MAGMA work done for cognitive agents, and we detail the interactions between such agents. We propose an Interaction Language associated with Interaction Protocols. A range from simple to complex interaction protocols is then presented. We illustrate the use of these interaction protocols for social and individual control issues in several application domains. We finally discuss our choices and future work. All this brings us at a stage where there is clearly still much work to do, but where we hope to dispose of an interesting framework, inspired by interdisciplinary studies, to approach effective Agent Oriented Programming.

1 Introduction

1.1 The MAGMA Approach

By localising the intelligence of the systems in the components and between the components of system, multi-agent systems try to validate the principle of the non-reducibility of the complexity. The objectives of the MAGMA group at LIFIA consist in the development of theoretical studies, software tools, and practical realisations for the decentralised simulation of complex systems, and for the distributed solving of complex problems. Most of the validation of the studies deals with the comparison with other approaches, at the level of the models and at the level of the tools, in a clear perspective of agent oriented programming.

Towards agent oriented programming. Modelling autonomous agents in a multi-agent world, meaning, our approach of the emerging paradigm of agent oriented programming, underlies that the operational part of how the solution is found (DPS) — how the equilibrium states of the systems are reached (DSS) — is taken in charge by the multi-agent system itself and no more by the conceiver / user. This point of view of decoupling declarative semantics and operational semantics, which remembers the one that generated logic programming, enables to consider the emerging paradigm of agent oriented programming as a natural evolution of object-oriented programming, a soon as objects become agents, when they are associated with perception capabilities, communication capabilities, reasoning and decision capabilities. These capabilities enable the agents to autonomously process the information relative to the problem to solve (DPS) — the system to simulate (DSS) —, the information relative to the other agents, to the domain, to the conceiver, and to the user.

The methodology. Our work, mainly a computer scientist view of it, focuses on the development of software tools to be enclosed in a common integrative environment. Through the application domains we have been investigating, the basic components at our multi-agent levels are the agents, the environments, the interactions, and the organisations. The agent models, or agent architectures, range from simple fine-grained automata to complex coarse-grained knowledge-based systems [Demazeau 90] [Demazeau 91b]. The environments are domain dependent, but at least, in our studies, they are always spatial environments. Interaction structures and languages range from physics-based interactions to speech acts [Demazeau 94a]. Organisations range from dynamic ones inspired by biological studies, to more governed by social laws ones inspired by sociological studies. For a given problem to solve — a system to simulate —, the user chooses the agent models, the environment model, the interactions models, and the organisation models to be instanciated. The specialisation of these generic tools in the context of the problem and regarding the application domain enables to build computational multi-agent systems.

Applications (Vision, Robotics, NLP, Télécom., T&C D, GIS)

Agents | **Environments** | **Interactions** | **Organisations**

Distributed System (DPSK, XENOOPS ...) , Standards

Network of workstations (Sun ...), Standards

The three statements. As assumptions for our research, we pose the three following statements • As we have just seen it, and from a declarative point of view, a multi-agent system will be composed of several agents, an environment, a set of possible interactions, and possibly at least one organisation. This first statement will be called *the declarative equation* • From a more computational point of view, the functions that will be ensured by the multi-agent system are enclosing the ones of the agents — enhanced by the programmed interactions between the agents and the environment / the other agents, with respect to the organisation — , in addition with the ones which results from the added value generated by the agents evolving in a multi-agent world, which are usually encompassed under the name collective intelligence. This second statement will be called *the functional equation* • As for being complete, and to try to reach the real notion of what could be agent-oriented programming, societies of agents should be able to be considered as coarser agents at a higher level of abstraction and should be handled as such [Boissier 92] [Pleiad 92]. This assumption that has to be operationalised, will constitute our third statement, *the recursion principle*.

the declarative equation

MAS = Agents + Environment + Interactions + Organisation

the functional equation

Fonction(MAS) = Σ Fonction(Agents) + Collective Fonction

the recursion principle

a MAS has to be considered as an agent at a higher level

The three sides of the research. • As for being able to tackle our research issues, the work has been academically splitted in two parts [Demazeau 91b]. The COHIA part (e.g. [Boissier 94b]) studies mainly deal with the study of the internal agent architectures (mainly cognitive ones) while the PACO part (e.g. [Demazeau 91a] [Demazeau 93]) studies mainly deal with the study of the external behaviour of agent groups (mainly reactive ones) evolving in a multi-agent world. This splitting might appear unclear, but in fact, when thinking to PACO as dealing with the study of the internal model of a society of agents, and remembering the recursion principle, it becomes obvious [Boissier 92]. • The agent models [Boissier 94b] [Occello 94], the environment models, the interaction models [Berthet 92] [Populaire 93] [Demazeau 94a], the organisation models [Sichman 94], are constituting as many toolboxes for the MAGMA integrative environment which ensures the computational assembling of the selected models when building a multi-agent system to solve a given problem — to simulate a given system —. Such systems are built above performant distributed systems such as DPSK [Cardozo 93] or XENOOPS [Bijnens 94] with respect to communication norms (TCP/IP, CORBA), that enables to implement the low-level interactions between the processes which are involved in the building of the multi-agent systems. In addition, these processes may be localised in different places in a network of computers, here again in respect with existing norms (UNIX, X, FRESCO). • The generic tools which we develop at the multi-agent level are experimented through the building of numerous systems in various application domains: Computer Vision [Boissier 94a] [Boissier 94c] [Demazeau 94b] and Robotics [Demazeau 91a] [Demazeau 93] [Hassoun 92], Natural Language Processing [Stéfanini 93] and Telecommunications [Koning 95], Town and Country Development [Ferrand 94] and Geographical Information Systems [Baeijs 95].

1.2 The Interaction Toolbox

One of the toolboxes from the MAGMA integrative environment consists in the one devoted to interactions between agents — and by extension, between the agents and the environment, as soon as one may consider the environment as being an active one.

From electrostatic forces to speech acts. Interaction structures and languages range from physics-based interactions to speech acts. Physics-based models like electrostatic forces permit to express simple interactions with attraction and repulsion, and such models are widely used to model the interactions between reactive agents. Illustrations of such types of interactions for reactive agents can be found in [Demazeau 91a] [Ferber 91] [Demazeau 93]. This kind of interactions is implemented by communicating through the environment, using it as a blackboard or a shared resource. Reactive agents do not encompass deliberative control, nor explicit reasoning, and nobody would expect such agents to hold structured conversations supported by speech acts as it is possible between cognitive agents. We will focus on this last type of interactions in the next section, which are usually implemented by direct point-to-point communication and more rarely through a shared resource.

Interaction versus Communication. In the MAS community, it is admitted that the term communication in MAS means more than it means in traditional Distributed Systems. Unfortunately, whereas communication already is standardised in this later, there is not yet a common agreement about how communication should be treated in MAS. How rich should be the syntax of a message to help its fast interpretation? Is it possible to define application independent protocols? Could one find a set of primitives and build protocols from these primitives? The answers to these questions have to be found at the MAS level and not only at the Distributed System level that might implement part of the communication between agents. To distinguish the MAS work with the standardised communication issues that exist in Distributed Systems, and for discussing about communication issues between agents, we will usually refer to *interaction* rather than to communication. Some of the answers we are looking for are illustrated by our approach to MAS. The interaction toolbox includes not only physics-based types of interactions, but also some cognitive ones, inspired, like most of such approaches, by speech acts.

2 Interactions between Cognitive Agents

2.1 The Speech Act Theory

In the MAS community, it is widely admitted that cooperation between reasoning agents needs explicit communication. Many proposals towards standardisation have been made these last years, but they usually deal with task-specific protocols. Hopefully, in some of our previous work [Berthet 92] [Demazeau 94a], in Finin's one [Finin 94], or in Burmeister's one [Burmeister 93], one may find steps towards standardisation.

Short notes about speech acts. Most approaches for interactions between agents in MAS have relied on Speech Act theory, drawn by Austin, then Searle [Searle 69], Ballmer and Brennenstuhl [Ballmer 81], and Vanderveken [Vanderveken 94]. In a speech act, one makes a distinction between elocutionary, illocutionary, and perlocutionary aspects. The plain utterance of a sentence (elocutionary act) is distinguished from its intended effects (illocutionary act) and its actual effects (perlocutionary act) on the receiver. The classification of illocutionary acts into assertives, directives, commissives, declaratives and expressives by Searle [Searle 69], has guided MAS researchers in choosing the nature of message types. Faced with the difficulties to classify all German speech acts, Ballmer and Brennenstuhl have suggested to group verbs into semantic categories, and have grouped semantic categories into models, that include an ordering of the categories according to their temporal relationship and degree of strength [Ballmer 81]. These categories and models have led to dialogues and appropriate frameworks for structuring dialogue protocols. Vanderveken has detailed and has completely formalised the seven fields in the structure of an interaction message [Vanderveken 94]. These works are very much inspiring for MAS research towards formalising general frameworks for cognitive interactions between agents that might not be human-like ones but also artificial ones.

The MAGMA approach. From the point of view of the computer scientist, if interaction is solely described in terms of sending and receiving messages, each agent must be able to infer what the sender intended by uttering the message. If messages are not structured, this inference could be inefficient. Starting from Searle's work, it seems that, in some way, there is no real problem to explicitly encode the elocutionary act and the illocutionary act with message types and this will lead us to choose about basic *types* and *strengths* of messages. Encoding the perlocutionary acts is a much more challenging tasks and involves to extend the theory of speech acts to a real dialogism [Brassac 94]. As for trying to solve this problem, we will assume that the sender will know about the internal architecture of the receiver, and we will distinguish between several *natures* of interactions. To fully implement the perlocutionary acts, and to be able to share the conversational context between the agents as it can be expressed by Vanderveken, we will extend the notion of dialogue protocols introduced by Ballmer and Brennenstuhl towards the notion of *Interaction Protocols* that we will be later able to instanciate to implement different contextual social behaviours of agents.

The initial ideas that describe our first steps towards the specification of the Interaction Language can be found in [Demazeau 94a] [Demazeau 94b]. In these earlier papers, and related ones [Boissier 94c], [Koning 95], the Interaction Protocols are presented outside the Interaction Language itself, which leads to some confusion about the fact that agents not only exchange interaction messages, but also the context of this interaction, namely the interaction protocol within which the message has been emitted. We hope that this possible misunderstanding cannot be done with the present contribution.

2.2 Our Approach

The Interaction Language. Messages should be bound by formal restrictions and structured for the ease of interpretation, for instance by employing message types, such that the intention of the sender could be immediately recognised from the message itself. This naturally leads to first distinguish between the *Communication Language* and the *Knowledge Representation Language*. The former mainly translates the message from a pure Distributed Systems point of view, and the latter surely vehicles at least the multi-agent domain knowledge (*Multi-Agent Language*) — which encompasses, as much as possible, the intention of the sender and its expectations —, but that also includes the *Application* domain *Language* (e.g. one for Computer Vision).

<interaction> ::= <communication> <knowledge representation>

<knowledge representation> ::= <multi-agent> <application>

The Communication Language. Described by the <communication> field, the Communication Language consists of different fields used by the communication layer of the system, that might be useful to be controlled from a multi-agent perspective. For instance one agent would like to implement a rendez-vous and thus, will need to force the corresponding interaction to be implemented in a synchronous mode. The presence of this <communication> field at the multi-agent level does not mean that we ignore the proper role of the communication layers at the Distributed Systems level. It first means, when referring to the MAGMA approach, that this <communication> field represents the necessary bridge with the Distributed System. It also means that far away to ignore the presence of the distributed systems capabilities, we would like that the agents may access them in an autonomous way, to better exploit these capabilities.

The fields which are composing the <communication> field are: • the receiver <to> and the sender <from> of the message — the contents of these fields are agent identifiers, or the keyword *broadcast* in case of the receiver —, the identity of the message <id>, the type of communication channel to use <via> — in the communication system we use, two kinds of communication channels are available: one for the emission of large amount of data, high-bandwidth channel, and another for simple message passing, like through mailboxes —, and the mode of communication <mode> — that is *synchronous* or *asynchronous* —.

<communication> ::= <from> <to> <id> <via> <mode>

The Application domain Language. It seems clear that at a given level of meaning, the complexity degree of the Multi-Agent Language is inversely dependent on the development degree of the Application Language that is used by the agents to exchange knowledge. Since we want to develop a quite complex Multi-Agent Language, this enables us to make use of simple Application Languages, like first-order ones. Such languages seem to be enough powerful and adequate for the particular application domains we are involved in, like the Computer Vision one. Nevertheless, and exactly because of our own real experience in building and using multi-agent systems in various application domains for various purposes, we do not believe about the technical usefulness nor the scientific realism to ignore the specificity of the application domains. In other words, and contrarily to efforts like the KIF one [Genesereth 94] or the Euroknowledge one, we do not think that it is possible nor suited to reach a normalisation of what is Knowledge without explicitly taking into account application domains. When referring to the MAGMA approach, it means that this <application> field represents the necessary bridge with the Application Domain. It also means that far away to ignore the presence of the Application domain knowledge, we would like that the agents may process it in an autonomous way, to better deal with it.

<application> ::= specific decomposition in application domain

The Multi-Agent Language. The Multi-Agent Language encompasses, as much as possible, the intention of the sender and its expectations. It also means that the field <multi-agent> gathers all the information that is related to the multi-agent level. In our approach, we have five kinds of information that are the <type>, the <strength>, and the <nature> of the interaction, as well as the <protocol> in the context of which the interaction act is emitted, and the <position> in which the receiver is situated when receiving the interaction act.

<multi-agent>::=<type><strength><nature><protocol><position>

3 The Interaction Language

3.1 The Basic Primitives: Type, Strength, Nature

The Type. As for the <type> of the interaction, we have adopted the primitives proposed by [Gaspar 91]. They consist of four possibilities: *present*, *request*, *answer*, *inform*. These four types have to be distinguished because of the different basic behaviours that they model from the sender or the receiver points of view. A *request* includes a change of state of the sender, waiting for an answer. An answer includes a change of state of the receiver, no more waiting for the answer. An *inform* includes no change of state for both the sender and the receiver. It might generate other informs, why not answers, in the spirit of perlocutionary acts, by it is not intended initially. A *present* includes a possible change of state of the sender and/or of the receiver. Typically, a present will enable to enter a society and to introduce itself to the other agents.

The Strength. The <strength> defines the priority of the interacting act. We borrow the terms filling this field from the Speech Act theory, restricting their meaning to this priority criterion: from *comm* (commanding, highest priority) to info (informing, lowest priority). Such a field enables an agent to express the priority of the interaction for itself, and which priority it should have for the receiver. The scale can be refined, and precisely, we have adopted a set of labels made explicit by Campbell and d'Inverno [Campbell 90] as a basis. Campbell and d'Inverno have tried to minimise the amount of information exchanged whatever could be the behaviour of the agents under consideration. For this purpose, they have tried to fully express the intentions an agent has when communicating with someone else, and they have built a taxonomy of the different tones of communication in twenty-one categories. From their set of tones, we have extracted a subset that is currently and experimentally suited for the different kinds of interactions we have to express: informing, information seeking, advising, expressing, bargaining, warning, persuading, and commanding. We have no real argument to defend this list of tones compared to other possible ones, except that this list has been sufficient for the experimental studies we have conducted. So, we are perfectly conscious that this list will be refined and maybe redefined in the future, to take into account future experimental needs, to generate a consistent theory, and, following a more cognitive science point of view, to better take into account the work that is currently done in the Speech Act theory.

The Nature. The <nature> is following an idea we have jointly introduced with Müller in [Demazeau 90], where we distinguished which kinds of information planning-like agent architectures may exchange: knowledge (or hypotheses), possible solutions (or plans), choices (or actions). Later work about dependence situations between agents — cf. social power, work jointly done with social psychologists

[Sichman 94] —, and further results in experimental vision systems [Boissier 94c], have led us to also consider the possibility of exchanging goals between agents. Generally speaking, and whatever is the assumed agent architecture of the receiver, the <nature> defines on which control layer of the receiver the content of the interaction has to be taken into account. To illustrate these layers, one can choose the ASIC control layers [Boissier 94b] and which are inherited from the Control Theory: *obs* (observation layer, exchange of hypotheses), *dec* (decision layer, exchange of goals), *ada* (adaptation layer, exchange of plans), and *com* (command layer, exchange of actions). As far as this would reveal possible, one would naturally like to define the <nature> independently of the agent architecture of the receiver. It is our strong belief that it is generally impossible, and only possible in the case of societies composed of homogeneous agents, thus having the same internal architecture — e.g. the BDI assumption in the AOP paradigm [Shoham 92] —. We prefer to just consider the layers of the ASIC architecture as an illustration of what an agent can do if it knows — because of built-in or acquired knowledge — about the ASIC type of the architecture of the receiver. Basically, we think that the <nature> strongly relies upon the knowledge or the capability of the sender to infer the internal architecture of the receiver, and thus cannot be generalised.

3.2 The Other Primitives: Interaction Protocols, Position

The usage of primitives like the type, the strength, and the nature, enables an agent to extract explicitly from the meaning of the message some information that is useful for the control of the information exchange and for the control of the whole society. Decoupling the intention of the sender from the message itself, giving explicit instruction to the control layer of the receiver at which the message itself should be taken, both issues constitute a first step, but it is not enough, since the agents also ought to know how to react to a message, or what the sender should expect after sending a message. All these additional requirements should be met collectively by a unique framework. We call such framework for structuring interactions among agents *Interaction Protocols.*

Interaction Protocols. An interaction protocol comes to restrict the different interactions an agent can link with other agents regarding the problems to solve. An interaction protocol is a transition network that defines, a priori, the set of possible transitions (shown as straight lines) which link a set of states (shown as circles) that the agents may alternately occupy according to the effective exchanged interactions acts. A transition may constraint the filling of some fields in the interaction act. It also may be labelled by a condition that an agent has to satisfy before using the transition. The fundamental difference between our Interaction Protocols and Petri Nets is that the next state cannot be determined by an external decision — i.e. it is not inferred from the current state of an agent and the general state of the network —, but the agent itself evaluates and chooses what to do next, the decision being a local one. This naturally involves that in some way, the message vehicles not only the interaction act, but also the nature of the conversation, represented by the field <protocol> as well as the state of the conversation the agents have reached, and that is represented by the <position>. By making explicit the presence of such a field in the interaction act we do not simply want to try to contribute to model the possible expectations from the receiver's point of view, with respect to the sender's intentions, and thus to try to operationalise the notion of perlocutionary acts. It also means that our approach enables to model the conversational context of the interaction from a pragmatic computer science point of view. In Speech Act theory, such a contextual information is assimilated as the preconditions to the speech acts. By incorporating the context in the interaction protocol, we are doing exactly the same, except that the context is shared through the interaction itself, and then, it prevents an agent to not understand the meaning of an interaction.

Processing the Interaction Protocol. The execution of any conversation between agents will thus typically involve the exchange of a basic protocol denoting the conversational frame that is chosen, the state of the interaction protocol itself, and the last instanciated act that has been emitted and that has enabled to reach this state. Receiving the interaction act, the agent is able to decode the conversation it is taking part (among the several ones it might have at the same time), it is able to identify the sender, its intention in the emission, its expectations. From these elements, it evaluates the situation, and reacts with respect to the possible interaction it might take, as mentioned in the corresponding interaction protocol.

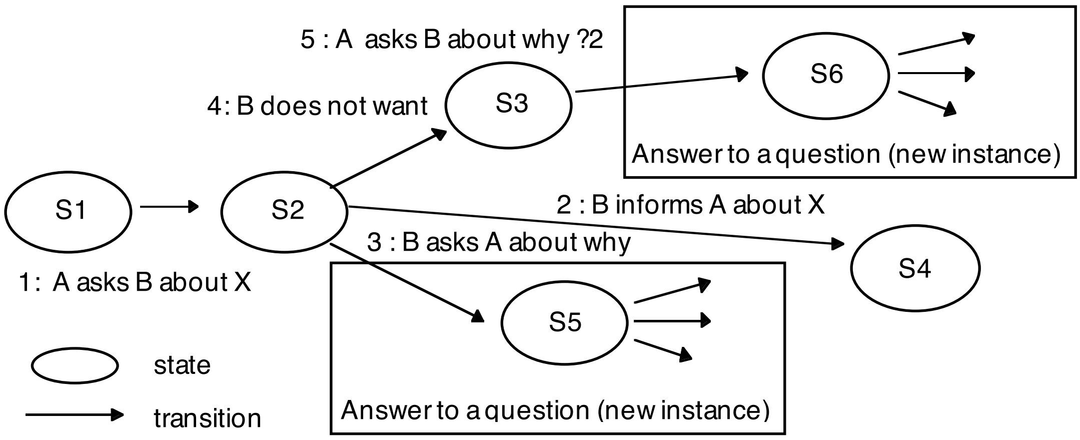
4. The Interaction Protocols

4.1 Basic Interaction Protocols

Classical interaction protocols in the MAS literature include the Contract Net Protocol [Smith 80], the Hierarchical Protocol [Durfee 90], the Introduction Protocol [Berthet 92], the SNAP protocol grounded by models issued from the linguistic study of the *struggle situation* in human societies. All of them have been developed in a specific context, and are dedicated to the implementation of one particular situation in the social control framework: coordination, negotiation, introduction, etc. We intend to offer a general framework in which such interaction protocols could be easily programmed and reused for social control issues.

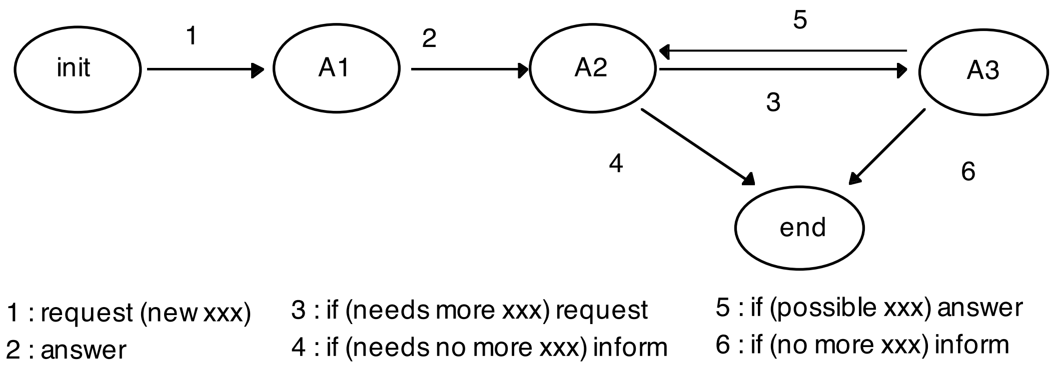
The complexity invariance of the Multi-Agent Language. We would like to point out the fact that the complexity degree of the Interaction Protocols is inversely dependent on the development degree of the complexity of the primitives of the Interaction Language — and so, of the Multi-Agent Language, assuming that a consensus could exist about what should figure in both the Communication Language and the Application domain Language —. Since we want to adopt rather simple and explicit primitives between agents, this leads us to be able to define quite complex interaction protocols. We just want here to illustrate the growing complexity of these protocols by presenting several of them.

The Everybody's Request Protocol. We attribute the initial design of this protocol to everybody, assuming that it does not belong in fact to anyone in particular.

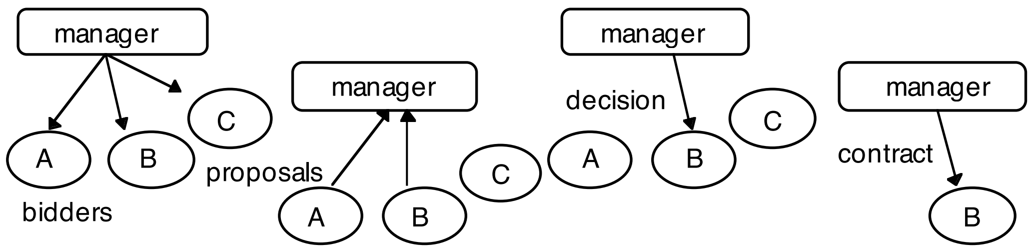


In this basic protocol, the receiver of a request either answers the question (2) — this is the usual way of doing and this is really the only case that should exist if we were only at the Distributed System level —, either starts a new instance of the Everybody's Request Protocol by asking the sender why it emits such a request (3). It may also not answer, either explicitly or because of not answering after some delay (4), and thus provoke the sender to start a new instance of the Everybody's Protocol by asking why the receiver does not want to answer (5). Choosing a protocol without the possibilities (3) and (4) would force the receiver to answer the request.

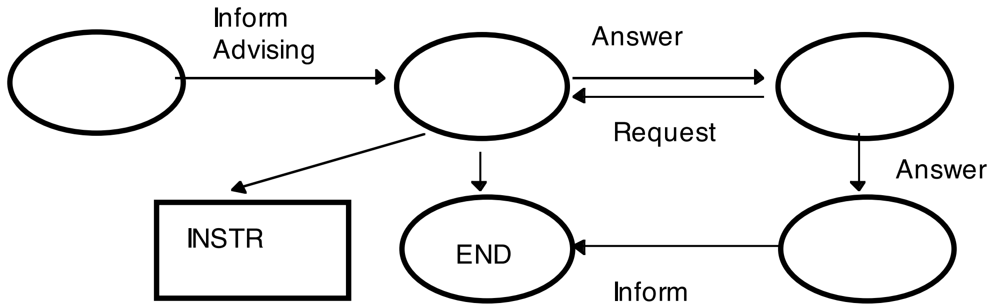
The Request-until-Satisfaction Protocol. The Everybody's Request Protocol is the probably one of the most popular protocols in MAS. An extension of it can be drawn when the receivers only partially fulfil the desires of the sender. A classical example of this happens when the sender in asking for a list of items. After being provided with it, the sender may ask the receiver to get more of these items. The receiver either provides the sender with more items, or signals him that there are no more available. At any stage, the sender may also inform the receiver that it got enough of them and then closes the conversation. Within the MAGMA project, this basic protocol has been used — with the restricted answer-request protocol just seen above — in the Computer Vision domain [Boissier 94a] [Boissier 94c], in the framework of Scene Understanding, to enable an agent, for instance, to ask another one for a list of features.



The Smith's Contract Net Protocol. This very classical negotiation protocol has been initially proposed by [Smith 80]. It enables to establish any kind of contract between a manager — one agent may dynamically take this role, as an example, for subcontracting one of its tasks — and several bidders — that, in our example, would be the agents for which the agent believes they may help him —. This negotiation protocol has then been later refined, complexified in several directions, but we just present here its general shape, without expressing it in our Interaction Language. The manager proposes a problem to solve, the bidders make proposals that are collected by the manager, the manager then chooses about who will be allocated with the problem, and establish a contract with him to be sure to recover the solution of the problem.



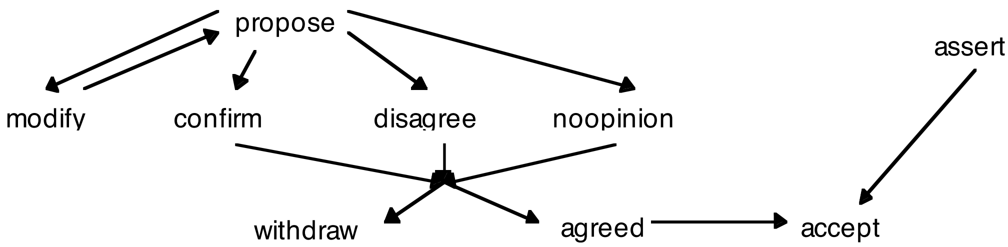
The Berthet's Doubt and Instruction Protocols. In the same direction than the Contract Net protocol, we have designed in [Berthet 92] two primitives that regularly compose conversations between agents. The first one, the Doubt Protocol, expresses the frequent existing stages within which the agents can express their doubt about what was exchanged, while the second one, the Instruction Protocol, enables them to ask for further information. The second one has been revised and has been presented above as the Request-until-Satisfaction Protocol. As for the first one, that is illustrated below, an agent receives an information with which it feels doubt. It can ask for advises by getting the opinion of other agents. From the answers it receives, it can enter the Instruction Protocol to get further information. It may also ask the initial sender of the information to provide with more reasons to ground the information it has previously sent. The initial sender can then answer either by trying to persuade or by giving up. These two protocols have been used to conceive the Introduction Protocol that is presented now .



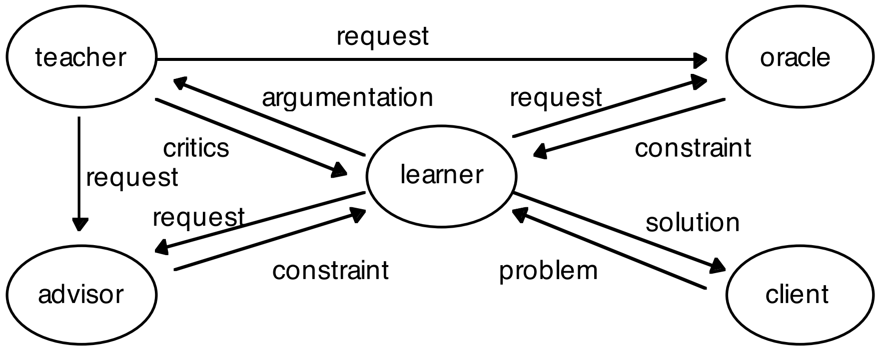
The Berthet's Introduction Protocol. Assuming that the representation of every agent entering a society includes a description of the several roles it could take, as well as the basic actions it is able to perform to realise each of these roles, the entering agent initialises a presentation to the agents in the society. Comparing their own roles and the corresponding basic actions, every agent detects • a possible redundancy with him, and then enters a Duplication Protocol to know how deep this duplication may be • alternatives to fulfil a role, and then starts a Competition Protocol to better know which of the agents will be the more performant in which cases • a possible incoherence between the respective knowledge of the agents about which role they can fulfil, and then enters an Incoherence Protocol to mutually learn about possible roles • nothing special, an simply starts an Information Protocol so that the agents can both update their mutual knowledge. This protocol is described in more details in [Berthet 92] and will not be illustrated here. As previously said, it uses the Doubt Protocol and the Instruction Protocol to build the four other protocols mentioned here.

-> same RO and same BA	-> duplication
(detection of redundancy)	
-> same RO and different BA	-> competition
(towards negociation, subcontracting)	
-> different RO and same BA	-> incoherence
(detection of inconsistencies, towards learning)	
-> different RO and different BA	-> information

The Sian's Cooperative Learning Protocol. One of the first learning protocols is the one proposed by [Sian 91]. In this protocol, an agent proposes a hypothesis to every agent that might be concerned. Each agent either confirms, modifies (by specialising or generalising), disagrees or has no opinion about this hypothesis. The initial sender counts the answers after some delay, and, using a global heuristic (shared by the other agents), finally decides to withdraw or to agree the hypothesis. In the latter case, it let know the other agents about the decided joint agreement about the hypothesis, so that each agent can locally update its knowledge. Within the MAGMA project, the initial Sian's protocol has been used in the Natural Language Processing domain [Stéfanini 93], in the framework of the Understanding of the Written French, as the basic protocol for making the several local experts agree about the meaning of parts of sentences.



The Individual Control and Learning MOSCA Protocol. The MOSCA Learning Protocol that is issued from studies about the mental schemes [Quinqueton 91] [Reitz 93] is a typical example of how an Interaction Protocol could be used at the individual control level and not only at the social control level. MOSCA has been conceived by considering the five roles an agent can successively have when learning. The transition between roles is dependent of the context and is accompanied by an explicit interaction. Concerning our Interaction Protocols, if such roles were considered as being fulfilled by several experts having one role each, MOSCA would have been a simple interaction protocol for social control, and more precisely for collaborative learning like the Sian's one. When considering the five roles as being fulfilled by a single agent, then the corresponding Interaction Protocols figures an Individual Control strategy. This makes a bridge between the social control level and the individual control level, and illustrates the recursion principle we have uttered at the beginning of the paper. Conceived as an interaction protocol used at the individual level, this protocol is used by the LIRMM to implement network managers in the Telecommunications domain [Koning 95].



4.2 MAGMA Applications of the Interaction Protocols

We briefly present here examples of use of the Interaction Protocols in several application domains. Most of these applications are reported in other papers that are referenced. There are two reasons for the presence of this subsection in this paper. On one hand, we would just like to illustrate the richness of the approach by showing the multiple usages of such Interaction Protocols for social control purposes. In each domain, and contrarily to existing systems where strategies are designed globally by the user and are embedded in the system to be executed as such, using interaction protocols enables each agent of the multi-agent system to dynamically choose the interaction protocols corresponding to the problem it has to solve, and adapt its reasoning and deciding processes in accordance to the evolution of the resolution context. On the other hand, it should also be clear that using protocols in a particular domain, introduces a domain knowledge with its proper semantics, and enables to provide a fully operational semantics to the protocols.

Strategies in Computer Vision Systems. In [Boissier 94a]□[Boissier 94c]□[Demazeau94b] we have shown how Interaction Protocols could be used to implement various strategies in Computer Vision, and how, in particular, scene understanding strategies or recognition / localisation strategies could be tackled dynamically by a distributed system, towards effective integration of visual modules. In some old paper, we have shown how such high-level vision systems can be viewed, conceived, and built as multi-agent systems [Boissier 92]. Two systems, VAP' and MAGIC, have been implemented using such a MAS approach (C++, CLIPS, DPSK). They use two basic interaction protocols — the restricted Everybody's Request protocol and the Request until Satisfaction Protocol — to implement find-and-track strategies (VAP') and reconstruction strategies (MAGIC).

Conflict resolution in Natural Language Processing. Ongoing work is performed in the MAGMA group to contribute to the Distributed Understanding of the Written French [Stéfanini 93].This Natural Language Processing work assumes the existence of cooperating experts that are individually able to give partial syntactical interpretations of parts of a sentence. The experts are emitting hypotheses and these hypotheses are checked and verified through negotiation and conflict resolution between the experts to determine the possible interpretation(s) of the sentence that will be agreed — or at least not disagreed — by every expert. A first system, TALISMAN, has been implemented using such a MAS approach (BIM PROLOG). It uses the Sian's Cooperative Learning Protocol to implement the negotiation strategy.

Negotiation in Town and Country Development. A recent project we are involved deals with the decision support for negotiating in a Town and Country Development framework [Ferrand 94]. A typical problem that is studied is land-use planning. Agents involved are administrations and decision makers (local, regional, national), manufacturers, associations, citizens, consultants. There are several qualitative and quantitative options. The issues are ecological, economical, and cultural ones. The actors have diverging interests and they have to interact, with respect to the existing laws, in order reach a decision. Investigations in the domain have just started and a system is being built. The problems to solve involve human actors, interactions, organisations, which needs adequate modelling before starting to get significant results. In every case, it provides a very rich framework to experiment negotiation strategies, like the Contract Net ones or Sian's ones, but also to develop new ones.

5 Conclusion

In this paper, we have presented the MAGMA approach to multi-agent systems, including a proposal of an integrative environment based upon the decomposition between agents, environments, interactions, and organisations. We have focused on the part of the MAGMA work done for cognitive agents, and we have detailed the interactions between such agents. We have proposed an Interaction Language associated with Interaction Protocols. A range from simple to complex interaction protocols has been then presented. We have illustrated the use of these interaction protocols for social and individual control issues in several application domains. Before concluding, we discuss some of our choices and present future work.

Guarantying the reception and the processing on an Interaction. One can guaranty the full reception and the full processing of any sent message through what we consider as being a loosening of constraints within the Multi-Agent Language. Practically speaking, putting the <strength> at its highest degree of priority, addressing the decision layer of the receiver through adequate filling of the <nature>, choosing suited <protocol> shared by every agent, and where the possibilities of reactions of the sender are restricted (e.g. the restricted answer-and-request protocol) enable to guarantee that the message will be received, correctly interpreted and adequately processed. This still assumes a good willing of the receiver, or its less important social role in the organisation than the one the sender fulfils.

The location of the Interaction Protocols. The Interaction Language, including the Interaction Protocols, is implemented in the Interaction Toolbox of the MAGMA integrative environment. Currently, communications between agents use the primitives of the DPSK distributed system. The Interaction Protocols are updated by the conceiver at a unique place. When starting one protocol, an agent comes to get a copy of the current release of the given protocol, and will interact with the other agents by the exchange of this release. In some way, this implementation is quite natural. When thinking to other domains (cf. the Driving Code for car-like driving, cf. the legal procedures and laws for justice issues), one clearly would like that these Interaction Protocols, considered as Social Laws, have to be recorded somewhere, in some database that would be shared by all the agents. That is what is currently done. Now, when thinking to learning issues, like specialising issues, learning such social laws by an agent may lead it to make its own representation of the laws. This is surely a very interesting work to do now, but one should clearly tackle the consequences of this: by customising the laws and making its own representation of them, the agent will alliterate the interaction protocol itself, and this would generate possible misunderstandings when interacting with someone else who would not have anymore the same representation of the given interaction protocol.

The interactions with other cognitive science disciplines. By making all our assumptions and choices in building our interaction language and interaction protocols, we simply want, from the computer scientist point of view, to build a complete and coherent computational framework in which our cognitive agents are able to interact. Our approach tries to be a pragmatic one. It uses results from other domains in Cognitive Science, neglecting some aspects that might not be useful from the computer science point of view, transforming others. At the same time, we hope that our approach, especially our proposal to deal with perlocutionary acts through the explicit exchange of a contextual protocol, or our distinction between interactions and organisations in the MAGMA approach, which is rather specific and unique in the MAS domain, will not only be developed further in the computer science domain, but will also generate fruitful exchanges with scientists from other cognitive sciences.

Towards agent oriented programming. • In the context of agent orient programming, and in difference with approaches such as the one developed by Shoham [Shoham 92], we are more general at the declarative level, because we do not assume any a priori model of agents, environments, interactions, or organisations, and we do believe that most of the applications of multi-agent systems do not require the use of a sophisticated BDI agent architecture. At the opposite, and contrarily to AOP, the interaction language we are tending to is not fully computational without the knowledge of the application domain. Despite of this, we believe that we dispose of a strong basis to be further developed in order to better model and deal with cognitive interactions between agents. • On the other hand, if we think to the proposed extension of using interaction protocols not only at a social control level (between agents) but also at the individual level (between sub-agents, the agent itself being considered as a society), as we have illustrated it by the work we are currently doing, we are really approaching the unavoidable recursion that should figure in any kind of agent-oriented programming languages, but that is not present by now in such so-called languages. Thus, we also strongly believe that our Interaction Language could also be a promising way to investigate further Agent Oriented Programming in .

Acknowledgements

This work would not have been possible without the scientific interactions and contributions of Olivier Boissier, Jean-Luc Koning, Michel Occello, Jaime Sichman, Marie-Hélène Stéfanini, and more generally of the past and current members of the MAGMA group who have contributed and who contribute to the theoretical studies in cognitive MAS, and who participate in the development of the MAGMA integrative environment. I am also very grateful to Christian Brassac, Cristiano Castelfranchi, Jean-Pierre Müller, Kurt Sundermeyer, Daniel Vanderveken and Walter Van de Velde for the influence they have on this work. The CNRS, Dr. Philippe Jorrand, the GDR-PRC IA, and the GDR PRC-CHM are also thanked for their support in this work.

References

[Baeijs 95] C. Baeijs, Y. Demazeau & L. Alvares "Application des Systèmes Multi-Agents à la Généralisation Cartographique", 3èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents, AFCET & AFIA, Chambéry, 1995.

[Ballmer 81] T. Ballmer & W. Brennenstuhl "Speech Act Classification: A study in the Lexical Analysis of Speech Activity Verbs", Springer Verlag, 1981.

[Berthet 92] S. Berthet, Y. Demazeau & O. Boissier "Knowing Each Other Better", 11th International Workshop on Distributed Artificial Intelligence, Glen Arbor, 1992.

[Bijnens 94] S. Bijnens, W. Joosen & P. Verbalten "Language Constructs for Coordination in Agent Space", 6th MAAMAW workshop, Demazeau, Perram & Müller eds., University of Odense, 1994.

[Boissier 92] O. Boissier & Y. Demazeau "A Distributed Artificial Intelligence View on General Purpose Vision Systems", in Decentralized A.I. 3, Demazeau & Werner, eds., North-Holland, Elsevier, Amsterdam, 1992.

[Boissier 94a] O. Boissier, Y. Demazeau, G. Masini & H. Skaf "Une architecture multi-agents pour l'implémentation du bas niveau d'un système de compréhension de scènes", 2èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents, AFCET & AFIA, Voiron, 1994.

[Boissier 94b] O. Boissier & Y. Demazeau "ASIC: An Architecture for Social and Individual Control and its Application to Computer Vision", 6th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Odense Science Park, Odense, 1994.

[Boissier 94c] O. Boissier & Y. Demazeau "MAVI: A Multi-Agent system for Visual Integration", 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Las Vegas, 1994.

[Brassac 94] C. Brassac & V. Chevrier, "Vers un examen du statut de l'interaction dans les systèmes multi-agents", Pragmatiques, Editions de l'Harmattan, 1994.

[Burmeister 93] B. Burmeister, A. Haddadi & K. Sundermeyer "Generic Configurable Cooperation Protocols for Multi-Agent Systems", 5th MAAMAW workshop, Ghedira and Sprumont eds., University of Neuchâtel, 1993.

[Campbell 92] J. Campbell & M. d'Inverno "Knowledge Interchange Protocols", in Decentralized A.I., Demazeau & Müller, eds., North-Holland, Elsevier, Amsterdam, 1990.

[Cardozo 93] E. Cardozo, J. Sichman, Y. Demazeau "Using the Active Object Model to Implement Multi-Agent System", 5th IEEE International Conference on Tools with Artificial Intelligence, Boston, 1993.

[Chang 92] M. Chang & C. Woo "SNAP: a communication level protocol for negotiations", in Decentralized A.I. 3, Demazeau & Werner, eds., North-Holland, Elsevier, Amsterdam, 1992.

[Demazeau 90] Y. Demazeau & J.-P. Müller "Decentralized A.I.", in Decentralized A.I., Demazeau & Müller, eds., North-Holland, Elsevier, Amsterdam, 1990.

[Demazeau 91a] Y. Demazeau "Coordination Patterns in Multi-Agent Worlds: Application to Robotics and Computer Vision", IEE Colloquium on Intelligent Agents, IEE, London, 1991.

[Demazeau 91b] Y. Demazeau & J.-P. Müller "From Reactive to Intentional Agents", in Decentralized A.I. 2, Demazeau & Müller, eds., North-Holland, Elsevier, Amsterdam, 1991.

[Demazeau 93] Y. Demazeau "La Plate-forme PACO et ses Applications", 2ème Journée Nationale du PRC-IA sur les Systèmes Multi-Agents, PRC-IA, Montpellier, 1993.

[Demazeau 94a] Y. Demazeau, O. Boissier & J.-L. Koning "Interaction Protocols in Distributed Artificial Intelligence and their Use to Control Robot Vision Systems", 6th International Conference on Artificial Intelligence and Information-Control Systems, Smolenice, Plander ed., World Scientific Publishing, 1994.

[Demazeau 94b] Y. Demazeau, O. Boissier & J.-L. Koning "Using Interaction Protocols to Control Vision Systems", 1994 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, 1994.

[Durfee 90] E. Durfee & T. Montgomery "A Hierarchical Protocol for Coordinating Multi-Agent Behaviour", AAAI National Conference, 1990.

[Ferber 91] J. Ferber & E. Jacopin "The framework of Eco-Problem-Solving" in Decentralized A.I. 2, Y. Demazeau & J.-P. Müller, eds., North-Holland, Elsevier, Amsterdam, 1991.

[Ferrand 94] N. Ferrand & Y. Demazeau "Multi-Agents et Multi-Décision en Aménagement du Territoire", 2èmes Journées du PIR Environnement du CNRS, Montpellier, 1994.

[Finin 94] T. Finin, R. Fritzon, D. McKay & R. McEntire "KQML as an Agent Communication Language", 3rd International Conference on Information and Knowledge Management, ACM, 1994.

[Gaspar 91] G. Gaspar "Communication and Belief Changes in a Society of Agents: Towards a Formal Model of Autonomous Agent", in Decentralized A.I. 2, Demazeau & Müller, eds., North-Holland, Elsevier, Amsterdam, 1991.

[Genesereth 94] M. Genesereth & S. Ketchpel "Software Agents", Communications of the ACM, vol. 37, 1994.

[Hassoun 92] M. Hassoun, Y. Demazeau & C. Laugier "Motion control for a car-like robot: potential field and multi-agent approaches", IEEE International Conference on Intelligent Robots and Systems, IROS 92, Raleigh, 1992.

[Koning 95] J.-L. Koning, Y. Demazeau, B. Esfandiari, J. Quinqueton, "Interaction entre Agents pour la Supervision de Réseaux en Télécommunications", 3èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents, AFCET & AFIA, Chambéry, 1995.

[Occello 94] M. Occello & Y. Demazeau "Building Real Time Agents using Parallel Blackboards and its use for Mobile Robotics", 1994 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, 1994.

[Pleiad 92] Pôle PLEIAD "Vers une Taxinomie du Vocabulaire pour les Systèmes Multi-Agents", 1ère Journée Nationale du PRC-IA sur les Systèmes Multi-Agents, PRC-IA, Nancy, 1992.

[Populaire 93] P. Populaire, Y. Demazeau, O. Boissier & J. Sichman "Description et Implémentation de Protocoles de Communication en Univers Multi-Agents", 1ères Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents, AFCET & AFIA, Toulouse, 1993.

[Quinqueton 91] J. Quinqueton, P. Reitz & J. Sallantin "Les schémas mentaux, un cadre conceptuel pour l'apprentissage a partir d'exemples", RFIA 91, Lyon, 1991.

[Reitz 93] P. Reitz & J. Quinqueton "Mosca: a multi-agent learning framework", ECML, 1993.

[Searle 69] J. Searle, "Speech Acts", Cambridge University Press, 1969.

[Shoham 92] Y. Shoham "Agent Oriented Programming", Artificial Intelligence, 1992.

[Sian 91] S. Sian "Adaptation based on Cooperative Learning in Multi-Agent Systems", in Decentralized A.I. 2, Demazeau & Müller, eds., North-Holland, Elsevier, Amsterdam, 1991.

[Sichman 94] J. Sichman, R. Conte, Y. Demazeau & C. Castelfranchi "A Social Reasoning Mechanism based on Dependence Networks", 12th European Conference on Artificial Intelligence, ECAI 94, Wiley, Amsterdam, 1994.

[Smith 80] R. Smith "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", IEEE transactions on Computers, Vol. 29, IEEE, 1980.

[Stéfanini 93] M.-H. Stéfanini & Y. Demazeau "Vers une architecture multi-agents pour le traitement automatique des langues naturelles", 1ères Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents, AFCET & AFIA, Toulouse, avril 1993.

[Vanderveken 94] D. Vanderveken "The Logic of Speech Acts", Cambridge University Press, 1994