FIPA Standardization Activities in the Software Engineering Area

Massimo Cossentino
Istituto di Calcolo e
Reti ad Alte Prestazioni (ICAR)
Consiglio Nazionale
delle Ricerche (CNR)
Viale delle Scienze,
90128 - Palermo (Italy)
cossentino@pa.icar.cnr.it

Gabriel Hopmans
Morpheus Software
P.O. Box 240
6200 AE Maastricht (The Netherlands)
also with
Communications Research & Semiotics
University Maastricht
6200 MD Maastricht (The Netherlands)
g.hopmans@mssm.nl

James Odell
James Odell Associates
3646 W. Huron River Drive
Ann Arbor
MI 48103-9489 (USA)
email@jamesodell.com

Abstract*— The Foundation of Intelligent Physical Agents (FIPA) aims at producing software standards to enable interworking between heterogeneous interacting agents and agent-based systems and one of its bodies is the Technical Committee. These Committees are devoted to produce specifications that will become part of the FIPA standards This paper describes the activities of three new FIPA Technical Committees: Interaction Protocols, Methodology and Modeling. The Interaction Protocols TC has the goal to produce new and more versatile conversation policies between agents and agent-based sytems. This paper describes one new proposed FIPA Interaction Protocol: Borda Count, which can be used to reach consensus between voting agents. The Methodology TC activity is devoted to the identification of a design methodology for MASs that could fit the greatest number of needs. The proposed approach is based on the method engineering that consists in the creation of a sort of meta-methodology that could be instantiated with a specific methodology for each problem. The Modeling TC is dealing with drawing the specifications for the future FIPA agent-based unified modeling language and in its work is starting from the existing experiences of UML and AUML. The main scope of this TC, in the first phase, will be related to some specific aspects of agent modeling such as structural and interaction notations. The activities of the three TCs are strongly interconnected and contain several overlaps. Their common aim is to contribute to the FIPA mission providing a substantial increase in the diffusion of agent-oriented solutions with the greater support given to the designer of agent systems.

I. Introduction

FIPA (Foundation for Intelligent Physical Agents) is a non-profit association whose mission is "The promotion of technologies and interoperability specifications that facilitate the end-to-end inter-working of intelligent agent systems in modern commercial and industrial settings".

The standardization activity is coordinated by the FIPA Architecture Board that has the responsibility of approving the work plans of the Working Groups and Technical Committees. While the Working Groups (such as the Ad Hoc) are mainly interested in the expression of informative specifications and the identification of application and test fields, the Technical Committees are devoted to the production of normative specifications. Seven of these committees are active by now, they are: Interaction Protocols, Methodology, Modeling, Ontologies, Security, Semantics, Services. Three of them (Interaction Protocols, Methodology and Modeling) belong to the software engineering area and in the following sections we will describe their activity and the related research and standardization problems.

The paper is organized as follows: in section two the activities of the Interaction Protocol TC (Technical Committee) will be discussed with a particular attention for the Borda Count proposed interaction protocol; in section three the work of the development methodology TC will be illustrated examining the problems related to the identification of a meta-model for the multi-agent systems and the related development methodologies. Section four contains the description of the modeling language technical committee that is currently drawing the first specifications for the AUML sequence and class diagrams. Finally in section five, some conclusions are drawn.

II. Interaction Protocols

The Interaction Protocol Technical Committee is established with the main goal to find a broader audience that has an interest in the development of new IP's. Activities of the TC are mainly related to new FIPA Interaction Protocols but the TC is also busy with the older ones like the IP library specification. The Interaction Protocols TC is working on a roadmap how FIPA should start a second generation of new Interaction Protocols. Several papers have shown the importance of Interaction Protocols and that for the medium-term future language and protocols will be more agreed and standardized. (See [1], [2], [3]) The IP TC is collecting common conversation policies for agents. During the FIPA eXperimental to Standard Technical Committee sessions it has been decided that the Interaction Protocol Library Specification needs an update. The IP TC is working on extending and updating this specification. This work has to be done in cooperation with the Modelling TC that is now working on the Interaction Diagrams of the Interaction Protocols and with the Methodology TC. For the IP library specification input upon how to model the conversation policies is needed. Also discussion about a need for associations between agent actions and agent capability ontology can result in valuable input for the IP library specification. One of the possible candidates for a new Interaction Protocol is the Borda Count Interaction Protocol. The term "Borda Count" is derived from the mechanism proposed by Borda [4], who recommended this election system that gave a better representation of what the people really want (better than the 'one man, one vote' system and the pairwise comparison).This protocol can be described as a mechanism that defines in principle that points are allocated to alternative strategies. In a collection of X alternatives X points will be allocated to the most preferred strategy, X-1 to the next best, and so on down to the least preferred strategy, which is allocated one point. The protocol requires that all voters have to rank their preferences among the X alternatives. The protocol is used then at a central location to add up the allocated points. The preferences are collected centrally to rank the scores given to each strategy, and to select the strategy with the maximum score as the winner. The Borda Count mechanism is identified as the unique voting method to represent the true wishes of the voters. [5] A new Interaction Protocol has been recently presented that is elaborately described and contains previous research on voting schemas in order for agents to arrive at a consensus choice or joint decision. The next subsection describes the draft of the Borda Count Interaction Protocol which is a first candidate for a new FIPA IP specification. The Borda Count mechanism is in general used in Politics and Economics but previous research and other agent-based projects also showed the relevance of the Borda Count mechanism (genetic algortithms, meeting scheduler, recommender system [6]). See for more details chapter 5 in [7]. The Borda Count proposal needs to be discussed in a broader audience soon to recognize the prerequisites to start the next generation of new FIPA Interaction Protocols. In the Borda Count Protocol one defines a winner in a set of alternatives and this set has been made known in advance. The task/action to be performed is ranking the alternatives and this is send as a request message to n agents by the Initiator agent. The participating agents receiving the request are able to generate proposals to perform the requested action/task as propose acts. Alternatively, the other agents may refuse to propose or send a not-understood. Once the deadline passes; the Initiator checks any received proposals. When the proposals send by the voting agents are valid the Initiator sends an accept-proposal (to k agents, where k can be equal to n if they are all valid) otherwise it will send a reject-proposal. After the initiator has received enough votes he starts to make the Borda Count calculation. The Borda Count calculation, as described above, takes place between the propose- and inform- acts in figure 1. The Borda Count calculation gives the Borda Count winner that represents the winning candidate (alternative). When the calculation is made by the Initiator he wants to propose this final result to the other agents as an inform act. The result is the winning alternative, which can be performed as action, since it represents an strategy. When the number one preference (the Borda Count winner) of the group is calculated the Initiator can refine it by iterating the mechanism . The Participants can have the ability to make another proposal when the Initiator decides to make a new request. (request 2 in the figure) It is upon the Initiator if he wants to refine the joint decision, if not he sends an inform message with the winner. The representation of the Borda Count IP is given in Figure 1.

III. Development Methodology

The FIPA Methodology Technical Committee work starts from the consideration that almost twenty different design methodologies can be found in literature for MASs (Multi-Agent Systems). This is the unquestionable prove that agents designer (just like objects designer) in accomplishing their different tasks, and solving specific problems in distinct production environments, often prefer to setup an own methodology specifically tailored for their needs instead of reuse existing ones. What seems to be widely accepted is that an unique specific methodology cannot be general enough to be useful to everyone without some level of personalization.

In this scenario we can identify two contrasting elements: first, in an interesting paper on object-oriented software (development) processes, Fuggetta [8] states that the research in this field is stuck and most technologies developed by the software process community have not been adopted by the industrial world. Second, the AgentLink community, in its roadmap [1] for agent based computing, embodying the feelings of a large part of the agent research and industrial community, has identified the designation of a standard in design methodologies as an essential demand.

In order to accomplish this request without neglecting the important warnings coming from the OO world we want to propose a quite open approach that allows the composition of a very large repository of human experiences (design process is first of all an human process) that could be expressed in terms of a standard notation (we are trying to refer to existing standards from OO and in case extend them).

We believe that agent-based systems are by themselves one of the aspects of the OO crisis solution and we are also strongly persuaded that the future standard in design methodologies will be a significant improvement in the agent research and industrial applications. The benefits will not only be limited to the dimension of the MAS that could be managed (this is not a secondary aspect if we think about the growing dimension of agent societies) but they will also be related to the possibility of integrating existing (even not agent-based) systems with new features on the fly [9].

In order to take advantage of the experiences done with existing methodologies we will adopt the method engineering paradigm. According to this approach, the development methodology is built by the developer (or by a method engineer) assembling pieces of the process (method fragments) from a repository of methods built up taking pieces from existing methodologies (Adelfe, AOR, Gaia, MESSAGE, PASSI,

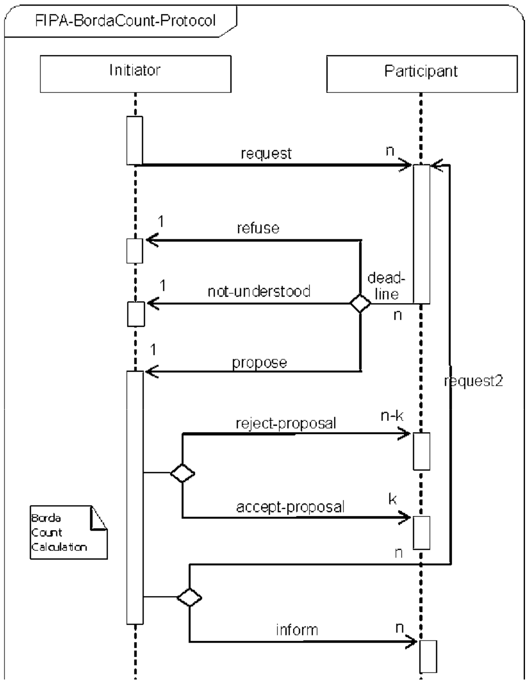

Fig. 1. FIPA Borda Count Interaction Protocol

Tropos, ...). In this way he/she could obtain the best process for his/her specific needs.

In the last years, the method engineering approach proved successful in developing object oriented information systems [10]. Its importance in the OO context should be evaluated considering not only the direct influence (not so much companies and individuals work in this specific way) but the indirect consequence that now, the most important and diffused development processes (for example RUP, the Rational Unified Process) are not rigid but they are a kind of framework within which the single designer can choose his/hers own path.

It could seem that introducing the method engineering paradigm in the AOSE context is a plain operation. But it is not so, because in the OO context the construction of method fragments (pieces of methodology), the assembling of the methodology with them and the execution of the design rely on a common denominator, the universally accepted concept of object and related model of the object oriented system. In the agent context, there is not an universally accepted definition of agent nor it exists any very diffused model of the multi-agent system.

We think that designing a system (object or agent-oriented) consists in instantiating the system meta-model that the designer has in his/hers mind in order to fulfill the specific problem requirements. This meta-model is the critical element in applying the method engineering paradigm to the agents world.

Referring to a MAS meta-model we mean a structural representation of the elements (agent, role, behavior, ontology,...) that compose the actual system with their composing relationships. Sometimes we can see that these concepts, for example the behavior, are used with slightly different meanings or granularity. The usefulness of the proposed methodological approach, therefore, depends on the availability of a standard definition of the MAS structure. As discussed below, we are currently exploring the process description of some diffused MAS design methodology with SPEM. Having these, we will be able of reasoning on the differences in the MAS meta-models that are present in the different approaches (not all of them specifically address FIPA agents). At the end we will (hopefully) obtain a large omni comprehensive meta-model of the MASs. This will include generic elements (e.g. the agent) but also specific ones (e.g. the cooperative agent referred in the ADELFE methodology).

During a real design process (Figure 2), the designer (or better the method engineer), before building his/hers own methodology, has to select the elements that compose the meta-model of the MAS he/she will build. In so doing he/she uses a CAME tool (Computer Aided Method Engineering tool) that offers a specific support for the composition of a methodology from existing fragments or with new ones. The availability of the MAS meta-model will help him/her both at a logical and practical level. First this will be useful in the method fragment selection phase (avoiding the selection of methods referring different elements) and secondly, the same fact of clearly declaring the structure of the system will allow the CASE tools to check for model coherence and to find not completely defined parts. Once composed the methodology the designer will perform the established process obtaining a model of the system that solves his/hers problem. Finally

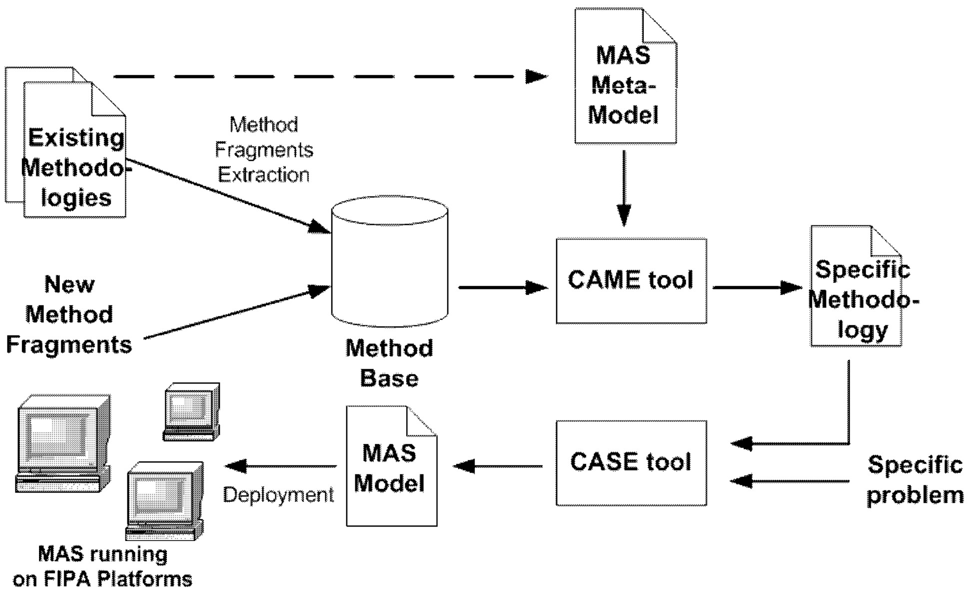

Fig. 2. The method engineering process adopted in the FIPA methodology technical committee

he/she could deploy the agents on the required platforms.

A. The Methodology TC Activity

The first step of this TC work will consist in the creation of the meta-model that will be used to describe the existing methodologies and the multi-agent system structure (MAS meta-model). An important contribution to this approach comes from an OMG specification, the Software Process Engineering Metamodel (SPEM). This is the natural candidate to be the meta-model adopted in this TC activity for processes since it is already an accepted standard in the OO context (and OO process are not too different from the AO ones). Moreover from the analysis of many existing approaches to these problem, Fuggetta [8] says that Process Modeling Languages "must be easy to use, intuitive, and tolerant"; all of these are properties that we can find in SPEM.

We are currently evaluating the possibilities offered by SPEM in the specific agent-oriented context obtaining interesting results. Several existing methodologies are being described with this meta-model and if specific needs will emerge we will evaluate the possibility of extending it.

From the descriptions of methodology processes we will extract the method fragments. A method fragment is a reusable part of a design process that taking some already designed pieces of the system produces a new part of the design following a precise procedure. These fragments will be collected in a method base. This introduces the second step of the methodology TC plan: the study of possible technological solutions for the implementation of this database in order to obtain a representation of the fragments that could be easily supported in a CASE/CAME (Computer Aided Software Engineering/Computer Aided Method Engineering) tool.

About the MAS structure meta-model we are currently collecting stimuli from the observation of the existing methodologies and from this, we will draw the first proposal that will be probably expressed in terms of an UML structural diagram.

The last crucial phase of the work will be the study of the method fragments composition strategies. Each method fragment produces an artifact that contributes to the construction of the complete MAS design model. Composing fragments coming from different methodologies implies considering that they may refer to different models of the system. For example they could address the concept of role in a slightly different way and as a consequence, reusing roles defined with one methodology in another context could bring to inconsistent or incomplete models. Anyway, this artifact structure is only one of the aspects of the problem we are dealing with. There is also a procedural point of view. In taking two method fragments from a repository and reusing them, we could find that they do not exactly match. It could be necessary to integrate them with some more activities that should complete the process.

In our approach we will face both the problems giving the right importance to the MAS meta-model and using it as a beacon for orienting the choices. The activity is now undergoing and the partial result consists of:

- A glossary including 86 proposed terms with a proposed definition for almost half of them.
- A first study on the MAS meta-model that describes the logical structure of the MAS to be designed with some methodologies.
- The process description (with SPEM) of some methodologies

Obviously all the method fragments will be converted to the use of the currently under development FIPA modeling language and this will facilitate their integration and will minimize the effort needed to comprehend and evaluate a larger number of possible solution strategies.

IV. Modeling Language

Multiagent systems (MAS) are often characterized as extensions of object-oriented systems. This overly simplified view has often troubled system designers as they try to capture the unique features of MAS systems using OO tools. In response, an agent-based unified modeling language (AUML) is being developed. The FIPA Modeling Technical Committee's goal is to be domain independent. Currently, it will examine those area where it has expertise: service-oriented architecture (SOA), business process management (BPM), simulation, real-time, AOSE, robotics, information systems. Other areas will be examined over time as further expertise becomes available. Instead of reliance on the OMG's UML, we intend to reuse of UML wherever it makes sense. We do not want to be restricted by UML; we only want to capitalize on it where we can. The general philosophy, then, is: When it makes sense to reuse portions of UML, then do it; when it doesn't make sense to use UML, use something else or create something new. Agent UML (AUML) synthesizes a growing concern for agent-based modeling representations with the increasing acceptance of UML for object-oriented software development.

A. Agent Software Modeling

The agent R&D community is increasingly interested in design methods and representational tools to support the associated artifacts. Multi-Agent System Engineering has been the focus of workshops at many agent conferences beginning with ATAL'97 and proceeding to current conferences, such as AAMAS 2003, WOA 03, and MATES 03. In response to this interest, the Modelling TC participants have initially identified sources of notations that should be considered for a FIPA agent-based unified modeling language (AUML). These current list of sources is as follows:

- UML 2.0
- AOR
- PASSI
- MESSAGE
- Tropos
- Adelfe
- Gaia
- BRIC
- Styx
- Prometheus
- Madkit
- OPM

This wide-ranging activity is a healthy sign that agent-based systems are having an increasing impact, since the demand for methodologies and artifacts reflects the growing commercial importance of our technology. Our objective is not to compete with any of these efforts, but rather to extend and apply a widely accepted modeling and representational formalism (AUML)-one that harnesses insights and makes them useful for communicating across a wide range of research groups and development methodologies.

B. UML

During the seventies, structured programming was the dominant approach to software development. Along with it, software engineering technologies were developed in order to ease and formalize the system development lifecycle: from planning, through analysis and design, and finally to system construction, transition, and maintenance. In the eighties, object-oriented (OO) languages experienced a rise in popularity, bringing with it new concepts such as data encapsulation, inheritance, messaging, and polymorphism. By the end of the eighties and beginning of the nineties, a jungle of modeling approaches grew to support the OO marketplace. To make sense of and unify these various approaches, an Analysis and Design Task Force was established on 29 June 1995 within the OMG. By November 1997, a de jure standard was adopted by the OMG members called the Unified Modeling Language (UML). This work has been recently extended in an effort to produce the next release, called UML 2.0. UML 2.0 was recommended by the OMG for standardization on 6 June 2003.

The UML unifies and formalizes the methods of many approaches to the object-oriented software lifecycle and supports the following kinds of models:

- **static models**- such as class, component, and package diagrams describe the static semantics of data and messages. Within system development, class diagrams are used in two different ways, for two different purposes. First, they can model a problem domain conceptually. Since they are conceptual in nature, they can be presented to the customers. Second, class diagrams can model the implementation of classes-guiding the developers. At a general level, the term class refers to the encapsulated unit. The conceptual level models types and their associations; the implementation level models implementation classes. While both can be more generally thought of as classes, their usage as concepts and implementation notions is important both in purpose and semantics. Package diagrams group classes in conceptual packages for presentation and consideration. Aggregations of classes are called components which can be expressed at both the conceptual and implementation levels of UML 2.0.
- **dynamic models**- including interaction diagrams (i.e., sequence, communication diagrams, and interaction overview diagrams), state charts, and activity diagrams.
- **use cases**- the specification of actions that a system or class can perform by interacting with outside actors.
- **implementation models**- such as physical component models and deployment diagrams describing the component distribution on different platforms.
- **object constraint language (OCL)**- is a simple formal language to express more semantics within an UML specification. It can be used to define constraints on the model, invariant, pre- and post-conditions of operations and navigation paths within an object net.
- **action models**- an action language would encompass both primitive actions and the control mechanisms pro-

vided by activities

C. AUML

Due to the extensive new work in improving UML, the FIPA Modeling TC has begun its work by examining the possibilities for the reuse and extension of UML for agent-based development. extensions to the following UML representations: packages, templates, sequence diagrams, communication diagrams, interaction overview diagrams, activity diagrams, class diagrams, deployment diagrams, and statecharts. Due to the current limitation of resources, the first phase is devoted to defining two modeling languages:

- **Class Diagrams** - Specify the internal behavior of an agent and relating it to the external behavior of an agent using and extending UML class diagrams.
- **Interaction Diagrams** - A generic term that applies to several types of diagrams that emphasize object interactions. These include collaboration diagrams, sequence diagrams, and the interaction overview diagram.

In addition, the Modelling TC participants have initially identified modelling areas that may be useful for representing and specifying agent-based systems. These areas are as follows:

- **Multi- vs. single agent** - The multiagent level of abstraction looks at several agents together with their relationships and/or interactions. The single-agent level of abstractions deals with one agent at a time. It describes the agent at the low, internal level before coding. For example, in a structural representation we have the agent main-class and the set of classes that realize the agent's tasks/behaviors .
- **Agent "class/component"** and implementation structure - *(subsumed on the Class Diagrams specification)*
- **Goal & soft goals** - Recommending representations of formalized goals and nonformal "soft goals" that can be useful for specifying agent-based systems. There are several ways in which goals might be expressed and several existing notations.
- **Use cases** - *(as specified in UML for now)*
- **Social aspects** - MAS system design can be inspired by human social phenomena. Furthermore, by computationally modelling social phenomena we can provide a better understanding of them. "Social" does not means only organization, roles, communication and interaction protocols, norms (and other forms of coordination and control); but it should be taken also in terms of spontaneous orders and self-organising structures.
- **Environment** - Without an environment, an agent is effectively useless. Cut off from the rest of its world, the agent can neither sense nor act. An environment provides the conditions under which an entity (agent or object) can exist. It defines the properties of the world in which an agent will function. Designing effective agents requires careful consideration of both the physical and communicational aspects of their environment.
- **Workflow/Planning** - MAS planning has been in the literature for over 15 years. However, there is no standard representation of it. By extending the UML 2.0 Activity Diagram, both planning and a MAS-based workflow approach can be expressed.
- **Levels of abstraction** - Agent systems can be seen by several angles. A possible view is how the system fits within the requirements. A second view would be how the agents in this system coexist, or in other words the basis of their society. A third view is the actual composition of the agent itself. And finally, how these agents are constructed in the implementation sense (classes and data structures). The levels of abstraction, then, indicates clearly in which angle the diagram is to be seen, and furthermore to define how different levels relate across diagrams.
- **Temporal constraints** - temporal properties such as how long an interaction must take; see Live Sequence Charts, UTC120) and Domains of time (e.g., real time vs simulation time)
- **Deployment and Mobility** - Mobile agent programming has been mainly technology driven, with a focus on implementing mobile agent platforms and small programming applications. This group will work on an extension of UML to provides language concepts for modeling mobility in analysis and design phase.

V. Conclusions

In this paper we presented the activities undergoing in the software engineering area within FIPA (Foundation for Intelligent Physical Agents). They are mainly centered on the work plans of three technical committee (interaction protocols, methodology and modeling) whose final purpose is to prepare the future standards in their specific field. The FIPA Interaction Protocol TC activities are mainly based on the proposal for the Borda Count Interaction Protocol. A small discussion is open to improve the proposal so far and to distinguish a general version and an iterated version of the protocol. Although during the FIPA activity, several proposals have been made for new Interaction Protocols, the TC has not received any work or references yet; therefore the TC opened a discussion about association ontologies (based on the idea presented in [11]). One of the soon upcoming activities that has to done on the FIPA Interaction Protocol Library Specification since this specification is deprecated while it should describe the use of Interaction Protocols in general.

The Methodology TC work is, by now, mainly centered on the identification of the MAS meta-model and the description of some of the most diffused methodologies in SPEM (Software Process Engineering Metamodel) in order to extract the method fragments from them. This activity will be followed by a study of the technological solutions that could be used to realized the method fragments repository. Another crucial point will be the study of possible guidelines for the methodology creation via the fragments assembling. Several approaches exist in literature about OO systems but in our specific context others could be explored like some ontology related ones.

The FIPA Modeling TC activities are in the first phase of a multiphase effort. The first phase has been presented in this paper, where the primary work is being done on AUML extensions to Interaction and Class Diagrams. Future phases will include other UML-based diagrams, e.g., Activity Diagrams, Use Case Diagrams, Deployment Diagrams, and State Machine Diagrams. As mentioned earlier, the FIPA Modeling TC activities are not limited to UML for their inspiration: we intend to reuse of UML wherever it makes sense. There is already some research being conducted to model such notions as agent goals, organizational structures, role modeling. These areas are not adequately supported by UML to support agent-based development. The efforts of the FIPA Modeling TC are ongoing and therefore not fully planned. We wish we knew exactly what will be required to model agent-based systems. However, the field of agents and agent-based systems is still in its early stages. As we understand better how to think, communicate, and represent our notions of agents, we will then be develop an richer and more expressive AUML.

FIPA entered its second term but several standard bodies and initiatives (like W3C, Semantic Web) are not familiar enough with FIPA applications/standards that should provide them with any additional value yet. We think that the FIPA audience is still too restricted and an effort is necessary to increase it. Opening communications with a broader audience and discussing FIPA specifications in a larger context can lead to wider results that can give an even greater contribution to the agent systems diffusion. In this scenario, the core activity of the FIPA members could be more related to final decisions upon the standard specifications, while the production of proposals and drafts could involve non-FIPA-members as well.

We consider that the discussed activities and consequent future standards will be a concrete step toward the maturity of agent-oriented systems. New developers will be interested in the introduction of agent technologies in their work since these systems will benefit of mature and diffused application platforms, a sufficient number of versatile interaction protocols, a specific and standard modeling language and a wide set of method fragments that can be used to assemble a specific methodology. The resulting increase in the diffusion of agent-oriented approaches will create a new market for specifically conceived CASE/CAME tools that will support the designer in all his/her activities, from analysis to coding and deployment referring to shared and consolidated standard specifications. This will further improve the advantages in using MAS and a virtuous cycle will be established.

REFERENCES

[1] M. Luck, P. McBurney, and C. Preist, *Agent Technology: Enabling Next Generation Computing. A Roadmap for Agent Based Computing*, AgentLink, Ed., 2003, http://www.agentlink.org/roadmap.

[2] D. W. Levine, "Fipa architectural challenges, meshing with the web," October 2001, http://www.fipa.org/docs/input/f-in-00042/.

[3] Y. Labrou, T. Finin, and Y. Peng, "The current landscape of agent communication languages," *IEEE Computer Society*, vol. 14, no. 2, 1999.

[4] J. Borda, *Mmoire sur les lections au scrutin.* Histoire de l'Acadmie Royale des Sciences, Paris, 1781.

[5] D. Saari, *Geometry of voting, studies in Economic Theory 3*, N. University, Ed. Springer-Verlag, 1994.

[6] R. Mukherjee, P. S. Dutta, and S. Sen, "Movies2go - a new approach to online movie recommendation," http://www.lumio.com/lc/ijcai/papers/mukherjee.pdf.

[7] G. Hopmans, "F-one racing-mas & the borda count interaction protocol, co-operating agents to achieve a collective goal in formula one racing," Ph.D. dissertation, University of Maastricht, 2002, http://www.fipa.org/resources/details.html#F-OneRacing.

[8] A. Fuggetta, *Software Process: A Roadmap.* ACM Press, 2000, ch. The Future of Software Engineering.

[9] F. Zambonelli and H. V. D. Parunak, "Sign of a revolution in computer science and software engineering," in *3rd International Workshop on Engineering Societies in the Agents' World.* LNAI, Sept. 2002.

[10] M. Saeki, "Software specification & design methods and method engineering," *International Journal of Software Engineering and Knowledge Engineering*, 1994.

[11] C. Stephen, N. Mariusz, M. Purvis, and P. Hwang, "Ontologies for interaction protocols," in *Proceedings of the 2nd Workshop on Ontologies in Agent Systems, First International Joint Conference on Autonomous Agents and Multi-Agent Systems, (AAMAS 2002),Bologna, Italy*, 2002, http://www.fipa.org/docs/input/f-in-00076/f-in-00076.pdf.