Future Challenges for Autonomous Systems

Helder Coelho[1]

Abstract The domain of intelligent creatures, systems and entities is suffering today profound changes, and the pace of more than a hundred meetings (congresses, conferences, workshops) per year shows there is a very large community of interest, eager of innovations and creativity. There is now no unanimity and homogeneity of the crowd, no convergence on what concerns scientific or technological goals, and recent surveys offer us strange results about the desires of industry and academy. However, observing recent conferences, we can work out some tendencies and move toward the future, yet conflicts are present concerning the aims the multiple communities pursue because some themes are relevant for several communities. Also, interleaving of areas generates points of friction between what must be done next.

1 Introduction

The NASA space programme is a good example for checking why autonomous systems became important and an aid to achieve complex aims. The Deep Space One mission started in 1999 (see also the real-time fault diagnosis system on the space shuttle by Georgeff and Ingrand) to validate technologies (e.g. autonomy), to support non-pre-planned goals, to make tests with simulated failures, and to open new avenues for more advanced applications. Examples of human-level intelligence are made today with software agents, surrounded by complex environments, a completely different kind than the single-task or smaller-scale agents of yesterday.

An agent is the sum of a system´s knowledge (represented with particular constructs) and the processes that operate on those constructs. The agent research field, within computing sciences, aims to produce more robust and intuitive artifacts, with the ability to deal with the unexpected. The building of these systems has turned out to be more complex than desired. This has pushed the whole community to refocus on the problem of the construction of autonomous

[1] University of Lisbon, Campo Grande, 1749-066 Lisboa, Portugal
hcoelho@di.fc.ul.pt

and multi-agent systems (MAS) by adopting less traditional software engineering principles, such as agent oriented programming languages and advanced software frameworks. Usually an agent can carry out several tasks simultaneously. In JADE (Java Agent DEvelopment framework) the tasks that an agent performs are encapsulated in a class called behaviour. This class contains an action method which executes the task and as it can be executed several times there is another method to decide when this loop ends. An agent may contain several behaviours which are scheduled in a cooperative way, using a round-robin cyclic algorithm. At each instant, the scheduler picks a behaviour and executes its action method. If the behaviour ends, it is removed from the agent´s list of behaviours. Otherwise, the scheduler picks another behaviour and repeats the same process. In the past, all behaviours in JADE were scheduled with the same priority, so the programmer could not assign different priorities to behaviours to give them different processor times. However, there are situations in which the use of priorities is needed in order to achieve a desired functioning of MAS. Two new behaviours that allow the use of priorities in JADE were implemented recently by (Suárez-Romero et al, 2006).

MAS are seen today as an appropriate technology to develop complex distributed software systems. The development of such systems requires an agent-oriented methodology that guides developers to build and maintain such complex entities. From the software engineering point of view, the systematic development of software should follow several phases. The number and purpose of these phases depend on the problem domain and the methodology. It is easy to find requirement elicitation, system analysis, architectural design, implementation, and validation/verification stages in most software methodologies, but the main focus of most of the agent-oriented methodologies (GAIA, Prometheus, Tropos, MaSE, SODA, MAS-CommonKADS, INGENIAS) has been on the analysis and design phases, paying less attention to implementation and testing. So, efforts have been made on languages for MAS and platforms and tools for multi-agent development. Yet, more focus must be given to individual agents, their organisations and coordination mechanisms, and shared environments, and this direction of work implies a more mature methodology with guidelines, concepts, methods and tools to facilitate the development. As a consequence, a unifying perspective of design, implementation and verification is desirable.

The world has changed. Modern systems are now very different from traditional information systems. They became open, made with autonomous, heterogeneous parts interacting dynamically. On account of this, some of the old software techniques were forced to leave the scene and to allow agents, adaptive reasoning and interactions to take the stage (Wegner, 1997).

The field of autonomous agents and multi-agent systems (AAMAS), or agents worlds, covers paradigms for software engineering and tools for understanding human societies. It contains also issues of distributed and concurrent systems, artificial intelligence, economics, game theory and social sciences. This is a large space of interactions, an inter- and multidisciplinary space.

Moreover, in recent years, task and team work were recognized as separable. This conviction supported the inquiry around a particular task and the interactions among agents, and pushed the need for a central mechanism for controlling the behaviour of agents in open environments. Autonomic computing became an important research topic aimed at answering the needs of a new range of problem domains, electronic institutions (eAdministration and eGovernment) in web communities, online markets, patient care-delivery, virtual reality and simulation. Open worlds increased in number and complexity challenging agent architects to facilitate automated decision making by all sorts of agents, including robots.

As a matter of fact, throughout the 90s we observed a move from algorithms to interactive computation because there was an imperative shift from mainframes to networks, wireless devices and intelligent appliances, and, at the same time, from number crunching to embedded systems and graphical user interfaces, and, also, from procedure-oriented to object-based and distributed programming.

New (virtual) information spaces built around digital cities, a kind of web portal (e.g. PortEdu), combine social, political and economic activities, including online forms, (profit and non-profit) data services shopping, education (e.g. AMPLIA environment for Medicine) and entertainment. Government regulations extended laws with guidance to corporate and public actions, requested norms (obligations, prohibitions and permissions) for disciplining social interactions, and attracting advertisers and businesses among private and public companies. City informatization emerged as government initiatives and data services on accumulating urban information were requested. Otherwise, they recognize the relevance of usability, they are well maintained, use proprietary software and rely on powerful search engines.

Institutions have shown they are a mechanism to make agent interactions more effective, structured, coordinated and efficient. New tools were created to assist in the design and verification processes. For example, landmarks play a potential role in building a bridge between the rigidity of the protocols and the flexibility of norms for regulating the behavior of autonomous agents.

In negotiation, everybody tries to reach a profitable result. But, the outcome of future business, or the success of social turns, depends heavily on the relationships among agents, the trust they have on others and their own good reputation. Trust and reputation models, constructed with values from past interactions with the environment (and extra information) now influence the agent´s decision-making process, even facilitate dealing with uncertain data.

In connected communities, software entities act on behalf of users and cooperate with infocitizens. Therefore, issues related to social and organizational aspects of agency are of keen importance, specially the questions of individual power, associated with decision-taking and reasoning (management of organizations, enforcement of norms), and of (adjustable) autonomy of agents in crisis situations.

In the following, we start by enumerating the current positive and negative aspects, observe the state of the art of autonomous agents and multi-agent systems, and go deep into the field to discover emerging trends and future challenges.

2 Today's Potentialities and Difficulties

The history of computing is marked by five ongoing trends: ubiquity, interconnection, intelligence, delegation and human-orientation, which are all associated with agents, a sort of computer entity capable of flexible, autonomous (problem-solving) action, situated in dynamic, open, unpredictable and distributed domains. Agents (synthetic and physical) are currently viewed as a metaphor for the design of complex distributed computational systems, as a source of technologies, and as simulation models of complex real-world systems, such as in biology or economics. In what concerns design, agents are mixed today with agent-oriented software engineering, architectures, mobile agents, infrastructures, and electronic institutions. The agent technologies are diverse, such as planning, communication languages, coordination mechanisms, matchmaking architectures, information agents (and basic ontologies), auction mechanism design, negotiation strategies, and learning.

There is a close connection between MAS and software architectures: a multi-agent system provides the software to solve the problem by structuring the system as a number of interacting autonomous entities embedded in an environment in order to achieve the functional and quality requirements of the system, and a software architecture is considered as the structure of a system which comprises software elements and the relationships among the elements. So, MAS are a valuable way to solve software intrincacies in a large range of possible directions to face problems. Typical architectural elements of a multi-agent system are agents, environments, resources, or services. The relationships between those elements are very diverse, ranging from environment mediated interaction between cooperative agents via virtual trails to complex negotiation protocols in a society of self-interested agents. This analogy suggests the integration of MAS with mainstream software engineering. Meanwhile current practice considers it as a radically new way of doing software.

MAS engineering has not yet addressed the issues of autonomy and interoperability in depth because FIPA-ACL infrastructures have adopted ad-hoc and developer-private communications assumptions, based upon reasons of communication efficiency or developer convenience. So the problem of interaction exists (agents are often hand-crafted) and requires new directions apart from those already adopted by the tools Jadex and Jason.

In the aftermath of a large scale disaster, agents' decisions range from individual (e.g. survival) to collective (e.g. victims' rescue or fire extinction) attitudes, thus shaping a 2-strata decision model (Silva and Coelho, 2007). However, current decision-theoretic models are either purely individual or purely collective and find it difficult to deal with motivational attitudes. On the other hand, mental-state based models find it difficult to deal with uncertainty. A way out is the making of hybrid decision models: i) the collective versus individual (CvI), which integrates both strata quantitative evaluation of decision making, and ii) the CvI-JI which extends the CvI model, using the joint-intentions formulation of teamwork, to deal with collective mental-state motivational attitudes. Both models have been evaluated from an experimental and case study based outlook that explores the tradeoff between cost reduction and loss of optimality while learning coordination skills in a partially observable stochastic domain.

At the end of the 80s, agents were capable of playing the role of intentional systems, due to the Bratman´s BDI (Belief-Desire-Intention) model, based upon Folk Psychology where behaviour is predicted and explained through the attribution of attitudes, such as believing, desiring, wanting, hoping, etc. Several BDI architectures (PRS, IRMA, dMARS) were implemented and explored in real problem-domains where issues such as organizations, interaction, cooperation, coordination, communication, competition or negotiation are the most interesting features.

The industrial strength of software is marked by a fundamental obstacle to take-up, the lack of mature agent software methodology, where joint goals, plans, norms and protocols regularly support coordination and interaction in organizations, and libraries of agent and organization models, of communication languages and patterns, or ontology patterns are available.

Over the past 20 years of research, logical theories of intelligent (rational) agents have been improved and refined and have served as the basis for executable specifications in order to implement diverse behaviours. Although this provides strong correctness, it became clear that those idealized specifications are inappropriate in practical situations because an agent has many resource-bounds (time and memory) to contend with. A way out is to represent and execute resource-bounded agents, and the reason justifying such a framework is the restriction of the amount of reasoning (temporal and doxastic) that the agent is allowed. For example, good candidates are the logic ATLBM (Alternating-time Temporal Logic with Bounded Memory), of Agotnes and Walther, able to describe the strategic abilities (decisions for certain circumstances) of coalitions of agents with bounded memory, imperfect recall and incomplete information (future inquiry: logical properties (complete axiomatizations) and computational complexity of the different logics), and the logic of situated-resource-bounded agents of Alechina and Logan.

The gap between realization (e.g. with BDI logic) and implementation (with AgentSpeak language) of intelligent agents remained unclear till today because both sides are superficially related: BDI notions in agent logics, modelled by abstract relations in modal logic, are not grounded in agent computations, and have no association with real behaviours. The current work of Meyer´s group at Utrecht University is around the quest for the holy grail to agent verification and intended to decrease the gap between the formal aspects and the realization/implementation of intelligent agents.

3 Technologies, Tools and Techniques

Theoretical and practical issues associated with the developing and deploying of multi-agent systems are now discussed within the interdisciplinary space of formal methods, programming languages, methodologies, techniques and principles (e.g. LADS´007 Workshop within the federated set of Workshops MALLOW). Along formal methods, current work includes the integration of two different concepts of belief (either as a probability distribution or as a logical formula) of Lloyd and Ng. Another good example of methodologies, in progress, is the MaSE for the development of embedded real-time systems of Badr, Mubarak and Göhner, well suited to aircrafts and industrial process controllers. Regarding tools and techniques, a modern example is associated with measuring the complexity of simulations of Klügl, because the variety of multi-agent applications is very large, in particular when compared with traditional simulation paradigms. The actual complexity of models is hidden, and there is a need to characterize it by introducing metrics for the properties of simulations. Finally, in what concerns programming languages, a nice example is the one on BDI agent organization by Hepple, Dennis and Fisher, because the BDI model style is common and there is agreement on the core functionality of agents. This research line aims at a unifying framework for the core aspects of agent organization, including groups, teams, roles and organizations as well. The purpose was to define a simple organizational mechanism, derived from the METATEM programming language, and show how several well known approaches can be embedded within it (the mechanism is intended to be independent of the underlying agent language).

Multi-agent systems (MAS) became one of the key technologies for software development today. This was achieved because the formal aspects of agency were deeply studied in a multi-disciplinary way with the help of logic and theoretical computer science (e.g. formal methods for verification and model checking, cooperation, planning, communication, coordination, negotiation, games, and reasoning under uncertainty in a distributed environment). Attacking agent organization is possible by the construction of a bridge between organizational theory and logics for social concepts governing MAS in open environments, as Virginia and Frank Dignum are doing. They are formalizing group organization capabilities and responsibilities, and relating two kinds of organizational structure, such as hierarchies and networks. Lorini, Herzig, Broersen and Trioquard are grounding power on actions and mental attitudes (beliefs, intentions) via Intentional Agency Logic.

MAS can be observed from the perspective of the Population-Organizational model, a minimal semantic model where the performance of organizational roles by agents and the realization of organizational links by social exchanges between agents are the key mechanisms for the implementation of an organization structure by a population structure. The structural dynamics of a MAS may then be modelled as a set of transformations on the system´s overall population-organization structure. Rocha Costa and Dimuro illustrated this approach by introducing a small set of operational rules for an exchange value based dynamics of organizational links.

In a case study of natural resources management, Adamatti, Sichman and Coelho proposed a new software architecture, called ViP-GMABS, which enabled virtual players to be associated in the GMABS methodology, a way to combine RPG (Role-Playing-Games) and MABS (Multi-Agent Based Simulation) techniques in an integrated way. The prototype ViP-Jogo Man was designed to be used as a group decision support system in a real problem domain because it encompasses complex negotiation processes and that methodology can be explored to handle conflict resolution.

All sorts of tools are thought to facilitate problem-solving and the engineering of good solutions for the owners of applications. Currently, in informatics at large, complex business tasks are turning into a popular issue, and increasingly there is a need for suitable aids to face several categories of domain-specific interventions. For example, web applications are frequent now in enterprises because they can be accessed via browsers in a standardized way. Also, the development of (mobile) agent-applications for open environments requires heterogeneous platforms to be interoperable, and agents are good for that particular context because they are able to execute and migrate through platforms, interact with each other independently from hosts, and interact with the agents of the application available on different platforms.

4 Emerging Trends: from AAMAS02 to AAMAS07

The main trends and drivers are now around the Semantic Web, Web services and service oriented computing, peer-to-peer computing, grid computing, ambient intelligence, self-systems and autonomic computing, and complex systems. There is choreography between technologies and simulation, having the design metaphor at the core of this space of conversations.

According to the AgentLink technology roadmap (2005) the current situation is associated with one design team, agents sharing common goals, closed agent systems applied to specific environments, ad-hoc designs, predefined communication protocols and language, and scalability only allowed in simulation (Luck, 2007). The projections are positive: 35% of software development will use agents and the rate of growth of adoption will increase until 2014 (penetration of 12% by 2010, or a third of long-run adoption level), yet object technologies will be more popular: not all applications of agent technologies will be coined as agent systems. So, in the short term we will have fewer common goals, use of semistructured agent communication languages (e.g. FIPA ACL), top-down design methodologies (Gaia) compared with today's ad-hoc designs, scalability extended to predetermined and domain-specific environments (today, scalability is only in simulation). In the medium term, we will have the design done by different teams, the use of agreed protocols and languages, standard agent-specific design methodologies, open agent systems in particular domains (bioinformatics, e-commerce), more general scalability, arbitrary numbers and diversity of agents in all such domains, and bridging agents translating between domains. However, we are far from design by diverse teams, truly-open and fully-scalable multi-agent systems, agents capable of learning appropriate communication protocols upon entry to a system, and protocols emerging and evolving through agent interactions.

Looking closer to the World Congresses AAMAS (from 2002 till 2007) we have great difficulties in analyzing the historical evolution of the field and in detecting shifts (new trends), because its organization is based upon possible sessions (space-time availability provided by local organizers) and not on existing tracks (or directions of research). Names attached to sessions are misleading and several papers were included without any reason (more care is required to prepare the sessions, and a new ontology of the field is mostly advisable). This is not the case for the International Joint Conference on AI, where areas are well fixed: from 1981 onwards the main three areas have been knowledge representation, reasoning and machine learning; the other ones go up and down according to their scientific potential and the financial aid assigned, allowing the community to infer the hot topics, shifts and trends.

The agents world is composed of multiple events, mainly workshops, where the community gets more or less strong links. There are also associated regions, like simulation, that adopt a similar form of organization without any well defined geography. For example, the fourth Conference of the European Social Simulation Association (ESSA´07, Toulouse, September 10-14, 2007) was structured into two components (plenary and parallel sessions) and along a continuum (queue) of themes, where social issues (resource sharing, reputation and communication, social influence, feature propagation in a population, market dimensionality, social power structures, historical simulation, rule changes, diffusion of innovation, impact of knowledge on learning, qualitative observation) and applications (policy, firms, organizations, economy, spatial dynamics, opinion and cultural dynamics) are overviewed.

In the mid-90s, there was an attempt made by Yves Demazeau to organize the whole field. He accepted the mainstream ideas of the objects and components movements, and advanced four units: agents (A) to cover the basic elements, environment (E) to cover the passive elements, interactions (I) to cover the means to exchange information and control among agents, and organizations (O) to cover the policies followed to constrain the interactions among agents up to 2003, he developed the idea of a MAS framework and methodology, based upon four vowels A, E, I and O, expanded later on with the U, for the users or the applications. The dynamics chart included the side of decomposition (identification, analysis), of modelling (choice, design) and of tools (programming).

The ATAL (Agents Theory, Architectures and Languages) Workshop was another attempt to characterize the field by dividing it into three main aspects (Wooldridge and Jennings, 1995): the agent contents, its organization and the available tools/infrastructures to build it up.

Another attempt, more general and abstract, to classify the research was made by (Sichman, 2003) along three independent directions (3D scenario): aims (resolution, simulation), description (theories, architectures, languages) and dimensions (agents, environments, interactions and organizations). The goal was to fix exactly each contribution (e.g. models, applications) in the context.

Luck and colleagues (2005), in the AgentLink roadmap, suggested also five components or good tracks of the Agents world: agent level (A), infrastructure and supporting technologies (E), interaction level (I), organization level (O), applications and industry (U), where A, I and O stand for the core set of technologies and techniques required to design and implement agent systems that are now the focus of current research and development.

We took the six conferences of AAMAS (Bologna, Melbourne, New York, Utrecht, Hakodate and Honolulu), from 2002 up to 2007, the whole list of themes (names of the sessions), and adopted the five vowels to analyze the evolution of Agents. Picking up the number of papers (we excluded the posters), attached to each theme, we pictured the state of the art along six consequent years.

The conclusions are the following: low relevance for tools (infrastructures and support technologies) and high relevance for applications (university and industry). The three agent technologies and techniques got 3/4 of the papers with agent or organization level topics at the top. For example, in AAMAS07 the ranking was: applications (25.6%), organization level (25.6%), agent level (23.3%), interaction level (22.6%), and infrastructures and support technologies (3%).

5 Future Challenges: Broad and Specific

Future challenges can be classified as broad or specific. Among the broad class, we may find creating tools, techniques and methodologies, automatic specification, development and management, integrating components and features, establishing trade-offs between adaptability and predictability, and linkage with other branches of computer science and other disciplines. Among specific challenges, we may find trust and reputation, virtual organization formation and management, resource allocation and coordination, negotiation, emergence in large-scale agent systems, learning and optimization theory, methodologies, provenance, service architecture and composition, and semantic integration.

Looking to the challenge of trust and reputation we find today: techniques for authentication, verification, validation; sophisticated distributed systems that involve action in absence of strong existing trust relationships; need to address problems of establishing, monitoring and managing trust; interactions in dynamic and open environments; and, trust of agents in agents (norms, reputation, contracts).

Organizing is a complex process, from the bottom-up or top-down point of view, and used in different contexts and applications. Reorganizing is a dynamic process with multiple dimensions and styles that can be predefined using an organizational language. Multi-agent organizations for current and next applications need to combine agent centered and organization points of view, to combine agent level and system level programming of organizational models, and to integrate and enforce dynamic and adaptive organizations.

An interesting direction is to design complex systems that evolve in changing environments with MAS. Self-organization is a promising paradigm to build up these systems adaptive where the collective function arises from the local interactions and the whole design grows bottom-up. The difficulty appears when discovering the right behaviours at the agent-level to make the global function emerge. Often, simulation helps designers to trap the correct behaviours during the design stage.

A complex problem was one of the most frequent targets chosen by the AAMAS-07 authors, potential candidates for the best paper. This is not strange because nowadays agent technologies are selected by large enterprises to help the problem-solving, yet building such a software system to achieve complex tasks is a rather difficult job because classical engineering techniques are not always useful. One of the hardest cases happens when the system must be composed of a large number of interacting entities and the global behaviour is said to emerge. The difficulty consists in linking the goal defined at the macroscopic level with the corresponding behaviour at the microscopic level.

Air traffic flow management is one of the fundamental challenges facing the Federal Aviation Administration (FAA) today. Finding reliable and adaptive solutions to the flow management problem is of paramount importance if the Next Generation Air Transportation Systems are to achieve the stated goal of accommodating three times the current traffic volume. This problem (Tumer and Agogino, 2007) is particularly complex as it requires the integration and/or coordination of many factors including: new data (e.g., changing weather info), potentially conflicting priorities (e.g., different airlines), limited resources (e.g., air traffic controllers) and very heavy traffic volume (e.g., over 40,000 flights over the US airspace). The air traffic flow simulator developed at NASA, used extensively by the FAA and industry, was selected to test a multi-agent algorithm for traffic flow management. An agent was associated with a fix (a specific location in 2D space) and its action consisted of setting the separation required among the airplanes going through that fix. Agents used reinforcement learning to set this separation and their actions speed up or slow down traffic to manage congestion. The results showed that agents receiving personalized rewards reduced congestion by up to 45% over agents receiving a global reward and by up to 67% over a current industry approach (Monte Carlo estimation).

Argumentation in agent systems has several hot research issues now, for example the study of protocol properties, combination of dialogue types (deliberation), multi-party dialogues, protocol design articulated with agent design, embedding in social context, and a framework for dialogue games.

Communication will open new avenues for research along the following directions: concepts for the semantics of both messages and conversations; standard languages for protocol description; tools for supporting a conversation policy construction and implementation; a complete semantics for agent communication; and, focus on multi-party communication.

Trust and reputation mechanisms are used to infer expectations of future behaviour from past interactions (Hermoso et al., 2006), for example in peer-to-peer systems. They are of particular relevance when agents have to choose appropriate counterparts for their interactions (e.g. reputation values about third parties), as it occurs with virtual (and regulated) organizations. The use of organizational structures helps to solve this difficulty and improves the efficiency of trust and reputation mechanisms by endowing agents with extra data to select the best agents to interact with. Certain structural properties of virtual organizations (e.g. governor agents, filters, protocols of sequential actions) able to limit the freedom of choice of agents are used to build an efficient trust model in a local way.

New applications need more research on mastering decentralized coordination and autonomic, adaptable systems, supporting heterogeneity, which involves dealing with semantics as a prerequisite for the quantum leap and based on infrastructure standards, modeling systems with parallel and possibly inconsistent goals and tasks, and exploration of Web 2.0 (e.g. 2^{nd} Life) as a playground for agents. The bet on ambient intelligence and Internet will be, for sure, a driver of new visions (decentralized setting, many devices, many users, dynamic sensor-actor networks, and a huge need for interaction).

Finally, there is a theme, serious games (SGs), that deserves to be put into focus, because it opens an interesting range of useful applications (e.g. e-democracy for integrated coastal area management, to support electronic debates and the implementation of contracts). SGs are a type of software developed with game technology and game design principles, for a primary purpose other than pure entertainment. Creating games that simulate functional entities (e.g. battles, but also processes and events at large) can be interesting not only for education (enterprise learning, professional training), but also for group-decision management (business intelligence), advertising, healthcare (see PortEdu portal, AMPLIA learning environment), public policy generation, or even politics. How can institutional or social power be structured, as part of the traditional hierarchies? Can organizational-dynamic games be of any utility for understanding democracy, the role of individuals and collectives and their relations with power? Machiavelli, Spinoza and Hobbes thought multitudes (a cooperative convergence of subjects) could be a substitute for groups dominated by a sole leader and support for constituent power. Subjects articulated via a network may generate a productive force and advance new ideas by working together, without any central boss. So, a whole of singularity becomes more productive (innovative) than the mass of individuals (representing a false unity).

The multitude or singular multiplicity presents advantages over other types of organization because each subject can speak for himself and cooperate with the others. As a matter of fact, the multitude is an active social agent, a multiplicity that acts (and works, disobeys) and a form of composition or contract among active individuals.

6 Conclusions

Distributed environments have grown dramatically creating specializations (e.g. various facets of multi-agent systems, semantic web communities) and technical (and technological) demands. It is important to identify key tracks and spaces of intersection between these expanding research areas, in order to discover synergies (points of confluence), redundancies, or difficult problems. If there is really a field of agents (at large), it will be desirable to draw a map containing its geographical elements (the so-called new set of categories to assemble the large queue of themes). So, it would be easier to organize new programmes of research, where strands of inquiry could be woven together to attack new problems. From example, the area of peer-to-peer knowledge sharing has a technological demand for more controllable, accessible, faster and larger scale systems. But this presents a challenge to traditional multi-agent coordination, ontology management and service architectures.

A nice example of a distributed environment is the case of autonomic electronic institutions which proved to be valuable to regulate open agent systems or agent societies. The rules of the game are fixed, i.e. it is known what agents are permitted and forbidden to do and under what circumstances, and the institutions can adapt their regulations to comply with their goals despite coping with varying populations of self-interested external agents. Self-management (organization, configuration, diagnosis, repair) becomes an interesting feature to face varying agents´ behaviours and helpful to support simulated agent populations.

The geography proposed by the AgentLink roadmap of (Luck et al., 2005) is so far the fastest way to analyze the evolution of the field of autonomous agents, and we applied it to the last two editions of AAMAS Conferences, where we found a stable situation concerning the main tracks of organization and agent levels.

There are increasing analogies between open problems in pairs of informatics areas, such as distributed systems and agent-based social simulation. Questions regarding micro to macro, cooperation and trust, or evolving network structures are central to software engineering, and modeling methodologies may be combined productively. Reputation, developed within electronic auctions, is also a key tool for partner selection, and can easily be exported into other domains from social networks to institutional evaluation. There is also interweaving of human and technological communities (e.g. ambient intelligence) during the construction of connected communities where software entities act on behalf of users and cooperate with infohabitants. These intersections occur often in the organization of conference sessions, where a paper could be classified under different categories and was inserted in a session on account of room availability. Such phenomena disturb the rankings and may falsify the conclusions.

Acknowledgements

A period of study at the Politechnic School of the University of São Paulo (USP) and the interaction with my colleague Jaime Sichman allowed the writing of this chapter. This work was financially supported by the LabMAg R&D unit, under the Portuguese FCT, and by the Brazilian CNPq BEV nº 170104/07-9. The chapter does not necessarily reflect the views of these two agencies.

References

Adamatti, D. F., Sichman, J. S., Coelho, H. : Using Virtual Players in GMABS Methodology: A Case Study in Natural Resources Management, Proceedings of the 4th Conference of the European Social Simulation Association (ESSA´07), Toulouse, September 10-14, 2007.

Agotnes, T., Walther, D.: Towards a Logic of Strategic Ability under Bounded Memory, Proceedings of the Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW´007), Languages, methodologies and Development tools for multi-agent systems (LADS´007) Workshop, September 3-7, Durham, 2007.

Alechina, N., Logan, B.: A Logic of Situated Resource-Bounded Agents, Proceedings of the Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW´007), Languages, methodologies and Development tools for multi-agent systems (LADS´007) Workshop, September 3-7, Durham, 2007.

Badr, I., Mubarak, H., Göhner, P.: Extending the MaSE Methodology for the Development of Embedded Real-Time Systems, Proceedings of the Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW´007), Languages, methodologies and Development tools for multi-agent systems (LADS´007) Workshop, September 3-7, Durham, 2007.

Costa, A. R., Dimuro, G. P.: Semantical Concepts for a Formal Structural Dynamics of Situated Multiagent Systems, Proceedings of the Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW´007), Coordination, Organization, Institutions and Norms in Agents Systems (COIN'007) Workshop, September 3-7, Durham, 2007.

Demazeau, Y.: Multi-Agent Systems Methodology, CNRS Leibniz IMAG, Rennes October 2, 2003.

Dignum, V., Dignum, F.: A Logic for Agent Organizations, Proceedings of the Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW´007), Formal Approaches to Multi-Agent Systems (FAMAS´007) Workshop, September 3-7, Durham, 2007.

Georgeff, M., Ingrand, F. F.: Real-time reasoning: the monitoring and control of spacecraft systems, Sixth Conference on Artificial Intelligence Applications, May 5-9, 1990.

Hepple, A., Dennis, L., Fisher, M.: A Common Basis for Agent Organisation BDI Languages, Proceedings of the Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW´007), Languages, methodologies and Development tools for multi-agent systemS (LADS´007) Workshop, September 3-7, Durham, 2007.

Hermoso, R., Bilhardt, H.. Centeno, R., Ossowski, S.: Effective Use of Organisational Abstractions for Confidence Models, Proceedings of the 7th International Workshop on Engineering Societies in the Agents World (ESAW06), G. O´Hare, A. Ricci, M. O´Grady and O. Dikenelli (Eds.), Dublin, September 6-8, 2006.

Klügl, F.: Measuring Complexity of Multi-Agent Simulations, an Attempt using Metrics, Proceedings of the Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW´007), Languages, methodologies and Development tools for multi-agent systemS (LADS´007) Workshop, September 3-7, Durham, 2007.

Lloyd, J. W., Ng, K. S.: Probabilistic and Logical Beliefs, Proceedings of the Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW´007), Languages, methodologies and Development tools for multi-agent systems (LADS´007) Workshop, September 3-7, Durham, 2007.

Lorini, E., Herzig, A., Broersen, J., Troquard, N.: Grounding Power on Actions and Mental Attitudes, Proceedings of the Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW´007), Formal Approaches to Multi-Agent Systems (FAMAS´007) Workshop, September 3-7, Durham, 2007.

Luck, M., McBurney, P., Shehory, O., Willmott, S. et al.: Agent Technology: Computing as Interaction, A Roadmap for Agent Based Computing, AgentLink III, 2005.

Luck, M.: General Introduction to Multi-Agent Systems, Tutorial slides of EASSS2007, Ninth edition of the European Agent Systems Summer School (EASSS), Durham, August 27-31, 2007.

Nakayama, L., Vicari, R. M., Coelho, H.: An Information Retrieving Service for Distance Learning, The IPSI BgDTransactions on Internet Research, Vol. 1, No. 1, 2005.

Sichman, J. S.: Raciocínio Social e Organizacional em Sistemas Multiagentes: Avanços e Perspectivas, Tese de Livre-Docência, Escola Politécnica da USP, Brasil, 2003.

Silva, P. T., Coelho, H.: An Hybrid Approach to Teamwork, Proceedings of VI Encontro Nacional de Inteligência Artificial (ENIA2007), XXVII Congresso da SBC, Rio de Janeiro, July 2-6, 2007.

Suárez-Romero, J., Alonso-Betanzos, A., Guijarro-Berdiñas, B.: Integrating a Priority-Based Scheduler of Behaviours in JADE, Proceedings of the 7th International Workshop on Engineering Societies in the Agents World (ESAW06), G. O´Hare, A. Ricci, M. O´Grady and O. Dikenelli (Eds.), Dublin, September 6-8, 2006.

Tumer, K., Agogino, K.: Distributed Agent-Based Air Traffic Flow Management, Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-07), Honolulu, July 16-18, 2007.

Wegner, P.: Models and Paradigms of Interaction, OOPSLA Tutorial Notes, 1995.

Wegner, P.: Why Interaction is More Powerful than Algorithms, Communications of the ACM, Vol. 40, Nº 5, pp. 80-91, 1997.

Wooldridge, M., Jennings, N.: Intelligent Agents: Theory and Practice, The Knowledge Engineering Review, Vol. 10, Nº 2, 1995.