Multi-Agent Systems and Complexity: Introduction

Peter McBurney

Department of Computer Science

University of Liverpool, UK.

In its brief history, computer science has enjoyed several different metaphors for the notion of *computation*. From the time of Charles Babbage in the nineteenth century until the mid-1960s, most people thought of computation as *calculation*, or operations undertaken on numbers. With widespread digital storage and manipulation of non-numerical information from the 1960s onwards, computation was re-conceptualized more generally as *information processing*, or operations not only on numbers, but also on text, sound and image data. This metaphor for computation is probably still the prevailing view among people who are not computer scientists. From the late 1970s, with the development of various forms of machine intelligence a yet more general metaphor of computation as *cognition* became widespread, at least among computer scientists. The fruits of this metaphor have been realized, for example, in the advanced artificial intelligence technologies which have now been a standard part of desktop computer operating systems, such as Microsoft Windows, since the mid-1990s. Intelligent, decision-making computer systems are now deployed across modern economies, from automated stock-trading systems to aircraft landing systems.

With the rapid growth of the Internet and the worldwideweb since 1992, we have reached a position where a new metaphor for computation is required: computation as *interaction.* In this metaphor, computation is something which happens by and through the communications which computational entities have with one another. Cognition and intelligent behaviour is not something which a computer does on its own, or not merely that, but is something which arises through its interactions with other intelligent computers to which is connected. *The network is the computer*, in SUN's famous slogan. This viewpoint is a radical reconceptualization of the notion of computation.

In this new metaphor, computation is an activity which is inherently social, rather than solitary, and that realization leads to a new ways of conceiving, designing, developing and managing computational systems. One example of the influence of this viewpoint is the emerging model of software as a service, for example in *Service Oriented Architectures*. In this model, computer applications are no longer "compiled together" in order to function on one machine (single user applications), nor distributed applications managed by a single organisation (such as today's Intranet applications), but instead are societies of components in which:

- The software components are viewed as *providing services to one another* rather than being compiled or run as a single programme.

- The software components and their services may be *owned and managed by different organisations*, and thus have access to different information sources, have different objectives, and have conflicting preferences.
- The software components are not necessarily activated by human users but *may also carry out actions in an automated and co-ordinated manner* when certain conditions hold true. These pre-conditions may themselves be distributed across components, thereby requiring prior co-ordination and agreement with or from other components.
- Intelligent, automated software components may even undertake *self-assembly of software and systems*, to enable adaptation or response to changing external or internal circumstances.

Such computer systems resemble those of the natural world and human societies much more than they do the arithmetical calculation programs taught in old-style computer programming classes. Accordingly, ideas and concepts from biology, sociology, economics, political science and statistical physics play an increasingly important role in contemporary computer science.

How should we exploit this new metaphor of computation as a social activity, as interaction between intelligent and independent entities, adapting and co-evolving with one another? The answer, many people believe, lies with agent technologies. An *agent* is a computer programme capable of flexible and autonomous action in a dynamic environment, usually an environment containing other agents. In this abstraction, we have encapsulated autonomous and intelligent software entities, called agents, and we have demarcated the society in which they operate, a multi-agent system. Agent-based computing concerns the theoretical and practical working-through of the details of this simple two-level abstraction, and the design, engineering and management of the systems which result.

One impact beyond Computer Science of this paradigm is the development of increasingly sophisted individual-based models of real-world phenomena in other disciplines. For example, the development of agent models in economics has enabled the emergence of a new branch of the discipline called agent-based computational economics, which has enabled simulation testing of new auction designs, of hypotheses about human economic behaviour, and of economic models of particular marketplaces. In particular, agent-based computational economics has supported the study of markets and economies as complex adaptive systems (e.g., Barnett *et al.* 2000), rather than as hydraulistic machines with deterministic relationships between flows. Similarly, agent-based or individual-based computational models are increasingly common in biology and ecology (Grimm *et al.* 2005).

This section presents a snapshot of the topics now explored by computer scientists and researchers in application domains using multi-agent approaches. The five chapters here arose from papers presented at the workshop on Multi-Agent Systems and Complexity:

- Jean-Pierre Georgé, Marie-Pierre Gleizes and Pierre Glize discuss co-operative self-organizing mechanisms;

- Carlos Gershenson discusses methodologies for designing and engineering self-organizing systems;
- Jeff Johnson and Pejman Iravani consider complexity in robot football, one of the great motivating problem domains in artificial intelligence; and
- Paul Marrow disusses the influence of ideas from biology in the design of software agents;
- Gilberto Tadeu Lima & Gustavo Gomes de Freitas explore the macrodynamics of financial stability and fragility through the use of agent-based computer simulations.

Readers interested in these topics are recommended to explore some of the books and papers cited below.

References:

W. A. Barnett, C. Chiarella, S. Keen, R. Marks and H. Schnabl (Editors) [2000]: *Commerce, Complexity, and Evolution.* Cambridge, UK: Cambridge University Press.

N. Boccara [2004]: *Modeling Complex Systems.* New York, NY, USA: Springer.

V. Grimm, E. Revilla, U. Berger, F. Jeltsch, W. M. Mooij, S. F. Railsback, H-H. Thulke, J. Weiner, T. Wiegand and D. L. DeAngelis [2005]: Pattern-oriented modeling of agent-based complex systems: lessons from ecology. *Science*, 310 (5750): 987—991.

M. Luck, P. McBurney, O. Shehory and S. Willmott [2005]: *Agent Technology: Computing as Interaction. A Roadmap for Agent Based Computing.* AgentLink III, the European Co-ordination Action for Agent-Based Computing, Southampton, UK.

F. Zambonelli and V. D. Parunak [2003]: Sign of a revolution in computer science and software engineering. In: P. Petta, R. Tolksdorf and F. Zambonelli (Editors): *Engineering Societies in the Agents' World.* Lecture Notes in Artificial Intelligence, Volume 2577. Berlin, Germany: Springer.