# Agent Systems and Applications*

Ruth Aylett[1], Frances Brazier[2], Nick Jennings[3],
Michael Luck[4], Hyacinth Nwana[5], and Chris Preist[6]

1. Centre for Virtual Environments, University of Salford, UK
2. Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands
3. Department of Electronic Engineering, Queen Mary & Westfield College, University of London, UK
4. Department of Computer Science, University of Warwick, UK
5. Advanced Research and Technologies Department, BT Laboratories, UK
6. Agent Technology Group, HP Laboratories, UK.

As the the number of deployed multi-agent applications increases, further and better experience with the technology is gained, enabling a strong evaluation of the field from a more practical perspective. In particular, questions relating to how the theory of multi-agent systems impacts on practice, and how the practical development itself compares with other technologies, can be answered in the light of a heightened level of maturity. Given the tensions between theoreticians and practitioners in computing in general, let alone their spats in AI or multi-agent systems in particular, the discussion on agent systems and applications was both vigorous and enthusiastic.

## 1 Characteristics of multi-agent applications

*What are the characteristics of applications which make them suitable for a multi-agent system solution? Are there applications which are unsuitable for multi-agent systems?*

The obvious response to the question of characteristics of suitable applications is that they are worth considering when the application requires both distribution and intelligence. Thus, a multi-agent approach would be sensible for problems that are inherently (physically or geographically) distributed where independent processes can be clearly distinguished. Such problems include, for example, distributed sensor networks, decision support systems, air traffic control, or other networked or distributed control systems. A distributed approach is not in itself enough, however, and there should also be requirements for intelligence or adaptivity in the sub-processes that involve explicit reasoning about behaviour, for example. If the problem can be solved by means of a look-up table at each node of a network, a multi-agent system would be excessive.

Physical distribution may not be the only reason for a distributed approach. Minsky's *Society of Mind* paradigm (Minsky, 1986) suggests the use of a multi-agent system where there is a wide range of reasonably self-contained pieces of functionality that require the use of AI, especially if they run asynchronously, or are distributed or independent in the sense of timing. In this way, a multi-agent approach might be applied to a single robot manipulator taking each joint as an agent, or to a single static system with sensors and effectors (an *immobot*) such as a smart building or a spacecraft.

*This report is the result of a panel discussion at the Second UK Workshop on Foundations of Multi-Agent Systems (FoMAS'97). All members of the panel are authors, listed alphabetically.

A range of further application areas that qualify for multi-agent solutions can also be enumerated. These include those requiring the interconnection and inter-operation of multiple autonomous, self-interested existing legacy systems, expert systems, and decision systems, or those requiring solutions that draw from distributed autonomous and selfish information sources, such as the Personal Travel Assistance demo from BT; those where the solutions draw from different distributed experts, such as health care provisioning, in which some central agent cannot possibly perform the task without help from other experts; problems that naturally cross organisational boundaries for which an understanding of the interactions among societies and organisations is needed; and problems where no single agent has a total view, but several agents have local views. The notion of *ownership* of information and strategies in the application is important here, and in particular when it is distributed over different organisational entities so that no single entity can (or does) have access to all the information. Final particular examples include traders in a marketplace (Chavez et al., 1997), and different entities in a business working on other's tasks (Jennings et al., 1996). In these situations, the problems to be tackled do not have one overall goal, but rather consist of balancing the (possibly conflicting) goals of different entities.

As with any other piece of technology, there are plenty of applications that do not require a multi-agent approach. Multi-agent systems are not required merely to produce modularity (though they reduce complexity), extra speed (though this may be an effect of their inherent parallelism), reliability (though they provide redundancy), flexibility or re-usability at Newell's knowledge level (Newell, 1982). In the same way, they are not required simply because a problem is too large for a centralised single agent due to resource limitations, nor because of the sheer risk of a centralised system, nor merely for reasons of efficiency, hetergenuous reasoning, etc.

For example, a payroll system might benefit in a software engineering sense from an object-oriented approach (providing modularisation and re-use), but such standard data-processing problems do not really need the communications overhead or functionality of a multi-agent approach. Such applications usually require neither distribution nor intelligence. In the same way, a small free-standing expert system used in a single location requires the intelligence but not the distribution, and the human interface is insufficiently complex to be worth thinking of it as an agent.

Finally, it is worth noting that a multi-agent system approach may be useful, though not necessary, when tackling problems that are easiest *visualised* in a way that appears to have the above characteristics, such as combat simulation (Rao and Selvestrel, 1992).

## 2 Applications development

*What does a multi-agent approach to applications development buy you over more standard approaches such as object-oriented, expert systems, or distributed computing approaches.*

In general, Object-oriented (OO) systems, expert systems, and distributed computing techniques do not offer solutions to the kind of problems for which multi-agent systems are used, for a range of reasons (Wooldridge, 1997).

OO techniques are good in general, but are rather low-level for intelligent applications. They can be used, for instance, to implement knowledge representations, but they do not themselves provide a knowledge representation. OO development methodologies can, however, be seen as a low-level underpinning for a multi-agent methodology.

The same might be said of distributed computing methodologies and indeed, many multi-agent systems (e.g., ADEPT) are built on top of distributed platforms such as CORBA. However, if it can be argued that if OO approaches are still relatively new, these are even newer and less generally accepted. Again, however, the level is wrong — for example, communications protocols do not operate at the high level of Speech Acts as one might wish for a multi-agent system. Yet distributed computing approaches could deal with some of the lower levels as well as providing some of the basic techniques (such as protocol definition and validation). An OSI model with an elaborated applications level might be ideal or, alternatively, current approaches to multi-agent applications development might form such a layer.

Expert systems are even more problematic in terms of development methodology. There is a reasonable consensus on the life-cycle, but while KADS has made some impression, many avoid it. Developments such as KRL and other movements for standardisation are very much in their infancy, while ontologies are helpful but not widely used. The most successful parts of knowledge engineering relate to knowledge acquisition (elicitation and structuring) and are not incompatible with a multi-agent approach in the case where such a system has to embody expert knowledge. This is not always the case, however.

More importantly, multi-agent applications require a *cooperation knowledge level* (Jennings and Campos, 1997), while expert systems (and others) typically operate at the symbol and knowledge levels (at most) (Newell, 1982). However, if using a *cooperation knowledge level* buys nothing for the application, it should be avoided. For example, Huberman's thermal economy multi-agent system has been criticised in that it provides an inferior solution to a single 'agent' control solution. Arguably, this is a clear *distributed problem solving* issue, but if it did not involve just *one* owner, then an inferior multi-agent solution might be justifiable.

The benefits of a specifically multi-agent methodology would be a reduction of the semantic gap between analysis on the one hand, and design and implementation on the other, leading to a reduction in the time to design and implement, with the usual trade-off between better expandibility and losses in execution efficiency and design specificity.

The Internet revolution is resulting in increased communication between distinct entities with different goals and distinct boundaries which must be secure. The new class of applications which service this need will inevitably use some form of multi-agent approach. Current methodologies emphasise top-down design, but multi-agent systems adopt a different approach — top-down within the agent, and bottom-up in the agent community.

If the utility of using a multi-agent solution (however it is measured, and it will depend on every new situation) is greater than the utility gained from a single 'agent' application, then such an approach should be used. In summary, multi-agent systems research can be regarded as developing a way of looking at problems, (a nascent design methodology) rather than a technology[1]. Hence multi-agent systems can, and do, use object-oriented, expert systems and distributed computing technologies to implement applications and toolkits that embody this approach. Multi-agent systems will not replace these technologies, but provide a different way of using them to tackle new kinds of problems. (See the panel report on methodological foundations from FoMAS'96 for a more detailed discussion of some of these issues (Fisher et al., 1997).

# 3 Killer applications

*Is there a killer application for multi-agent systems and, if so, what is it?*

[1] Note that there is some multi-agent systems technology, in particular in dealing with the social aspects of problem-solving (e.g., coopertaion coordination, etc).

The search for a *killer application* for multi-agent systems brings quite different responses. Nwana and Brazier suggest applications such as interactive collaborative design, air-traffic control, network management, command and control, and distributed data mining.

By contrast, Aylett and Preist take a different stance. While recognising the killer application as the 'holy grail' of the agent world, Preist believes that it does not yet exist because the problems where multi-agent systems are necessary are only just appearing. Current inter-organisation applications do not require multi-agent systems, for two reasons. First, they *support* human communication to carry out tasks, rather than *automating* tasks completely (e.g., FastParts component trading). Second, they are proprietary systems that link a small number of companies in a predetermined way (e.g. the WalMart Quick Response system (Kalakota and Whinston, 1996)). Nevertheless, Preist does suggest applications that are both automated and open such as on-line automated trading as possibilities.

Aylett questions the need for *killer* applications as such, but sees computer pets as catching the popular imagination, while in the corporate world, information retrieval agents ride on the back of Internet hype. In these cases, the *agenthood* is obvious, but for other multi-agent applications only the results are obvious, and only a few people will ever care about how they were achieved.

## 4 Barriers to the uptake of technology

> *Are there barriers to be crossed before a broader uptake of multi-agent systems technology will be seen? For example, do we need to have a stock of specialised tools or development methods (such as languages, etc) for multi-agent systems for them to be a widely-accepted class of system technology?*

Barriers to the uptake of multi-agent systems technology can be divided into two classes, which we can call social and technical barriers. Environmental barriers cover issues relating to public relations, training and experience, while technical barriers cover tools and techniques. These classes of barrier are inter-related but are considered in turn below.

In order for a technology to be used 'in anger', there must be a clear notion of its purpose and usefulness. Typically, this is obtained via small industrial trial projects that train those in the field in the concepts and methods and allow some basic experience to be laid down. There is no substitute for this process. Only if some clear gains are realised during the course of it will any further progress be made. Slick and expensive commercial packages are not required at this stage, but it must be possible to use cheap or public domain software reasonably easily, even if this has limited functionality. Government initiatives and industrial clubs are a good way of getting this process off the ground. Large high-tech organisations (such as BT, Siemens, Daimler-Benz and Logica) are already successfully carrying out this process but it is not yet active in the wider industrial context

The existence of trained people helps significantly. Industry now contains many people who studied expert systems at university and a growing number who also studied neural nets and fuzzy logic. But how many are there who studied agents? This barrier should not be underestimated.

One outcome of this process — if it is successful — is a demand for better tools and methods. At the same time, particular tools can have the effect of increasing experience and thus demand as the experience with expert system shells has shown. Limited as they were, expert system shells popularised the technology widely, and laid the basis for a second generation of more sophisticated tools.

On the technical side, there are a number of tools and techniques that might push back the barriers, and in particular concerning the interaction between agents. Because inter-organisational applications are one of the major drivers for the uptake of multi-agent systems, some form of inter-agent communication standard is necessary, either pre-designed, or emerging *de facto* from a community. The standards are likely to be problem-specific initially with, for example, the possibility of a standard language of trade negotiation emerging. If a pre-designed standard is to be adopted, however, it is important that it is easy to use and is designed with the upcoming needs of the business community in mind. Some efforts in this direction are already underway with, for example, FIPA, Agent Society, MAF, W3 consortium, OPS, etc. Note the related impact CORBA is having with Distributed Objects.

Other technical areas of concern include issues of security so that agents can communicate securely and be trusted with important business tasks; the need for clear, non-*ad hoc* development methodologies addressing *real* agent issues to support the design and development of multi-agent systems; and tool-kits to facilitate their construction with good debugging environments, hence ZEUS (Nwana et al., 1998) and dMARS.

While all these aspects might contribute to a greater uptake of agent technology, one thing that must not be standardised or preordained is the internal design of agents. As agent communities will cross organisational boundaries, it is important that different organisations are free to adopt whatever internal approach they choose.

Nwana also sounds a note of warning, suggesting that a new field is only defined by its *problems*, not its *methods*. The field of multi-agent systems is falling into the trap that has befallen artificial intelligence, of deluding itself that the methods (e.g., cooperation, achieving rationality, competition, believable agents, conceptual and theoretical foundations, multi-agent planning, negotiation) are the really important issues. It is the problems that are foremost important.

## 5 Parallels with other technologies

*Are there parallels with other applications and development technologies that can illuminate the future of multi-agent systems, such as OO technology, or are multi-agent systems something fundamentally different?*

All the panellists agreed that multi-agent systems are not fundamentally different. Indeed, many solutions to multi-agent systems problems already exist in several parts, and researchers need to do *creative* synthesising of some of these already invented wheels. Where necessary, 'new wheels' may be invented to link up the old ones. This position is vindicated by the ZEUS project which synthesises in one system many old wheels (Nwana et al., 1998). Other successful projects such as dMARS (d'Inverno et al., 1998) and ADEPT (Jennings et al., 1996) also vindicate this position. The new wheels, such as coordination and visualisation, are simply there to link up old concepts. As a result, the whole then becomes greater than the sum of its different parts, and hence novel, but this does not mean it is fundamentally different.

Other technologies such as Distributed Objects, the Internet, OO techniques, etc., have interesting histories from which to learn. For example, without HTML, HTTP, CGI standards, the Internet would not be the phenomenon it is today. Similarly, CORBA has made Distributed Objects a worthy investment, and a potential route to solving the legacy problem. These technologies are successful because at their core, they are quite simple, and all the complexity resides at their edges. At a more detailed level, however, while OO technology may provide a means to implement agent systems, it is not aimed at providing a conceptual representation of knowledge, reasoning, intelligent behaviour and interaction.

## 6 Objective measures of benefits and drawbacks

> *How can the benefits and drawbacks of a multi-agent system solution be assessed in an objective manner?*

In trying to find ways of assessing the contribution of multi-agent systems, one view argues that this is only really with experience. Strictly speaking, we need to know how good or bad the multi-agent approach is in comparison with the best alternative solutions. Nearly always, the answer is unknown because resources do not allow multiple approaches to be tried. Moreover, in the real world there are usually any number of different ways of solving a particular problem and it is impossible to compare them in any objective fashion. In practice, this is where the 'suck it and see' applications mentioned above come in. From experience of producing small systems in a variety of different organisations to solve a variety of different problems, it becomes possible to generalise about the benefits and drawbacks and to see what the evidence is for what was claimed to be good and bad.

Again, the comparison with expert systems or neural nets is illuminating. In the former case, the benefits were expected to be in the area of efficiency (because of fewer experts needed) but turned out to be in the area of effectiveness (with better use of existing experts and more widespread distribution of expertise). In the case of neural nets, some comparisons have been made with equivalent mathematical and statistical techniques, and the benefits are less in performance and more in the time and skill required for development and extension. Without the experience in both of these cases, we would not have been at all sure what to start measuring.

The alternative view is that though rigorous evaluation is the academic answer to this, the market will eventually be the most objective evaluator. Companies are buying into Distributed Object technologies because there are clear needs and benefits for them, and the same is true of multi-agent systems, which will not be taken up unless they demonstrate their need and potential.

In the short term, we need to define criteria for judging multi-agent solutions. Do they offer a *value-added* solution to other conventional applications such as control, expert systems, distributed problem solving approaches, and blackboard approaches? This simply requires the definition of *value-added* which could range from excellent (where no solution was probable without a multi-agent approach), through minimal (where a multi-agent solution is marginally of value) to poor (where conventional approaches offer better solutions).

Again, Nwana points out the dangers, suggesting a critical appraisal of some current efforts whose benefits are questionable. He cites Stuart Russell who recently argued against what he calls *premature mathematization*:

> "There is always a danger, in this sort of claim, that its acceptance can lead to 'premature mathematization', a condition characterized by increasingly technical results that have increasingly little to do with the original problem" (Russell, 1997).

According to Nwana, multi-agent systems research is in danger of falling into the same trap that Russell asserts of AI.

Despite this warning, the prospects look good. Indeed many researchers in the field have acknowledged the difficulties that arise when formal theoretical work is not used to inform practice, and *vice versa*, and there is an emerging trend of research that aims to bring these two strands together e.g. (Jennings, 1995; Luck et al., 1997; Rao, 1996) . In a sense, difficult experiences with AI in general over the last thirty years or so have provided the impetus for a strong effort to avoid similar problems in multi-agent systems. The development of real systems and applications forms a major part of the field and, as has been suggested, the experiences gained can and should be fed back into the basic research. Relative youth here can be seen to be an advantage in this way. Ultimately, the enthusiasm for agent systems and applications, coupled with cautious and perhaps suspicious voices such as Nwana's, suggests a productive future.

## References


A. Chavez, D. Dreilinger, R. Guttman and P. Maes, 1997. "A Real-Life Experiment in Creating an Agent Marketplace" In *Proceedings of the Second Conference on Practical Applications of Agents and Multi-Agent Systems*, 159-178.

M. d'Inverno, D. Kinny, M. Luck and M. Wooldridge, 1998. "A Formal Specification of dMARS" In *Intelligent Agents IV: Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages*, Singh, Rao and Wooldridge (eds.), Lecture Notes in Artificial Intelligence, 1365, 155–176, Springer-Verlag.

M. Fisher, J. Müller, M. Schroeder, G. Staniford, and G. Wagner, 1997. "Methodological Foundations for Agent-Based Systems" *The Knowledge Engineering Review*, 12(3): 323–329.

N. R. Jennings, 1995. "Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems using Joint Intentions" *Artificial Intelligence* 75(2):195-240.

N. R. Jennings and J. R. Campos, 1997. "Towards a Social Level Characterisation of Socially Responsible Agents" In *IEE Proceedings on Software Engineering*, 144(1): 11-25.

N. R. Jennings, P. Faratin, M. J. Johnson, T. J. Norman, P. O'Brien, M. E. Wiegand, 1996. "Agent-based business process management", *International Journal of Cooperative Information Systems*, 5(2&3):105-130.

R. Kalakota and A. B. Whinston, 1996. *Frontiers of Electronic Commerce* 36-37, Addison Wesley.

M. Luck, N. Griffiths, and M. d'Inverno. "From agent theory to agent construction: A case study" In *Intelligent Agents III: Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages, Lecture Notes in Artificial Intelligence, 1193*, pages 49–63. Springer-Verlag, 1997.

M. Minsky, 1986. *The Society of Mind* Simon and Schuster, New York NY.

A. Newell, 1982. "The Knowledge Level" *Artificial Intelligence* 18(1):87–127.

H. S. Nwana, D. T. Ndumu and L. C. Lee, 1998a. "ZEUS: An advanced tool-kit for engineering distributed multi-agent systems". In *Proc. 3rd Int. Conf. Practical Appl. Intelligent Agents and Multi-Agent Technology*, 377-391.

A. S. Rao. "Agentspeak(l): BDI agents speak out in a logical computable language" In W. Van de Velde and J. W. Perram, editors, *Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World, LNAI 1038*, 42–55. Springer-Verlag: Heidelberg, Germany, 1996.

A. Rao and M. Selvestrel, 1992. "Agent-Oriented Architecture for Air Combat" In *Proceedings of the Australian Joint Conference on Artificial Intelligence*.

S. J. Russell, 1997. "Rationality and Intelligence" *Artificial Intelligence* 94(1):57–77.

M. Wooldridge, 1997. "Agent-based Software Engineering". In *IEE Proceedings on Software Engineering*, 144(1):26–37.