Autonomous Agents and Multi-Agent Sytems, 9, 203–252, 2004

A Manifesto for Agent Technology: Towards Next Generation Computing[1]

MICHAEL LUCK mml@ecs.soton.ac.uk
School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

PETER MCBURNEY p.j.mcburney@csc.liv.ac.uk
Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK

CHRIS PREIST chris.preist@hp.com
HP Labs, Filton Road, Stoke Gifford, Bristol BS34 8QZ, UK

Abstract. The European Commission's eEurope initiative aims to bring every citizen, home, school, business and administration online to create a digitally literate Europe. The value lies not in the objective itself, but in its ability to facilitate the advance of Europe into new ways of living and working. Just as in the first literacy revolution, our lives will change in ways never imagined. The vision of eEurope is underpinned by a technological infrastructure that is now taken for granted. Yet it provides us with the ability to pioneer radical new ways of doing business, of undertaking science, and, of managing our everyday activities. Key to this step change is the development of appropriate mechanisms to automate and improve existing tasks, to anticipate desired actions on our behalf (as human users) and to undertake them, while at the same time enabling us to stay involved and retain as much control as required. For many, these mechanisms are now being realised by agent technologies, which are already providing dramatic and sustained benefits in several business and industry domains, including B2B exchanges, supply chain management, car manufacturing, and so on. While there are many real successes of agent technologies to report, there is still much to be done in research and development for the full benefits to be achieved. This is especially true in the context of environments of pervasive computing devices that are envisaged in coming years. This paper describes the current state-of-the-art of agent technologies and identifies trends and challenges that will need to be addressed over the next 10 years to progress the field and realise the benefits. It offers a roadmap that is the result of discussions among participants from over 150 organisations including universities, research institutions, large multinational corporations and smaller IT start-up companies. The roadmap identifies successes and challenges, and points to future possibilities and demands; agent technologies are fundamental to the realisation of next generation computing.

Keywords: agent technology, roadmap, trends, survey, challenges.

1. Introduction

As the computing landscape moves from a focus on the individual standalone computer system to a situation in which the real power of computers is realised through distributed, open and dynamic systems, we are faced with new technological challenges and new opportunities. The characteristics of dynamic and open environments in which, for example, heterogeneous systems must interact, span organisational boundaries, and operate effectively within rapidly changing circumstances and with dramatically increasing quantities of available information, suggest that improvements on the traditional computing models and paradigms are required [131]. In particular, the need for some degree of *autonomy* [80], to enable components to respond dynamically to changing circumstances while trying to achieve over-arching objectives, is seen by many as fundamental. While this notion is not intended to suggest an absence of control, some application contexts offer no alternative to *autonomous* software. In practical developments, *Web Services*, for example, now offer fundamentally new ways of doing business through a set of standardised tools, and support a service-oriented view of distinct and independent software components interacting to provide valuable functionality. In the context of such developments, agent technologies have become some of the primary weapons in the arsenal aimed at addressing the emergent problems, and managing the complexity.

Because of the horizontal nature of agent technology, it is likely that the successful adoption of agent technology in these areas will have a profound, long-term impact both on the competitiveness and viability of IT industries, and also on the way in which future computer systems will be conceptualised and implemented.

1.1. Agents as a design metaphor

Agents provide designers and developers with a way of structuring an application around autonomous [79], communicative elements, and lead to the construction of software tools and infrastructure to support the design metaphor [65]. In this sense, they offer a new and often more appropriate route to the development of complex systems, especially in open and dynamic environments. In order to support this view of systems development, particular tools and techniques need to be introduced. For example, methodologies to guide analysis and design are required; agent architectures are needed for the design of individual components, and supporting infrastructure (including more general, current technologies, such as Web Services) must be integrated.

1.2. Agents as a source of technologies

Agent technologies span a range of specific techniques and algorithms for dealing with interactions with others in dynamic and open environments. These include issues such as balancing reaction and deliberation in individual agent architectures, learning from and about other agents in the environment and user preferences, finding ways to negotiate and cooperate with agents and developing appropriate means of forming and managing coalitions. Moreover, the adoption of agent-based approaches is increasingly influential in other domains. For example, multi-agent systems can provide faster and more effective methods of resource allocation in complex environments, such as the management of utility networks, than any human-centred approach. Similarly, the use of agent systems to *simulate* real-world domains may provide answers to complex physical or social problems which would be otherwise unobtainable, as in the modelling of the impacts of climate change on various biological populations, or modelling the impact of public policy options on social or economic behaviour [15].

1.3. Overview

This paper has arisen out of the work of the AgentLink project, a key focus of which was the development of a technological roadmap to provide a focussed, up-to-date assessment of how the agent field can and should develop. In so doing, it addresses:

- briefly, the background to agent technology – its origins and focus;
- the state-of-the-art in agent technology – the current situation, in terms of technology and applications, including commercial success stories and failures;
- a long-term vision for the field – where we will be if agent technology succeeds, the commercial opportunities and the impact that can be expected from this success;
- the technology gaps between the state-of-the-art and the long-term vision; the problems that we need to solve in order to realise the long-term vision, the different techniques that are being applied in order to bridge these gaps; and
- a discussion of the implications of the study in specific terms for different sets of stakeholders – the regulatory implications, the implications for research (such as content gaps, funding gaps, etc), and the implications for industry (such as skills gaps, collaboration opportunities, etc).

This paper thus describes the current state-of-the-art of agent technologies and identifies trends and challenges that will need to be addressed over the next 10 years or so to progress the field and realise the benefits. In Section 2, we review the current use of agents both as a design metaphor and as a source of technologies, before considering the relationships with other disciplines and agent applications. Section 3 then offers specific indications of the possible ways in which the field might develop over the next 10 years, broken into four distinct phases. From this particular timeline, Section 4 identifies several *technological* challenges that arise, while Section 5 considers the complementary challenges for the research and development *communities*. Finally, in Section 6, we review application opportunities that either demand or suggest the use of agent technologies for full realisation.

2. Current state-of-the-art: research and development

2.1. Agents as design

The use of agents as an abstraction tool, or a metaphor, for the design and construction of systems provided the initial impetus for developments in the field. On the one hand, agents offer an appropriate way to consider complex systems with multiple distinct and independent components. On the other, they also enable the aggregation of different functionalities that have previously been distinct (such as planning, learning, coordination, etc.) in a conceptually embodied and situated whole. Thus these notions provide a set of technology areas that relate directly to these abstractions in the design and development of large systems, of individual agents, of ways in which agents may interact to support these concepts, and in the consideration of societal or macro-level issues such as organisations and their computational counterparts.

2.1.1. Agent-oriented software engineering. Work on methodologies and software engineering for agent systems exploits synergy from the interaction with existing communities such as the software and knowledge engineering communities, and has a strong emphasis on practical use in industry. The main goal is to determine how agent qualities affect software engineering, and what additional tools and concepts are needed to apply software engineering processes and structures to agent systems [129].

Specific areas of interest here have included:

– requirements engineering for agent systems;
– techniques for specification of (conceptual) designs of agent systems;
– verification techniques;
– agent-oriented analysis and design;
– specific ontologies for agent requirements, agent models and organisation models;
– libraries of generic models of agents and agent components;
– agent design patterns;
– validation and testing techniques; and
– tools to support the agent system development process (such as agent platforms).

To date, this work has largely concentrated on analysis and design methods, development tools and languages for programming and communication [78, 124]. Although substantial progress has been made in recent years after an initial absence of such supporting work, most developments are still largely at the prototype stage. There are experiments and some case-studies but these have been *ad hoc* rather than methodical and systematic in testing.

A related but distinct aspect under the broad heading of agent-oriented software engineering is concerned with the use of agents in the development of complex distributed systems, as opposed to the application of traditional software engineering to agents. This work adopts the *metaphor* of agents, or the *design view* by which agents provide a natural and elegant means to manage complexity. The complexity of many systems arises from the interactions between the components of the system, and an agent paradigm provides a natural way to model such interactions. The agent abstraction may be applied not just to represent technological components of implemented systems, but also to the modelling and design of complex systems that may be implemented in the most appropriate fashion. Of course, given the initial complexity that motivates an agent approach, it seems likely that the entire development lifecycle in such cases will benefit from agents.

2.1.2. Agent architectures. Agent architectures are the fundamental engines underlying the autonomous components that support effective behaviour in real-world, dynamic and open environments [39].

Initial efforts in the field of agent-based computing focused on the development of intelligent agent architectures, and the early years established several lasting styles of architecture. These range from purely *reactive* (or *behavioural*) agents that operate in a simple stimulus-response fashion, such as those based on the Subsumption Architecture of Brooks [17] at one extreme, to more *deliberative* agents that reason about their actions, such as the class of belief-desire-intention (BDI) agents (for example, [40] that are increasingly prevalent (including in commercial products such as JACK [63] from Agent Oriented Software), at the other extreme. In between the two lie *hybrid* combinations of both, or *layered* architectures, which attempt to involve both reaction and deliberation in an effort to adopt the best of each approach. Increasingly more sophisticated agents than the traditional BDI kind are have also been developed, but the benefits of the increased sophistication is largely confined to well-defined areas of need rather than offering general solutions.

2.1.3. Mobile agent systems. Many researchers and programmers see agents as programs roaming a network to collect business-related data in order to help users to buy goods, or implement platform-independent code-on-demand, for example [22]. This need for mobile agents is acknowledged, and builds on European strengths, but mobility brings added security problems. The research effort concentrates on how to guarantee termination, security or exactly-once protocols [120, 123]. To protect against malicious hosts, agents should contain time-limit validity, and electronic money with an expiration date. A key issue that needs to be addressed here is *administrability* of mobile agent systems through, for example, authorisation policies; this has been a major reason why mobile agents have not yet been taken up by the mainstream. Note also that hosts need to be protected as well as agents.

End users already encounter the situation that, while ample bandwidth is available on the backbones of network service providers, their experience is limited by the constraints of the infamous *last mile*. Mobile agents may improve the end user experience by offloading application-specific filtering, media adaptation, and other pre-processing to a node with high bandwidth connectivity. This is particularly interesting for mobile phones and portable devices.

One of the commercial application areas in which the added value of mobile agents is very high is large-scale distributed or decentralised system integration with highly adaptive and dynamic business logic. Existing solutions are generally centralised, pulling everything onto one platform, limiting the complexity and changes that can be handled. A decentralised agent approach divides and conquers complexity by pushing a large part of the business logic out onto source systems so that much monitoring and aggregation can be done on each. This distributes workload and increases robustness because the local processing can be performed independently of other systems, resulting in fewer and more relevant interactions with these systems, at a higher level of abstraction. In turn, mobility, mainly single-hop, is the answer to the increasing need for flexibility and adaptability in business logic. Agents can easily be deployed to source systems, carrying new database drivers, code to interact with new application or file types, or new data processing rules. Software is updated at the component-level, at runtime, proving a level of dynamism and flexibility that goes far beyond current release policies. Agent communication and behaviour capabilities complete the picture, being very well suited to high-level service-based interactions, the decentralised implementation of business logic, and for adapting and handling change in their environment. A nice property of the dynamic, component-level approach is that it naturally fits step-by-step system integration, with each step resulting in added value for the business. This is a particularly significant advantage in the current economic climate, in which many companies have seen mega-projects fail.

For example, Global IDs Inc in the US offers a next-generation product suite for data integration based on the Tryllian mobile agent platform. Their data integration products are capable of simultaneously monitoring many hundreds of enterprise systems for relevant changes in data or meta-data, by deploying mobile agents onto those systems. The agents tap into local databases or applications, keep track of changes, can pre-process data and only forward relevant events or structured derived data to centralised collectors – in real time if required. The mobility of the agents allows highly customised functionality, which can be dynamically updated. Thus, the business user can change the business rules that are being executed at any point in time, while only relevant drivers and adapters are transferred to a source system. Agents can asses the impact of changes in the business rules and handle that impact throughout the integration process.

Other potential applications of mobile agent systems are in the management of complex distributed networks with differentiated components, such as mobile telecommunications networks. Mobile networks typically have several types of network components: base stations, which interface directly with the mobile devices of end-users; base-station controllers, which aggregate traffic from a number of base stations; and mobile switching centres, which switch calls between the network and other networks. The optimal design of a mobile network – that is, the best location of each type of network component – depends heavily on the pattern of network traffic. One challenge of many mobile networks is that this pattern changes by time of day or by day of the week; traffic during business hours may be concentrated in city centres, for instance, while at other times it is more dispersed. Thus an optimal network design may be one which changes with the pattern of traffic. Moving network components is not usually feasible, but we may be able to move their functionality: for example, a number of agents, each representing a network component, could move around the physical network as the traffic pattern changed. Other arguments in favour of mobile code include the avoidance of network latency, for example to increase fairness in applications with bounded response times (such as auctions).

Since the inception of the notion of mobile agents, research and development has concentrated primarily on research in their *basic* operation, including the separation of those components of a mobile agent that must be transported with it, different migration models, the abstractions required for location-awareness and basic security. Notable cornerstones of mobile agent work have been Telescript from General Magic [126] (now superseded), Aglets from IBM [73], Mole from the University of Stuttgart [12], enago from IKV++, TACOMA from the Universities of Tromso and Cornell [68], as well as D'Agents from Dartmouth College [57], though there are many others.

2.1.4. Agent infrastructure. Agent infrastructure is concerned with developmental and operational support for agent systems. Middleware technologies (e.g., [7, 90]) aim to address issues such as ad-hoc networking while taking into account the heterogeneity of the environment. Mobile agent systems research has made important contributions in terms of efficient code mobility mechanisms and resource discovery mechanisms.

Middleware can be described as the software layer that resides between the underlying host and network operating system on the one hand, and the application layer on the other. Its purpose is to provide a common set of programming interfaces that developers can use to create distributed systems [8].

In the last few years, several new technologies have emerged that are aimed specifically at the ad-hoc networking that is central to the support of significant agent-based systems. These include Jini, UPnP and Salutation, for example, which define discovery and registration protocols that allow for dynamic discovery. Similarly, markup languages such as XML and RDF(S), along with standardised ontologies, provide a means for resource description and manipulation of this data at a semantic level.

Agent infrastructure also provides management functionality through such mechanisms as Jini leasing, which controls access to registry services, communication support from underlying transport mechanisms to robust protocols for information exchange, and security support to ensure that agents are properly authenticated and suitably authorised to perform their required actions.

Much of this amounts to a matter of leveraging existing work for application to agent-based computing, one of the most salient current examples being Web Services (e.g., [5, 93, 98]. There are also now a large number of agent development environments and toolkits (including 10–15 implementations of the latest FIPA standards and several high profile commercial toolkits such as Agent Oriented Software's JACK, and Lost Wax's agent framework). While not all of the available tools are sufficiently mature for mission critical usage (especially the non-commercial offerings), such systems are providing researchers and developers with vital tools for rapid prototyping and testing of agent systems. In a similar vein, initiatives at the community level offer important potentially standardised infrastructure, or standards for infrastructure, that can provide a critical enabler for development of scalable interoperable systems. These include, but are not limited to, the following.

– Base technologies:
  - The Extensible Markup Language (XML) is the universal format for structured documents and data on the Web. It was designed for ease of implementation and for interoperability with both SGML and HTML.
  - The Resource Description Format (RDF) is a framework for describing and interchanging metadata.
– eBusiness:
  - ebXML aims to standardise XML business specifications. By providing an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure and consistent manner.

  - RosettaNet is a consortium of major technology companies working to create and implement industry-wide eBusiness process standards. RosettaNet standards offer a robust nonproprietary solution, encompassing data dictionaries, an implementation framework, and XML-based business message schemas and process specifications for eBusiness standardisation.
- Universal plug and play:
  - Jini network technology provides simple mechanisms that enable devices to plug together to form an emergent community in which each device provides services that other devices in the community may use.
  - uPnP offers pervasive peer-to-peer network connectivity of intelligent appliances and wireless devices through a distributed, open networking architecture to enable seamless proximity networking in addition to control and data transfer among networked devices.
- Web services
  - UDDI is an industry initiative aimed at creating a platform-independent, open framework for describing services, discovering businesses, and integrating business services using the Internet. It is a cross-industry effort driven by platform and software providers, marketplace operators and eBusiness leaders.
  - SOAP provides a simple and lightweight mechanism for exchanging structured and typed information between peers in a decentralised, distributed environment using XML.
  - WSDL/WSCL: WSDL provides an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages, thus enabling the automation of the details involved in applications communication. WSCL allows the abstract interfaces of Web services, i.e., the business level conversations or public processes supported by a Web service, to be defined.

2.1.5. Electronic institutions. As the complexity of the real-world increases, there is a need to incorporate organisational concepts into computing systems, with the purpose of considering organisation-centered design. Electronic institutions provide a computational analogue of human organisations in which agents interact through roles that are defined as specified patterns of behaviour [47, 122]. Similarly, virtual organisations can potentially take advantage of the new electronic environments through coalition formation among disparate partners to form aggregate entities capable of offering new, different or better services than might otherwise be available [89]. Agent technology can help enterprises reduce their operational costs and speed-up time to market by helping distributed business processes run smoother and in a better coordinated fashion. This has particular application to supply chain and workflow management issues.

To design such systems requires a theory of organisation design, and knowledge of how organisations may change and evolve over time. Sociological organisation theory and social psychology are clearly important inputs to the design. Moreover, for the design of open multi-agent systems, political theory may be necessary [85]. Open systems permit the involvement of agents from diverse design teams, with diverse objectives, which may all be unknown at the time of design of the system itself. How the system as a whole makes decisions or agrees on joint goals will require the adoption of specific political philosophies, such as whether issues are subject to simple majority voting or transferable preference voting, etc. These aspects of multi-agent system design are still in their infancy, and much interaction between agent technologies on the one hand, and sociology, organisation design, political science and social choice theory on the other, will be required to achieve mature technologies.

2.2. Agent technologies

2.2.1. Overview. Agent-based computing has been a source of technologies for a number of research areas, both theoretical and applied. These include distributed planning and decision-making, automated auction mechanisms and learning mechanisms. Moreover, agent technologies have drawn from, and contributed to, a diverse range of academic disciplines, in the humanities, the sciences and the social sciences.

2.2.2. Multi-agent planning. Problem decomposition for distributed execution is one of the earliest areas to achieve success, drawing on developments in traditional planning. Issues studied here include ensuring that the distinct plans of different agents in a system do not conflict, attempting to optimise the overall plan schedule, and the decomposition and distribution of local planning tasks from a central goal [34]. In *partial global planning* (PGP), for example, agents form abstractions of local plans to use to inform other agents of plan steps of interest [43]. In this way, a partial global plan can be constructed that minimises redundancy, improves coordination, etc. Related work has addressed different strategies for maximising group performance with planning, execution, monitoring, communication and coordination.

More recent efforts have provided models of team or group activity in which agents collaborate towards specific objectives. Much of this work has been based on explicitly formulated theoretical models of joint intentions and commitment strategies, which provide underpinning theories in this area. Similarly, work on *coalition formation* in this respect [111, 118] underpins more general considerations of virtual organisations.

2.2.3. Agent communication languages. The power of agent systems depends on inter-agent communication. Powerful agents need to be able to communicate with users, with customers, with system resources, and with each other if they are to cooperate, collaborate, negotiate and so on. Common agent languages hold the promise of diverse agents communicating to provide more complex functions across the networked world. Indeed, as agents grow more powerful, their need for communication increases [19, 113].

The two agent communication languages with the broadest uptake are KQML and FIPA ACL. KQML was developed in the early 1990s as part of the US government's ARPA Knowledge Sharing Effort, and is a language and protocol for exchanging information and knowledge, which has been used extensively [49]. The Foundation for Intelligent Physical Agents (FIPA) is a non-profit organisation aimed at producing standards for the interoperation of heterogeneous software agents. The unproductive *standards war* scenario that might have arisen at one point seems now to have been avoided, with the most active participants supporting the FIPA effort, which incorporates many aspects of KQML [72]. The FIPA standard for communications is the FIPA Agent Communications Language (FIPA ACL) [50], which has a defined syntax and semantics based on speech act theory from the philosophy of language [9, 110].

Europe has been a prime mover in the FIPA standardisation effort, which seeks to address interoperability concerns through a sustained programme. This is one area in which the visibility of agent technology is strong, with some of the most active take-up efforts from early adopters as, for example, is illustrated by the Agentcities initiative.

Despite their merit, KQML and FIPA ACL only deal with agent-to-agent communication. If we understand an agent as something that can act on behalf of a human or an organisation, human-computer interface issues will be crucial for the acceptance of agent technology. Questions remain of how a task can be delegated from a user to an agent, how user preference structures can be transferred to agents, and how the state of task execution can be adequately monitored and controlled by the user.

2.2.4. Coordination mechanisms. For many years, researchers have been working on problems associated with inter-agent processes [21, 31], but the relationship between the different elements is still under debate. Practical work on developing multi-agent systems, however, has brought a lot of progress, ranging from the simple but effective Contract Net Protocol [30, 114] in the 1970s to more recent work with, for example, market mechanisms in coordination [109], and the investigation of properties such as fairness and truthfulness, and their utility and applicability to optimising coordination among agents [45, 46]. After more than a decade of work, research on coordination languages and models is focusing on the development of case studies, allowing the impact of different classes of coordination models and languages to be fully appreciated and compared. One very successful example is the use of agents on the DaimlerChrysler production line, in which dynamic agent coordination achieved savings of 10%.

Recently, formal dialogue games, which have been studied in philosophy since the time of Aristotle [6], have found application as the basis for interaction protocols between autonomous agents [84]. Dialogue games are formal interactions between two or more participants, in which participants "move" by uttering statements according to pre-defined rules. Dialogue-game protocols have been proposed for agent team formation, persuasion, negotiation over scarce resources, consumer purchase interactions and joint deliberation over a course of action in some situation [3, 4, 35, 36, 82].

2.2.5. Matchmaking architectures and algorithms. Rather than require individual agents in a multi-agent system to identify their own partners for cooperation, other specially designed agents may provide assistance [71]. Matchmakers are agents that maintain a continually updated repository of information about agents currently in the system, their capabilities, and other relevant information. Agents contact the matchmaker, describing a task in the hope of finding a capable agent to assist. Brokers take this to another level of sophistication in accepting tasks from requesting agents, assigning them to others, and possibly also prioritising and minimising cost, depending on the particular broker. Particularly important to note is that unlike more traditional yellow pages services, these agents can perform partial matches, providing much greater flexibility than might otherwise be available [32]. They provide an effective means for mediating the interactions between agents in an open system. The RETSINA architecture [116] offers just one example of a concrete platform offering such matchmaking and brokering services.

2.2.6. Information agents and basic ontologies. In the knowledge management arena, an increasing number of companies are realising that their own intranets are valuable repositories of corporate information, but without an understanding of how to apply it effectively this information is likely to be useless. Knowledge management is concerned with the acquisition, maintenance and evaluation of the knowledge of an organisation, but demands tools that foster productive collaboration while capturing, representing and interpreting the organisation's knowledge resources. This kind of knowledge can enhance adoption of best practice, highlight new business opportunities, and speed up the identification of market dynamics and sales opportunities. At present, companies employ largely manual processes, though initial applications are being developed.

Information agents typically have access to multiple, heterogeneous and geographically distributed information sources, in the Internet and corporate intranets, and search for relevant information, on behalf of their users or other agents. This includes retrieving, analysing, manipulating, and integrating information available from geographically distributed, distinct, autonomous information sources [59]. An intelligent information agent should pro-actively acquire, mediate, and maintain the relevant information on behalf of users or other agents [70]. Acquisition and managing information may also imply the *purchase* of information where appropriate, filtering, monitoring and updating, as well as data mining for some high-function tasks. The agent should be able to present both unified views, and different perspectives of the information, to the user. These processes will involve fusing heterogeneous data.

This can also be seen as a move from Enterprise Application *Integration* (EAI) to Enterprise Application *Collaboration*, which is not so much concerned with information management as with process management. Agent technology is a technology that helps to improve processes. Knowledge management using agents is just a means to this end.

2.2.7. Sophisticated auction mechanism design. In the very near future, a boom in agent-mediated auctions, a long-established and well-understood trading mechanism, is anticipated. Currently agents can recommend, but do not yet authorise an agreement. Fully automated negotiations [66] will come first in areas where the problem can be specified simply, each trade is of relatively small value, the process is repeated often, and interactions are repeated very fast. Businesses buying bandwidth for virtual private networks, and electric power capacity, provide good examples [27, 88].

There are several key drivers of agent-mediated auctions. One is the rapid recent growth in the use of auctions on the Internet. Although much attention is given to consumer auction sites such as e-Bay, the most spectacular growth has been in B2B applications, primarily for corporate procurement. General Electric Corporation (GE) of the USA, for example, purchased over US$ 6 billion worth of goods and services via on-line auctions in 2000. With such large proportions of corporate transactions being conducted via on-line auctions, it is a natural progression to attempt to automate these via agent-mediated auction. In one recent example in which Volvo auctioned contracts over the Internet for wooden packaging material with the assistance of Accenture and Trade Extensions, savings of 7.1 million Swedish Kronors were made, around 4% on previous years, while the number of suppliers was reduced from 15 to 6, offering extra benefits.

Another driver is the use of *program trading* in stock markets. Program trading involves the buying and selling of stocks via automated computer programs, which execute instructions of their human principals according to pre-defined rules of procedure. Although the prices of technology stocks are currently at a low, the markets on which they are traded, such as NASDAQ in the USA, have record levels of transactions. These transactions are primarily conducted by software programs, with fewer and less-frequent involvement by human traders. About one-third of NASDAQ trades are now executed by electronic trading programs. Despite their value, these programs have little flexibility and responsiveness, and a new set of capabilities, that may be provided by use of agent technologies, is required.

2.2.8. Negotiation strategies. Negotiation using agents will become a key part of next generation e-commerce and supply chain systems. Indeed, they are already being used for simple negotiation; Lost Wax and Cap Gemini have developed a supply chain management demonstrator which includes negotiation strategies. Drawing on theoretical economics and game theory, much academic work has been carried out on negotiation strategies, though as yet only very simple strategies have been deployed. Strategies exist for negotiating price in many-to-many environments (such as double auctions) and for participating in multiple one-to-many auctions simultaneously. Strategies also exist for carrying out multi-parameter negotiation in a one-to-one situation. Preliminary simulation-based studies have been carried out to determine the effectiveness of such strategies, but the extensive testing that would be required by industry has yet to be undertaken. In contrast to the theoretical study of strategies in auctions and games undertaken in economics, for example [92], there has been relatively little work to date on effective strategies in more complex mechanisms, such as those using argumentation [102, 103]. These mechanisms typically allow participants to state and question the reasons for proposals, not simply accept or reject the proposals themselves. Considerable theoretical and simulation work is required to identify effective strategies in different argumentation contexts.

The kind of functionality outlined in these two sections is also crucial at a lower level, when dealing with *quality of service* aspects. This can be found in 3G mobile telecommunication networks (bandwidth versus reliability versus cost, for example), but it will also play an important role in smart services in general. Currently, service level agreements are set up and often monitored by hand, and this must change.

2.2.9. Learning. When designing agent systems, it is impossible to foresee all the potential situations an agent may encounter and specify behaviour optimally in advance. Agents must therefore learn from, and adapt to, their environment. This task is more complex when the agent is situated in an environment that contains other agents with different (and in many cases unknown) capabilities, goals, and beliefs. Multi-agent learning, (the ability of agents to learn how to communicate, cooperate, and compete) becomes crucial in such domains [2].

Learning is increasingly being seen as a key quality of agents, and research into learning agent technology, such as reinforcement learning and genetic algorithms, is now being carried out across Europe. Applications of learning agent technology have been especially successful in the areas of personalisation and information retrieval, and promising results have been achieved in the areas of robotics and telecommunications. More effort will be needed, however, to make learning an inherent part of commercial agent applications. One of the problems to tackle is the safety of learning agents, since trust in self-adapting agents is still a major hurdle.

Over the years, learning and adaptation has occupied researchers from disciplines such as artificial intelligence, machine learning, information retrieval and HCI, and in the agent domain, work has also concentrated around other areas including adaptive user interfaces, user profiling, and personalisation techniques. For example, there is significant commercial interest in personalisation, both as a means of delivering targeted products and services to customers, and as a way of exploiting the opportunities of *pervasive* computing, which refers to the ‘anywhere, anytime, on any device’ model of computing. Gathering information to support personalisation, and adapt it over time, implies machine learning.

2.3. Links to other disciplines

In addition to computing science, agent technologies have drawn on the work of a number of other disciplines, both theoretical and applied. In some cases, this flow has been in both directions, with an agent perspective leading to new insights or new research directions in the other discipline. The most important links are with the following.

Philosophy: A number of areas of philosophy have been influential in agent theory and design. The philosophy of beliefs and intentions [16, 100], for example, led directly to the BDI model of rational agency, used to represent the internal states of an autonomous agent. Speech act theory [9, 110], a branch of the philosophy of language, has been used to give a semantics to various languages, including the agent communication language FIPA ACL of FIPA [24, 50].

Similarly, argumentation theory [44] – the philosophy of argument and debate, which dates from the work of Aristotle [6] – is now being used by the designers of agent interaction protocols for the design of richer languages, able to support argument and non-deductive reasoning [19, 95]. Issues of trust and obligations in multi-agent systems [58, 75] have drawn on philosophical theories of delegation and norms [86].

Logic: As in Computer Science generally, recent years have seen a flowering of applications to agent technologies of formal logic, particularly modal and temporal logics [128]. Logics of knowledge and belief (epistemic logics) have been used to represent the internal states of agents in a computational manner, as in the BDI model [48, 104]; deontic logics have been used to represent obligations and norms in agent systems [86]; dynamic and process logics have been used to reason about the interactions between agents, e.g., in modelling the formation of coalitions between agents engaged in some activity [94, 97].

Economics: In applying agent technology to distributed resource allocation problems, such as the management of an electricity network, agent technology has naturally drawn on economic theory. Examples include game theory [92], which studies the properties of formalised economic interactions between participants, and mechanism design theory, which considers the problem of the optimal design of resource allocation mechanisms [107]. Auctions are the most common mechanism studied in economics, and the combination of economic theory, operations research and computer science has led to the emergence of a new discipline, *computational auction design*. The rapid growth of electronic auctions has facilitated this emergence, and led to an interaction with agent technology in order to automate such auctions and the mechanism design process itself [99].

Social sciences: Although perhaps less developed than for economics, various links between agent technologies and the social sciences have emerged. Because multi-agent systems are comprised of interacting, autonomous entities, issues of organisational design and political theory become important in their design and evaluation. Because prediction of other agents' actions may be important to an agent, sociological and legal theories of norms and group behaviour are relevant [25], along with psychological theories of trust and persuasion [54]. Moreover, for agents acting on behalf of others (whether human or not), preference elicitation is an important issue, and so there are emerging links with marketing theory where this subject has been studied for several decades.

Biology: Biological metaphors for computation have been very influential in computer science over the last three decades as, for example, in the development of evolutionary computation and neural network processing. Indeed, the agent metaphor itself may be seen as partly biologically-inspired, with a system of interacting software components being viewed in the same way as an eco-system of autonomous living entities. Conversely, multi-agent system models have found application for the simulation of biological systems, such as fish populations in the North Sea, in a similar manner to their use to simulate socio-economic domains [15].

These examples show the diversity of interfaces between agent technologies and other disciplines. For agent systems applied to simulation of corporate or public policy decision domains, many of these connections are present simultaneously.

2.4. Application and deployment

2.4.1. Overview. Potential applications of agent-based systems can be divided into three broad categories:

- Assistant agents, such as agents engaged in gathering information or executing transactions on behalf of their human principals on the Internet. The Trading Agent Competition (TAC), where agents seek to book hotels and make travel arrangements for their principals, provides an example of this type of application [125].
- Multi-agent decision systems, where the agents participating in the system must together make some joint decisions. For instance, a system of agents representing the various components of a telecommunications network may jointly seek to allocate scarce resources across the network, such as call-connections, and thereby manage the operation of the network. The joint decision-making mechanism used by the agents involved may be an economic mechanism, such as an auction, or an alternative mechanism, such as one based on argumentation.
- Multi-agent simulation systems, where the multi-agent system is used as a model to simulate some real-world domain. Typically, multi-agent models are used for domains with many different components, interacting in diverse and complex ways, and where the system-level properties are not readily inferred from the properties of the components. Examples of such domains include: human economies, human and animal societies, biological populations, road-traffic systems, computer networks, and games (such as the agent-based *Creatures* [56]).

The distinction between the first type of application and the other two is between a single agent and a multi-agent system, although agents in the first case may need to interact with many other agents. Decisions in the second and third cases are taken in some sense collectively, not individually as in the first case. The main distinction between the second and the third types of application is that the second has as its goal the system, comprised of agents, whereas the third has as its goal the *understanding* that comes from the system. In addition, in many simulations, the agents provide an appropriate representation of real world components, while in the second type of application, agents are used for what they do. A consequence of this is a distinction between multi-agent systems which themselves take decisions and those which only provide advice to human decision-makers. Deployment of multi-agent systems of the second type in situations where real decisions are taken generates, of course, a host of ethical and philosophical issues. How should decisions of the system be assessed? Who is responsible if decisions taken by the system are at fault? Under what circumstances should humans feel confident about the decision-making activities of these systems? The difficulty of these questions is one reason, perhaps, why multi-agent systems have yet to find great employment in decision-making roles.

2.4.2. Industrial and commercial applications. Though many industrial and commercial applications have been developed [96], the main areas in which agent-based applications have been reported are as follows: manufacturing, process control, telecommunication systems, air traffic control, traffic and transportation management, information filtering and gathering, electronic commerce, business process management, human capital management, skills management, (mobile) workforce management, defence, entertainment and medical care.

For example, in manufacturing, applications have addressed areas of configuration design of manufacturing products, collaborative design, scheduling and controlling manufacturing operations, controlling a manufacturing robot, and determining production sequences for a factory. In process control, which is a natural application for agents, by virtue of controllers being autonomous reactive systems, several applications have been developed. Perhaps the best known of these is ARCHON, a software platform for building multi-agent systems that has been applied in several domains, including electricity transportation management and particle accelerator control [23]. Other such systems have been developed for monitoring and diagnosing faults in nuclear power plants, spacecraft control, climate control and steel coil processing control.

In eCommerce, full automation through agents is still not with us, but an increasing amount of trade is being undertaken by agents, and there are already several interesting applications. These include: a simple electronic marketplace called Kasbah, in which agents buy and sell goods [20]; BargainFinder, an early application (no longer available) which was an agent that discovers the cheapest CDs; Jango, a personal shopping assistant able to search on-line stores for product availability and price information; and so on.

The telecommunications sector has also seen a significant amount of effort on agent technology since 1992. European funding programmes ACTS and EURESCOM devoted specific research lines devoted to agents and addressed issues such as their application to telecommunications services, service management and workflow, and methodologies for agent development. Participants included the principal European telcos: BT, Telecom Italia, Telefonica, Portugal Telecom, Telia.

In supply chain management, Lost Wax and Cap Gemini have developed an agent-based demonstrator in which aircraft are serviced, covering routine and emergency demands for mobile service engineers. Engineers have different capabilities to respond, according to their location and training. Service vans provide engineers with mobility and can carry a particular inventory of spares and specialist tools. Additional spares and tools are held in local or regional depots with differing logistical arrangements and lead times. Occasional, unplanned events, such as changes in the state of readiness for war or the elimination of resources require the system to adapt to the new environment immediately. The application is modelled as a set of interacting autonomous agents executing in the Lost Wax agent framework, which provides an application programming interface (API) through which agents interact with the environment and each other.

In the entertainment and leisure sector, agents have been used to develop computer games such as the highly successful Creatures [56], which provides a rich, simulated environment containing a number of synthetic agents that a user can interact with in real-time. They have also been used in cinema to play out roles analogous to those played by real, human actors, as in Titanic. More recently, the second in the *Lord of the Rings* film trilogy, *The Two Towers*, achieved visually impressive battle-scenes by using the *Massive* agent system. Although the battle scene was broadly predetermined, the movement and action of each individual character is controlled by perceiving and responding to the artificial environment and to other characters. The agents can learn over time and their behaviour can change. In this way, convincing and effective scenes emerge through the autonomous actions of computational agents.

2.4.3. ***Simulation applications.*** Multi-agent systems offer strong models for representing real-world environments with an appropriate degree of complexity and dynamism [28]. For example, simulation of economies, societies and biological environments are typical application areas [15].

Agent-based simulation is characterised by the *intersection* of three scientific fields, namely agent-based computing, the social sciences, and computer simulation. The social sciences study interaction among social entities and include social psychology, management, policy and some areas of biology. Computer simulation concerns techniques for simulating phenomena on a computer, such as discrete event, object-oriented, and equation-based simulation. Interesting and relevant work occurs in related *intersection* areas, including:

– social sciences and agent-based computing (social aspects of agent systems);
– computer simulation and agent-based computing (multi agent based simulation); and
– social sciences and computer simulation (social simulation).

There are two broad approaches within the agent-based social simulation research community. One is based on defining logical systems to underlie social interaction (the *foundational* model), and the other on observing social processes and modelling those (the *representational* model). The first approach is more influenced by social sciences, and the second by social simulation. These approaches can be used collaboratively.

Scientists find computer simulation useful when addressing changes that cannot be easily forecast, but typically the causes can be identified retrospectively. Flight simulators used to train pilots have a similar approach. They teach pilots how to respond appropriately to types of unexpected events. Scenario analysis by business strategists and social policy analysts can have much the same purpose. An agent-based social simulation analysis can be more flexible and responsive than alternative modelling methods. For example, an agent-based social simulation analysis of climate change (following the Kyoto agreement) can capture the development of social pressures as the outcome of individual choices and social interaction. Information about how humans react in extreme circumstances may also help to make agents more robust.

There are three broad application areas in agent-based social simulation, as follows.

– Social structures and institutions, where observation and evidence are used to help set up the model. Sometimes these simulations help to develop plausible explanations of observed phenomena, sometimes to help in the design of organisational structures, or inform policy or managerial decisions. For example, the selection of product features by a company engaged in new product design may be based on an agent-based simulation model of the marketplace in which the new products will be sold; here, the agents represent consumers and choose between alternative product offerings on the basis of awareness, price, brand reputation, information they receive from other agents, etc.

- *Physical systems*: examples include agent-based models of intelligent buildings [29], of traffic systems [14], and of biological populations [41]. Research into the impacts of climate change on various biological populations, for instance, has been undertaken by means of multi-agent simulation models.
- Software systems of all types, currently including eCommerce and information agency. Traffic on a new telecommunications network, for instance, may be forecast by means of a multi-agent system simulation of predicted user behaviour.

For example, ant-inspired agent-based simulations of complex supply chains have been used by EuroBios to assist logistics analysts and plant schedulers at Air Liquide in making better decisions. Modern supply chains, with webs of relationships rather than a narrow pipeline, do not lend themselves well to traditional optimisation techniques. Non-linear agent-based simulations, in which all entities in the supply chain can be modelled and tracked, can provide better results. At Southwest Airlines, agent-based simulations of cargo routing revealed many missed opportunities to load cargo, and enabled a cut in multiple handling of freight by 75% and an increase in revenue of $10 million.

As is evident, simulation covers a range of phenomena from the most applied (such as manufacturing processes, traffic systems, information and control systems) to the most abstract (such as social dimensions, belief, trust, duty and right).

2.4.4. The commercial context. The commercial potential for agent-based systems was identified early by several major players. IBM saw agent systems as able to add value to underlying systems and developed a number of agent engines, including early work on the Aglets mobile agent system [73], and more recently their Autonomic Computing programme, which is concerned with may issues related to agents such as self-healing software, for example. Hewlett Packard was also involved as an early player, with simple task automation agents in its NewWave desktop environment, and went on to develop the eSpeak agent development tool, though this is now largely defunct. In 1997, Siemens released MECCA, the first FIPA-compliant agent platform.

Developments in consumer electronics also fuelled agent-based systems. Some high-profile undertakings failed to work commercially at the time, but can be informative to suppliers looking at related markets now. The US start-up General Magic proposed its Telescript agent language, to support the development of agents that could migrate across Telescript-enabled nodes [126]. Although somewhat overtaken by events, this work underpins current mobile agent research.

Other high-profile companies that adopted agent technology in one form or another included Firefly, an offshoot of the MIT Media Lab in the US, which provided personalised end-user Web interfaces, and Autonomy in the UK, which used complex pattern recognition systems to provide information management agents. (Although these companies have been both hugely successful and much less successful at different times, they no longer present themselves in agent terms. Perhaps one reason for this is the convergence of areas in which the technologies underpinning agents are being seen as a fundamental part of computing as a whole rather than agents in particular).

The late 1990s enthusiasm for new technology, which also included such things as consumer interface agents, was excessive and somewhat misleading, given the limited capacity to reproduce end user preferences at the time. But agents were even then a powerful and flexible way of structuring software. IBM and some other systems developers saw the potential for agents to provide a powerful, flexible new set of user-friendly functions. They also saw agents supporting integration of older software with new, agent-based systems, through agent "wrappers".

By the end of the 1990s, IT industry pundits recognised that in reality there were many different kinds of agents. Agents to support complexity in real time command and control systems need different qualities from personalised user interface agents for consumer Web access, for example.

Indeed, the current commercial environment offers a much more sober assessment of the value of agent technology, and an increasing number of sustained efforts both to use agent technology as part of mainstream software development and to adopt it for commercial advantage. The range of corporate entities with a stake in this space is varied.

At one end of the spectrum, the major players such as IBM, Microsoft, Siemens, HP Labs, BTexact, etc, all continue to invest in the R&D of basic technology as well as trying to find commercial application in products. More focused start-ups such as Tryllian in the Netherlands, Agent Oriented Software and Magenta in the UK and IKV++ and Living Systems in Germany, for example, offer specific agent products that are the cornerstones of their business. A more recent model of the general software development house, which also offers expertise in the development of agent platforms for particular purposes, in the context of traditional software development, has also begun to emerge. Lost Wax in the UK and Whitestein Technologies in Switzerland, are examples. Finally, what might be called user organisations are also recognising the benefits, with DaimlerChrysler providing perhaps the best example of a real quantified advantage through their use of agent technology for car production in scheduling on the factory floor.

A large number of companies are implicitly working on, or using, agent technology, blending it into established practices and existing technologies. In this sense, the specific agent aspects are less visible, but this suggests an increasing maturity of the technology rather than any failure. Indeed, as companies mature in their market approach, the varied public perceptions of agents, which often have an academic angle, seldom match what these companies are offering, and the agent technologies moves to the background, being an enabler rather than a sales argument.

3. The broad agent roadmap

3.1. Predictions

In any high-technology domain, the systems deployed in commercial or industrial applications tend to embody research findings somewhat behind the leading edge of academic research. Multi-agent systems are no exception to this, with currently-deployed systems having features found in published academic research and prototypes of 3–5 years ago. By looking at current academic research interests and areas of focus, we are able to extrapolate future trends in deployed systems.

Accordingly, we have identified four broad phases of the future development of deployed multi-agent systems as summarized in Table 1. These phases are, of necessity, only indicative, since some companies and organisations will be leading users of agent technologies, pushing applications ahead of these phases, while many others will be laggards. We aim to describe the majority of deployed applications at each time period. Note that this view on timescales takes the *development* view rather than the *research* view in that typically research is about 3–5 years *ahead* of development in this context. At the same time, the predictions are bold, and relate to the beginning of development rather than full and successful take-up.

The time phases are distinguished along five dimensions:

– the degree to which the participating agents share common domain knowledge and common goals;
– the degree to which participating agents are designed by the same or diverse design teams;
– the nature of the communications languages and interaction protocols used by the agents participating in the multi-agent systems (which can range from *ad hoc* languages through fixed standardised languages, to emergent languages);
– the scale of the system, such as how many agent participants can be supported by the system, how many users, or the complexity of the system as a whole; and
– the design methodologies (if any) used for the design of the system – for example, while there are currently established object-oriented development methodologies, no such routes exist for agent-oriented systems, which must either use unsuitable or *ad hoc* methods.

3.2. Phase 1: Current

Multi-agent systems in current deployment are typically designed by one design team for one corporate environment, with participating agents sharing common high-level goals in a single domain. These systems may be characterised as *closed*. The communications languages and interaction protocols are typically *in-house* protocols, and are defined by the design team prior to any agent interactions. Systems are usually only scalable under controlled, or simulated, conditions (though efforts are underway to ensure to address this, and Tryllian's agent platform, for example, can run 30,000 active agents). Design approaches tend to be *ad hoc*, inspired by the agent paradigm rather than using any specific methodologies. Examples of the systems developed in this phase are those for the management of utility networks.

It is likely that, for the foreseeable future, there will be a substantial commercial demand for *closed* multi-agent systems because of the security concerns that arise from open systems. While progress in this respect will change the nature of agent systems, the importance of closed, well protected systems must not be underestimated.

3.3. *Phase 2: Near-Term Future (c. 2004–2005)*

In the next phase of development, systems will increasingly be designed to cross corporate boundaries, so that the participating agents have fewer goals in common, although their interactions will still concern a common domain. However, despite this diversity, all participating agents are designed by the same team designing the system and will share common domain knowledge. Increasingly, standard agent communications languages, such as FIPAACL, are used, but interaction protocols remain non-standard. These systems are able to handle large numbers of agents in pre-determined environments, such as those of Grid applications and Agentcities. Development of these systems will increasingly use top-down methodologies, such as the GAIA [130] methodology, or middle-out methodologies supporting applications based on service-oriented architectures. Example systems developed in this phase include those to enable automated scheduling coordination between different departments of the same company, closed-user groups of manufacturing suppliers engaged in electronic procurement activities, or network-centric operations.

3.4. *Phase 3: Medium-Term Future (c. 2006–2008)*

In the third phase, multi-agent systems will permit participation by heterogeneous agents, designed by different designers or teams. Any agent will be able to participate in these systems, provided their (observable) behaviour conforms to publicly-stated requirements and standards. However, these open systems will typically be specific to particular application domains, such as B2B eCommerce or Bioinformatics. The languages and protocols used in these systems will be agreed and standardised, perhaps being drawn from public libraries of alternative protocols. These libraries will likely differ by domain. Ontologies, in particular, will be important to master this semantic heterogeneity.

The systems will scale to large numbers of participants, although typically only within the domains concerned. The third phase will see the development of bridge agents, able to translate between separate domains. Thus, for example, a multi-agent system for automated meta-analysis of research results in some area of biology will be able to utilise bridge agents to undertake commercial negotiations when interaction with an eCommerce system is required, say for access to information protected by patent. In the third phase, system development will proceed by standard agent-specific design methodologies, including templates and patterns for different types of agents and types of agent systems. Semantic issues related to, for example, coordination between heterogeneous agents and access control, are of particular importance here.

Examples of systems in this phase will be corporate B2B electronic procurement systems permitting participation by any supplier, rather than closed user groups.

3.5. *Phase 4: Long-term Future (c. 2009 onwards)*

The fourth phase in this projected future will see the development of open multi-agent systems spanning multiple application domains, and involving heterogeneous participants developed by diverse design teams. Agents seeking to participate in these systems will be able to learn the appropriate behaviour for participation in the course of doing so, rather than having to prove adherence before entry. Although standard communications languages and interaction protocols will have been available for some time, systems in this phase will enable these to emerge by evolutionary means from actual participant interactions, rather than being imposed. Of course, such languages, protocols and behaviours may be mere refinements of previously-developed standards, but will be tailored to their particular contexts of use.

By this phase, systems will be fully scalable in the sense that they will not be restricted to arbitrary limits (on agents, users, complexity, etc.). As with the previous phase, systems development will proceed by use of rigorous agent-specific design methodologies. Multi-agent systems deployed in this phase, for example, will support fully ambient computing.

4. Technological challenges

4.1. Challenges: Summary

Arising from this picture of the future of agent research, we see a number of broad technological challenges for research and development over the next decade. These are summarised in the table below, with each challenge being described in more detail in the sub-sections which follow. Each of these subsections includes a table that attempts to identify key sub-challenges, with indications of when they will attract successful attention from the research and development communities, in relation to the short term (ST), medium-term (MT) and long-term (LT) future discussed above. In particular, the tables suggest that long-term issues are worthy of strategic investment and effort while short-term issues are largely already addressed or are being addressed.

One important issue that we do not consider explicitly, but which merits substantial consideration relates to the design of business models for vendors, resellers and customers of agent technology: how can the different parties involved make money with agents?

4.2. Increase quality of agent software to industrial standard

One of the most fundamental obstacles to the take-up of agent technology is the lack of mature software development methodologies for agent-based systems. Clearly, basic principles of software and knowledge engineering need to be applied to the development and deployment of multi-agent systems, but they also need to be augmented to suit the differing demands of this new paradigm.

At present, many existing agent applications are developed in an ad hoc fashion, following little or no rigorous design methodology and with limited specification of the requirements or design of the agents or of a multi-agent system as a whole. To develop methods with which both the requirements of such systems, and the systems

Table 1. Phases of development for agent systems.

Phase of development	Key features of deployed systems
Phase 1: Current	– Closed agent systems applied in a specific corporate environment – Predefined, in-house protocols and languages – Ad hoc design inspired by agent paradigms – Implicit organisational context – Scalability largely in simulation
Phase 2: Short-term future (c. 2004–2005)	– Cross-boundary systems with participating agents known in advance – Semi-structured languages (e.g., FIPA) and non-standard protocols – Large numbers of agents interacting in a pre-determined environment (e.g., Grid applications, Agentcities) – Explicit but fixed organisational context – Top-down design methodologies (e.g., GAIA)
Phase 3: Medium-term future (c. 2006–2008)	– Open systems in specific domains (e.g., bioinformatics, eCommerce), with bridging agents or (ontology mappers) translating between domains – Participation by any agent able to satisfy publicly-advertised standards – Agreed protocols and languages – Serious, large-scale grid systems in single domains – Open organisational context enabling dynamic virtual organisations – Use of standard agent-specific design methodologies.
Phase 4: Longer-term future (c. 2009 onwards)	– Truly open and fully-scalable agent systems – Agents learn appropriate protocols and behaviour upon entry to system – Languages, protocols, and behaviours emerge from actual agent interactions – Evolving organisational structure with multiple, dynamic, interacting organisations – Self-modifying agent communications languages

themselves, can be modelled and specified at a conceptually acceptable level of detail, characteristics of real-world multi-agent applications need to be identified, in relation to specific domains. Such specifications describe the semantics of systems without concern for implementation details, providing a basis for verification, validation and testing of properties of the systems in the light of the specified requirements. These properties can relate to the functionality of the system behaviour, but properties that are sometimes called *non-functional* (such as scalability, performance, reliability and robustness) must also be addressed.

From an analysis point of view, systems including agent technology require *dedicated basic concepts and languages*. In particular, concepts representing dynamic aspects (e.g., time, action), locality aspects (e.g., position in a space), and concepts representing mental state (e.g., belief, desire) are needed. At the highest level, for example, coordinational, interactional, organisational and societal concepts such as joint goals, joint plans, society norms, interaction protocols, and organisation forms, must be able to be expressed. Moreover, at the level of the individual agents, representational elements are required for basic agent concepts such as observation, action, communication, beliefs, desires, goals and plans. Both functional and non-functional properties need to be covered.

From the design point of view, the key to facilitating the engineering of complex agent systems is reusability. To this end, designers must be provided with libraries of:

– generic organisation models (e.g., hierarchical organisations, flat organisations);
– generic agent models (e.g., purely reactive agent models, deliberative BDI models);
– generic task models (e.g., diagnostic tasks, information filtering tasks, transactions);
– communication languages and patterns for agent societies;
– ontology patterns for agent requirements, agent models and organisation models;
– interaction protocol patterns between agents with special roles;
– reusable organisation structures; and
– reusable knowledge bases.

Finally, at a tool level, software developers will require sophisticated yet easy-to-use agent-oriented CASE environments to help them in all aspects of the system development process, including the design, testing, maintenance and visualisation of agent-oriented systems. Some systems already have rudimentary elements of these.

For example, initial efforts in this area have attempted to develop an Agent UML (AUML) [91], with some success. The key task now is to ensure that there is support for developers through industrial strength tools and community building activities to provide access points. Here the challenges are technological in terms of tool support, methodological in providing ways to use the tools to support overarching development of agent systems, and societal in raising awareness and providing training support through, for example, a stock of case-studies that is resonant with developers.

Importantly, the success of future developments is likely to be ensured not by considering agents in isolation, but through their integration with evolving (and current) systems integration technologies (such as Jini and UDDI). Agent technologies are particularly relevant at higher levels of interaction relating to communication, ontologies, content and semantics, whereas business integration frameworks focus on the provision of scalable and robust solutions to the lower levels, including protocols, syntax, distributed computing APIs, directory services etc. It is important to build on current efforts to ensure that these are interoperable. The challenges in support of industrial strength agent systems are summarized in Table 2.

4.3. Provide effective agreed standards to allow open systems development

Much of the standardisation effort in the agent community has fallen to the FIPA and the Object Management Group (OMG), which are the premier agent standardisation bodies, although the former is the significant active organisation [26]. Importantly, as technologies converge, other non-agent standards are becoming

Table 2. Challenges in support of industrial strength agent software.

Industrial strength software	Now	ST	MT	LT
Peer to peer	•	•	•	•
Web services	•	•	•	•
Agent UML	•	•	•	•
Better development tools		•	•	•
Generic designs for coordination			•	•
Libraries for agent-oriented development			•	•

increasingly relevant [127]; over the next few years, there will be a much larger role for the less rich, but more widely adopted, Web services standards. Standards efforts in related fields have been discussed above.

The core mission of the FIPA software agent standards consortium is to facilitate the interoperation and interworking between agents across multiple, heterogeneous agent systems. To this purpose, FIPA has been working on specifications that range from agent platform architectures to support communicating agents, semantic communication languages and content languages for expressing messages and interaction protocols that expand the scope from single messages to complete transactions. The core message of FIPA is that through a combination of speech acts, predicate logic and public ontologies, standard ways of interpreting communication between agents can be offered that respect the intended meaning of the communication.

Currently, FIPA's main activities are focusing on the following.

- Promoting to *standard* status a core set of FIPA specifications.
- Building a service model for representing, modelling, discovering and using services.
- Developing a new semantic framework to reflect the needs of verifiability and conformance. In particular, the objective is to adopt or define a semantic framework that can give an account of FIPA's existing communicative acts and interaction protocols as well as a number of additional constructs.
- Creating new specifications to ensure that interoperability between FIPA-compliant agent platforms and platform fragments can be maintained in *ad hoc* networks.
- Standardising ontology modelling, representations and use within agent systems.

Also required for open systems development will be public libraries of interaction protocols designed for specific interactions (such as [37]). These may use existing agent communications languages, as do the Contract Net, English Auction and Dutch auction protocols when implemented using FIPA ACL. They may however be implemented in ad-hoc communications languages, as many of the various dialogue game protocols for agent argumentation currently tend to do. As more sophisticated interactions become common in open agent systems, there will be a need for libraries of such interaction protocols, available for re-use.

Table 3. Challenges in support of standards for agent systems.

Agreed standards	Now	ST	MT	LT
Peer to peer	•	•	•	•
Web services	•	•	•	•
FIPA ACL	•	•	•	•
Better development tools		•	•	•
Flexible business/trading languages			•	•
Libraries of interaction protocols			•	•
Tools for evolution of communications languages and protocols				•

In addition to standard languages and interaction protocols, open agent societies will require the ability to collectively evolve languages and protocols specific to the application domain and to the agents involved. Some work has commenced on defining the minimum requirements for a group of agents with no prior experience of each other to evolve a sophisticated communications language, but this work is still in its infancy. Research in this area will draw on linguistics, social anthropology, biology, the philosophy of language and information theory. The challenges for agent standards are summarized in Table 3.

4.4. Provide semantic infrastructure for open agent communities

At present, information agents exist in academic and commercial laboratories, but are not widely available in real world applications. The move out of the laboratory is likely to happen over the next 10 years, but requires the following:

- a greater understanding of how agents, databases and information systems interact;
- investigation of the real-world implications of information agents (for example, including the economic effects of shopbots [81]); and
- development of benchmarks for system performance and efficiency.

Moreover, a much higher degree of automation than is currently available in dealing with knowledge management is needed for information agents. In particular, this demands:

- new web standards that enable structural and semantic description of information; and
- services that make use of these semantic representations for information access at a higher level.

The creation of common *ontologies*, thesauri or knowledge bases play a central role here, and merits further work on the formal descriptions of information and, potentially, a reference architecture to support the higher level services mentioned above.

Through the convergence of DAML and OIL [61], a new standard for ontology representation and reasoning has been established as OWL [62], but there are still

Table 4. Challenges in support of semantic infrastructure for open systems.

Infrastructure for open communities	Now	ST	MT	LT
Semantic description	•	•	•	•
Data integration and semantic web	•	•	•	•
Semantic interaction		•	•	•
Web mining		•	•	•
Agent-enabled semantic web (of services)			•	•
Shared, improved ontologies				•

known limitations. Additionally, the development of ontologies themselves raises interesting questions that have yet to be answered. How much of the domain semantics need to be explicitly encoded, and what is the separation between ontologies and the implementations of agents that use them? The level of this separation has important consequences as ontologies change over time and agents have to be able to cope with the change. How important is inference, and what kinds of inference mechanisms are needed?

In summary, although the use of OWL suggests a timely convergence of standards for wider adoption, the development of shared ontologies within this framework is critical – they must be published, hosted, and used more widely in order to establish even limited use of shared terminology and representations. In particular, generic tool and service support for enabling the sharing of ontologies will become increasingly important in developing critical mass. Additionally, a whole set of questions relating to enforcing consistent modelling approaches when developing ontologies must be answered, but other communities (e.g., Semantic Web) are working on these, but we shall not consider them further here. The challenges identified are summarized in Table 4.

4.5. Develop reasoning capabilities for agents in open environments

4.5.1. Virtual organisations. At present, *organisational* approaches (e.g., [89]) do not adequately handle the issues inherent in open multi-agent systems, namely heterogeneity of agents, trust and accountability, failure handling and recovery, and societal change. Human societies have successfully coped with similar issues by creating institutions that establish norms for group dynamics in open systems. The next challenge for agent-based computing is to develop appropriate representations of analogous computational concepts to the norms, legislation, authorities, enforcement, etc., that can underpin the development and deployment of dynamic electronic institutions. As mentioned above, agent researchers will need to draw on political science and sociology to develop sophisticated and effective agent societies.

4.5.2. Coalition formation. Similarly, virtual organisations involve dynamic coalitions of small groups that can provide more services and make more profits than an individual group. Moreover, such coalitions can disband when they are no longer effective. At present, coalition formation for virtual organisations is limited, with such organisations largely static. The automation of coalition formation [101] will save both time and labour, and may be more effective at finding better coalitions than humans in complex settings.

Although coalition formation has been addressed in game theory for some time, it has typically been centralised and computationally infeasible, suffering from a number of important drawbacks. For example, it is only applicable for small numbers of agents, and generally favours one big coalition, limiting the scope of the application. Recent work using a dialect of modal propositional dynamic logic (PDL) to model games and interactions has permitted the representation of coalitions and may prove valuable in formalising reasoning about coalitions of agents [97].

Emerging computation infrastructures such as the Grid are now providing a greater need for effective work in virtual organisations to facilitate higher-level applications. Indeed, virtual organisations have been identified by [52] as the tool with which to unwrap the power of the Grid, and agent-based computing offers the means to underpin it. Similarly, emerging work on Web services, eBusiness workflow systems and (e.g.) Agentcities all have long-term aims of supporting dynamic formation of virtual organisations.

4.5.3. Negotiation and argumentation strategies. To date, research into negotiation can be considered point work, with particular efforts or examples rather than a more coherent science of negotiation strategy. Strategies identified by economic or game theoretic reasoning, for example, tend to be specific to the auction or game mechanism involved. This makes their identification and deployment something of a black art, without any over-arching and computational theory. It also limits implementation possibilities: for example, it is not yet possible to define a computational agent capable of effective negotiation in any arbitrary negotiation context.

Moreover, as mentioned above, research into negotiation and deliberation mechanisms which are more complex than auctions and game-theoretic mechanisms is still in its infancy. Research into argumentation mechanisms [103], for example, and the strategies appropriate for participants under them, is also needed before these mechanisms will achieve widespread deployment.

For commercial deployment of negotiation and argumentation strategies, we need the following.

- Rigorous testing of existing algorithms in realistic environments to identify their strengths and weaknesses.
- Development of an over-arching theory or methodology to identify which algorithmic techniques should be deployed in which circumstances.
- Development of algorithms for negotiating in more complex environments, for example using argumentation.
- Development of efficient argumentation engines, to include domain-specific argumentation strategies.
- Development of techniques for allowing users to specify their preferences and desired outcome of negotiation in complex environments.
- Development of techniques to enable agents to identify, create and dissolve coalitions in multi-agent negotiation and argumentation contexts.

Table 5. Challenges in support of agent reasoning in open environments.

Reasoning in open environments	Now	ST	MT	LT
Organisational views of agent systems		•	•	•
Norms and social structure			•	•
Theory and practice of negotiation strategies			•	•
Enhanced understanding of agent society dynamics			•	•
Theory and practice of argumentation strategies			•	•
Automated eScience systems				•

4.5.4. Domain-specific models of reasoning. In many domains of human research activity, there is an information explosion currently occurring. In genomics and proteomics, for example, we are facing an ever-growing avalanche of information, for which automated analysis procedures are required [18]. One approach to this problem is the development of automated scientific inferencing procedures which would consider all the data and evidence available, and from this automatically generate new knowledge in the form of justified scientific conclusions. However, evidence for hypotheses in scientific domains is rarely ever initially conclusive in one direction or another, but may support multiple, competing hypotheses. One could therefore view this automated inference engine as a multi-agent system, where different agents propose hypotheses and marshall evidence to support them in argument with one another. To create such systems will require, in addition to agent technologies, models of scientific reasoning and of scientific argument; these may well be domain-specific. The creation of this vision of automated eScience will require collaboration between agent researchers and both domain experts, such as genome biologists, and philosophers of science.

In all this work on developing reasoning abilities of individual agents (shown in Table 5), however, an overarching issue remains the need to ensure that there is an appropriate agent response to the mental conditions arising from autonomous intentional decision-making [38]. This work relates to a development of the agent architectures described earlier, with deeper and better analysed models.

4.6. Develop agent ability to understand user requirements

At the architecture level, future avenues for learning research include developing distributed models of profile management, as well as more general distributed agent learning techniques rather than just single agent learning in multi-agent domains. Other research communities have considerable expertise in the elicitation of user preferences and utilities, and future research in agent technologies should draw on this work. One community is marketing theory which, over the course of the last 50 years, has developed an impressive range of proven, practical techniques for elicitation of desires and preferences from potential consumers [74]. Another community is the Uncertainty in Artificial Intelligence (UAI) community, where techniques have been developed over the last decade to obtain expert probabilities and utilities needed for calibration of probabilistic belief networks [67, 105]. Finally, the field of user modelling has focused on the need to elicit what job needs doing in the first place.

At the centre of every personalisation technology is some form of user *profile* which represents the information needs and preferences of the user. Such profiles can take a variety of forms, ranging from sparse-vectors of document ratings to rich, highly structured representations based upon XML. A profile may be located entirely within the locus of the user's control, e.g., on their own PC or PDA, or may be retained as one of many such profiles on a server controlled by a service provider. The majority of existing work on agent-based personalisation makes the simplifying assumption that profiles are held in some central repository where they can be processed, compared, and otherwise manipulated; recently, some research has begun to challenge this assumption by moving to a more distributed approach. Particularly interesting questions to answer here include how to deal with the security or privacy of user profiles, and how users can find relevant information if they don't want to reveal too much of their profiles.

Developing approaches to personalisation that can operate in a standards-based, pervasive computing environment presents many interesting research challenges, including how to integrate machine learning techniques (for profile adaptation) with structured XML-based profile representations. Another area deserving of greater activity is that of distributed profile management, a task for which the agent-based paradigm should be well suited. The impact of the emerging Semantic Web on approaches for wrapper induction and text-mining also requires careful study.

4.7. Develop agent ability to adapt to changes in environment

Even though learning technology is clearly crucial for open and scalable multi-agent systems, it is still in early development. While there has progress in many areas, such as evolutionary approaches and reinforcement learning, these have still not made the transition to real-world applications. Reasons for this can be found in problems of scalability and in user trust in self-adapting software. In the longer term, learning techniques are likely to become a central part of agent systems, while the shorter term offers application opportunities in areas such as interactive entertainment, which are not safety-critical (see Table 6).

Many agent research areas have been looking mainly at non-adaptive technology. However, with increasing maturity of these areas, learning techniques will increasingly move towards the center stage in these areas. Examples of areas where learning will receive more attention in the near and middle-term future are communication, negotiation, planning and coordination, and information and knowledge management.

While learning techniques for single agents are relatively well-advanced, the area of multi-agent learning still needs more work, particularly in relation to issues of scalability. Many test application domains are overly simplistic, and it is questionable as to how well these methods would work in complex and large-scale real-world applications. Hierarchical approaches that represent, and reason over, environment states at different levels of granularity seem promising in overcoming these problems.

Table 6. Challenges in support of agent adaptation to environments.

Learning technologies	Now	ST	MT	LT
Adaptation	•	•	•	•
Evolving agents	•	•	•	•
Personalisation	•	•	•	•
Distributed learning		•	•	•
Hybrid technologies		•	•	•
Self organisation			•	•
Run-time reconfiguration and redesign				•

Most work to date on learning techniques has been focused on reinforcement learning and evolutionary approaches [121]. While these techniques are naturally suited to agents, some limitations have been encountered, especially regarding convergence speed towards the desired result. Inductive learning, while being the focus of much machine learning research outside the agent community, has mainly seen application in information retrieval and data mining. Research in hybrid methods (e.g., relational reinforcement learning combining inductive logic programming and Q-learning) will bring the different branches together and result in techniques that are likely to overcome many of the current limitations.

Aside from the personalisation aspects of learning and adaptation that have formed the basis for much current work, the development of advanced technologies for personal information management raises a number of important social issues. Privacy is an obvious concern for many users. Achieving truly pervasive technology, with support for personalisation, should move society closer to the goal of universal information access, by making information accessible on the widest range of platforms in a form that is tailored to the needs of the individual.

Issues here involve the relationship, and integration, of agents with the Semantic Web, to address an explicit gap at present. They also relate to the need to ensure that emerging profile standards (such as CPEXchange) provide appropriate support for adaptive technologies.

4.8. *Ensure user confidence and trust in agent systems*

Although currently deployed agent applications often provide good security, for agents autonomously acting on behalf of their owner, several additional factors need to be addressed as summarized in Table 7. First, considerable effort must still be put into issues of security in open agent systems. Efforts by other communities are tackling some aspects here, but more on specific agent security concerns needs to be done. Second, collaboration of any kind, especially in situations in which computers act on behalf of users or organisations, will only succeed if there is trust. For this trust to be given requires a variety of factors to be in place.

– A user must have confidence that an agent or group of agents which represents them within an open system will act effectively on their behalf – it must be at least as effective as the user would be in similar circumstances. For this to occur, the agent must have an accurate model of the goals and preferred outcomes of the user (as discussed earlier) and must implement appropriate behaviour in a robust and reliable way. There are two approaches to this that are used by the software engineering community, and are being extended to cover open agent systems by current research. First, formal methods can be used to prove desirable properties of an agent in a given class of open systems. For example, it could be demonstrated that a trading agent will never enter into a loss-making deal. This is ideal, in that the user can be certain that the agent will act abiding by these properties. However, in practical systems it is often not possible to undertake such a formal analysis. An alternative approach is through extensive reliability testing, where the environment in which an agent is to be deployed is simulated and a large variety of scenarios are played out to ensure that the behaviour of the agent is appropriate. While test methodologies for standard software are well-understood, research is required to adapt them for use on agents which are to be deployed in an open system [76].

– Agents must be secure and tamper-proof, and must not reveal information inappropriately (as, for example, with bank account details). There is much work on system security, cryptography and privacy which can be exploited and adapted for use in agent technology. Also, it must not be possible for another agent to pretend to represent the user. Again, work outside the agent community in areas such as digital signatures and certificates, non-repudiation protocols, and contract verifiability are important.
– If a user is to trust the outcome of an open agent system, they must have confidence that agents representing other parties or organisations will behave within certain constraints. For example, if an agent makes an agreement to trade with another agent, the user should have confidence that the other end-user will indeed abide by the agreement made. The agent community is exploring different approaches to solving this problem, as follows.
  - Reputation mechanisms to assess the past behaviour of particular agents or users, to allow avoidance of untrustworthy agents in future [108].
  - The adoption of norms (social rules) by all members of an open system, and the enforcement of sanctions against agents that transgress them [75]. This may involve a third-party to enforce the norms and impose sanctions appropriately.
  - In certain circumstances, it is possible to design self-enforcing protocols, which ensure that it is not in the interests of any party to break them. (For example, *incentive compatible* negotiation mechanisms ensure that it is best to tell the truth).
  - The use of electronic contracts to represent and enforce agreements between several parties. As well as technical problems associated with the representation and automation of contracts, there is also the important legal issue of the status of electronic contracts negotiated between automated entities.

Beyond the technical challenges are issues relating to trust in adopting agent technology in the first place. In order to encourage adoption and use of agents, we must ensure that full control is placed with the user, and is only relaxed to give agents autonomy as user trust in agents is built up. To facilitate this, initial applications for internet commerce, for example, will require the internal functioning of an agent to

Table 7. Challenges in support of user confidence and agent trust.

Trust and reputation	Now	ST	MT	LT
Reputation mechanisms	•	•	•	•
Self-enforcing protocols	•	•	•	•
Reliability testing for agents		•	•	•
Security and verifiability for agents		•	•	•
Electronic contracts		•	•	•
Norms and social structure			•	•
Fomal methods for open agent systems			•	•

be visible and adjustable by users, to enable the user to predict how the agent will behave in the future.

5. Challenges for the agent community

Distinct from the more technical challenges that can be seen as clear advances in technological research or development as enumerated above, there are also important community activities that can either contribute to the success of the field's development or can prove a constraining influence. Technology alone is not enough – the connections with, and influences from, related technology and application areas, as well as related initiatives and developments with their own impetus, are critical. A sustained programme of development must seek to engage in these strategic community goals.

5.1. *Leverage underpinning work on similar problems in computer science*

The concepts underlying the field of agent-based computing are not unique to this particular branch of computer science and, as existing software technology becomes more sophisticated and moves up the application stack, it will increasingly intersect with many sub-areas of agents. For example, distributed object technologies share many similar but less sophisticated abstractions, and employ similar notions of brokers facilitating interaction between components.

Similarly, existing approaches to software engineering, including work on verification and validation, have much to offer to current efforts to develop agent-specific development methodologies that are tailored to the particular needs of this paradigm. More generally, theories of agent interaction have yet to draw in detail on abstract theories of distributed computation, an area of considerable research effort in theoretical computer science in recent years [1, 55].

It is important that stronger links with these traditional areas of computing are established and reinforced in order to leverage underpinning work from these related areas, while also providing scope for work in the development of agent-based systems correspondingly to inform them. More specifically, a convergence of technologies can be seen in visions of Ambient Intelligence [42], for example, in which the adoption of agent technologies is implicit in the fabric of the environment that requires the underpinning of other supporting fields of computing.

In particular, current work on the Semantic Web [59], Web services and on Grid and peer-to-peer computing are intimately entwined with work on agent-based systems. It is difficult to imagine, for example, how the virtual organisations envisaged as the top level of support for Grid applications can be developed without involving agents and agent technologies [119] (see below). Similarly, Semantic Web work on ontologies, for example, can facilitate agent interoperability, while agent-based computing offers appropriate computational entities for traversal, retrieval and processing in that environment.

Issue: Existing software technology is moving up the application stack towards agents.
Recommendations:
– Build bridges with the distributed systems, software engineering and object technology communities.
– Develop agent tools and technologies on top of existing standards.
– Engage in related (lower level) standardisation activities (e.g., UDDI, WSDL, WSFL, XLANG, OMG CORBA and other widely used industrial strength middleware).
– Clearly articulate the relationship with distributed software engineering.
– Explore and clarify relationships between agent theories and abstract theories of distributed computation.

5.2. Link with related areas in Computer Science dealing with different problems

In a similar fashion, related disciplines that typically address different problems to those tackled by agent-based computing can also provide valuable inputs to research and development of agent-based systems. Particular examples of relevant areas include the fields of artificial life and computational biology, which have developed an arsenal of techniques that may be appropriate for application to agent-related problems. The building of bridges to different disciplines should thus also focus on areas with different target problems, so that techniques may be adopted wholesale, or in hybrid approaches to offer new solutions.

This can be clearly observed in the potential that artificial life techniques, for example, offer to agent-based simulation, and to learning and adaptation in such contexts. Similar work is being undertaken within the robotics and animations communities. Another example is the recent experience gained by the UAI community in the elicitation of user preferences and utilities for calibration of probabilistic influence modes, such as Bayes nets. These methods of elicitation may have direct application for agent elicitation of human preferences, utilities and decision-procedures.

Issue: Other disciplines are addressing different problems with techniques that may be applied to agent systems.

Recommendations:
– Build bridges, especially to the artificial life, robotics, UAI, logic programming and the *traditional* mathematical modelling communities.
– Develop agent-based systems using hybrid approaches.
– Develop metrics to assess the relative strengths and weakness of different approaches.

5.3. Extend and deepen links with other disciplines

As mentioned above, agent theory and practice has benefited from connections with other disciplines. For many disciplines these links are already deep and fruitful, such as those with economics, logic and philosophy. For other disciplines, there exists potential to develop closer connections. For instance, agent systems designers could draw more extensively than they have done on political theory and sociology in the design of agent societies, and on decision theory in the assessment of performance of agent systems. Similarly, agent-mediated electronic commerce has yet to make great use of the models and techniques developed by marketing theorists for preference elicitation or the diffusion of innovations.

Both sides can potentially benefit from such connections. Indeed, new disciplines can even emerge, as appears to be happening in computational auction theory, where theoretical economists, computer scientists and operations researchers have each made crucial contributions [117]. One example of a potential new discipline is a rigorous theory of distributed decision-making, which would seek to develop performance assessment criteria for multi-agent systems engaged in decision-making. Such a theory would draw on agent theory, the philosophy and sociology of delegation, organisation behaviour and decision theory.

Issue: Prior research findings and practical learning from other disciplines are relevant to the design and deployment of multi-agent systems technologies.
Recommendations:
– Seek to maintain and deepen existing strong connections with related disciplines, including economics, game theory, logic, philosophy and biology.
– Build new connections with sociology, anthropology, organisation design, political science, marketing theory and decision theory.

5.4. Encourage industry take-up

Commercial deployment of agent systems is currently confined to early adopters in some segments of industry and government, such as utility companies, and agent systems have yet to achieve widespread deployment in operating environments. There are several possible reasons for this. One is that although there are several relatively mature, development-oriented platforms (with fewer agent-specific capabilities) and some richer, more research-oriented agent platforms, most platforms are still too immature for operational environments.

This factor will be alleviated in due course, as technology moves from pure research to development [77]. To further this trend, we recommend that working prototypes of commercial agent systems be developed for specific industry sectors and made available for commercial use. Additionally, there may be a lack of awareness of the potential applications of agent systems. To facilitate greater awareness, we recommend that a set of early adopter case studies be prepared, both successful and unsuccessful, with an analysis of the reasons for success or failure. Such case studies should also include assessments of the resources and timescales required, and the factors critical to their successful deployment. These case studies may then be distributed to potential user organisations, such as large corporations and government agencies throughout Europe, and publicised to national computer industry associations and societies.

Another possible reason for a lack of deployment is the cost of system development and implementation, both in direct financial terms and in terms of required skills and timescales. High deployment costs are a feature of any new technology, and if this is the case for agent systems, it would not be surprising. As agent design tools and standard methodologies are developed, and as development teams gain greater experience, these costs should fall. To ensure that these experiences are disseminated beyond the early adopter community to other organisations, we recommend that best practices for agent-oriented development and deployment be identified and publicised. Similarly, we support contribution to the maintenance and development of strong standards for improving the interoperability of multi-agent systems developed in heterogeneous contexts and having to interact in an open and distributed environment.

An alternative approach, which may offer a better route, and which some companies are adopting, is to fit their agent work into existing models as much as possible. For example, the Tryllian toolkit requires Java skills rather than AI or agent-specific skills, while Whitestein leverages J2EE and its associated skills base. In either case, however, the value of agent technology is that it allows complex problems to be solved more quickly and more easily. Others suggest that it also enables easier deployment and quicker results and return on investment.

The lack of industrial take-up can also be understood through the absence of a migration path. We cannot hope to establish agent technology radically and from scratch. Instead, we need to show how industry can migrate to agent-based solutions gradually, while protecting existing investments in hardware, software, and skills. The alternative approach of the paragraph above in which products gradually evolve from established practices and technologies, offers perhaps the best option for more immediate take-up.

The success of the AgentLink conference on *Agents for Commercial Applications*, held in London in January 2002, and demonstrated by the presence of over 60 different commercial organisations, shows the wide extent of commercial interest in agent systems and the possibilities for further dissemination of agent technologies and deployment experiences.

Issue: Encourage increased deployment of agent technology.

Recommendations:
- Build prototypes spanning organisational boundaries (potentially conflicting).
- Encourage early adopters of agent technology, especially ones with some risk.
- Develop catalogue of early adopter case studies, both successful and unsuccessful.
- Provide analyses of reasons for success and failure cases.
- Identify best practice for agent oriented development and deployment.
- Support standardisation efforts.
- Support early industry training efforts.
- Provide migration paths to allow smooth evolution of agent-based solutions, services, systems, and products from today's solutions, services, systems, and products.

6. Application opportunities

Finally, there are a number of existing and emerging application domains for agent technologies and multi-agent systems. We present a brief description of several of these domains, to demonstrate their wide range and diversity. They indicate the potential impact of agent-related technologies on human life and society, and the fact that agent technologies will play an increasingly important role in underpinning Europe's transition to an information-intensive society and economy. Many other very exciting and potentially very large areas for application are already being investigated, such as health care and manufacturing, but we will not examine them further here.

6.1. Ambient Intelligence

The notion of Ambient Intelligence (AmI) has largely arisen through the efforts of the European Commission in identifying challenges for European research and development in Information Society Technologies [42]. Aimed at seamless delivery of services and applications, it relies on three identified pillars of ubiquitous computing, ubiquitous communication and intelligent user interfaces, yet it offers perhaps the strongest motivation for, and justification of, agent technologies. The AmI vision describes an environment of potentially thousands of embedded and mobile devices (or software artefacts) interacting to support user-centred goals and activity. This suggests a component-oriented view of the world in which the artefacts are independent and distributed. The consensus is that autonomy, distribution, adaptation, responsiveness, and so on, are the key characterising features of these AmI artefacts, and in this sense they share the same characteristics as agents.

In particular, these AmI artefacts are likely to be function-specific (though possibly configurable to tasks) and will, of necessity, need to interact with numerous other AmI artefacts in the environment around them in order to achieve their goals. Interactions will take place between pairs of artefacts (in one-to-one cooperation or competition), between groups of artefacts (in reaching consensus decisions), and between artefacts and the infrastructure resources that comprise their environments (such as large-scale information repositories, or other supporting resources, possibly through agent encapsulation). Interactions like these enable the establishment of electronic institutions or virtual organisations, in which groups of agents come together to form coherent groups able to achieve some overarching goals.

Importantly, interactions will also occur between artefacts and users, potentially requiring greater sophistication in interface issues [115], and in user understanding (and modelling). Also, the openness of the system, and its heterogeneity, will require the employment of learning and adaptation techniques, since many properties of the environment and other agents cannot be known at design time.

Though largely included in the previous points, the environment provides the infrastructure that enables AmI scenarios to be realised. On the one hand, artefacts offering particular services can be distinguished from issues concerning facilitating services such as the physical infrastructure needed to support effective interaction through sensors and actuators, and the physical connectivity for supporting quick and efficient interactions, for example. On the other, they can also be distinguished from issues relating to the virtual infrastructure needed to support resource discovery, large-scale distributed and robust information repositories (as mentioned above), and the logical connectivity needed to enable effective interactions between large numbers of distributed artefacts and services, for example.

Two particularly important points to note in relation to the pervasiveness of artefacts in the environment relate to scalability (and more particularly, device scalability), or the need to ensure that large numbers of artefacts and services are accommodated, and the need to ensure that the heterogeneity of artefacts and services is facilitated by the provision of appropriate ontologies to enable the effective interactions mentioned above. To address all these aspects of AmI will require efforts to provide solutions to issues of operation, integration and visualisation of distributed sensors, *ad hoc* services and network infrastructure: a formidable challenge.

6.2. *Bioinformatics and Computational Biology*

An emerging domain of application for agent technologies is in the Biological Sciences. One of these is in the use of multi-agent systems for simulation modelling of biological systems, in a manner similar to their use for the simulation of socio-economic and public policy domains. Examples of such applications have ranged from simulation of within-cell information-signalling processes, where the agents represent the various molecular components of the cell [69], all the way to the simulation of macro-level biological systems, such as fish populations in the North Atlantic. As with simulation models in public policy domains, their use in biology enables the assessment of alternative assumptions and input parameters, and the development of a better understanding of the dynamics of complex interacting systems.

Another area of application in biology is in Bioinformatics. With the information explosion caused by genomics and proteomics there is a great need for automated information-gathering and information-inference tools. Information-gathering agents may provide assistance to human researchers in finding appropriate research literature or in conducting automated or semi-automated testing of data. In addition, data mining agents may present human researchers with a set of potential hypotheses that can be induced from the data sources. In particular, the kinds of resources available in the bioinformatics domain, with numerous databases and analysis tools independently administered in geographically distinct locations, lend themselves almost ideally to adoption of a multi-agent approach. Here, the environment is open and distributed with resources entering and leaving the system, there are large numbers of interactions between entities for various purposes, and the need for automation is substantial and pressing. Some early work in this direction, using agents for genome analysis, is demonstrated by the GeneWeaver project in the UK [18], and work using DECAF in the US [33]. More substantial work is now underway on the use of agents as part of a large-scale eScience project on a Bioinformatics Grid testbed [87], also in the UK.

Automated inference from biological data is still very immature. As mentioned above, a potential longer term application of multi-agent systems technologies is the use of agents engaged in reasoned argument to achieve resolution about ambiguous or conflicting experimental evidence, in a manner similar to the way in which human scientists do currently. This area of automated eScience is probably a decade or more from achievement, but will draw on the agent negotiation and argumentation mechanisms developed for distributed resource allocations problems, such as those found in eCommerce. As an example of the application of these techniques, consider the inference of genomes for animal or plant species. Obtaining the complete genome by experimental methods for all species is a task which could take hundreds of years, and so, in the interim, we may need to rely on inference of the genome of one species on the basis of comparison with those of related species. Since inferences from these other genomes may well conflict with one another, reasoned argument between intelligent agents, acting on behalf of scientists, in a multi-agent system, may provide a means to generate an overall consensus.

6.3. *Grid Computing*

The high-performance computing infrastructure, known as the Grid [51], for supporting large-scale distributed scientific endeavour has recently gained heightened and sustained interest from several communities, as a means of developing eScience applications such as those demanded by e.g., the bioinformatics scenarios described above, the Large Hadron Collider facility at CERN, engineering design optimisation, and combinatorial chemistry [53]. Yet it also provides a computing infrastructure for supporting more general applications that involve large-scale information handling, knowledge management and service provision. Typically, Grid computing is abstracted into several layers, which might include: a data-computation layer dealing with computational resource allocation, scheduling and execution; an information layer dealing with the representation, storage and access of information; and a knowledge layer, which deals with the way knowledge is acquired, retrieved, published and maintained.

It is natural to view large systems in terms of the services they offer, and consequently in terms of the entities providing or consuming services. Grid applications, in which typically many services may be involved, spread over a geographically distributed environment, which new services join and existing ones leave, thus very strongly suggest the use of agent-based computing. In this view, agents act on behalf of service owners, managing access to services, and ensuring that contracts are fulfilled. They also act on behalf of service consumers, locating services, agreeing contracts, and receiving and presenting results. Just as in the AmI vision, agents will be required to engage in interactions, to negotiate, and to make pro-active run-time decisions while responding to changing circumstances. In particular, agents will need to collaborate and to form coalitions of agents with different capabilities in support of new virtual organisations. As mentioned earlier, such virtual organisations have been identified by Foster [52] as the tool with which to unwrap the power of the Grid.

Initially geared towards high performance computing, grid computing is now being recognised as the future model for service-oriented environments, within and across enterprises. The impact will be larger than just virtual organisations – a global company is much like such a virtual organisation and will require similar technology.

6.4. Electronic Business

To date agents have been used in the first stages of eCommerce, product and merchant discovery and brokering [112]. The next step will involve moving into real trading – negotiating deals and making purchases. This stage will involve considerable research and development, including generating new products and services such as market-specific agent shells, payment and contracting methods, risk assessment and coverage; quality and performance certification, security, trust, and individualisation. Researchers will need to look to different fields that have dealt with interaction problems, such as game theory, economics and sociology.

It can be argued that the real impact of electronic commerce will be on a dramatic change in the supply chain [64]. If a consumer can contact directly the producer instead of a reseller it might produce an increase in efficiency of the overall supply chain. These changes in the supply chains will permit new markets to appear, old markets to change and the participation of new players. These observations raise some broader questions about eCommerce in general, and the speeding-up effects of agents in particular. Consumers who are excluded from the eCommerce loop may find their prices and choice become worse.

In the very near future a boom in agent-mediated auctions is expected. The auction is a long-established and well-understood trading mechanism, and available agent technology can support such agent-mediated auction houses.

Although agents can recommend, they do not yet authorise agreement. However, fully automated negotiations will soon be seen in areas where the problem can be specified simply, each trade is of relatively small value, the process is repeated often, and interactions are repeated very fast. Businesses buying bandwidth for virtual private networks, and electric power capacity, provide good examples. (This will also require the emergence of new business models, and resellers to add value, for example by intelligent solutions.)

As mentioned earlier, automated trading is being driven partly by the growth of electronic auctions for business-to-business procurement and partly by program trading on stock exchanges.

In the short term, travel agencies and retailing will be the primary business-to-consumer application domains using agent technology in eCommerce. One of the current efforts aimed at driving this forward can be seen in the TAC, which offered a sophisticated problem domain of multiple auctions for agents to compile travel packages for customers. Such initiatives can highlight the potential of agent technology for a wider audience, while at the same time contributing to the more rapid development of the field in a specific application and problem domain.

Here, one interesting segment is supply chain management for virtual and trans-national enterprises. On the other hand, it can be foreseen that agent technology in this market will enable small and medium enterprises to collaborate and form coalitions in much more flexible ways, almost regardless of geographic location.

In the longer term, full-supply chain integration is the aim. According to a PricewaterhouseCoopers report,[2] there were over 1000 public eMarkets and around 30,000 private exchanges at the start of 2001. Although the baseline domains exist, the lack of standards and uniformity of these platforms constrains what can currently be achieved, but offers a real challenge and opportunity for deployment of agent systems over the next 5–10 years.

6.5. Simulation

An important category of applications of multi-agent systems is in simulation of natural or artificial societies [15]. These applications include education and training systems, scenario exploration and policy systems, and entertainment systems.

6.5.1. Education and training. Multi-agent systems provide a natural basis for training of decision-makers in complex decision-making domains [106]. For example, defence simulations using multi-agent systems, can enable military planners, strategic defence staff and even operational staff to gain experience of complex military operations through simulations and war games [13, 60]. These simulated experiences are obtained instead of, or in addition to, experiences gained in actual military operations. Similarly, decision-makers in other complex and dynamic environments can gain valuable experience through exercises which simulate their real-world domain using multi-agent systems. Applications include marketplaces subject to rapid change, such as telecommunications markets undergoing deregulation, and markets for fast-moving consumer goods, such as breakfast cereals, where consumer tastes and competitor activities can lead to market turbulence. In these applications, as for those in defence, multi-agent systems may simulate over a few hours the dynamics of an actual market which could occur over several years, and so give trainee decision-makers rapid exposure to many diverse experiences. In addition, as the military example reveals, the decision-maker is allowed to learn through his or her mistakes, without these having real-world consequences.

6.5.2. Scenario exploration. Social simulation is somewhat unusual in that it does not require many of the challenges listed earlier to be addressed in order for it to succeed in the timescales considered in this report. Since simulations are by their nature closed (even though they may model open systems) they are almost immediately enabled. However, there are many open issues to be resolved before agent-based simulation models can be applied more widely to public policy domains. For example, there is as yet no general understanding of what constitutes *good* performance by a multi-agent system, except perhaps in some domains. There is no guarantee, for example, that an agent society in which different species of agents co-evolve in the course of their interactions with one another will progress in any sense; later generations of a species may be less fit than earlier generations of that same species when pitted against earlier generations of their competitor species. In such a case, at which time-point should the simulation be terminated? Different termination points may lead to different assessments of system performance and different recommendations to policy makers. Indeed, the question of performance assessment of multi-agent systems is part of a larger, mostly open, question of performance assessment of decision support systems in general.

In addition, many applications of agent systems to public or social policy domains involve the development of alternative scenarios, the outcomes of which are used to guide human decision-makers. But at present there is no formal theory of scenarios and scenario analysis, which would tell us how to construct scenarios, how many scenarios to construct and how to reason between and across their outcomes [83]. Developing formal theories of scenarios and rigorous methods of performance assessment for multi-agent systems will require collaboration between computer scientists, philosophers and decision theorists, as well as the domain experts to which these systems are applied. It may be the case that formal theories of scenario analysis may draw on recent work in theoretical computer science attempting to develop a mathematical theory of simulation, such as [11].

6.5.3. Entertainment. Multi-agent systems as social simulations are also of increasing importance in entertainment applications [10]. These applications range from single-(human)-player computer games to multi-player games, where the other players may be both humans and agents. The popularity of social simulation games, such as Maxis, SimCity, for example, where human players construct artificial societies, show the potential for multi-agent simulation applications. Potential applications also exist in other interactive media, such as interactive movies, television and even books, where viewers and readers may have their own avatar participate in the story and may interact with fictional characters directly.

7. Conclusions

In this paper we have sought to provide a manifesto for the progress of research, development and deployment of agent technology, with a view to supporting future computing efforts. Based, in part, on the work undertaken in the AgentLink II thematic network from 2000 to 2003, we have reviewed the current state of activity in the field, and have outlined a path through identifiable problems and challenges yet to be faced. Our timeline considers the next 10 or so years, and is intentionally bold in its projections. We do not claim that these will be achieved, but that these are possible outcomes that we should strive for, and should use to focus our collective efforts as a community. More importantly, in order to be able to face the technical challenges identified, we have argued that several social (or community) challenges must also be met.

Although we believe that the problems and challenges identified here provide a strong and clear set of immediate and longer-term objectives, the analysis itself is limited. We have focused primarily on research and development aspects, and have taken a rather research-influenced approach. By contrast, market and industry trends have been only fleetingly incorporated, and there are likely to be many similar omissions. Rather than invalidate the work here, this suggests instead that there is further work to be done on other aspects of agent technologies, and on updating the analysis as more substantive and broader assessments take place. Indeed, as part of the work of AgentLink III (see www.agentlink.org for details), we aim to address these points, and to develop the roadmap further.

It is clear that there are many opportunities for the application of agent technologies. Indeed, it is perhaps true to say that the value of such technologies has never been greater than at the present moment, with the serendipitous convergence of several fields of research and technological developments. Yet the possibility of failure remains; we see this manifesto as just one of many ways to set out a path to follow that will enable us to avoid that.

Acknowledgments

This paper is based on the AgentLink roadmap for agent technology, a key part of the work of AgentLink II, which was funded by the European Commission (IST-1999-29003). The roadmap was developed over a long period, with contributions from numerous different sources. The effort was led by the authors of this paper, who collated and edited contributions from the community, especially AgentLink SIG coordinators, and wrote the major parts. The community as a whole was involved through the completion of questionnaires, formal reviewing, and informal discussion, and included the following, to whom we are very grateful: Christine Guilfoyle, Sonia Bergamaschi, Paul Davidsson, Frank Dignum, Pete Edwards, Matthias Klusch, Daniel Kudenko, Scott Moss, Paolo Petta, Volker Roth, Carles Sierra, Franco Zambonelli, Liliana Ardissono, Ronald Ashri, Bernhard Bauer, Albert van Breemen, Phil Buckle, Bernard Burg, Monique Calisti, Jonathan Dale, Mark d'Inverno, Jim Doran, Ed Durfee, Bruce Edmonds, Stan Franklin, Matjaz Gams, Francisco Garijo, Marie-Pierre Gleizes, Pierre Glize, Rune Gustavsson, Vincent Hilaire, Nick Jennings, W. Lewis Johnson, Menno Jonkers, Beatriz Lopez, Eduard Lukschandl, Zakaria Maamar, Viviana Mascardi, Philippe Massonet, Stuart Middleton, Jörg Müller, Eric van Nijhuis, Steve Osborn, Agostino Poggi, George Rzevski, Michael Schroeder, Susan Thomas, Philip Turner, Paul Valckenaers, Gary Vickers, Gerhard Weiss, Steve Willmott, Ken Woghiren, Chris Wright, Agentcities and FIPA. Related projects that provided a more general technological context include the EPSRC-funded myGrid (GR/R67743/01) and Comb*e*Chem (GR/R67729/01).

Notes

1. This is a revised version of the AgentLink Roadmap, developed as part of the activities of AgentLink II, the European Commision's Network of Excellence for Agent-Based Computing.
2. E-markets: realism, not pessimism, PriceWaterhouseCoopers, 2001.

References

1. S. Abramsky, "Semantics of interaction: An introduction to game semantics," in A. M. Pitts and P. Dybjer, (eds.), *Semantics and Logics of Computation*, Cambridge University Press: Cambridge, UK, 1997, pp. 1–31.
2. E. Alonso, M. D'Inverno, D. Kudenko, M. Luck, and J. Noble, "Learning in Multi-Agent Systems," *The Knowl. Eng. Rev.*, vol. 16, no. 3, pp. 277–284, 2001.
3. L. Amgoud, N. Maudet, and S. Parsons, "Modelling dialogues using argumentation," in E. Durfee (ed.), *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS 2000)*, IEEE Press: Boston, MA, USA, 2000a, pp. 31–38.
4. L. Amgoud, S. Parsons, and N. Maudet, "Arguments, dialogue, and negotiation," in W. Horn (ed.), *Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI 2000)*, IOS Press: Berlin, Germany, 2000b, pp. 338–342.
5. A. Ankolenkar, M. Burstein, J. R. Hobbs, O. Lassila, D. L. Martin, S. A. M. Drew McDermott, S. Narayanan, M. Paolucci, T. R. Payne, and K. Sycara, 2001, "DAML-S: Web Service Description for the Semantic Web," in I. F. Cruz, S. Decker, J. Euzenat, and D. L. McGuinness, (eds.), *The First Semantic Web Working Symposium*, Stanford University: California, 2001, pp. 411–430.
6. Aristotle, *Topics*, Clarendon Press: Oxford, 1928.
7. K. Arnold, B. O'Sullivan, R. W. Scheifler, J. Waldo, and A. Wollrath, *The Jini Specification*, Addison-Wesley, 1999.
8. R. Ashri, M. Luck, and M. d'Inverno, "Infrastructure support for agent-based development," in M. d'Inverno, M. Luck, M. Fisher, and C. Preist, (eds.), *Foundations and Applications of Multi-Agent Systems*, vol. 2403 of *LNAI*, 2002, pp. 73–88.
9. J. L. Austin, *How To Do Things with Words*. Oxford University Press: Oxford, UK, 1962.
10. R. Aylett and M. Luck, 2000, "Applying artificial intelligence to virtual reality: Intelligent virtual environments," *Appl. Artif. Intell.*, vol. 14, no. 1, pp. 3–32, 1962.
11. C. L. Barrett and C. M. Reidys, "Elements of a theory of computer simulation I: Sequential CA over random graphs," *Appl. Mathe. Comput.*, vol. 98, pp. 241–259, 1999.
12. J. Baumann, H. K. Rothermel, M. Strasser, and W. Theilmann, "MOLE. A mobile agent system," *Software – Prac. Exp.*, vol. 32, no. 6, 575–603, 2002.
13. J. Baxter and R. Hepplewhite, 1999, "Agents in tank battle simulations," *Commun. ACM.,* vol. 42, no. 3, pp. 74–75.
14. A. L. C. Bazzan, J. Wahle, and F. Klügl, "Agents in traffic modelling: From reactive to social behaviour," in *Proceedings of the 23rd Annual German Conference on Artificial Intelligence: Advances in Artificial Intelligence*, Springer-Verlag, 1999, pp. 303–306.
15. N. Boccara, *Modeling Complex Systems*, Graduate Texts in Contemporary Physics, Springer: New York, NY, USA, 2004.
16. M. E. Bratman, *Intention, Plans, and Practical Reason*, Harvard University Press: Cambridge, MA, USA, 1987.
17. R. Brooks, "Intelligence without Representation," *Artif. Intell.*, vol. 47, no. 1–3, pp. 139–159, 1991.
18. K. Bryson, M. Luck, M. Joy, and D. Jones, "Agent interaction for bioinformatics data management," *Appl. Artif. Intell.*, vol. 15, no. 10, pp. 917–947, 2001.
19. N. Chaib-draa and F. Dignum, "Trends in agent communication language," *Comput. Intell.*, vol. 2, no. 5, pp. 89–101, 2002.
20. A. Chavez and P. Maes, "Kasbah: An agent marketplace for buying and selling goods," in *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*. Practical Application Company: London, UK, 1996, pp. 75–90.

21. P. Ciancarini, R. Tolksdorf, and F. Zambonelli, "A survey of coordination-middleware for XML-centric applications," *The Knowl. Eng. Rev.*, vol. 17, no. 4, 2003.
22. W. R. Cockayne, and M. Zyda, *Mobile Agents*, Manning Publications, 1997.
23. D. Cockburn and N. R. Jennings, "ARCHON: A Distributed Artificial Intelligence System for Industrial Applications," in G. M. P. O'Hare and N. R. Jennings, (eds.), *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons, 1996. pp. 319–344.
24. P. R. Cohen, and C. R. Perrault, "Elements of a plan based theory of speech Acts," *Cogn. Sci.* vol. 3, pp. 177–212, 1979.
25. R. Conte and C. Dellarocas, *Social Order in Multiagent Systems*, Kluwer Academic Publishers, 2001.
26. J. Dale and E. Mamdani, "Open standards for interoperating agent-based systems," *Software Focus*, vol. 1, no. 2, 2001.
27. R. K. Dash, D. C. Parkes, and N. R. Jennings, "Computational mechanism design: A call to arms," *IEEE Intell. Syst.*, vol. 18, no. 6, pp. 40–47, 2003.
28. P. Davidsson, "Multi agent based simulation: Beyond social simulation," in S. Moss and P. Davidsson, (eds.), *Multi-Agent-Based Simulation*, vol. 1979 of *Lecture Notes in Computer Science*, Springer, pp. 97–107, 2001.
29. P. Davidsson and M. Boman, "Saving energy and providing value added services in intelligent buildings: A MAS approach," in D. Kotz and F. Mattern, (eds.), *Agent Systems, Mobile Agents, and Applications*, vol. 1882 of *Lecture Notes in Computer Science*, Springer, pp. 166–177, 2002.
30. R. Davis and R. G. Smith, "Negotiation as a metaphor for distributed problem solving," *Artif. Intell.*, vol. 20, no. 1, 1983.
31. K. Decker and V. Lesser 'Designing a family of coordination algorithms,' in *Proceedings of the First International Conference on Multi-Agent Systems*, pp. 73–80, 1995.
32. K. Decker, K. Sycara, and M. Williamson, "Middle-agents for the internet," in *Proceedings of the 15th Joint Conference on Artificial Intelligence*, vol. 1. Morgan Kaufmann, 1997, pp. 578–573.
33. K. Decker, X. Zheng, and C. Schmidt, "A multi-agent system for automated genetic annotation," in *Proceedings of the Fifth International Conference on Autonomous Agents*, 2000, pp. 433–440.
34. M. E. desJardins, E. H. Durfee, and C. L. O. nad Michael J. Wolverton, "A survey of research in distributed, continual planning," *AI Magazine*, vol. 20, no. 4, pp. 13–22, 1999.
35. F. Dignum, B. Dunin-Keplicz, and R. Verbrugge, "Agent theory for team formation by dialogue," in C. Castelfranchi and Y. Lespérance, (eds.), *Intelligent Agents VII: Proceedings of the Seventh International Workshop on Agent Theories, Architectures, and Languages (ATAL 2000)*, Springer: Berlin, Germany, 2000, pp. 150–166.
36. F. Dignum, B. Dunin-Keplicz, and R. Verbrugge, "Creating collective intention through dialogue," *Logic IGPL*, vol. 9, no. 2, pp. 305–319, 2001.
37. M. d'Inverno, D. Kinny, and M. Luck, 1998, "Interaction protocols in agentis," in *Proceedings of the Third International Conference on Multi-Agent Systems*, IEEE Press, 1998, pp. 261–268.
38. M. d'Inverno, and M. Luck, "Understanding autonomous interaction," in W. Wahlster (ed.), *ECAI'96 - Proceedings of the 13th European Conference on Artificial Intelligence*. John Wiley and Sons, Ltd., 1996, pp. 529–533.
39. M. d'Inverno and M. Luck, *Understanding Agent Systems*. Springer, (2nd edn.), 2004.
40. M. d'Inverno, M. Luck, M. Georgeff, D. Kinny, and M. Wooldridge, "The dMARS architecture: A specification of the distributed multi-agent reasoning system," *Auton. Agents Multi-Agent Syst.*, 2004.
41. R. Duboz, èric Ramat, and P. Preux, "Scale transfer modeling: using emergent computation for coupling an ordinary differential equation system with a reactive agent model," *Syst. Anal. Model. Simul.* vol. 43, no. 6, pp. 793–814, 2003.
42. K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.-C. Burgelman, "Scenarios for ambient intelligence in 2010," Technical report, Information Society Technologies, European Commission, 2001.
43. E. H. Durfee and V. R. Lesser, "Partial global planning: A coordination framework for distributed hypothesis formation," *IEEE Trans. Syst. Man, Cyber*, vol. 21, no. 5, pp. 1167–1183, 1991.
44. F. Eemeren, R. Grootendorst, F. S. Henkemans, J. A. Blair, R. H. Johnson, E. C. W. Krabbe, C. Plantin, D. N. Walton, C. A. Willard, J. Woods, and D. Zarefsky, *Fundamentals of Argumentation Theory*, Lawrence Erlbaum Associates: Mahwah, NJ, USA, 1996.

45. U. Endriss and N. Maudet, "Welfare engineering in multiagent systems," in A. Omicini, P. Petta, and J. Pitt, (eds.), *Engineering Societies in the Agents World (ESAW-2003): Post-Proceedings*. Springer: Berlin, Germany, 2004.
46. U. Endriss, N. Maudet, F. Sadri, and F. Toni, "On optimal outcomes of negotiations over resources," in M. W. J. S. Rosenschein, T. Sandholm and M. Yokoo, (eds.), *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*, ACM Press: New York, NY, USA, 2003, pp. 177–184.
47. M. Esteva, J. A. Rodrúguez-Aguilar, C. Sierra, P. Garcia, and J. L. Arcos, "On the formal specifications of electronic institutions," in *Agent Mediated Electronic Commerce, The European AgentLink Perspective*, Berlin, Springer-Verlag: 2001, pp. 126–147.
48. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, *Reasoning about Knowledge*, MIT Press: Cambridge, MA, USA, 1996.
49. T. Finin, Y. Labrou, and J. Mayfield, "KQML as an agent communication language," in J. Bradshaw (ed.), *Software Agents*, MIT Press: Cambridge, MA, USA, 1997, pp. 291–316.
50. FIPA, "Communicative act library specification," Standard SC00037J, Foundation for Intelligent Physical Agents, 2002.
51. I. Foster, and C. Kesselman, (eds.), *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1998.
52. I. Foster, C. Kesselman, and S.Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *Int. J. High Perform. Compu. Appl.* vol. 15, no. 3, pp. 200–222, 2001.
53. J. G. Frey, M. Bradley, J. W. Essex, M. B. Hursthouse, S. M. Lewis, M. M. Luck, L. Moreau, D. C. D. Roure, M. Surridge, and A. Welsh, "Combinatorial chemistry and the Grid," in F. Berman, G. Fox, and T. Hey, (eds.), *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley and Sons: Chichester, England, 2003, pp. 945–962.
54. D. Gambetta, (ed.), *Trust: Making and Breaking Cooperative Relations*, Blackwell: Basil, 1988.
55. E. Goubault, "Geometry and concurrency: a user's guide," *Math. Struct. Comput. Sci.*, vol. 10, no. 4, pp. 411–425, 2000.
56. S. Grand, D. Cliff, and A. Malhotra, "Creatures: Artificial life autonomous software agents for home entertainment," in *Proceedings of the First International Conference on Autonomous Agents*, ACM Press, 1997, pp. 22–29.
57. R. S. Gray, G. Cybenko, D. Kotz, R. A. Peterson, and D. Rus, "D'Agents: Applications and performance of a mobile-agent system," *Software – Prac. Exp.*, vol. 32, no. 6, pp. 543–573, 2002.
58. N. Griffiths, and M. Luck, "Coalition formation through motivation and trust," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM Press: 2003, pp. 17–24.
59. J. Hendler, "Agents and the Semantic Web," *IEEE Intell. Syst.*, vol. 16, no. 2.
60. R. Hill, J. Chen, J. Gratch, P. Rosenbloom, and M. Tambe, "Intelligent agents for the synthetic battlefield," in *Joint proceedings of the Fourteenth National Conference on Artificial Intelligence and the Ninth Conference on Innovative Applications of Artificial Intelligence*, Providence, RI, 1997, pp. 1006–1012.
61. I. Horrocks, "DAML+OIL: A Reason-able Web ontology language," in *Proceedings of the 8th International Conference on Extending Database Technology*. Springer-Verlag, 2002, pp. 2–13.
62. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen, "From SHIQ and RDF to OWL: The making of a web ontology language," *J. of Web Semantics*, vol. 1, no. 1, pp. 7–26, 2003.
63. N. Howden, R. Ronnquist, A. Hodgson, and A. Lucas, "JACK Intelligent Agents: Summary of an agent infrastructure," in T. Wagner and O. Rana (eds.), *The 5th International Conference on Autonomous Agents, Workshop on Infrastructure for Agents, MAS and Scalable MAS*, 2001, pp. 251–257.
64. M. N. Huhns, L. M. Stephens, and N. Ivezic, "Automating supply-chain management," in *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, ACM Press: 2002, pp. 1017–1024.
65. N. R. Jennings, "An agent-based approach for building complex software systems," *Communi. ACM*, vol. 44, no. 4, pp. 35–41, 2000.

66. N. Jennings, P. Faratin, A. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge, "Automated negotiation: Prospects, methods and challenges," *Group Decis. Negot*. vol. 10, no. 2, 2001.
67. F. V. Jensen, *Introduction to Bayesian Networks*, Springer: New York, NY, USA, 1996.
68. D. Johansen, K. J. Lauvset, R. van Renesse, F. B. Schneider, N. P. Sudmann, and K. Jacobsen, "A TACOMA retrospective," *Software – Prac. Exp*., vol. 32, no. 6, pp. 605–619, 2002.
69. S. Khan, R. Makkena, F. McGeary, K. Decker, W. Gillis, and C. Schmidt, "A multi-agent system for the quantitative simulation of biological networks," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM Press, 2003, pp. 385–392.
70. M. Klusch, "Information agent technology for the internet: A survey," *Data Knowl. Eng*., vol. 36, no. 3, 2001.
71. M. Klusch and K. Sycara, "Brokering and matchmaking for coordination of agent societies: A Survey," in A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, (eds.), *Coordination of Internet Agents: Models, Technologies, and Applications*, Springer:Verlag, 2001, pp. 197–224.
72. Y. Labrou, T. Finin, and Y. Peng, "The current landscape of Agent Communication Languages," *IEEE Intell. Syst*., vol. 14, no. 2, 1999.
73. D. B. Lange and M. Oshima, *Programming and Deploying Java Mobile Agents Aglets*, Addison-Wesley Longman Publishing Co. Inc, 1998.
74. G. L. Lilien, P. Kotler, and K. S. Moorthy, *Marketing Models*, Prentice-Hall: Englewood Cliffs, NJ, USA, 1992.
75. F. Lopez y Lopez, M. Luck, and M. d'Inverno, "Constraining autonomy through Norms," in *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, ACM Press, 2002, pp. 674–681.
76. C. K. Low, T. Y. Chen, and R. Rœnnquist, "Automated test case generation for BDI agents," *Auton. Agents Multi-Agent Syst*., vol. 2, no. 4, pp. 311–332, 1999.
77. M. Luck, "From definition to deployment: What next for agent-based systems?," *Knowl. Eng. Rev*., vol. 14, no. 2, 1990.
78. M. Luck, R. Ashri, and M. d'Inverno, *Agent-Based Software Development*, Artech House: Norwood, MA, 2004.
79. M. Luck and M. d'Inverno, "A formal framework for agency and autonomy," in *Proceedings of the First International Conference on Multi-Agent Systems*, AAAI Press/MIT Press, 1995, pp. 254–260.
80. M. Luck, S. Munroe, and M. d'Inverno, "Autonomy: Variable and generative," in H. Hexmoor, C. Castelfranchi, and R. Falcone, (eds.), *Agent Autonomy*, Kluwer, 2003, pp. 9–22.
81. P. M. Markopoulos, and J. O. Kephart, "How valuable are shopbots?," in *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM Press, 2002, pp. 1009–1016.
82. P. McBurney, R. Eijk, S. Parsons, and L. Amgoud, "A dialogue-game protocol for agent purchase negotiations," *J. Auton. Agents Multi-Agent Syst*., vol. 7, no. 3, pp. 235–273, 2003.
83. P. McBurney and S. Parsons, "Formalizing scenario analysis," in A. Darwiche and N. Friedman, (eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Eighteenth Conference (UAI-2002)*, Morgan Kaufmann: San Francisco, CA, USA, 2002, pp. 327–334.
84. P. McBurney and S. Parsons, "Dialogue game protocols," in M.-P. Huget (ed.), *Communication in Multi-Agent Systems: Agent Communication Languages and Conversation Policies*, Lecture Notes in Artificial Intelligence 2650. Springer: Berlin, Germany, 2003, pp. 269–283.
85. P. McBurney and S. Parsons, "Engineering democracy in open agent systems," in A. Omicini, P. Petta, and J. Pitt, (eds.), *Engineering Societies in the Agents World (ESAW-2003): Post-Proceedings*, Springer: Berlin, Germany, 2004.
86. P. McNamara and H. Prakken, (eds.), *Norms, Logics and Information Systems: New Studies in Deontic Logic and Computer Science*, IOS Press: Amsterdam, The Netherlands, 1998.
87. L. Moreau, S. Miles, C. Goble, M. Greenwood, V. Dialani, M. Addis, N. Alpdemir, R. Cawley, D. D. Roure, J. Ferris, R. Gaizauskas, K. Glover, C. Greenhalgh, P. Li, X. Liu, P. Lord, M. Luck, D. Marvin, T. Oinn, N. Paton, S. Pettifer, M. V. Radenkovic, A. Roberts, A. Robinson, T. Rodden, M. Senger, N. Sharman, R. Stevens, B. Warboys, A. Wipat, and C. Wroe, "On the use of agents in a bioInformatics grid," in S. Lee, S. Sekguchi, S. Matsuoka, and M. Sato, (eds.), in *Proceedings of the*

Third IEEE/ACM CCGRID'2003 Workshop on Agent Based Cluster and Grid Computing. Tokyo, Japan, 2003, pp. 653–661.

88. J. Nicolaisen, V. Petrov, and L. Tesfatsion, "Market power and efficiency in a computational electricity market with discriminatory double-auction pricing," *IEEE Trans. on Evol. Comput*., vol. 5, no. 5, pp. 504–523, 2001.
89. T. J. Norman, A. Preece, S. Chalmers, N. R. Jennings, M. M. Luck, V. D. Dang, T. D. Nguyen, V. Deora, J. Shao, W. A. Gray, and N. J. Fiddian, "CONOISE: Agent-based formation of virtual organisations," in *Research and Development in Intelligent Systems XX: Proceedings of AI2003, the Twenty third International Conference on Innovative Techniques and Applications of Artificial Intelligence*, 2003, pp. 353–366.
90. S. Oaks, B. Traversat, and L. Gong, *JXTA In a Nutshell*. O'Reilly and Associates, 2002.
91. J. Odell, H. Van Dyke Parunak, and B. Bauer, "Representing agent interaction protocols in UML," in P. Ciancarini and M. J. Wooldridge, (eds.), *First International Workshop, AOSE 2000 on Agent-oriented Software Engineering*, Springer-Verlag, 2001, pp. 121–140.
92. M. J. Osborne, and A. Rubinstein, *A Course in Game Theory*, MIT Press: Cambridge, MA, USA, 1994.
93. M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara, "Semantic matchmaking of web services capabilities," in I. Horrocks and J. A. Hendler, (eds.), *International Semantic Web Conference*, vol. 2342 of *LNCS*, Springer, 2002, pp. 333–347.
94. R. Parikh, "The logic of games and its applications," *Ann. Discrete Math*., vol. 24, pp. 111–140, 1985.
95. S. Parsons, and P. McBurney, "Argumentation-based communication between agents," in M.-P. Huget (ed.), *Communication in Multi-Agent Systems: Agent Communication Languages and Conversation Policies*, Lecture Notes in Artificial Intelligence 2650, Springer: Berlin, Germany, 2003, pp. 164–178.
96. H. V. D. Parunak, "A practitioners review of industrial agent applications," *Auton. Agents Multi-Agent Syst*., vol. 3, no. 4, pp. 389–407, 2000.
97. M. Pauly, "Logic for social software," Ph.D. thesis, Centrum voor Wiskunde en Informatica, Universiteit van Amsterdam: Amsterdam, The Netherlands (Institute for Logic, Language and Computation Dissertation Series DS-2001-10), 2001.
98. T. R. Payne, R. Singh, and K. P. Sycara, "Importing the semantic web in UDDI," in C. Bussler, R. Hull, S. A. McIlraith, M. E. Orlowska, B. Pernici, and J. Yang, (eds.), *Web Services, E-Business, and the Semantic Web, CAiSE 2002 International Workshop*, vol. 2512 of *LNCS*, Springer, 2002, pp. 225–236.
99. S. Phelps, P. McBurney, S. Parsons, and E. Sklar, "Co-evolutionary auction mechanism design: A preliminary report," in J. Padget, D. C. Parkes, N. M. Sadeh, O. Shehory, and W. E. Walsh, (eds.), *Agent Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, Springer: Berlin, Germany, 2003, pp. 123–142.
100. J. L. Pollock, *Cognitive Carpentry: A Blueprint for How to Build a Person*, MIT Press: Cambridge, MA, USA, 1995.
101. C. Preist, C. Bartolini, and A. Byde, "Agent-based service composition through simultaneous negotiation in forward and reverse auctions," in *Proceedings of the 4th ACM Conference on Electronic Commerce*. ACM Press, 2003, pp. 55–63.
102. I. Rahwan, P. McBurney, and E. Sonenberg, "Towards a theory of negotiation strategy (a preliminary report)," in S. Parsons and P. Gmytrasiewicz, (eds.), *Fifth Workshop on Decision-Theoretic and Game-Theoretic Agents*, AAMAS: Melbourne, Australia, 2003, pp. 73–80.
103. I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and E. Sonenberg, "Argumentation-based negotiation," *Knowl. Eng. Rev. (In press)*, 2004.
104. A. S. Rao and M. P. Georgeff, "Modeling rational agents within a BDI-Architecture," in J. Allen, R. Fikes, and E. Sandewall, (eds.) *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning KR91*, Morgan Kaufmann: San Mateo, CA, USA, 1991.
105. S. Renooij, "Probability elicitation for belief networks: issues to consider," *Knowl. Eng. Rev*., vol. 16, no. 3, pp. 255–269, 2001.

106. J. Rickel and W. L. Johnson, "Integrating pedagogical capabilities in a virtual environment agent," in *Proceedings of the First International Conference on Autonomous Agents*, ACM Press, 1997, pp. 30–38.
107. A. E. Roth, "The economist as engineer: game theory, experimentation, and computation as tools for design economics," *Econometrica*, vol. 70, no. 4, pp. 1341–1378, 2002.
108. J. Sabater, and C. Sierra, "Reputation and social network analysis in multi-agent systems," in *Proceedings of the First International Joint conference on Autonomous Agents and Multiagent Systems*. ACM Press, 2002, pp. 475–482.
109. T. W. Sandholm, "Distributed rational decision making," in G. Weiss, (ed.), *Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence*, MIT Press: Cambridge, MA, USA, 1999, pp. 201–258.
110. J. Searle, *Speech Acts: An Essay in the Philosophy of Language*, Cambridge University Press: Cambridge, UK, 1969.
111. O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artif. Intell.*, vol. 101, no. 1–2, pp. 165–200, 1998.
112. C. Sierra and F. Dignum, "Agent-mediated electronic commerce: Scientific and technological roadmap," in F. Dignum and C. Sierra, (eds.), *Agent Mediated Electronic Commerce: The European AgentLink Perspective*, vol. 1991 of *Lecture Notes in Computer Science*, Springer, 2001, pp. 1–18.
113. M. P. Singh, "Agent communication languages: Rethinking the principles," *IEEE Comput.*, vol. 31, no. 12, pp. 40–47, 1998.
114. R. G. Smith and R. Davis, "Frameworks for cooperation in distributed problem solving," *IEEE Trans. Syst. Man, Cyber.*, vol. 11, no. 1, 1980.
115. N. Streitz, "Interaction design for the disappearing computer," in E. H. L. Aarts, R. Collier, E. van Loenen, and B. E. R. de Ruyter, (eds.), *Ambient Intelligence*, Vol. 2875 of *Lecture Notes in Computer Science*, Springer, 2003, pp. 351–355.
116. K. Sycara, M. Paolucci, M. van Velsen, and J. Giampapa, "The RETSINA MAS Infrastructure," *Autonomous Agents and Multi-Agent Systems*, vol. 7, no. 1–2, pp. 29–48, 2003.
117. L. Tesfatsion, "Agent-based computational economics: Growing economies from the bottom up," *Artif. Life*, vol. 8, no. 1, pp. 55–82, 2002.
118. F. Tohme and T. Sandholm, "Coalition formation processes with belief revision among bounded-rational self-interested agents," *J. Logic Comput.*, vol. 9, no. 6, pp. 793–815, 1999.
119. D. Trastour, C. Bartolini, and C. Preist, "Semantic web support for the business-to-business e-commerce pre-contractual lifecycle," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 42, no. 5, pp. 661–673, 2003.
120. C. F. Tschudin, "Mobile agent security," in M. Klusch, (ed.), *Intell. Inform. Agents*, Springer, 1999. pp. 431–446.
121. K. Tuyls, "Learning in multi-agent systems: An evolutionary game theoretic approach," Phd thesis, Vrije Universiteit, Brussels, Belgium, 2004.
122. W. W. Vasconcelos, J. Sabater, C. Sierra, and J. Querol, "Skeleton-based agent development for electronic institutions," in *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM Press, 2002, pp. 696–703.
123. G. Vigna, "Cryptographic traces for mobile agents," in G. Vigna, (ed.), *Mobile Agents and Security*, vol. 1419 of *LNCS*, Springer-Verlag, 1998, pp. 137–153.
124. G. Weiss, "Agent orientation in software engineering," *Knowl. Eng. Rev.*, vol. 16, no. 4, pp. 349–373, 2002.
125. M. P. Wellman, A. R. Greenwald, P. Stone, and P. R. Wurman, "The 2001 Trading Agent Competition," *Electron. Markets*, vol. 13, no. 1, 2002.
126. J. E. White, "Mobile Agents," in J. M. Bradshaw (ed.), *Software Agents*, AAAI Press / The MIT Press, 1997, pp. 437–472.
127. S. Willmott, U. Cortes, and D. Cabanillas, "Standards for Agent Development," in M. Luck, R. Ashri, and M. d'Inverno, (eds.), *Agent-Based Software Development*, Artech House: Norwood, MA, 2004.
128. M. J. Wooldridge, "The logical modelling of computational multi-agent systems," Ph.D. thesis, University of Manchester, Manchester UK, 1992.

129. M. Wooldridge and P. Ciancarini, "Agent-Oriented software engineering: The state of the art," in P. Ciancarini and M. Wooldridge, (eds.), *Agent-Oriented Software Engineering*, vol. 1957 of *LNCS*, Springer, 2001, pp. 1–28.
130. M. Wooldridge, N. R. Jennings, and D. Kinny, "The gaia methodology for agent-oriented analysis and design," *Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 3, pp. 285–312, 2000.
131. F. Zambonelli and H. Parunak, "Signs of a revolution in computer science and software engineering," in P. Petta, R. Tolksdorf, and F. Zambonelli, (eds.), *Engineering Societies in the Agents World (ESAW 2002)*, Springer: Berlin, Germany, 2003, pp. 13–28.