

A perspective on software agents research

HYACINTH S. NWANA and DIVINE T. NDUMU

Applied Research & Technology Department, British Telecommunications Laboratories, Martlesham Heath, Ipswich, Suffolk IP5 3RE, UK (email:{hyacinth, ndumudt}@info.bt.co.uk)

Abstract

This paper sets out, ambitiously, to present a brief reappraisal of software agents research. Evidently, software agent technology has promised much. However, some five years after the word ‘agent’ came into vogue in the popular computing press, it is perhaps time the efforts in this fledgling area are thoroughly evaluated with a view to refocusing future efforts. We do not pretend to have done this in this paper—but we hope we have sown the first seeds towards a thorough first five-year report on the software agents area. The paper contains some strong views not necessarily widely accepted by the agent community.¹

1 Introduction

The main goal of this paper is to provide a brief perspective on the progress of software agents research. Though ‘agents’ research had been going on for more than fifteen years before, agents really became a buzzword in the popular computing press (and also within the artificial intelligence and computing communities) around 1994. During this year several key agent-related publications appeared. An Ovum² report entitled *Intelligent agents: the new revolution in software* (Ovum, 1994) wildly speculated on market sector totals for agent software and products by the year 2000. The first of now several special issues of the *Communications of the ACM* on agents appeared in 1994 and it included articles like Maes’ now-classic paper on “Agents that reduce work and information overload” (Maes, 1994) and Norman’s conjectures on “How might people interact with software agents” (Norman, 1994). Then there was Reinhardt’s *Byte* article on “The network with smarts” (Reinhardt, 1994). Indeed, during late 1994 and throughout 1995 and 1996, we saw an explosion of agent-related articles in the popular computing press. It is no coincidence that this explosion coincided with that of the World Wide Web (WWW).

The field has clearly matured since the publication of certain key papers and books including Wooldridge and Jennings (1995), Nwana (1996), Bradshaw (1997), Jennings and Wooldridge (1998) and Nwana and Azarmi (1997), amongst many others. Several annual and biennial conferences now grace the area, including the *International Conference on Multi-Agent Systems* (ICMAS), the *International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology* (PAAM), and the *International Conference on Autonomous Agents* (AA). ICMAS, PAAM and AA held their first conferences in 1995, 1996 and 1997, respectively. These conferences and numerous other agent-oriented national and international workshops, many “agent” special issues of learned journals, agent books, agent standardisation initiatives such as FIPA,³ the newly established *Journal of Autonomous and Multi-Agent Systems* (JAMAS), Huhns

¹ The views expressed in this paper are those of the authors only, and do not necessarily reflect the views of our organisation.

² Ovum is a respected UK-based market research company.

³ <http://drgo.cselt.it/fipa/>

and Singh's (1998) *Readings in Agents*, initiatives such as the AgentLink⁴ Network of Excellence, etc. all bear testimony to a quickly maturing area. But how much real progress has indeed been made since 1994 towards the goals agent technology promised? This is the subject of this brief paper.

For the purposes of this paper, we break the software agents domain into two: multi-agent systems and autonomous interface/information agents, just as the AA and the PAAM conferences do.

The breakdown of the paper is thus as follows. Section 2 revisits some of the promises of multi-agent technology and briefly evaluates some of the progress that has been made in this sub-area over the last five years. In doing so, we critique generally some of the current MAS work. Section 3 does likewise for personal and information agents which in this paper also subsumes mobile agents. Section 4 spells out some further general viewpoints on the future of the software agents work. Section 5 concludes this paper.

2 Multi-agent systems: promises and reality

The hypothesis/goal of Multi-Agent Systems (MAS) is clear enough and has been proven in many multi-agent prototypes across the globe: creating a system that interconnects separately developed agents, thus enabling the ensemble to function beyond the capabilities of any singular agent in the set-up. Much important ground in such systems had been covered well before 1994, and it is truly a matter of debate how much *real* progress has been made since. Of course, there has been much consolidation of pre-1994 work. However, let us revisit the promises more closely whilst simultaneously interjecting with some of the reality.

Let us ground the multi-agent systems hypothesis in a real application which many can relate to, and which we can also reuse later on in the paper. Consider the case of a traveller wishing to arrange a trip from a town outside London to a city on the East Coast of the USA. This is the sort of activity—surely—that agent researchers promise people will be able to delegate to their personal travel software agents, in the same vein as we do to our human secretaries. Today, it is still largely the case that the secretary would consult other human travel agents, who in turn contact yet others to arrange our flight and itinerary. These others include hotel agents, railway agents, rental car agents, etc. However with much of the information now being found online, but being owned by different stakeholders who all want to make money from the information and service, the *promise* is that our Personal Travel Agent (PTA) will negotiate with other software agents representing the interests of the different stakeholders. This way, our itinerary gets generated with minimal or possibly without any human intervention, unless changes to the itinerary are required. This is truly a multi-agent problem wherein the inter-operation of separately developed and self-interested agents provide a service beyond the capability of any agent in the set up, and in the process all or most gain financially. Economically speaking, all these agents have comparative advantages over each other due to specialisation, and trading their services is good for all. This lofty goal generates some critical challenges that the hype of software agents and many papers in the literature cast aside. We describe some of the main problems next, and comment on how far the challenges they pose have been realised.

2.1 The information discovery problem, current reality and challenges

In the travel scenario outlined, the PTA must obtain travel information such as timetables and seat availability information from the databases of appropriate travel companies. Therefore, it needs

⁴AgentLink is Europe's Network of Excellence for agent-based computing. See <http://www.agentlink.org>

some means of discovering what relevant resources exist and where they can be found. In other words, it needs the Internet equivalent of Yell!⁵ or Yahoo!⁶ indices that provide “yellow pages” type service listings, rather than web-site directories. The reality is, as usual, different. Given the frequency that Internet addresses change, it is wise to avoid hard-coding the addresses of resources into agents. Hence agent systems solve the information discovery problem by delegating responsibility for maintaining indices of available resources to specialised types of agent that serve whole agent communities. To find a particular resource an agent typically asks a *Facilitator*; which is a “yellow pages” service like the “Yahoo!” directory. Once it has identified an information provider, an agent obtains its network location by querying an *Agent Name Server*. This provides a “white pages” service, mapping machine names to their physical location on a network just as Internet Domain Name Servers translate Internet addresses to Internet Protocol (IP) numbers. The key issue of note here is that there is much unavoidable *drudgery* involved in solving the information discovery problem today. These indices all have to be maintained and updated. On the Internet the problem is ameliorated with tools such as Yahoo. If one had to develop a multi-agent application for business process engineering, much of the drudgery of tackling the information discovery problem will have to be done by the agent designer “from scratch” and manually too. How do agents learn for themselves to discover new information resources in open environments like the Internet? How do they decide which sources are no longer reliable or up-to-date? How would the indices be kept current without having them designed *a priori*? Can some agents in the society spot opportunities and decide to specialise on servicing other agents’ information needs—in effect, new facilitators or information brokers emerge without being designed *a priori* by agent designers? Our point is simply this—much of the information discovery problem is currently tackled manually—and as long as this persists, it serves as a limiting factor to deploying multi-agent systems for real since they will essentially remain closed systems. As a human owner of your software PTA, you would expect it to learn and discover other cheaper and more reliable information resources, and not only go back to those few agents it knows who may be out of date and unreliable. We have seen precious little work addressing the general form of the information discovery problem within the context of a truly open environment.

2.2 The communication problem, current reality and challenges

Suppose a solution to the information discovery problem exists, and the PTA is able to locate a flight reservation service on the Internet. To exploit this resource, the PTA must communicate with the reservation system; this raises three challenges. First, both parties must adopt a common transport protocol that enables the request and results to be transmitted, as shown by the pipe in Figure 1. Fortunately, solutions to this problem abound, e.g., the TCP/IP protocol (Transmission Control Protocol/Internet Protocol), which facilitates the transfer of information across the Internet. The next challenge is to devise a common communication language that both parties understand. Currently, the lingua franca of the Internet is the Hypertext Mark-up Language (HTML), whose standard format enables browser software to interpret pages on the World Wide Web. However, HTML web pages were designed for presentation of information for human perusal, and so valuable information is typically intermingled with formatting instructions—necessitating “wrapper induction” software programs (Kushmerick, 1997), to parse the pages and extract the information from the formatting instructions. Clearly, in our travel scenario, a language geared towards direct machine-to-machine communication is preferable. With the transport and structure of messages agreed, the final challenge is for the PTA and flight reservation system to agree on the list of terms to be used in the context of messages, and the meanings of these terms. This obstacle is known as the ontology problem, and is perhaps the main problem preventing widespread

⁵<http://www.yell.co.uk>

⁶<http://www.yahoo.co.uk>

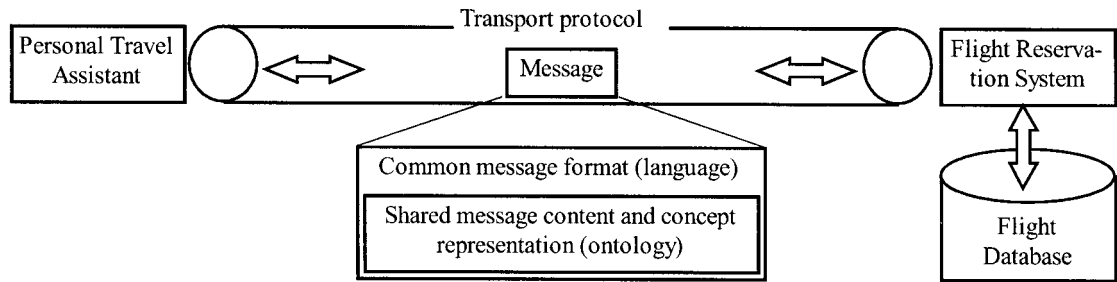


Figure 1 The three prerequisites for effective communication. For the PTA to make a flight reservation there must be some means of transmitting the request to the flight reservation system. For both parties to understand one another, their messages must be in a common language that is grounded in a shared ontology. Once a message is received, the reservation system will use its knowledge of the ontology to translate the request into the terms used within the flight database

inter-operability of heterogeneous, distributed cooperating systems;⁷ we expand on this problem next. The reality here is that there are genuine efforts to tackle the communication problem. Whilst protocols like TCP/IP can facilitate information transfer between machines, no standard language exists to express the structure and content of messages. This problem requires the communicating parties to agree on what instructions, assertions, requests, etc. will be supported, and what syntax is used. A number of inter-agent communication languages have been proposed, notably KQML (Knowledge Query and Manipulation Language) (Finin & Labrou, 1997) and FIPA ACL (Agent Communication Language). These languages are based on speech act theory performatives (Searle, 1969), wherein the speaker's intent of the effects of a message on the hearer is communicated by specifying the type of the message, e.g., *ask*, *tell* or *achieve* message types. Furthermore, most of the languages do not specify a syntax or semantics of the contents of the messages, with the rationale being that different application domains may require different content languages. Nonetheless, a number of general-purpose content languages have been developed, e.g., KIF (Knowledge Interchange Format) (Genesereth & Fikes, 1994), typically used with KQML, and FIPA SL the preferred content language for use with the FIPA ACL.

Despite these efforts, we have some concerns. The agent community in our view either seems not to be taking this problem too seriously (judging from the general apathy we observe towards initiatives like FIPA) or they believe FIPA and other bodies will resolve such issues for them. Perhaps some researchers believe this is a rather boring issue which they can assume in their demonstrators that it will be solved or that it will just go away—it will not. Communication is at the heart of cooperation and competition germane to multi-agent systems. The problem of standardising knowledge-based issues is very dissimilar from standardising protocols in the 7-layer OSI model say—and very non-trivial too. We are involved in FIPA, and despite some heroic efforts on its part, the FIPA ACL would end up suffering from the same problems that befell KQML. It is still ambiguous and vague. KQML also has had mis-identified or missing performatives which FIPA ACL is trying to address. However, this culminates in a situation where we still do not know when to use ACL's myriad of performatives—we frequently get away in our demonstrators with just using a few. However, the provision of a precise, formal semantics is not the answer either. The average agent designer would not have to become an expert in modal logics before being expected to use the FIPA ACL. So one is left with the troubling question of how one tests for FIPA conformance? What hope is there really of achieving agent inter-operability? Without addressing this problem, rolling out multi-agent applications *en masse* is out of the question. We touch on this issue several times

⁷ For example, progress towards large scale electronic commerce is plagued by the lack of inter-operable product catalogues (for a discussion of this topic see "Catalogs for the Digital Marketplace". Research Report No. 97-03, CommerceNet, 1997. Available at <http://www.commerce.net/research/research.html>).

later in this paper. However, we end this section with an open question—do we really need performative-based ACL or can we do better without one, e.g., by standardising collaboration protocols only?

2.3 The ontology problem and the current sad reality

The ontology problem can be illustrated by returning to the scenario depicted in Figure 1. For the PTA and flight reservation system to interact, both systems must agree on and then share common definition of travel-related concepts. The ontology (or concept definitions) specifies the terms each party must understand and use during communication, e.g., definition of a journey leg, identities of airlines, types of fares, etc. Creating an ontology involves explicitly defining every concept to be represented; take journey legs for example, what is the attribute of a leg, and what does each attribute mean pragmatically? We must also ask what are the constraints on valid attribute values, and how are the attributes of one concept related to those of another? Whilst some general-purpose ontologies have been proposed for use in inter-agent communication, like Cyc (Lenat, 1994), the current trend is towards the provision of editors for creating domain-specific ontologies and converters for translating between ontologies (e.g. see Gruber (1993)). The argument being that, firstly, most general-purpose ontologies are unlikely to include the intricacies of all possible domains; and secondly that they are likely to be bloated and unnecessarily complex for most applications.

However, in our view, current reality and research truly fails to appreciate the magnitude of this problem. The ontology issue has always been considered secondary to other issues such as cooperation, negotiation, formalisation and logics for beliefs, desires and intentions, etc. You will find few papers and research projects truly addressing this issue. Perhaps this is a case of sheer escapism. Whither the ontology problem? Not a chance! This problem is at the core of the agent inter-operability issue—is it realistic to expect knowledge and cooperation level inter-operability without a significant degree of ontological sophistication of the agents concerned? We are now convinced that FIPA ACL and all other such attempts are, at worst, bound to fail or, at best, bound to only enjoy limited and short-lived success due to our skirting round the ontology issue. A problem stems from what is sometimes referred to as the *interaction problem* between domain and task. The issue is, when designing a multi-agent system, in order to define the *domain ontology*, you need to know the purpose or *task* for which it will be used. Hence, to plan a trip, I need to be aware of ontology concepts such as planes, flights, airports that may be different in other tasks. Worse still, even communicating or requesting the help of another agent requires some clear bilateral understanding of the context or task, so that the right ontology gets used. If an agent does not recognise that my PTA is trying to plan a trip for me, how would it know how to interpret “first leg”, say? “First leg” has a clear meaning within the travel context and may mean something else in another context. In short, there is much unavoidable interaction between the communication layer, the task layer and the content (or domain) layer during agent-to-agent communication. In KQML or FIPA ACL, we presume a no interaction hypothesis between these three layers—but this is patently unrealistic. Hence, again we ask—what prospects are there for true agent inter-operability without domain ontology standardisation? Is FIPA skirting round the standardisation issue by not tackling the real thorny ontology problem first and squarely? To be fair, FIPA is increasingly becoming aware of the ontology problem, but perhaps not its magnitude. Perhaps, as we suspect, there is no answer to this problem in the short term necessitating the defining or limited domain ontologies for use within limited tasks within limited contexts. Either way, the dearth of papers tackling the ontology problem is quite remarkable. Perhaps we should expend efforts building special purpose standard ontologies for particular contexts and tasks. However, it should be obvious from the discussion so far that our hope to achieve open multi-agent applications (on the WWW say) without having addressed the ontology problem is rather naïve.

2.4 *The legacy software integration problem, current reality and challenges*

Legacy systems are large pieces of software, based on older technologies, and generally not designed for inter-operability with other systems. The flight database of Figure 1 is an example of a legacy system. The legacy system problem involves devising some mechanism that enables legacy systems to communicate with external systems like the PTA. As Figure 1 suggests, the solution is likely to involve a proxy, e.g., the flight reservation system, capable of translating requests made using the shared ontology into queries in the internal language of the legacy system, and then translating the results received from the legacy system back into the shared ontology.

Genesereth and Ketchpel (1994) discuss the problem of integrating legacy software with agent systems, and suggest three possible solutions to the problem. First, rewriting the software—a costly approach. Secondly, through use of a separate piece of software called a *transducer* that acts as an interpreter between the agent communication language and the native protocol of the legacy system. Lastly, using the *wrapper* technique where the legacy program is augmented with code that enables it to communicate using the inter-agent language. We know of no other approaches but these to solve this problem. Our point here is that this relates squarely to the ontology problem, and there is much drudgery to address the legacy software integration problem. There are only a few efforts around the globe attempting to automate this process, i.e., that of facilitating the reverse engineering of legacy systems to throw up the ontology, tasks and wrappers required of agent applications. Clearly, with the millions of legacy systems out there, this is a pressing concern.

2.5 *The reasoning and coordination problem and current reality*

Now suppose that solutions to the aforementioned problems exist, and the PTA is able to successfully retrieve timetable information from various travel companies. Now the PTA must plan a journey, which is likely to consist of several legs, from the traveller's home to their desired destination. This process necessitates reasoning, planning and constraint satisfaction capabilities in order to deal with timing and other dependencies in the travel itinerary while attempting to satisfy the traveller's preferences. In essence, the PTA needs to coordinate the services of the different service providers so that as a whole, the service providers behave coherently in their attempt to provide an integrated service. For instance, the PTA should be able to reason that the flight and hotel accommodation should be booked prior to considering local transportation, and ensure that a taxi is booked to coincide with the arrival of the train.

Again researchers have had some considerable successes tackling the coordination and reasoning problem. The reasoning problem is handled in most multi-agent systems using AI techniques such as rule-based inference, classical planning, various logic formalisms, and constraint satisfaction techniques. Coordination to ensure coherent functioning of the multi-agent society is also handled with a variety of approaches including organisational structuring, contracting, multi-agent planning, negotiation and coordination. Nwana et al. (1996) presents a gentle introduction to the literature on coordination of multi-agent systems. We believe the challenges here are being adequately tackled, not least because we are borrowing from and consolidating established AI work. However, perhaps it still makes more sense to carry out such “borrowing and consolidation” within the context of real or quasi-real world problems.

2.6 *The monitoring problem*

The problems of information discovery, communication, ontology and planning manifest themselves again during post-plan monitoring. To react proactively to external events that might jeopardise successful execution of a planned trip, the PTA needs to discover relevant news sources, and periodically monitor them for news on events that might affect the trip. Next, it needs to evaluate whether and how any such events might jeopardise the trip, and then determine corrective actions (which might involve re-planning) to counter their effects. The Internet now has many such

information sources, like weather reports,⁸ reports on road traffic conditions⁹ and news on airport strikes.¹⁰ As such news sources present information in natural language, the key challenge is to build ontologies rich enough to represent the underlying concepts involved, so allowing agents to reason with the information obtained.

Once again, because of the dependence of the monitoring problem on the problems of information discovery, communication, ontology, reasoning, and even the legacy software problem, no general-purpose *total* solution can be proposed to this problem. Thus, it is typically resolved in an application-specific manner using a combination of techniques. In general, however, solutions are based on software sensor mechanisms that react to changes in the sensed environment and notify the monitoring agent of these changes.

2.7 *What conclusions do we draw from the above problems and the reality of the state-of-the-art?*

Our views on this question probably make for controversial reading, but we believe they are both frank and real. We must point out that they also reflect work that we have carried out ourselves, much of it unpublished. Seen from the viewpoints of the sorts of problems above which truly plague the deployment of multi-agent systems, we are forced to truly question what progress we have made beyond 1994. Many of the solutions we employ for the reasoning and coordination problem were known pre-1994 but many papers in the agents literature report on this at the expense of the other problems. The ontology problem has been largely sidelined and so has the information discovery problem. FIPA is making some brave but questionable efforts at addressing the communication problem, and the monitoring problem is quite idiosyncratic to the task anyway. Few real efforts are really facilitating the legacy software integration problem.

All in all, our assessment of progress in multi-agent systems since the agents craze is a rather gloomy one. It is no wonder that there are hardly any deployed multi-agent systems to date beyond such work as that carried out in the ARCHON project (Wittig, 1992) on transforming standalone expert systems into a community of cooperating agents (Jennings et al., 1993). There was one key ingredient in the apparent success of this project—there existed a real world problem to be solved. Likewise, the OASIS agent-based air-traffic control system that was being field tested at Sydney airport (Rao & Georgeff, 1995) also set about to address a real need. Oasis has since been dropped due to legacy and political problems.

We therefore sadly conclude that in place of real progress in multi-agent systems, there has been much re-invention of wheels, and worse still too much “re-labelling” of old concepts. There is a vast amount of extremely relevant and first-class literature out there published in the late 1970s and 1980s, which MAS researchers can just draw upon without having to re-invent them all over again. We believe that addressing such multi-agent problems as those above will provide for real progress. Instead, we see many papers and projects that clearly in our view fall in certain key respects such as the following.

Many papers still appear to confuse the following: agents and objects; distributed computing and agent-based computing; object-oriented systems, expert systems and agent-based systems. Object-oriented computing, distributed computing, etc. do not in themselves offer solutions to multi-agent problems. Why? Because distributed computing modules are usually passive and dumb. Also, their communications are usually low-level while Multi-Agent Systems (MAS) require high-level messages. Lastly, and importantly, MAS applications require a cooperation-knowledge level while these systems (as OO, expert systems, etc.) typically operate at the symbol and knowledge levels (at most) (Newell, 1982). In many papers we have seen, it appears some researchers choose to introduce a cooperation knowledge level so as to arrive at a multi-agent solution where it is truly arguable that

⁸ <http://www.meto.govt.uk/sec3/sec3.html>

⁹ <http://www.rac.co.uk/html/jou/fjourney2.htm>

¹⁰ News channel.

it is warranted. Indeed, if the cooperation knowledge level buys you nothing for the application, avoid it. For example, Huberman's thermal economy MAS has been criticised that it provides an inferior solution to a single "agent" control solution. Arguably, a distributed problem solving solution is sufficient here. Had the problem not been totally "owned" by one owner, then an inferior MAS solution is arguably justifiable.

We also point out that many efforts attempt to provide multi-agent solutions to the wrong problems. The following in our view does *not* necessitate the employment of multi-agent systems. It is important that these are pointed out, not least because some papers appear in the proceedings of agent conferences and workshops which suffer from these criticisms. Multi-agent systems are *not* really needed:

- where multi-agent system solutions are only used to enhance modularity (which reduces complexity), speed (due to parallelism), reliability (due to redundancy), efficiency and flexibility;
- where problems are simply too large for a centralised single "agent" to perform due to resource limitations or the sheer risk of having one centralised system;
- where multi-agent system solutions are used mainly to address problems requiring heterogeneous reasoning—perhaps use a blackboard approach here.

These seem to us to be Parallel AI and/or Distributed Problem Solving concerns, but there are papers addressing them that go under the banner of agents. In our view, problems requiring multi-agent solutions include the following (we do not aim to be exhaustive here). Problems requiring the interconnecting and inter-operation of multiple, autonomous, "self-interested" existing legacy systems, e.g., expert systems or decision support systems. Similarly, problems whose solutions draw from distributed autonomous experts, e.g., in health care provisioning or electricity distribution. Problems that are inherently distributed, e.g., geographically as in distributor sensor networks or air-traffic control. Problems whose solutions require the collation and fusion of information, knowledge or data from distributed, autonomous and "selfish" information sources, e.g., BT's personal travel assistance prototype application (Ndumu et al., 1998). Such systems also go under the name of cooperative information systems. If we were to generalise, we would say multi-agent solutions would at the very least require problems that tackle distributed resource allocation problems.

At the risk of offending many esteemed academic agents researchers, many of whom we know very well and collaborate with, we dare mention the following. Too many academic MAS researchers with a few notable exceptions do *not* really "own" (and hence do not appreciate) real MAS problems. Solution merchants looking for problems, drawn in by the hype of the domain? Furthermore, from the viewpoint of practitioners, some of the issues addressed by academics are rather pedestrian, again because of a failure of understanding and/or facing up to the real problems required of multi-agent system designers. We have in mind issues such as agent rationality arguments, logics, formalisation of belief, desire and intentions, etc.—for their own sake—which buys us nothing, and looks dangerously half-baked, hollow and impractical, particularly in the absence of a real problem. We say this with some trepidation because academic researchers require the freedom to do research for its own sake—and they should not be shackled by real world concerns. However, our point here is a rather very important one of premature formalisation. Stuart Russell argues in his 1997 AI journal article against what he calls "premature mathematization" in AI. He writes:

"There is always a danger, . . . , that . . . can lead to 'premature mathematization', a condition characterised by increasing technical results that have increasingly little to do with the original problem" (Russell, 1997)

We totally agree with this sentiment, and MAS research is falling into the same trap that Russell asserts of AI. Much literature on agent methods and techniques is published which appears to us "prematurely mathematized", leading towards increasingly technical results that bear little resemblance to any real world multi-agent problems that we can recognise.

In this regard, we emphasise a key caveat that we fervently believe in. A new field is *only* defined

by its *problems*, not its *methods/techniques*. We argue strongly that MAS has to some degree been falling into the trap that befell AI—that of deluding itself that its methods and techniques (e.g., cooperation, rationality theories, agent languages, conceptual and theoretical foundations, multi-agent planning, negotiation) are the real important issues. No! They are not! It is the problems that they are meant to solve as in air-traffic control or electronic commerce that are of foremost importance.

It is also important to point out that despite the hype, MAS are *not* fundamentally different. Indeed, we will assert strongly that a lot of the solutions to MAS problems are already out there already in bits and pieces. What MAS researchers need to do is a *creative* synthesising of some of these already-invented wheels. Where necessary, “new wheels” may be invented to link up the old ones. This is the philosophy we adopted in our ZEUS agent tool-kit work, which really just synthesises in one system many old wheels (Nwana et al., 1999). In carrying out this project, we pilfered unashamedly from a wide and diverse literature, including agent communication languages, distributed object technologies, the coordination, cooperation, negotiation literature, some views from rational agency, visual programming, planning, scheduling, methodological issues, research on ontologies, automatic code generation, HCI design, agent visualisation, etc. We contributed a few new wheels. The whole then becomes greater than the sum of its different parts, and hence novel! But this does not mean it is fundamentally different.

2.8 Some further research challenges on multi-agent systems

Drawing from all we have covered in this section, we believe there are many hurdles to be crossed before a broader uptake of multi-agent systems technology will be seen? We believe we need the following.

- First, researchers need to appreciate and address the sort of core multi-agent problems we highlight: the information discovery problem, the communication problem, the ontology problem, the legacy software integration problem, the reasoning and coordination problem and the monitoring problem. Most importantly, the solutions to these problems need to be formed in the context of solving real or quasi-real world problems.
- We need clear, non-*ad hoc*, methodologies—not the classic waterfall-type agent papers that already exist—but methodologies that address the core issue of facilitating the creation of an agent solution to a problem working from the problem and its domain down towards implementation using agent technology. This is in contrast to current attempts at methodologies that seem to emphasise the technology first, and then try to work from the technology to prototypical problems and domains. Such approaches are in general counter-intuitive and impractical, and most often fail to define a path from technology to prototypical problems and domains. We may require new languages to document the methodologies, perhaps by developing a new swathe of agent-based patterns. A pattern is not a heuristic; rather, they are good tools for documenting models, designs and frameworks. They also happen to be good tools for analysis, and we have begun looking at patterns for those roles in-house. In the AA-98 conference, Kendall (1998) and Aridor and Lange (1998) described work that began addressing this problem, although they emphasised pattern-based methodologies for agent design at the source level. A more appropriate approach, championed by Kendall and others, utilises role-based patterns that emphasise the problem and domain by looking at the roles of the various participants in a domain. Next, the particular responsibilities and collaborations of the participants.
- Tool-kits are absolutely vital to the MAS industry to facilitate their construction, engineering and standardisation, hence work such as ZEUS (Nwana et al., 1999) and dMARS. With an area of the sheer technical complexity as is the case in multi-agent systems, it makes much sense to encapsulate much engineering know-how into such tool-kits and frameworks. These tool-kits must have good debugging environments, as ZEUS poorly attempts to provide.
- Research on non-functional issues—currently being ignored—should commence in earnest. By

“non-functional issues” we refer to work on scalability, stability and performance of multi-agent systems. The paper by Lee et al. (1998) provides just a starting point towards such important work.

- Concerted research towards solving the ontology problem is absolutely crucial. The non-interaction hypothesis with respect to the ontology problem needs to be revisited and its ramifications explored. Is FIPA skirting round it—when it should be tackling it squarely? Without some resolution of the ontology problem, what are the real prospects for agent interoperability via the current ACL efforts? Legacy problems reign supreme—arguably a key bottleneck to the uptake of MAS technology. Research towards tools and efforts to automate this process is now an urgent concern.
- Standards are of course vital (cf. the impact CORBA is having with Distributed Objects). Hence, FIPA is a truly important initiative. However, as we have noted, is it developing the right standards? Would they be ignored, adopted or rejected? Are agent researchers supporting FIPA as they should?
- We end this section with a call for more efforts towards evaluating multi-agent systems. Recently, we searched largely unsuccessfully for literature to help us address this problem. Without such efforts that will force us to evolve evaluation metrics, how do we know we are making any real progress? In the short term, we as MAS researchers should define criteria for judging MAS solutions. Do they offer “value-added” solutions to other conventional applications and approaches, e.g., expert systems, distributed problem-solving approaches, blackboard approaches? This seems to us to be a clear, albeit general criterion. We only need to define “value-added” which could range from “excellent” (i.e. no solution was probable without an MAS approach), through “minimal” (where an MAS solution is marginally of value) to “very poor” (where conventional approaches offer better solutions). In the short term too, we need to visualise and debug multi-agent systems—and this is a non-trivial task. Ndumu et al. (1999) reports on some of our efforts towards this goal. Academics would argue that rigorous evaluation is essential. However, in the long term, the most objective criteria for evaluating MAS will be the market—would people pay for their employment in industry? Companies are buying into Distributed Object technologies because there are clear benefits which accrue from them; they fill a need. The same will be true of MAS if they are successful in the long term. They will not be taken up otherwise.

Our view is that most of the papers on multi-agent systems that have been submitted to all the past three PAAM conferences have, by and large, failed to tackle the sorts of problems we have highlighted so far. And PAAM is a conference for practical applications of agent technology. However, at PAAM98, all the invited speakers clearly appreciated the sort of views we have expressed in this paper. We were truly glad to see this. Mike Huhns, Jeff Rosenschein, Danny Lange and Richard Grant (from Autonomy) all gave excellent talks which derived from their need to address real world issues.

Selected papers from PAAM98 were edited by the authors, and have recently appeared as a special issue of the *International Journal of Applied Artificial Intelligence* on multi-agent systems (Nwana & Ndumu, 1999). The papers were those we judged addressed at least some of the sort of issues we have described so far. Ndumu et al. (1998) report on a personal travel assistance demonstrator that we have developed, and explains in some detail how we addressed the core multi-agent issues described earlier in that specific application. Nwana and Azarmi (1997) present papers that expound on several of the multi-agent and personal agent issues.

3 Personal and information agents: promises and reality

This section covers the two agent types that arguably led to the “agents craze” witnessed post 1994: personal and information agents. It covers the promises and reality of “agents that reduce work and information overload” (Maes, 1994).

3.1 Personal agents: promise, reality and challenges

Personal or Interface agents were championed by MIT's Pattie Maes in her seminal 1994 *Communications of the ACM* paper. Maes argued for user interfaces that go beyond responding to direct manipulation, i.e., towards Kay's dream of having indirectly managed human-computer interfaces by delegation to proactive and helpful software agents. Thus, Maes' *promise* was the provision of personal assistants that help and collaborate with the user at the user interface in some work environment, e.g., a spreadsheet or an operating system. Such interface agents would provide assistance by monitoring the user's actions in the interface, i.e., "watching over the shoulder of its user", learn new "short cuts", and suggest better ways of doing the task. Maes argued correctly that learning was invaluable to this task. We call this the strong personal agent hypothesis.

Let us explore this strong hypothesis further. Imagine you were working with some new operating system, say Windows NT or Linux. Your Personal Agent (PA) would need to be capable of the following.

- The PA would need to "know" the idiosyncrasies of the particular operating system reasonably or fully well in order to provide adequate suggestions.
- Importantly, the PA would need to continually model its users. Such user modelling would be vital for any personalisation of the assistance it provides to its user.
- Perhaps more importantly, the PA also needs a clear understanding of the various tasks the user is carrying out at the interface. It needs to be able to infer or ask what the user's *intention* at any time is—so that it is able to provide appropriate assistance. In short, the PA almost needs a deep cognitive model of the user.
- Implicit in Maes' strong assumption is the task of intelligent tutoring and mentoring, requiring the PA to have four key knowledge modules: domain knowledge on the application, a model of its user, strategies for assistance and a catalogue of typical problems users exhibit in that environment (Nwana, 1990).

In retrospect, Maes' strong hypothesis presumed that the intelligent tutoring problem had been solved, and this is definitely not the case. It is not surprising then what has happened since. It seems clear to us that Maes' original strong hypothesis is yet to be proved, even in demonstrator applications. We detect that the strong hypothesis has been quietly dropped and a much weaker substitute put in its place. Furthermore, it is truly debatable that there is *always* a real need for such personal agents. Most of us have experienced Microsoft's "Bob/paper clip". If you are like us who find it an annoyance most of the time, you can only conclude the hypothesis that such personal agents are needed is truly questionable. As far back as 1994, Mitchell et al. (1994) had also noted that

"it remains to be demonstrated that knowledge learned by systems like CAP¹¹ can be used to significantly reduce their user's workload" (p. 90).

This still remains an open question as per 1999.

There seems to us to be an issue here—that of the interplay between the nature of the task and the modelling or learning required. If the task is quite mundane as Maes' original paper alluded to, is it really worth anyone's while having a personal agent for it that requires modelling the user? On the contrary, if the task is quite complex (as in using some complicated software package), can a PA provide any really useful assistance without a deep cognitive model of the user and the task? There is an issue of a fine balance here in our experience. We guess that these issues emerged as problems leading to the adoption of a much weaker hypothesis. Hoffmann and Stamos-Kaschke (1998) question many of the underlying assumptions of personal agents.

Naturally, the weakened hypothesis has been proved by some of the demonstrators that have emerged from Maes' and other researchers' labs. They include prototypes for movie recommenda-

¹¹ CAP was a Calendar Apprentice agent developed at CMU that assisted the user manage and schedule its meetings.

tions, buying CDs, books, matchmaking agents, the Remembrance Agent (Rhodes & Starner, 1996), etc. The weak hypothesis no longer emphasises “looking over your shoulder” within some environment and learning; rather it emphasises having some initial profile of the user that is used to represent the user, and the only environment being explored today is the web—not spreadsheets, calendars or operating systems. Other issues emerge in this weakened form of the hypothesis, notably issues of trust, privacy, security and, inevitably, the ontology issues.

Such issues and others such as standardisation are at the heart of current research into personal agents. For example, there is the Open Profiling Standard (OPS) initiative (Hensley et al., 1997), that is attempting to standardise the profile information required by personal web agents.

There are thus outstanding several challenges for personal agent researchers.

- What currently is the key personal agent hypothesis that we are working towards—now that the strong hypothesis has either been dropped, found wanting or found to be difficult to realise? Do we really need personal agents all the time? Have their assumptions been tested in specific applications?
- Are personal agents complementary to “believable agents” defined as interactive versions of quality characters in traditional artistic media like film? In such work, there are different goals to achieve—the need to model emotions, intentions, social behaviours and discourse. The AA conferences report on such work on believable agents.
- Other desirable features of personal agents include having a common user profile and adaptability (i.e., being able to learn a user’s preferences and habits over time and adapt to the changing needs of the user). How do you detect interest migration? Information sharing amongst personal agents belonging to the same user makes much sense as well as agent collaboration to perform tasks for their users. However, such work is still in its infancy.
- Naturally, issues of trust, information privacy, security and ontology remain important research issues. For example, how do you maintain an up-to-date profile and how do you test its usefulness whilst keeping it private and secure? Users must *trust* their personal agents in order to volunteer information to them. Work towards standardisation such as OPS is important.
- There is little work towards personal agent frameworks where issues such as those in the previous two bullet points are addressed together. A personal agent framework would identify and implement common components of personal agents. It would include a common user profile, provide adequate and acceptable levels of privacy and security for personal profiles, and facilitate different personal agents of a single user to collaborate with each other. It will also facilitate the collaboration of personal agents between users or groups of users. Some work to develop such a framework is reported in Crabtree and Soltysiak (1999).
- Most importantly, empirical studies and experiments need to be carried out with such agents in order to test both their assumptions and usefulness. There is truly a dearth of such work in the literature that reports on real users’ views on specific personal agents. This is an area truly ripe for iterative design, to continuously enhance the quality of our personal agents. The complete near-absence of such work in the literature is truly worrying.

3.2 Personal information agents: promise, reality and challenges

The *raison d’être* for information agents is unarguable: to evolve tools to help manage the explosive growth of information, particularly online. Other factors are the volatility of the web and the fact that legitimate business data can increasingly be found on the web too (witness the growth of E-Commerce sites). Manually searching the web is intrinsically ineffective and inefficient. The underlying hypothesis and *promise* is that, somehow, we would greatly ameliorate the problem using information agents. The *reality* is that, unless you consider meta-search engines like Metacrawler¹² to be information agents, we are still well away from achieving this promise.

¹²<http://www.metacrawler.com>

We believe that really useful information agents would tackle tasks such as the following:

“Compose a report on the recent financial results of the six main competitors to our company using Internet-based sources, spending only £400 of access fees”.

Admittedly, this is quite a complex task. However, consider these from Sharpe (1996):

“Gather information from Internet sites on golf”

“Filter out mail apart from that relating to the pension plan for the past two days”

“Filter out post from Bob and summarise it for me, presenting it when I have a 15-minute gap in my schedule”

Some others from Bradshaw (1997):

“On what date in February did I have lunch with John?”

“Distribute this draft to the rest of the group and let me know when they have read it”

“Whenever a paper is published on agents, order a copy for my library”, to which your information agent may reply “Shall I order papers on XXX as well?”

“Keep me updated on the Iraq crisis”

Most would agree that really useful information agents would resolve such queries. However, the reality is that virtually no such information agent system exists, and that the state-of-the-art work in this regard is arguably still Etzioni and Weld’s (1994) Internet softbot.

It is easy to see why. Such tasks require that we resolve the information discovery problem, the ontology problem, the legacy software problem, the reasoning problem and the monitoring problem described earlier. We do not describe the details as we have done so already. Some query language would need to be used into which the above queries would be translated. The information would need to be collated and presented in some understandable format, in some cases summarised. The task would also have to be planned.

Resolving other mundane issues is currently a matter of sheer tedium. This is because currently, the lingua franca of the Internet is the Hypertext Mark-up Language (HTML), whose standard format enables browser software to interpret pages on the World Wide Web. However, as noted earlier, HTML web pages were designed for presentation of information for human perusal, and so valuable information is typically intermingled with formatting instructions—necessitating “wrapper induction” software programs, to parse the pages and extract the information from the formatting instructions. This typically amounts to “screen scraping”. If your agent manages to learn the structure of some page, and the page changes, it would need to re-learn the new structure. Even the evolution of XML/EDI framework for processing documents and exchanging transactions does not take away the problems. However, it does go somewhere to addressing some.

Our point here is simple—these hard information agent problems are largely not being tackled in order to resolve the sort of queries above. At his invited talk at PAAM98, Jeff Rosenschein of the Hebrew University of Israel and AgentSoft echoed the same views. He explained how AgentSoft’s LiveAgent Pro¹³ technology and tools are addressing such issues. He described how the tool is integrative, customisable, action-oriented and users can initiate actions. So to satisfy the query “find the home number of the CEO of IBM”, the agent first selects websites (i.e., information discovery and ontology) to combine to satisfy the goal. Next, it applies AI planning techniques (i.e., reasoning and coordination) to work out how to resolve the query. Planning here is truly hard because of uncertainty (of gathering information), integrating planning and execution and the existence of destructive and other undesirable activities. It employs wrapper induction techniques to learn about websites and supports agent negotiation. This is one of the more comprehensive efforts we know of which attempts to tackle the information agent problems squarely. Since the work is commercially sensitive, it is not clear what progress has been made since Rosenschein’s presentation at PAAM98.

¹³<http://www.geocities.com>

As for the challenges of information, they are similar to those of multi-agent systems less the communication problem in some instances. Furthermore, we really need to work towards being able to satisfy the sort of queries we have noted at the beginning of this section. Our view again is not much progress has been made, and that a significant amount of drudgery is unavoidable in order to make progress in this area. The ramifications of any successes in this area on the general area of knowledge management would be immense.

3.3 Mobile information agents: promise, reality and challenges

We know of no other technology that, *prima facie*, is so seductive and promises so much, but on closer scrutiny (and indeed so far in practice) appears would deliver so little. The key hypothesis is that agents need not be stationary. This is largely because the benefits of mobile agents are largely non-functional; mobility is neither necessary nor sufficient for agenthood. Danny Lange of General Magic, chief designer of Aglets and a key champion of mobile agent technology, posited “seven good reasons for using mobile agents” at his PAAM98 invited talk (Lange, 1998). They were that mobile agents reduce the network load, they overcome network latency, they encapsulate protocols, they work asynchronously and autonomously, they adapt dynamically, they are naturally heterogeneous and they are robust and fault tolerant. Danny presented a resolute defence of mobile agents work, arguing they are an emerging technology that makes it easier to design, implement and maintain distributed systems.

Danny Lange’s arguments notwithstanding, any field which can show off literally hundreds of experimental mobile agent development systems but with no clear applications that would *not* exist without mobile agents, must be in some crisis. We make this bold statement after having explored mobile agents ourselves. We are obviously over-emphasising our point here, but mobile agents are clearly still a solution with no clear problem. Note that we distinguish between mobile agents and mobile objects. Experimental systems include Odyssey,¹⁴ Aglets,¹⁵ Concordia¹⁶ and Voyager.¹⁷ Standardisation efforts are also underway within the Mobile Agent System Interoperability Facility (MASIF).

Let us be a bit clearer. Mobile agents seem to take us back from a more declarative position to writing software to a more procedural/imperative one, arguably a backwards step. However, this is probably unfair. More relevantly, there are many viewpoints to mobile agents. First, you can see mobile agents as “mobile” FIPA-ACL or KQML-type *messages*. Many papers in the literature clearly take this view, arguably providing a benefit if network latency is a problem. So instead of messaging a server in New York from London when the US is rising, it makes more sense to launch a mobile agent in such a case. Secondly, you may see a mobile agent as a mobile *object*, as in the object-oriented sense. This is the view if you just use the serialisation/de-serialisation classes in Java—objects can be serialised, sent across the network and unserialised at the other end. However, all this buys is object mobility. In our view, a significant part of the mobile agent craze is all about no more than this. Thirdly, there is the view of mobile agents as mobile *processes*. This was the original Telescript view, where agents were essentially mobile processes. It is arguable whether any mobile process is an agent, but we will avoid this debate here. Lastly, you could have mobile objects that truly encapsulate protocols, strategies, etc. These would pass as mobile agents. Mobility here is also an extension of the agent’s autonomy. The reader should note here as we have that—no matter how seductive mobile agents appear and which of these views the designer adopts—they still have to contend with all the problems we described in section 2: information discovery, communication, ontology, legacy software, monitoring and, indeed, more. This is because the true mobile agent has to worry about issues like remote code execution. The designer also has to worry about issues of

¹⁴http://www.genmagic.com/html/agent_overview.html

¹⁵<http://www.trl.ibm.co.jp/aglets>

¹⁶<http://www.meitca.com/HSL/Projects/Concordia>

¹⁷<http://www.objectspace.com/voyager>

remote execution versus code on demand, or weak or strong migration; the latter refers to a situation where the entire agent state (i.e., data and execution state) is transferred along with code to the next location. Strong migration can be very time consuming and expensive for large agents with multiple threads—which they would have to be if they are to be any use. Weak migration, on the other hand, leads agent designers to design more complicated agents. These are all before the obvious issues of security are considered.

All in all, there is a strong *prima facie* case for simply ignoring mobile agents. We do not advocate this though. We just point out that really useful mobile agent applications may also emerge when we have fully developed their static counterparts. Mobile agents just bring an additional set of problems on top of those we have with static agents already.

As for challenges for mobile agents, their usefulness truly needs to be proved in real applications whilst simultaneously proving that their usefulness exceed their obvious possible security risks. Other obvious research challenges include the following:

- Research has to be carried out towards mechanisms for agent security; much of this is already on going.
- Mobile agents clearly require transactional management support structures if they will be of any use. What sort of transaction models should be employed? Consider an electronic cash carrying agent which migrates to some supplier's server. Both the server's state and the agent's state change after completion of the payment transaction. Now just suppose the transaction failed—we must be able to recover the states just before the transaction begun such that the agent can return to its owner, and the server's state is “rolled-back”.
- Many truly useful mobile agents would tend to be large, and the costs of sending large agents across the network may well be greater than its benefits. Empirical research to point out where and when mobile agents are truly valuable still requires doing.
- How do mobile agents collaborate? Research into how they are controlled, i.e., control structures, is still in its infancy.

Again, such core problems as we have described for personal agents, information agents and mobile agents are only *infrequently* addressed in papers submitted to PAAM and other agent-related conferences.

4 Other miscellaneous viewpoints

Notwithstanding the fairly negative views propounded in the preceding sections, our observations of the agents research, development and fledgling commercial exploitation scene suggests the field is set to mature to the point of addressing some of the problems we have outlined. We base our arguments on the fact that there now appears to be clear *market sector* segmentation of the field into areas such as:

- electronic commerce—ostensibly providing value-added and personalised support and recommendation to online retail, supply chain integration and personalised customer interaction;
- corporate Intranets or managed Extranet solutions (e.g., the AmericaOnline, BT's LineOne or Excite's online communities), again providing value-added and personalised searching and collation support;
- personal assistants such as General Magic's Portico system wherein personal agents allegedly provide a service which filters your emails and communications, routes your telephone calls or answers them, books appointments for you, etc. General Magic is supposed to have rolled out this service—we are not aware so far of its progress;
- resource allocation and management systems, for example, for telecommunications network management, production planning, air-traffic control, etc. and
- agent-based middleware—to mediate between applications and network-layer applications.

The emphasis on market sector problems suggests to us that the R&D work will be forced to tackle

real-world problems, thus refocusing efforts towards addressing the issues of information discovery, communication, ontology, collaboration and reasoning, and monitoring that we described earlier.

It is worth remarking that in spite of the wild speculation about the revenues to be accrued from agent technology by the turn of the century, one year to the day, it remains unclear who is or will make money out of agents and how. Most of the personal and information agent start-up companies such as Autonomy, Verity, AgentSoft, Excite/Jango and FireFly are still loss leaders. Many of the current offerings are frankly no more than meta-search engines. Further, they make their money not by providing a core *agent* service, but by providing a value-add to a core service, thereby helping to pull in further customers and locking them in (which in itself is not a bad thing!). In fact, this begs the question—is there any real core agent service or product? Would there ever be one? Hence, it is really difficult to compute what the core financial figures due to agent technology are. The situation is rather worse for collaborative agents, with few if any companies, and arguably none raking in any revenues, except perhaps the commercial arm of the Australian AI Institute (AAIL)—but then again AAIL is a non-profit organisation. For agent researchers though, it seems a good idea to start up companies in order to deliver real solutions—this is a sure way of discovering the real problems. Besides, if you are lucky, it could be subjected to a strategic buy-out as has been the case with Etzioni and Weld's Jango, FireFly and recently AgentSoft, which has been bought twice over. This could be a sure way of getting rich if you can succeed to attract venture capital to start up the venture in the first place.

To summarise, the sheer difficulty of the problems inherent in the agent approach suggests to us that rather than having a revolutionary impact on society as was hyped, the promises of the approach will be realised in an evolutionary manner. Consider the sheer difficulty of developing an industry acceptable ontology for knowledge-level commerce for even a product as simple as cars! Consider the headache that would emerge if we were to get the car industry to standardise their ontologies.

5 Conclusion

This paper has very briefly evaluated the recent progress on realising the promises of agent technology. Our evaluation has highlighted the fact that not much discernible progress has been made post 1994, perhaps because researchers have failed to address the practical issues surrounding the development, deployment and utilisation of industrial-strength production systems that use the technology. We note that once greater effort is placed on building useful systems, not just prototype exemplars, the real problems inherent in information discovery, communication, ontology, collaboration and reasoning, will begin to be addressed. We are confident that the field will mature to this point, since already there is evidence of greater market sector segmentation into clear application-focused areas—where there are financial imperatives, the research will follow!

The clear message of this paper is in our (hopefully) cogent argument that the devil in realising the promises of agent technology is in the details. As long as the myriad of numerous¹⁸ national and international agent workshops and conferences continue to fail to tackle them, progress will remain arguable and debatable.

Finally, we draw from Donald Schon's (1983) provocatively, original book which encourages "reflection-in-action" on the part of all professionals and researchers. We believe that the entire thesis and contribution of this paper (beyond our contention that not enough progress has been made in the agents area) is captured in this book when Schon describes the dilemma of "rigor or relevance":

"there is a high, hard ground where practitioners¹⁹ can make effective use of research-based theory and technique, and there is a swampy lowland where situations are confusing 'messes' incapable of technical solution. The difficulty is that the problems of the high ground, however great their technical interest, are

¹⁸ There is real case for arguing for a reduction in the number of agent events if you accept our contention that they do not add much value over what is out there already.

¹⁹ Read 'researchers'.

often relatively unimportant to clients or to the larger society, while in the swamp are the problems of greatest human concern. Shall the practitioner stay on the high, hard ground where he can practice rigorously, as he understands rigor, but where he is constrained to deal with problems of relatively little social importance? Or shall he descend to the swamp where he can engage the most important and challenging problems if he is willing to forsake technical rigor?" (Schon, 1983, p. 42)

We believe Schon in this quote succinctly captures a key dilemma in trying to make agent technology a genuine revenue machine or paying proposition. Are agent researchers forsaking relevance for rigor? There is ample evidence to suggest so. As agent researchers, we need to descend into the "swamps" to confront the real problems that plague the realisation of the potential of agent technology.

References

- Aridor, Y and Lange, DB, 1998. "Agent design patterns: elements of agent application design" *Proc Autonomous Agents 1998* pp. 108–115.
- Bradshaw, JM (ed), 1997. *Software Agents* MIT Press.
- Crabtree, B and Soltysiak, S, 1998. "Identifying and tracking changing interests" *Int J Digital Libraries* **2** 38–53.
- Davies, NJ, Weeks, R and Revett, MC, 1997. "Information agents for the World Wide Web" in Nwana, HS and Azarmi, N (eds) *Software Agents and Soft Computing: Concepts and Applications* Springer-Verlag, pp. 81–99.
- Etzioni, O and Weld, D, 1994. "A softbot-based interface to the Internet" *Communications of the ACM* **37**(7) 72–76.
- Finin, T and Labrou, Y, 1997. "KQML as an agent communication language" in Bradshaw, JM (ed) *Software Agents* MIT Press, pp. 291–316.
- Foner, L and Crabtree, IB, 1997. "Multi-agent matchmaking" in Nwana, HS and Azarmi, N (eds) *Software Agents and Soft Computing: Concepts and Applications* Springer-Verlag, pp. 100–115.
- Genesereth, MR and Fikes, RE (eds), 1992. "Knowledge Interchange Format, Version 3.0" Reference Manual, Computer Science Department, Stanford University. (Available at <http://www-ksl.stanford.edu/knowledge-sharing/papers/README.html#kif>)
- Genesereth, MR and Ketchpel, SP, 1994. "Software agents" *Communications of the ACM* **37**(7) 48–53.
- Gruber, T, 1993. "A translation approach to portable ontology specifications" *Knowledge Acquisition* **5**(2) 199–220.
- Hensley, P, Metral, M, Shardanand, U, Converse, D and Myers, M, 1997. *Proposal for an Open Profiling Standard* Technical Note W3C, June. (Available as <http://www.w3.org/TR/NOTE-OPS-FrameWork.html>)
- Hoffmann, U and Stamos-Kaschke, J, 1998. "Mission impossible? User meets agent" *Proc Practical Application of Intelligent Agents and Multi-Agent Technology* London, pp. 173–189.
- Huhns, MH and Singh, MP, 1998. *Readings in Agents* Morgan Kaufmann.
- Jennings, NR and Wooldridge, M, 1998. *Agent Technology: Foundations, Applications, and Markets* Springer.
- Jennings, NR, Varga, LZ, Aarnts, RP, Fuchs, J and Skarek, P, 1993. "Transforming standalone expert systems into a community of cooperating agents" *J Intelligent and Cooperative Information System* **2**(3) 289–318.
- Kendall, EA, Murali Krishna, PV, Pathak, CV and Suresh, CB, 1998. "Patterns of intelligent and mobile agents" *Proc Autonomous Agents 1998* pp. 92–99.
- Kushmerick, N, 1997. "Wrapper Induction for Information Extraction" *PhD Thesis*, Technical Report No: UW-CSE-97-11-04, Department of Computer Science & Engineering, University of Washington, USA.
- Lange, DB, 1998. "Mobile agents: Environments, technologies and applications" *Proc Practical Application of Intelligent Agents and Multi-Agent Technology 1998* London, pp. 11–14.
- Lee, L, Nwana, HS, Ndumu, DT and De Wilde, P, 1998. "The stability, scalability and performance of multi-agent systems" *BT Technology J* **16**(3) 94–103.
- Lenat, DB, 1995. "CYC: A large-scale investment in knowledge infrastructure" *Communications of the ACM* **38**(11) 33–38.
- Maes, P, 1994. "Agents that reduce work and information overload" *Communications of the ACM* **37**(7) 31–40.
- Ndumu, DT, Collis, JC and Nwana, HS, 1998. "Towards desktop personal travel agents" *BT Technology J* **16**(3) 69–78.
- Ndumu, DT, Nwana, HS, Lee, LC and Haynes, HR, 1999. "Visualization and debugging of distributed multi-agent systems" *Applied Artificial Intelligence J* **13**(1/2) 187–208.
- Newell, A, 1982. "The knowledge level" *Artificial Intelligence* **18** 87–127.
- Norman, D, 1994. "How might people interact with agents?" *Communications of the ACM* **37**(7) 68–76.

- Nwana, H and Azarmi, N (eds), 1997. "Software agent and soft computing: towards enhancing machine intelligence" *Lecture Notes in Artificial Intelligence* **1198** Springer-Verlag.
- Nwana, H, 1996. "Software agents: an overview" *The Knowledge Engineering Review* **11**(3) 205–244.
- Nwana, HS, 1990. "Intelligent tutoring systems: an overview" *Artificial Intelligence Review* **4**(4).
- Nwana, HS, Ndumu, DT, Lee, LC and Collis, JC, 1999. "ZEUS: A toolkit for building distributed multi-agent systems" *Applied Artificial Intelligence J* **13**(1/2) 129–185.
- Nwana, HS, Lee, L and Jennings, NR, 1996. "Coordination in software agent systems" *BT Technology J* **14**(4) 79–88.
- Nwana, HS and Ndumu, DT (eds), 1999. Special Issue of *Applied Artificial Intelligence J* **13**(1/2).
- O'Brien, P and Wiegand, M, 1997. "Agents of change in business process management" in Nwana, HS and Azarmi, N (eds), *Software Agents and Soft Computing: Concepts and Applications* Springer-Verlag, pp. 132–145.
- Ovum, 1994. *Intelligent Agents: the New Revolution in Software* Ovum Report, Ovum Publications, London.
- Rao, AS and Georgeff, MP, 1995. "BDI agents: from theory to practice" *Proc 1st International Conference on Multi Agent Systems* San Francisco, California, pp. 312–319.
- Reinhardt, A, 1994. "The network with smarts" *Byte* October, 51–64.
- Rhodes, BJ and Starner, T, 1996. "Remembrance agent: a continuously automated information retrieval system" *Proc Practical Application of Intelligent Agents and Multi-Agent Technology* London, pp. 487–496.
- Russell, S, 1997. "Rationality and intelligence" *Artificial Intelligence J* **94**(1–2) 57–77.
- Schon, D, 1983. *The Reflective Practitioner: How Professionals Think in Action* Basic Books.
- Searle, JR, 1969. *Speech Acts*. Cambridge University Press.
- Sharpe, R, 1996. "Software agents" *Cambridge Market Intelligence IT Report* Cambridge Market Intelligence Ltd.
- Wittig, T (ed), 1992. *ARCHON: An Architecture for Multi-Agent Systems*, Ellis Horwood.
- Wooldridge, M and Jennings, NR, 1995. "Intelligent agents: theory and practice" *The Knowledge Engineering Review* **10**(2) 115–152.