An Analysis of the Dynamics of Adaptive Multiagent Systems, with an Application to Global Information Exchange Systems

José M. Vidal
Computer Science and Engineering
University of South Carolina
Colombia, SC, 29223
http://jmvidal.ece.sc.edu
vidal@sc.edu

March 8, 2001

1 Introduction

In the past twenty years we have seen an increasing emphasis on the study of the dynamics of complex systems such as the human immune system and the economy. Some of this work is reflected in volumes such as (Kauffman, 1993) (Resnick, 1994) (Axelrod, 1997). The belief is that we need to understand the relationships between disparate phenomena in order to determine the long-term repercussions of local actions. For example, economic policies like sanctions can have dire consequences on the environment because they might set off a sequence of events that result in undesirable consequences.

The rise of Internet and pervasive computing are prompting the need to understand and engineer systems of similar complexity. These systems, however, are composed of articial agents which implement machine learning techniques and will be refered to as adaptive multiagent systems. Examples range from simple adaptive embeded systems for pervasive computing and systems for dynamic resource allocation under uncertainty to grandiose world-wide electronic marketplaces of agents, global information exchanges, and "global brains" (Brooks, 2000) that accrue humanity's knowledge.

In this proposal I detail my plan to study the dynamics of adaptive multiagent systems in order to develop a predictive theory of their dynamics and a methodology for the engineering of utility-based adaptive multiagent systems. I also present an system architecture and development plan for an incentive-compatible information exchange system. This system will use the engineering techniques I develop as well as serve as a first step towards realizing the visionary scenarios described above. Finally, my teaching plan incorporates the decentralized mindset into the teaching of software enginerring and develops an applied curricula and supporting materials for the nascent field of agent-based software engineering.

1.1 Adaptive Multiagent Systems

An adaptive multiagent system is composed of articial (software) agents that use machine-learning techniques to improve their behavior over time. The behaviors it exhibits are sometimes those of a naturally-occurring complex adaptive systems (Holland, 1995) but, since artificial agents have different characteristics from animals (e.g., they learn in a different way and do not reproduce), their emergent behavior is different. Since the agents in these systems are constantly learning about the effects of their actions, the world they live in, the behaviors of the other agents, and the effect that these behaviors have on the system, it is a challenge to build systems that will exhibit a specified behavior.

A simple example is the use of machine-learning agents in the iterated Prisoner's Dilemma (Axelrod, 1984). Another example is the e-commerce system with agents buying and selling from each other and learning about the quality of goods and services received (Kephart et al., 2000). In this system the agents could learn models of other agents and can make better decisions about who to trade with and who to avoid (Vidal et al., 1998).

Since the agents in an adaptive multiagent system are constantly changing their behavior to incorporate the new learned knowledge and they interact with each other in complex ways, it is hard to determine how the system as a whole will behave. Currently the only general way to find out what will happen is by actually taking the time to build the whole system and running it. However, it is very important to determine beforehand what to expect in terms of system dynamics. For example, we would not want to build a system where the agents end up not communicating with each other, or where they communicate so much they overwhelm the communications infrastructure, or where a few agents are destined to become dictators of the whole system, etc.

1.1.1 Current Work

Recently, we have seen some research that tries to understand at a higher level how to build these systems. For example, the Collective Intelligence framework (Wolpert and Tumer, 1999) gives a formal way of determining the reinforcement to be given to individual reinforcement learning agents in order to speed up the convergence of the system to a given global utility function. That is, given a utility function for the system's behavior (e.g. it is best if only x people go to the bar at once) the author show how to derive the *wonderful life utility* reward for each agent. This reward speeds the agent's learning because it eliminates any contributions from the other agent's and gives the agent a reward proportional to its contribution to the global utility.

Another example is our CLRI framework (Vidal and Durfee, 2001) (Vidal and Durfee, 1998b) which shows how to parameterize the abilities of the agents' learning algorithm in order to determine the expected behavior of the system, thereby shedding some light into the dynamics of adaptive multiagent systems. The CLRI framework describes an agent i's **behavior** at time t with the function $\delta_i^t(w)$ which is a mapping from every world state to the action that i will take in that state. We also define Δ_i^t to be the behavior the agent should have, that is, the behavior that would give i the best possible payoff. This function is referred to as the **target function** for agent i. Finally, $e(\delta_i^t)$ is i's **error**, or the probability that i will take an action that is different from the one it should take, as given by Δ_i^t.

The learning problem the agent faces is to change its $\delta_i^t(w)$ so that it matches $\Delta_i^t(w)$. If we imagine the space of all possible decision functions, then agent i's δ_i^t and Δ_i^t will be two points in this space, as shown in Figure 1. The agent's learning problem can then be re-stated as the problem of moving its decision function as close as possible to its target function, where the distance between the two functions is given by the error $e(\delta_i^t)$. This is the traditional machine learning problem.

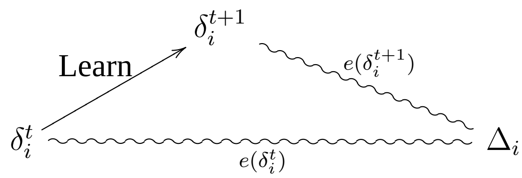

Figure 1: The traditional learning problem.

However, once agents start to change their decision functions (i.e., change their behaviors) the problem of learning becomes more complicated because these changes might cause changes in the other agents' target functions. We end up with a moving target function, as seen in Figure 2. In these systems, it is not clear if the error will ever reach 0 or, more generally, what the expected error will be as time goes to infinity. Determining what will happen to an agent's error in such a system is what we call the **moving target function problem**, which we address in this article. However, we will first need to define some parameters that describe the capabilities of an agent's learning algorithm.

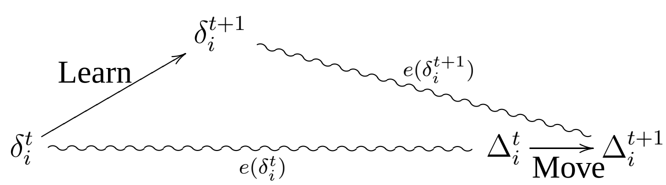

Figure 2: The learning problem in learning MASs.

My CLRI framework shows how to calculate the expected error for adaptive multiagent systems given that we can describe the agents' learning algorithms using three parameters: **change rate**, **learning rate**, and **retention rate**; and we can describe the effects of one agent's behavior changes on another agent's target function using the **impact** parameter. These results are very encouraging but represent only a first step towards my goal of engineering adaptive multiagent systems. Specifically, the mapping from a general machine-learning algorithm, e.g., Q-Learning (Watkins and Dayan, 1992), classifier systems (Holland, 1995), or neural networks, cannot always be achieved and always seems to introduce some error.

2 Plan of Work

My career goal is to develop techniques for the engineering of adaptive multiagent systems and to promote an understanding of emergent and distributed phenomena among students. My research plan is divided into two parts. The theoretical part, detailed in Section 2.1, involves the development of new theories for the understanding of adaptive multiagent systems. This work, if successful, will have a deep societal impact as explained in Section 2.1.4. Section 2.2 presents the applied part of my research. It sketches an implementation plan for a family of incentive-compatible information exchange systems. These systems will allow users to form webs of trust between themselves and to harness the knowledge of the whole group when needed. Since this part is programming-intensive, I will require the help of two graduate students and funds to cover the cost of their equipment.

2.1 Research Plan: Engineering Adaptive Multiagent Systems

The theory I am developing views agents as utility maximizing or rational entities in much the same way economists view humans as rational utility maximizing agents. However my agents must learn which behaviors lead them to the highest payoff. Of course, as one agent adapts by changing its behavior, this might affect the utility that all the other agents receive.

As the agents modify their behavior the system designers will be concerned with the global utility of the system. The designer of the system defines a global utility for the system and then designs the system, to the best of his abilities and within the given constraints, so it will behave in a way that maximizes this utility. The global utility can be as simple as just the sum of the individual agent's utilities, or it can be a more complex function of the agents' behavior.

We can get a better understanding of the dynamics of such a system if we picture a 2-dimensional plane where each point represents a possible behavior for an agent. The points are connected to each other via directed links, each connection representing the possibility that the agent could learn and modify its behavior, moving it from one point to the other. We can then visualize a third dimension that represents the utility the agent will receive if it engages in the given behavior.

An agent's learning can then be viewed as a path in this landscape. Depending on the agent's learning algorithm, we can often assume the agent generally goes "up" on the utility landscape. That is, it tries to maximize its utility. However, since the other agents are also changing their behavior, this means the agent's landscape might change after each step and each of its steps might change other agents' landscapes.

Furthermore, we can take a global view where each point in the xy plane is a vector that contains all the agents' behaviors. In this landscape the z axis is the global utility. This is he picture the system designer is most concerned with. Does the system behavior converge to a global optima? What behavior does it engage in? The answers to these questions will depend on the particulars of the learning algorithm used. My research will answer these questions.

For example, in the simple case where the global utility is simply the sum of the individual agents' utilities then we know that if an agent takes a step "up" this step will translate into a step up in the global utility, as long as its change in behavior does not reduce the utility that other agents will receive.

In general, we define a system as **factored** (Wolpert and Tumer, 1999) if any change in an agent's behavior which gives it a higher utility also leads to a higher global utility. This might not happen if an agent's change in behavior reduces other agents' utility. This is formally captured by the **impact** parameter I define in my CLRI framework.

My research plan is to continue the study of the dynamics of adaptive utility-maximizing agents in multiagent systems. I will continue to expand the current set of tools I have created as well as incorporate other tools such as Wolpert's "wonderful life utility" (Wolpert and Tumer, 1999) and Kauffman's NK-landscapes, as shown in Section 2.1.1, and continue to use agent-based based simulations (Vidal and Durfee, 1998a) (Vidal et al., 1998) of increasing complexity in order to complement the theory, as explained in Section 2.1.2. All this in an effort to develop a principled method for the engineering of adaptive multiagent systems, explained in Section 2.1.3. Finally, some of the possible impacts of this research are presented in Section 2.1.4.

2.1.1 Utility Landscapes

Utility landscapes are exemplified by Kauffman's NK landscapes (Kauffman, 1993). An NK-landscape has N agents. Each agent has a state and a utility for each of the possible global states, where a global state is the set of all the N agents' states. An agent's utility is a function of the state of K of its neighbors. The global utility is the sum of the individual agent utilities. For example, if K=0 then an agent's utility depends only on its own state, so the agent can simply change its state to the state that given it the highest utility. At the other extreme, when K=N-1 an agent's utility depends on everyone else's state. The K=0 landscape takes the form of a very high and smooth peak with no local optima. The K=N-1 landscape is very rugged. In fact, the utility or height of adjacent points is not correlated in any way.

NK landscapes where inspired by biological evolution and have been used to understand the dynamics of evolving biological systems. Other results (Kauffman, 1995) have shown that the global optima is more likely to be reached in rugged landscapes if the agents form "patches" where, instead of trying to maximize their own utility they try to maximize the utility of their patch. I have used this insight, and the mathematical tools that come with it, to develop algorithms for dynamic resource allocation in multiagent systems[1]. I will also study the application of these mathematical to the analysis of artificial multiagent systems.

2.1.2 Agent-Based Simulations

Agent-based simulations are important technique for understanding the dynamics of these systems. They are well-established tools for this type of research (Axelrod, 1984) (Axelrod, 1997) (Holland, 1995) (Vidal and Durfee, 1998a) (Vidal et al., 1998) which provide insights when theory fails, or when the appropiate theory has not yet been developed.

I plan to continue the development of these simulations in an effort to gain a better understanding of the impact that different machine learning techniques have on the dynamics of different adaptive multiagent systems. Specifically I am concerned with the problems of convergence— does the system behavior become stable? robustness— does the system continue to exhibit the same behavior under small disturbances? and timing— how long does it take for the system to converge?

One example is the simulator[2] I have developed for a dynamic resource allocation domain. This domain consist of target agents and follower agents in a two-dimensional space. The agents have a limited radar range as well as a limited communications range. Nevertheless, the follower agents are able to coordinate using only local information so that the target agents consistently have at least two follower agents following them.

The protocol I designed and implemented enables each agent to communicate the information needed to *recuce its dependence on other agents' states*. This is a key insight gained from these simulations. An agent can use communications in order to make its utility more independent of

[1] `http://jmvidal.ece.sc.edu/targetshare/`

[2] `http://jmvidal.ece.sc.edu/targetshare/demo/info.html`

the other agents' state (i.e., in terms of the NK landscape this amounts to using communications to reduce K), and I have shown one effective way to accomplish it.

This and other simulators I have built[3] form part of my ongoing research into the analysis of dynamic multiagent systems. As part of the proposed research I will extend the later simulations to cover other more complex domains and include more sophisticate learning techniques such as Q-learning (Watkins and Dayan, 1992), decision trees, and genetic algorithms (Holland, 1995).

2.1.3 Engineering Adaptive Multiagent Systems

A general engineering discipline for the design and development of multiagent systems is emerging. That is, in the same way that object-oriented programming developed its own Unified Modeling Language (UML) (Fowler et al., 1999) and design patterns (Gamma et al., 1995), the field of multiagent research is now ready for similarly powerfull design tools. In fact, the challenge has already been made explicit by experienced researchers in this field (Jennings, 2000) (Petrie, 2000) (Huhns, 2000) to develop this new agent-based software engineering approach.

My research will develop a general utility-based approach to multiagent systems engineering. The basic steps of this approach have already been hinted-at in this proposal. We first consider each agent as endowed with its own utility function and we describe the desired global behavior using another utility function. Then we decide how the agents will behave. The agents can be either selfish utility-maximizing agents, they can form teams (or "patches") and maximize the utility of the group, or they can be made to maximize the global utility. Some of these choices might not be possible in some problem domains. Communication and coordination strategies are chosen in order to align the agents' behavior with the global utility. Finally, a machine learning algorithm might be used in some agents in order to increase their effectiveness over time and alighn their actions with their utility and with the global utility.

Deciding on which type of agent behavior (e.g., selfish or team-oriented), coordination or communication protocol, and machine learning algorithm to use for a particular implementation is still an open problem. The proposed research, however, will shed some light into this question. Specifically, once we can view the agent interactions as search in an n-dimensional utility landscape (Section 2.1) and we have the tools to analyze the dynamics of this search, then plotting the effect of any combination of behavior/protocol/learning algorithms should be a mostly automatic task.

That is, when the proposed resarch is finished I will be able to provide tools, either mathematical or programmatic, which will take as input parameters describing the proposed system (e.g., team-based agents with size of five, a contract-net (Smith, 1981) protocol, and reinforcement learning with learning rate of $.8$), and deliver a description of the expected system dynamics.

2.1.4 Impact of this Research

The theoretical research proposed in this section will enable the development of a wide range of multiagent systems. These include the allocation of weapons to targets in a military domain, the allocation of goods an services in an electronic commerce application (Vidal et al., 1998), and the development of a manufacturing factory scheduler (Baker et al., 1999). These systems could be made more robust and dynamic with the use of adaptive agents, if we could predict how local adaptivity affects global behavior. The coming emergence of nanotechnologies and of pervasive

[3]See for example: `http://jmvidal.ece.sc.edu/applets/circle/` and (Vidal and Durfee, 1998a)

computing, where every electronic gadget is on the network, brought about by new technologies like Jini and Bluetooth will also enable the deployment of a mirriad of new adaptive multiagent systems.

The proposed research will also enable us to take one more step towards the goal of a "global brain" (Brooks, 2000) (Heylighen and Bollen, 1996) (Heylighen, 1999) where the "sum" of the knowledge is made accessible to anyone, at any time. The WWW has allowed us to place a lot of information online. We must now endeavor to trasform this information into knowledge. One way to accomplish this is with the use of intelligent learning agents that can share and seek information in an intelligent fashion. For example, a doctor will have an agent that holds his information and answers his queries by asking other agents that it trusts (from past experience), or asking those agents to recommend other agents. Another example is a democracy where every person constantly expresses an opinion, via his agent, on those subjects he cares about so our leaders are always able to carry out the actual will of the people. Imagine a government that makes funding decisions by instantenously aggregating the knowledge of all the appropriate researchers in the field, rather than relying on the knowledge of a handful of reviewers.

2.2 Research Plan: A Global Information Exchange Implementation

Of course, the idea of a global brain or World Wide Mind (WWMind) which enables the aggregate of human knowledge to be accessible in a context-sensitive manner by anyone at any time is currently only science-fiction. However, I believe we can expect it to be built within the next few decades and the system I propose in this section is one step in that direction. Please note that I do not propose to tackle all the technologies that will be needed to develop a WWMind. I will, instead, concentrate on the dynamics of learning agents as they form opinions of each other which, in turn, lead to virtual "communities of trust".

2.2.1 System Design: Information Is Money

Lately, it has become popular to think of the future of the Internet as consisting of millions of agents buying and selling services from each other (Kephart et al., 2000) (Vidal et al., 1998). While this idea is attractive and well-justified given current trends, there some major obstacles that must be conquered. Namely, such agents will need to transact real money. This means that each agent will need complex encryption and micropayment (e-money) software installed. Furthermore, the ability to trade real money will put these agents in unknown legal grounds. If the agent loses a million dollars (one cent at a time), who is liable?

While these barriers might be overcome for some closed systems, I believe there is a better solution. The solution I propose is a system where agents exchange information instead of money. That is, the agents use the information they have as money and trade it for the information they need. This is enabled by the use of learning agents that model each other's past behavior and change accordingly.

Specifically, I propose to build a prototype multiagent system where agents exchange preferences/ratings over pieces of information (e.g., books, research papers, websites, etc.) The agents do not exchange money, they use information itself as a currency. In order for the system to work, the agents must learn about each other and form dynamic communities of trust. Each agent generates a set of evaluations over data and over the other agents. Through repeated interactions with

D	Set of documents, where $d \in D$
E	Set of evaluations, where $e \in E$ and $e = (d, u)$, where u is a utility.
E_i	Set of evaluations done by i.
$U_i(e)$	Utility agent i derives from evaluation e
$U_i(j)$	Utility agent i expects from an evaluation e_j from j.

Table 1: Summary of the notation used.

the other agents, each agent learns which ones provide it with better information, under the assumption a few bad interactions are not fatal to an agent. Over time, each agent will create its own community of trusted agents.

More formally, let D be the set of documents the agents will evaluate, where $d \in D$. For example, each one of these documents could be a URL. The set of all evaluations is E, where $e \in E$. An evaluation e is defined as pair $e = (d, u)$, consisting of a document d, and a utility value u. The set of all agents is A, were $i \in A$. Subscripts match these evaluations to agents, and superscripts identify individual evaluations. For example $e_i^1 = (d_i^1, u_i^1)$ is agent i's evaluation number 1, which gives document d_i^1 a utility of u_i^1. The utility value u_i^1 in evaluation (d_i^1, u_i^1) tells us how much agent i liked the document d_i^1. In order for agent i to produce this evaluation it must have "read" (or, somehow, processed) document d_i^1.

Each agent i can expect to receive a utility $U_i(e)$ from an evaluation e. The value of the evaluation e_i to i is given by $U_i(e_i)$. The utility value has two components: **private value** and **common value**. The private value refers to how much value i can derive *directly* from e_i. The private value of e_i for i is zero, since in order for i to produce e_i it must have already consumed d_i so it already knows the utility of that document. The common value is the value that i can expect from trading e_i with other agents. A lower bound for this value is 0, in the case all other agents have also already consumed d_i. An upper bound is given by the sum of the utilities that all other agents assign to that evaluation. Formally, this means that

$$0 \leq U_i(e_i) \leq \sum_{j \in A_{-i}} U_j(e_i). \tag{1}$$

We expect that the actual value will lie somewhere in between since it is likely that some other agents have already consumed d_i, or do not attribute any value to i's evaluations.

$U_i(j)$ represents the utility i expects to get from an evaluation done by j. More formally,

$$U_i(j) = \frac{1}{|E_j|} \sum_{e_j \in E_j} U_i(e_j). \tag{2}$$

Of course, i cannot calculate this value, since its calculation requires knowledge of all of j's evaluations. Instead, the protocol and algorithms presented in Section 2.2.2 will try to approximate this value by using machine learning techniques.

Notice that, while the agent's evaluations are limited to be over the set of documents, this is not a necessity. Specifically, agent i could evaluate agent j and produce $e_i = (j, u_i)$ where $u_i = U_i(j)$. Agent i can then give this evaluation to another agent k. In effect, agents can evaluate each other, and make these evaluations available to others.

2.2.2 The Protocol

The protocol is defined for pairs of agents. One agent asks a question, the other one receives the question and decides how to answer it. The questions are all of the form “Can you give me an evaluation for document d?”.

Before agent i asks for an evaluation, it must first determine who it will question. The agent can ask one or more other agents for evaluations. The decision of whom to ask is made by stochastically choosing agents from two sets.

1. The set of agents that i has information about. That is, those for which $U_i(j)$ exists. The choice among these is made stochastically, giving more weight to the agents with higher values.

2. The set of agents that i does not have any information on.

Choosing between these two sets is the classic exploit vs. explore dilemma (Russell and Norvig, 1995, Chapter 20.5). That is, the agent must decide if it should exploit the information it has, or if it should explore unknown opportunities in the hope that they will lead to even higher payoffs. This decision is made by using the standard solution of exploring a small percentage of the time. That is, i will sometimes ask an unknown agent for an evaluation.

After accumulating a number of evaluations, agent i will decide whether or not to consume the document d. If it decides to consume and evaluate it then i can update its $U_i(j)$ values for all agents j which it asked. This update can be done using reinforcement learning (Kaelbling et al., 1996) (Watkins and Dayan, 1992) (Sutton, 1988). If i does not receive an evaluation from j, it will also update its models of j, using a low utility value that reflects the fact that j never responded.

When agent i receives a request for an evaluation, it must determine whether to honor it or not. Agent i will be more likely to answer the questions posed by those agents $j \in A$ that have higher $U_i(j)$ values. As before, i will also sometimes answer the requests from agents about whom it knows nothing. This technique implements a strategy similar to Tit-for-Tat, which has been shown to be an evolutionary stable strategy which limits the proliferation of parasites (Axelrod, 1984). That is, an agent will tend to deny requests from agents that have ignored the agent’s request. Similarly, an agent will honor requests from other agents that have honored its requests with quality evaluations. Other work (Sen, 1996) has also shown that this type of reciprocity between agents can promote cooperative behavior among self-interested agents.

For example, imagine that a greedy agent joins this system and decides never to answer requests, or answer only with incorrect evaluations. At first this agent might do well because the other agents do not know anything about it. However, as time passes, they will learn that this agent never sends any replies and will ignore the greedy agent’s requests. Since the greedy agent receives no replies, it will eventually leave the system. The existence of an absolute quality is not assumed. That is, each agent is free to have its own opinion about the utility of each document.

The self-enforcing behaviors proposed will lead to the formation of small stable overlapping communities where agents interact more often with other agents that have the same or similar opinions. These communities are expected to last for long periods of time since agents are should maintain the same, or similar, opinions over time. This stability results in a preservation of the neighborhood. Recently in (Cohen et al., 1999) it has been shown that such context-preservation, as it is called by the authors, is a key ingredient in the promotion of cooperation among self-interested agents. Therefore, it is expected that the agents within these groups will have big incentives for cooperating and providing useful evaluations to others.

We must also consider deep-modeling agents. For example, if all agents j think highly of agent i (i.e., $U_j(i)$ is high for all other agents j), then it is likely that i will always get its requests answered. This works fine, as long as i is not aware of its high status. The moment i becomes aware of its position, it can decide to stop honoring as many requests as it used to. The only way for i to find out about its high status is to somehow build models of the other agents. These models quickly become nested agent models to some arbitrary depth (e.g., what i thinks about what j thinks about i). My previous work (Vidal and Durfee, 1998a) has shown that these nested models suffer from decreasing returns, so we can expect agents to only use shallow models.

In summary, given my previous experimentation (Vidal, 1998) and other results from economics (Avery et al., 1999), complex systems (Cohen et al., 1999), and artificial intelligence (Sen, 1996), I feel confident that the protocols described will support the creation of communities of self-interested but cooperating recommender agents. However, I am certain that, as I study it in more depth and share it with others, many problems and shortcomings will be found. But I believe these can be overcome with due diligence. If this project is successful the payoffs will be great and far-reaching— it would give individuals the ability to quickly find useful information, on a specific subject, within a distributed domain. It will also form the first step towards the WWMind vision.

2.2.3 The System Architecture: Using Existing Technologies

During the last year I have studied the available technologies in order to determine the best system architecture. There are several possibilities. We can require each user to run his own small webserver which would allow two-way communication. Another possibility is to build the system on top of one of the currently popular peer-to-peer distribution systems such as Freenet[4] and Gnutella[5]. In fact, the popularity of these systems is a reflection of the need for peer-to-peer communications in the Internet, something that is hampered by the WWW's client-server architecture. As these systems mature, I will continue to explore the possibility of implementing the proposed system using their technology.

One final possibility, which I have investigated in more depth, is to build an applet-based system. This system would consist of a small Java applet that will be downloaded by the user's browser. This applet will implement the agent we have described. The agent could, for example, use the user's bookmarks and assume that the user gives these URLs a high utility.

The agent will engage in the protocol described in Section 2.2.2, asking other agents for URLs they find interesting. The agent will ask the user to evaluate these URLs, i.e., give them a utility value. These evaluations will provide the input for the agent's learning mechanism. A more passive approach would be to constantly monitor the user's bookmarks; if the user ads one of the recommendations as a bookmark then he must give it a high utility value.

The agents will be implemented using the JATLite[6] agent framework which allows browser-based applets to communicate with each other using KQML (Finin et al., 1997). Although using the FIPA ACL[7] is another possibility. The content language will be the Resource Description

[4]See `http://freenet.sourceforge.net`
[5]See `http://gnutella.wego.com`
[6]`http://java.stanford.edu`
[7]`http://www.fipa.org`

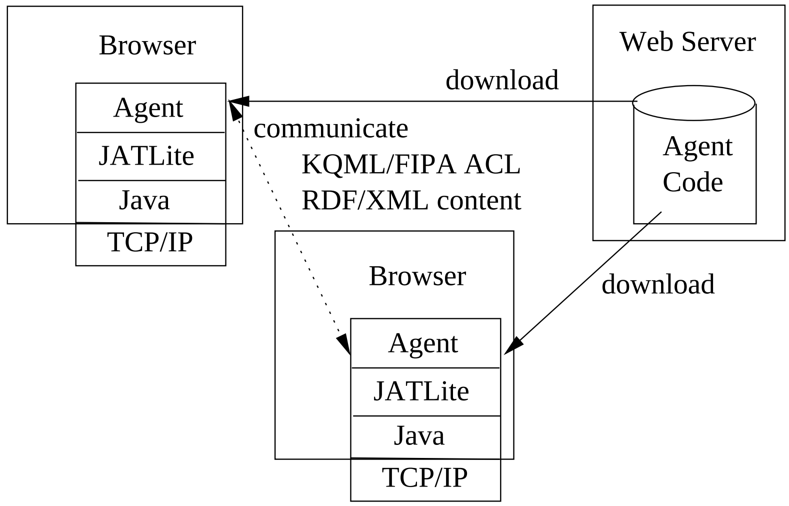

Figure 3: Proposed architecture for URL recommender system.

Framework. RDF is a W3C proposed *"foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information on the Web.*[8]*"*. It uses XML syntax, to which it adds some semantics. As such, RDF will provide the necessary syntax, semantics, and inter-operability with the rest of the WWW.

JATLite provides a way for applets that were downloaded from the same server but now run on different browsers to talk to each other. Under the traditional client/server model implemented by web servers, clients (browsers) can only talk to servers (web). That is, clients cannot talk to other clients. JATLite provides a way to bypass this limitation. JATLite also provides name-server support, which allows a new agent to immediately find other existing agents. A picture of the proposed architecture is shown in Figure 3.

The applets will recommend URLs to the users of the system and request feedback from them. Because of protocol restrictions, the first deployments of the system will have to be limited to communities of users with common interests, for example, researchers in the multiagent systems community, or people interested in the Linux operating system. This initial deployment should provide enough data to further improve the protocols and system. Later deployments might incorporate an ontology and encompass a wider user base.

As a preliminary experiment in using these technologies and understanding user preferences, I have developed, without grant support, a program that transforms user's Netscape bookmarks into a customizable website which incorporates RSS[9] content channels, a cgi-bin search script, and simple recommender-system techniques. The program is called `bk2site`[10] and it is being used extensively across the world.

Development of this program has acquainted me with the technologies necessary for the development and maintenance of the proposed system and it has also taught me some important lessons about the attitudes that people have towards this kind of information. Most `bk2site` users publish a subset of their bookmarks. In email conversations with them it is clear that they

[8] `http://www.w3.org/TR/REC-rdf-syntax/`
[9] `http://my.netscape.com/publish/help/mnn20/`
[10] `http://bk2site.sourceforge.net`

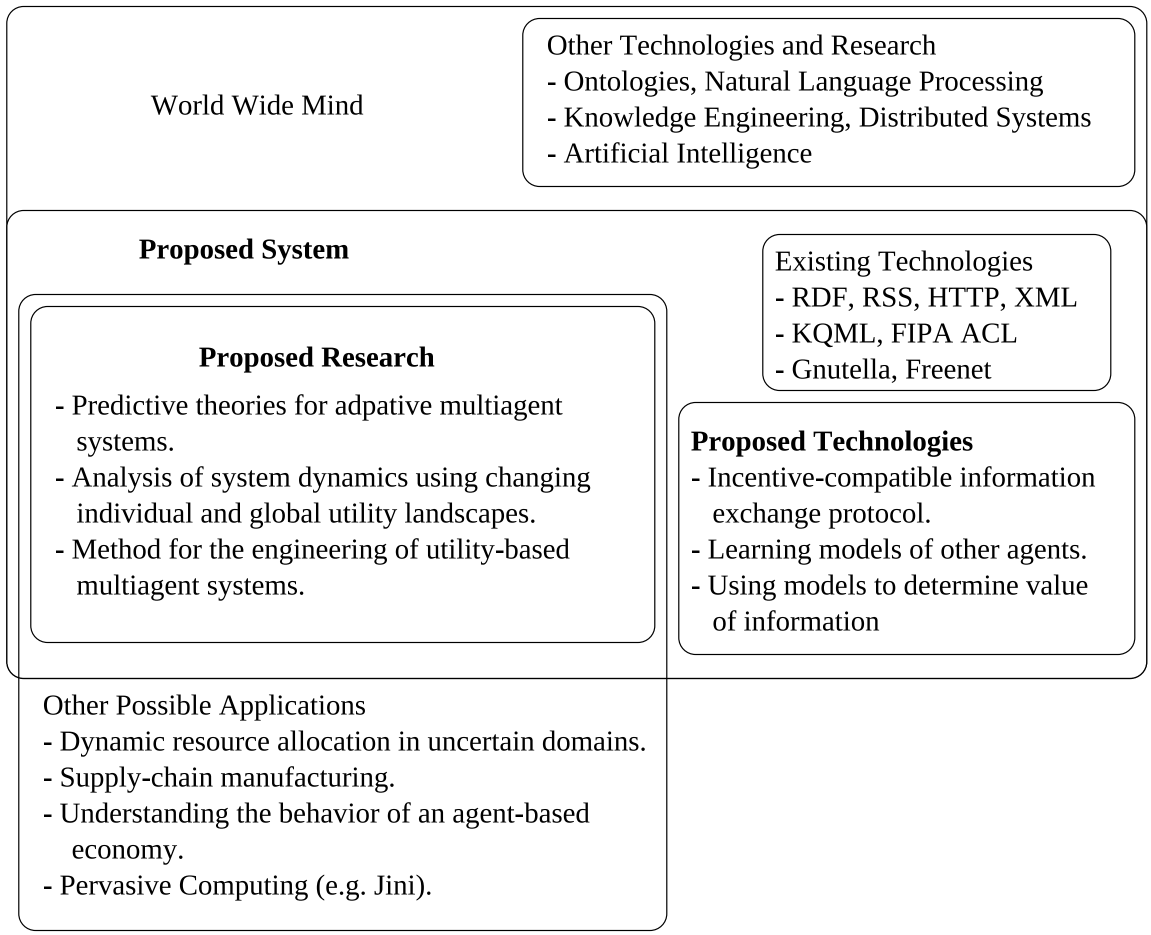

Figure 4: Summary of the proposed research, placing it within the context of long-term goals such as the World Wide Mind. The areas with boldface titles are my proposed research.

welcome the ability to easily publish their own preferences on the web, rather than simply consuming someone else's ideas. Most, however, also have a clear separation between their private information and their public info, a separation they wish to maintain. A few users, like myself at `http://www.multiagent.com` and AgentWeb at `http://agents.umbc.edu`, have set up more polished websites where the knowledge of a community is aggregated.

From these experiences it is clear that many people want to share some of their knowledge, often without any direct reward, as long as their privacy is maintained. `bk2site` enables sharing, in a limited way. The next step is to leverage and organize all these disparate opinions into a system that aggregates and delivers all this knowledge to its users in an as-needed basis—the system I propose to build.

2.2.4 Some Common Questions and Criticisms

Will it scale? Yes. It will scale in terms of the number of agents and documents in the same way that a human community scales. That is, each agent will have a small community of trusted agents. If it cannot find the recommendation it needs within that community it will ask one of the agents to recommend another agent, this agent might recommend another one, and so on. This technique has been already been shown to work (Kautz et al., 1997).

What are your plans for the empirical evaluation of the proposed system? Since the system will integrate new functionality as it is requested by the users and as we learn about the capabilities and limitations of our approach, it is impossible at this time to developed a detailed evaluation plan. Also, for the purpose of this research, I am not as concerned with the usability of the system and how it compares to other products, as I am with the formation of communities of trust among agents and the response from individual users. Therefore, my tests will look at the dynamics of the system, not at GUI and usability issues.

What if people change their opinions? or the network becomes disconnected? or a malicious agent does X? This is a small system which will have to make some "unreasonable" assumptions such as: people do not change their opinions, the network is reliable, and malicious agents only have limited powers of deception. That is, I do am not proposing to build the WWMind. However, I expect the system to still be useful, even if it does not solve all the problems and does not address all possible circumstances.

Isn't the proposed system just a commercial product? I want to make it clear that this is a research project, not a product which might be funded by industry. The theoretical research is clearly only of immediate academic interest. Even the proposed prototype implementation is too high-risk for industry. In fact, even if it were to be completely successful, it only implements an open protocol which allows agents to exchange some limited information in a restricted domain. Similar to the early Internet and the early WWW, this project is simply about enabling entities, in this case agents, to communicate with each other. The only difference is that my agents will speak a more semantically-sophisticated language and will engage in more complex adaptive behaviors.

2.3 Research Plan Summary

I have summarized the proposed research plan in Figure 4. The proposed research is necessary, but not sufficient, for the development of the proposed system and other long-term high-reward visionary undertakings such as the WWMind. In the short-term, my research will have a deep impact on the understanding, engineering, and analysis of adaptive multiagent systems which, given the growing networked world, will only continue to increase in number

References

Avery, C., Resnick, P., and Zeckhauser, R. (1999). The market for evaluations. *The American Economic Review*. to appear.

Axelrod, R. (1997). *The Complexity of Cooperation*, chapter Evolving New Strategies, pages 10–29. Princeton University Press.

Axelrod, R. M. (1984). *The Evolution of Cooperation*. Basic Books.

Baker, A. D., Parunak, H. V., and Erol, K. (1999). Agents and the internet: Infrastructure for mass customization. *IEEE Internet Computing*, 3(5):62–69.

Brooks, M. (2000). Global brain. *New Scientist*.

Cohen, M. D., Riolo, R. L., and Axelrod, R. (1999). The emergence of social organization in the prisoner's dilemma: How context-preservation and other factors promote cooperation. Technical report, The Santa Fe Institute. Santa Fe Report: 99-01-002.

Finin, T., Labrou, Y., , and Mayfield, J. (1997). KQML as an agent communication language. In Bradshaw, J. M., editor, *Software Agents*, chapter 14. The MIT Press.

Fowler, M., Scott, K., and Booch, G. (1999). *UML Distilled, Second Edition: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns : Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Heylighen, F. (1999). Collective intelligence and its implementation on the web: algorithms to develop a collective mental map. *Computational and Mathematical Organization Theory*, 5(3):253–280.

Heylighen, F. and Bollen, J. (1996). The world-wide web as a super-brain: from metaphor to model. In Trappl, R., editor, *Cybernetics and Systems*.

Holland, J. H. (1995). *Hidden Order*. Addison-Wesley.

Huhns, M. N. (2000). Agent teams: Building and implementing software. *IEEE Internet Computing*, 4(1).

Jennings, N. R. (2000). On agent-based software engineering. *Artificial Intelligence*, 117(2):277–296.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

Kauffman, S. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Pres.

Kauffman, S. (1995). *At Home in the Universe*. Oxford University Press.

Kautz, H., Selman, B., and Shah, M. (1997). Referral web: combining social networks and collaborative filtering. *Communication of the ACM*, 40(3):63–65.

Kephart, J. O., Hanson, J. E., and Greenwald, A. R. (2000). Dynamic pricing by software agents. *Computer Networks*, 32(6):731–752.

Petrie, C. (2000). Agent-based software engineering. In *Proceedings of the Practical Application of Intelligent Agents and Multi-Agents*.

Resnick, M. (1994). *Turtles, Termites and Traffic Jams*. The MIT Press.

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Sen, S. (1996). Reciprocity: a foundational principle for promoting cooperative behavior among self-interested agents. In *Proceedings of the Second International Conference on Multiagent Systems*, pages 322–329, Menlo Park, CA. AAAI Press.

Smith, R. G. (1981). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44.

Vidal, J. M. (1998). *Computational Agents That Learn About Agents: Algorithms for Their Design and a Predictive Theory of Their Behavior*. PhD thesis, University of Michigan.

Vidal, J. M. and Durfee, E. H. (1998a). Learning nested models in an information economy. *Journal of Experimental and Theoretical Artificial Intelligence*, 10(3):291–308.

Vidal, J. M. and Durfee, E. H. (1998b). The moving target function problem in multi-agent learning. In *Proceedings of the Third International Conference on Multi-Agent Systems*.

Vidal, J. M. and Durfee, E. H. (2001). Predicting the expected behavior of agents that learn about agents: the CLRI framework. Technical report. A revised version will appear in Autonomous Agents and Multi-Agent Systems Journal.

Vidal, J. M., Mullen, T., Weinstein, P., and Durfee, E. H. (1998). The UMDL service market society. In *Proceedings of the Second International Conference on Autonomous Agents*.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8:279–292.

Wolpert, D. H. and Tumer, K. (1999). An introduction to collective intelligence. Technical report, ACM Computing Research Repository. cs.LG/9908014.