
An Agency-Based Framework for Electronic Business

Larry Kerschberg and Sonali Banerjee

Center for Information Systems Integration and Evolution,
Department of Information and Software Engineering,
School of Information Technology and Engineering
George Mason University, MSN4A4, Fairfax, Virginia, 22030-4444, USA

Abstract. This paper presents an overview of current trends in Electronic Business (E-business), and how an enterprise can use the Electronic Marketplace for strategic advantage. The role of cooperative information agents is discussed within the context of E-business. An agency-based framework is presented for E-business in the class of logistics and supply systems. Such systems are enjoying wide acceptance in commercial and military application domains, as witnessed by the phenomenal growth in Internet stocks for companies such as Amazon.com. We also show how agent-based systems provide scalable, configurable and evolutionary solutions for such applications.

1. Introduction

The rapid strides made in computer science and information technology have enabled business organizations to reach new customers and open new markets in an Electronic Marketplace. The Internet and World Wide Web facilitate searching for and sharing information, sharing crucial business data, and building better business-to-business relationships.

Electronic commerce is a more cost-effective way of doing business than building new stores or warehouses and hiring more people to run them. It has streamlined processes, simplified sharing of critical data and lowered organizational barriers—reaching beyond a company’s walls and into the operations of its customers, suppliers and business partners. Shopping in an interactive world provides more options, customized goods and services based on user preferences, a record of previous activity as well as customer-based feedback, permits customers to “try before they buy,” and even involves them in product design and delivery.

Internet technology can deliver what the customer wants - timeliness, accuracy and lower cost. Customers previously had to call a customer service representative, while now orders can be taken and tracked online, and a customer's credit status can be verified online. This provides a tighter *integration* between consumers and producers, thereby eliminating the traditional middleman, and replacing him by the *infomediary*, as witnessed by "business portals" such as Yahoo!

The Internet provides a new medium and mechanism to establish business associations, collaborations, and alliances. One might say that the Internet and Web are about *leveraging relationships for strategic advantage*, as witnessed by the recent mergers and acquisitions, such as AOL, Netscape and Sun Microsystems, @Home and Excite, Yahoo!, Geocities, and Broadcast.com, AT&T, TCI, and MediaOne, etc. This new medium, called the Electronic Marketplace, enables an entirely new range of entrepreneurial possibilities, with future enterprises providing valued-added cooperative, end-to-end virtual and real services.

Cooperative information agents will play a major role in the Electronic Marketplace. We now discuss some possible roles for such agents. Subscriptions to information sources such as online investing, auctions, and brokerage services, allow people (or agents) of similar interests to meet, discuss, provide recommendations and suggestions, negotiate and conduct business. Moreover, monitoring services provide alerts, based on user-specified critical events, conditions, or new information and knowledge, pertinent to a user's decision-making criteria. Users get immediate feedback and triggered alerts, so that the end-to-end experience is perceived to be of higher quality, because a *relationship of trust* is being established. The quality of the product, commodity, or service, as well as the information, knowledge, communication, delivery mechanisms is perceived to be of high quality because of the ongoing interactive and evolutionary nature and mutual nurturing of the customer-to-business or business-to-business relationship. Quality of service issues will play a large role in creating "brands" in Electronic Business.

Electronic business, or E-business, has brought about changes in marketing strategies. E-business is a new vehicle for building relationships of mutual benefit between businesses, with their customers, and their suppliers. Information can be easily sifted, facts checked immediately and third party evaluations obtained rapidly. This brings to the forefront the quality of goods and services provided. Brands can be established or destroyed very quickly, and this is less biased by the size of the business or the amount of money it can invest on advertisement and marketing. Marketing itself is more narrowly targeted with highly customized products and services, as for example the customer profiles maintained by Amazon.com to recommend new books to returning customers. Continuing with the Amazon.com example, customers register their payment and delivery preferences, thereby enabling "One-Click Shopping", whereby the shopping basket is checked out, order tracking number assigned, and periodic status reports sent (the date the order was filled and the date it left the warehouse) via email. The Electronic Marketplace raises customer expectations regarding quality-of-service issues; issues such as the quality of the customer shopping experience, product and service quality, web site performance, and content management, will be receiving increasing attention.

Today E-business is having an impact on the way companies do business. These impacts involve the re-engineering of existing business processes, such as marketing, order processing, billing, payment, inventory and delivery, to support distributed solutions across the enterprise and throughout the entire supply chain. E-business enables not only business transactions, but also information exchange. E-Business has also raised business-to-business commerce to unprecedented levels. In 1998, Cisco Systems transacted about US\$5 Billion of business through its Web site, and Boeing provides seven-days-a-week, 24-hours-a-day parts ordering for the world's airlines through its site.

In general, an E-business framework has a number of major functions or components:

Day to Day Operations

- interactive business and financial transactions including e-forms, email and electronic faxes;
- electronic cataloguing;
- order placing;
- order fulfillment and shipping;
- logistics and electronic order-tracking services;
- automatic invoicing and payment services;
- electronic funds transfer; and
- the exchange of multimedia information and product designs, such as CAD/CAM drawings, photos, and schematics.

Long Term Operations

- vendor registration and contracting;
- demand forecasting and order scheduling;
- advertising of products and services;
- data mining of consumer information and transactions; and
- customer profiling and customization of advertisements to have a more effective impact on prospective customers.

A fully-automated demand-driven inventory management and just-in-time delivery system can be achieved by integrating these components with guaranteed overnight (or on-time) delivery. The advent of the Internet, its popularity and accessibility, has increased radically the number of organizations doing business electronically and has brought about increased competition and made global commerce a reality.

Today virtually every major information technology supplier is involved in electronic business. Apple, Boeing, IBM, HP, Cisco, Compaq, Microsoft, Oracle, SAP, and Sun Microsystems are some of the giant E-business facilitators that offer a wide range of products and services. The advent of the service-based approach to E-business promotes the formation of *virtual enterprises*, capable of providing specialized services within a broader end-to-end virtual and real environment. An example is the use of overnight and guaranteed delivery services such as FedEx and UPS to ship products from warehouses to customers.

A more striking shift is occurring at the physical level of enterprise cooperation. Recently Dell Computer, Inc. announced plans to build a factory employing 3,000 people, adjacent to the airport at Nashville, Tennessee, home of FedEx's hub. Clearly, this *strategic* decision by Dell to co-locate with FedEx so as to be closer to a world-class overnight delivery service, indicates the need to reduce the time between when an on-line order is taken and the time the computer is shipped to the customer. In addition, FedEx is expanding its reach deep into the supply and manufacturing chains of companies like Fujitsu and Hewlett-Packard, for whom it is now assembling computers from parts delivered just-in-time to FedEx's Nashville facility, and then shipping them to customers. FedEx is also handling computer repair for these companies.

In this paper we present an agency-based architecture for an Electronic Marketplace for logistic applications, and more specifically, study the role of intelligent software agents in facilitation, negotiation, brokerage, monitoring, alerting and performance verification, for a federation of customers, producers, catalog curators, and contracting agents. This framework and architecture is modular, scalable, and should foster the creation of virtual enterprises, in this and similar application domains. Section 2 examines the role of agents in electronic business. Section 3 discusses agent architectures and compares single versus multiagent schemes. Section 4 depicts our agency-based architecture for a class of logistics applications. Section 5 presents conclusions and suggestions for future research. Appendix 1 presents a table describing the many commercial, research and prototype software for agent specification, configuration and deployment.

2. The Role of Agents in Electronic Business

An agent is a program regulating a certain aspect of its environment. Autonomous agents are self-starting mechanisms with specific goals [28]. An intelligent agent is an attempt to model a software agent [23] after human cognitive behavior.

Agents are useful in electronic business because they are goal-driven adaptive problem solvers, and therefore act as proactive 'catalysts'. They can make contact, establish connections, negotiate and facilitate interactions. They can disseminate information to a wide audience or automatically target information to those with special interests [43]. Intelligent autonomous software agents may be used to filter information [39], integrate information from heterogeneous sources [30, 31], automate stereotypical behavior and regulate transactions for efficient business interactions [27]. The use of agents removes the need of having the user initiate all tasks explicitly and to monitor all events.

An agent could provide a common interface for the entire business process. Agents can be configured automatically with appropriate knowledge bases (ontologies), task-specific information, negotiation and communication protocols for its mission. The advantage of using agents for E-business is that they are scalable, modular, robust, composable and configurable. This enables the application system requirements for a particular enterprise to be built (and altered) fairly rapidly. The agent system would provide the infrastructure needed for a seamless connection of enterprises and would serve as an "engine" or "shell" for transactions between databases involved in the different commerce applications.

Using agents for E-business increases productivity by allowing processes to be done concurrently and autonomously. The increased throughput becomes critical to performance with greater proliferation of business on the Internet. Furthermore, by modeling software agents to behave like human agents, consumer resistance to change from traditional to virtual environments is likely to be overcome.

Most enterprises have legacy systems and also access information on the Internet. E-business requires access to diverse data residing in multiple, autonomous, heterogeneous databases and the integration of that data into coherent information that can be used by the enterprise. The problem is complicated by the fact that the data may be multimedia with diverse temporal and spatial granularity [58, 59, 63, 64] and stored in diverse formats [62]. The semantics of the data may conflict across multiple sources and the data may be of uncertain quality, and the reliability of the source may be questionable. Cooperative information agents can play a major role in the intelligent integration of information and its associated problems mentioned above.

3. Agent Architectures

In this section we discuss several agent architectures, and discuss their pros and cons vis-a-vis E-business. First we examine single-agent systems, then multiagent systems. We introduce *agencies* of agents. Also examined are federated architectures and research issues of agent coordination and communication.

3.1 SINGLE AGENT SYSTEM

An agent system may consist of a single agent or multiple agents. To many people a single agent approach is more intuitive than multiagent approaches. Although it might seem that single agent systems should be simpler than multiagent systems, the opposite is in fact true. Distributing control among multiple agents allows each agent to be simpler. No one agent has to be able to complete a given task on its own. A single agent could very well turn out to be more complicated than a multiagent system if its tasks are complex. Single agent systems should be reserved for domains that require centralized control. To perform more complex tasks, agents may be organized as part of a community of cooperative problem solvers.

Multiagent systems have several agents who model each other's goals and actions. There may be direct interaction among agents in a multiagent environment. This interaction could be viewed as environmental stimuli or may be direct agent interaction (communication) separate from the environment. From an agent's perspective, multiagent systems differ from single agent systems most significantly in that other agents can determine the environment's dynamics. In addition to the uncertainty that may be inherent in the domain, other agents affect the environment in unpredictable ways.

3.2 MULTIAGENT SYSTEMS

A single agent to model E-business transactions would be very complicated and would unnecessarily burden the agent. Having multiple agents could speed up a system's operation by providing a method for concurrent processing. Since the domain is easily decomposed into components—several independent tasks that can be handled by separate agents—it could benefit from a multiagent architecture. Furthermore, the parallelism can help deal with limitations imposed by time-bounded reasoning requirements.

While parallelism is achieved by assigning different tasks or abilities to different agents, robustness is a benefit of multiagent systems that have redundant agents.

If control and responsibilities are sufficiently shared among different agents, the system can tolerate failures by one or more of the agents. Another reason for choosing a multiagent system architecture is scalability. Since they are inherently modular, it is much easier to add new agents to a multiagent system than it is to add new capabilities to a monolithic single agent system. Systems, whose capabilities and parameters are likely to change over time, or across agents, can also benefit.

From a programmer's perspective the modularity of multiagent systems can lead to simpler programming. Rather than tackling the entire task with a centralized agent, programmers can identify subtasks and assign control of those subtasks to different agents. The problem of splitting a single agent's time among different parts of a task is also solved. We conclude therefore that a multiagent architecture is more amenable to the needs and requirements of the E-business domain.

There are several forms of organizations of multiple agents, namely:

- Homogeneous non-communicating agents.
- Heterogeneous non-communicating agents.
- Communicating agents with any degree of heterogeneity.

Each multiagent organization introduces new issues and complications. If a task can be accomplished with non-communicating agents, then adding communication only makes the system more complicated. The type of multiagent system to be used depends on the characteristics of the domain. The simplest possible system that is effective within the domain should be used. Relevant domain characteristics include: the number of agents; the amount of time that may be allocated; whether or not new goals arrive dynamically; the cost of communication; the cost of failure; user involvement; and environmental uncertainty. Decker distinguishes three different sources of uncertainty in a domain [17]: the transitions in the domain itself might be non-deterministic; agents might not know the actions of other agents; and agents might not know the outcomes of their own actions.

Multiagent Homogeneous System without Communication. The simplest multiagent system consists of homogeneous non-communicating agents. All agents have the same internal structure including goals, domain knowledge, and possible actions. They also have the same procedure for selecting among their actions. The only differences among agents are their inputs, the actual actions they take and possibly their location. Although the agents have identical capabilities and decision procedures, they may have limited information about each other's internal state and sensory inputs. Thus they may not be able to predict each other's actions. Several information retrieval and information filtering agents (Meta-crawler, Copernic, etc.) have this kind of architecture.

Multiagent Heterogeneous System without Communication. Heterogeneous agents in a multiagent domain add a great deal of potential power, but at the price of added complexity. Agents might be heterogeneous in any of a number of ways, from having different goals to having different domain models and actions. An important aspect of

heterogeneous agent systems is whether agents are *benevolent* or *competitive*. Even if they have different goals, they may be co-operative towards each other, or they may actively attempt to inhibit each other. An agent can serve its own self-interests by establishing a reputation for being cooperative. When working on a task with another cooperative agent, the two can benefit from a sense of trust for each other. Cooperation is also an issue when certain resources are required simultaneously by the competing agents. Designers of multiagent systems having limited resources must decide how the agents will share those resources. In another paper in these proceedings, we present novel ways that agencies can negotiate for constrained resources in an optimal fashion.

With heterogeneous agents, the problem of modeling others is very complex. The goals, actions, and domain knowledge of the other agents may be unknown and thus will need to be modeled. Without communication, agents are forced to model each other strictly through observation.

Multiagent Heterogeneous System with Communication. The full power of a multi-agent system can be realized when agents are given the ability to communicate with one another. With the aid of communication, agents can coordinate much more effectively. Communication also helps with resource allocation bottlenecks by allowing the heterogeneous agents to post constraints to each other's blackboards and schedulers, thus allowing them to coordinate without the aid of a centralized agent.

When agents have similar goals, but different abilities, they can be organized into a team or "agency." Each agent then plays a separate role within the team. With such a benevolent team of agents that are heterogeneous with respect to their abilities, one must provide some method for assigning different agents to different roles. This assignment might be obvious if the agents are very specialized.

Since e-business benefits from software agents that are specialized to perform particular tasks, cooperation is required for successful of the job at hand. A heterogeneous multi-agent system is therefore, probably the best architecture for e-business. The different agents in the system may be organized into "agencies" in order to accomplish complex tasks. The agencies may be established ones that are organized along functional lines, or they may be formed dynamically.

Agency Organization. There are several possible organizational structures for heterogeneous multiagent systems such as:

- Uncoordinated teams _ where a matchmaker is queried about potential agents that can perform a task and then the request is sent to an agent selected at random.
- Economic markets _ where agents bid for tasks based on some pricing function that may include its current load, resources required in order to perform the task, and are selected based on price, reliability and other utility characteristics.
- Federations _ where agents give up individual autonomy to a centralized facilitator or broker that handles requests.
- Dynamic Teams _ where agents form teams based on some input parameter for the duration of a single task.

This is not an exhaustive list. Studies in organizational theory indicate that no one structure can be said to be "best" for all organizations. The structure must adapt to the task

environment based on the uncertainty associated with input and output measurements, causal relationships in the environment, time lapse between decision and feedback, etc.

3.3 FEDERATED ARCHITECTURES

A federated approach using cooperating agents rather than a monolithic system is more amenable to the needs of both information integration and electronic business [30, 31]. Agents are mobile, and can act autonomously, communicate, cooperate, adapt and reason based on rules and knowledge [24]. Thus, a system of trustworthy cooperating information agents, each having the requisite knowledge to perform its tasks within the larger problem-solving framework [17, 18, 42], would provide a flexible architecture for E-business. The approach proposed in this paper is to build a federation of configurable intelligent agents, which provide services, such as catalog access, consumer report information, product query formulation, facilitation, brokerage, negotiation, contracting, mediation, information integration and source wrapping, to enable the enterprise to realize its E-business goals. The agents in a federation may cooperate with each other and function as an agency, with agents being autonomous but sometimes surrendering autonomy in order to collaborate efficiently.

In an environment where we have multiple agents concurrently performing multiple tasks, we require a mechanism by which we can regulate the sequence in which tasks are performed so as to optimize productivity. This requires us to determine how to model an agent's activities and how to detect dependencies between those activities.

3.4 RESEARCH ISSUES IN MULTIAGENT SYSTEMS

Problem-solving should be a cooperative endeavor in Multiagent Systems [7, 41]. Multiple copies of knowledge and data, possibly in differing formats, may have to be maintained by the system. There are several facets to the heterogeneity of information in systems: syntactic, control and semantic. Syntactic heterogeneity refers to the myriad of knowledge representation formats, data definition formats to represent both knowledge and data. Control heterogeneity arises from the many reasoning mechanisms for intelligent systems including induction, deduction, analogy, case-based reasoning, etc. Semantic heterogeneity arises from disagreement on the meaning, interpretation and intended use of related knowledge and data. The advantage of different knowledge representations and information about the same thing being represented in different ways in the knowledge database is that such alternative views are helpful in creativity while syntactic differences can be resolved by translation agents given the computational equivalence of the representation formalisms.

Agent Coordination. A crucial problem in agent-based systems [6, 32] is to coordinate autonomous cooperating agents that are working towards common and/or individual, and possibly conflicting goals [10, 15, 17, 28, 29, 44]. In multiagent environments, the need to coordinate agents may arise from several reasons [28]: 1) to manage dependencies among agents' actions; 2) to adhere to and maintain global constraints; and 3) to facilitate problem solving in cases where no one agent has sufficient competence, resources or information to solve the entire problem. Coordination is the process of managing dependencies between activities in order to avoid conflicts while having maximum concurrency [8, 9, 17, 28, 29, 44, 46, 45]. Coordination involves task decomposition, resource allocation, synchronization, group decision making, communication and the preparation of common objectives.

The major coordination issues are: distributed planning [20], coalition formation [33-36], negotiation mechanisms [3, 7, 16, 37, 47] and the protocols [7, 56] by which agents can coordinate and manage commitments [28]. There are two approaches to coordination: implicit and explicit coordination [8, 9, 21, 55].

Agent coordination and control [5, 19, 24, 25, 38, 42] may be accomplished through:

1. mediated means using planning algorithms and load balancing [13, 14, 48],
2. reactive means based on the perception of the various autonomous agents [11, 12],
3. proactive means by the spontaneous formation of coalitions with the autonomous agents themselves taking the initiative [33-36, 41], or
4. competitive means which are market based [60].

Controls may be enforced through role-based access control mechanisms [1, 2, 50-52], by task reassignment, agent adaptation and evolution [18, 22, 26, 40, 49], or agent termination. These decisions may be made by monitoring resource consumption, by means of a cost/benefit optimization model [4, 57], or may be based on performance criteria.

Agent Communication. With the aid of communication, agents can freely exchange information in order to help them coordinate much more effectively. Communication costs arise due to limited bandwidth, the consumption of reasoning time, and the time required for the communicative acts. There is a tradeoff between lower communication costs and better decisions. With communicating agents, systems can become arbitrarily complex and highly controlled. Of course this may cause communication bottlenecks and performance may be prohibitively low.

From a practical point of view, the communication might be broadcast or posted on a “blackboard” for all to interpret, or it might be targeted point-to-point from one agent to another specific agent.

In all communicating multiagent systems, and particularly in domains that include agents built by different designers, there must be some set language and protocol for the agents to use when interacting. Independent aspects of protocols are information content, message format, and coordination conventions. Among many others, existing language protocols for these three levels are: KIF for content [23], KQML for message format and COOL for coordination. There has been a lot of research done on refining these and other communication protocols.

Communication capability of agents may be considered as an “action” or speech acts [13]. When an agent transmits information to another agent, it has an effect just like any other action would have. Thus, within a planning framework, one can define preconditions and post-conditions (effects) for communicative acts. When combined with a model of other agents, the effect of a communication act might be to alter an agent’s belief about the state of another agent or agents. Agents can learn to choose among a set of social behaviors that include broadcasting and listening.

When agents communicate, they may decide to cooperate on a given task or for a given amount of time. Such commitments to each other involves agreeing to pursue a given goal, possibly in a given manner, regardless of the cost to the agent. Commitments can

make systems run much more smoothly by providing a way for agents to “trust” each other. Therefore, plans can be formed which depend on the timely completion of specific tasks by different agents that are specialized in the performance of the given task.

4. The Agency-Based Electronic Marketplace

We propose an agency-based architecture for the Electronic Marketplace. The architecture consists of a collection of cooperating multi-layered agencies, each consisting of cooperating agents. This system is designed to eliminate the need for large warehouses of food, clothing and pharmaceuticals for the different supply sites of the supply chain organization, and rather to have selected prime vendors bid on orders, negotiate special terms and deliver goods directly to customers. The enterprise has both in-house transportation facilities for routine/bulk transportation between physical stores as well as negotiated contracts with external commercial transportation services for rapid transportation of specific products to individual customers. The aim here is to maintain optimal enterprise-wide inventory levels.

Since customer satisfaction has strategic implications, it is very important to closely monitor transactions. In order to ensure that customers do not go elsewhere because of inventory shortages, it becomes necessary to track the supply of critical items, and to be able to take necessary corrective action in a timely fashion. This is done automatically by the deployment of special agents, called sentinels that monitor the system, inform logisticians and decision-makers, and trigger alternative contingency plans to take care of a possible crisis [4, 53, 54, 61]. Figure 1 provides an overview of the Electronic Marketplace architecture.

Each agent-based subsystem, called an *agency*, may be considered a federation of agents, each with specific goals and a specific functional role in the Marketplace. Therefore, each agency can be considered as a collection of cooperating autonomous agents with particular expertise. The following discussion outlines the constituent agents and the role they play.

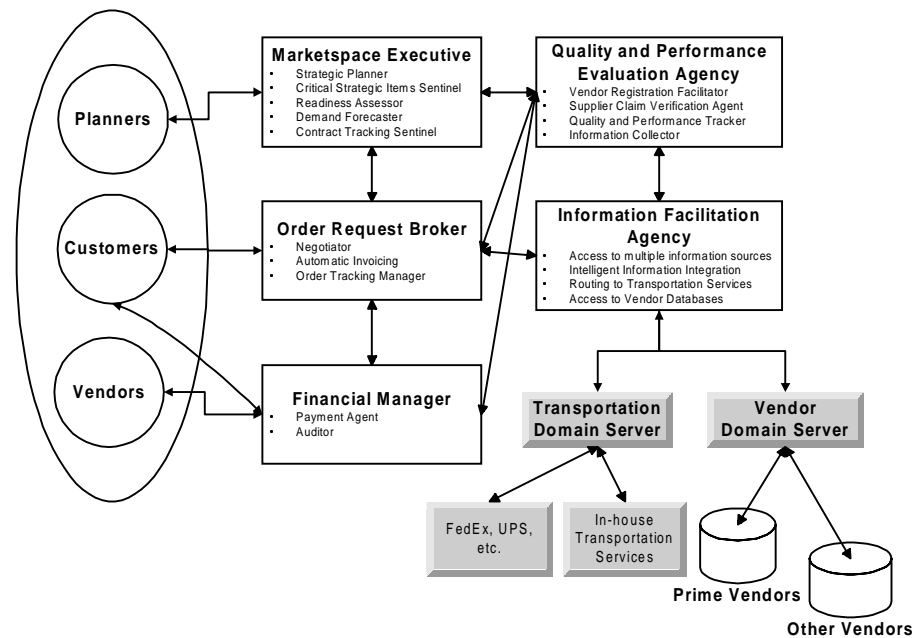


FIGURE 1. Agency Architecture for the Electronic Marketplace

4.1 MARKETSPACE EXECUTIVE

This agency handles planning and monitoring of the Marketplace. It ensures that probable crisis situations are detected early and avoided by taking suitable action. This also acts as an interface for the organization's planners and strategists who are aware of the long term goals, expectations and strategies.

Strategic Planner. Strategic planning is performed within the interface Marketplace Executive agency. It allows planners to: provide domain knowledge, define long term plans and system objectives, define rules and constraints, identify critical items and situations to be monitored, define the cause-effect relationships, establish rules and courses of action for the sentinels, and obtain reports on all aspects of the system at any time. The desired features of the interface would be to provide heuristic support for all information searching scenarios and providing suggestions for building better queries.

Critical Strategic Items Sentinel. This agent is actually a collection of sentinels operating concurrently, continuously monitor the state of the system, tracking the critical and strategic items.

Readiness Assessor Sentinel. This agent ensures that strategic plans can be fulfilled by the in-house and externally contracted resources. It monitors the activities within plans,

resource utilization, and workflow, ensuring that crisis situations are avoided and contingency plans are in place for critical situations.

Demand Forecaster. This agent breaks down plans into resources and materiel that need to be provided. It also builds the contract terms that will be necessary.

Contract Tracking Sentinel. This agent is a demon that monitors the system to alert against any deviation from the contracts made so that necessary alerts are flagged and triggers are fired. This allows the system to detect any deviation from plans already in place so that necessary action can be taken in a timely fashion to ensure crisis control.

4.2 ORDER REQUEST BROKER

This agency handles the day-to-day functions of the Marketplace. It ensures that transactions run smoothly without interruptions, and takes corrective action to handle inconsistencies. This also acts as an interface for the organization's operators who perform the individual transactions by providing optimized customer order - broker mapping, suggestion on building better queries to the user, improving overall order fulfillment and delivery turn around time, and by allowing the creation of personalized user profiles. This agency may be considered to have the following agents:

Negotiator. This agent would not only provide optimized broker-supplier associations, but also would negotiate with a registered supplier to issue a contract in accordance with the specifications and terms specified by the customer.

Automatic Invoicing. This agent is the supply receiver and payment requester. It consults a database containing records of contracts made and supplies obtained against these contracts. It acknowledges the receipt of merchandise based on UPC (bar) code scans and a contract identification code that indicates the partial or complete fulfillment of a particular contract. Upon receipt of merchandise it sends a request for payment to the appropriate agent in the financial broker.

Order Tracking Manager. This agent takes over the contract after the negotiator has completed negotiations, tracks progress made, and maintains contact with the supplier to ensure that the contract will be fulfilled according to schedule. In case of any problem indicated by the supplier, this agent sends the pertinent information to the sentinel tracking contract terms in the Marketplace executive and the quality and performance tracker in the quality and performance evaluation agent.

4.3 FINANCIAL BROKER

An issue vital to the electronic commerce architecture is a secure payment scheme. There are a number of institutions and organizations involved in research to develop not only a secure form of payment, but also to make the Internet a secure place for business transactions. Some of the major institutions include CyberCash, DigiCash, and First Virtual Bank. Besides the security issues that need to be addressed, a viable interface has to be developed to make payments for purchases with a pay-on-delivery paradigm in mind. This would involve automated invoicing, electronic transfer of funds and auditing of transactions. The financial broker is the agency that handles the financial transactions of the organization. It consists of the following agents:

Payment Agent. This agent removes the need for invoices to be prepared and approved. When the automatic invoicing agent acknowledges the receipt of merchandise corresponding to a particular contract, funds are automatically transferred to the supplier account and a record of the financial transaction is sent to the auditor. Depending on the contract terms, the payment may be made incrementally as items are delivered or deferred until the order is completed.

Auditor. This agent maintains records of all the financial transactions and performs audits.

4.4 QUALITY AND PERFORMANCE EVALUATION AGENCY

This agency provides not only the “yellow pages” of registered vendors of the organization, but also tracks performance of the vendors, builds a level of confidence in different suppliers, verifies supplier claims and obtains referrals. The agency consists of the following agents:

Vendor Registration Facilitator. This agent enables new vendors to make their services available to the client. The agent serves as a broker and establishes the logistics between the vendor and the customer.

Supplier Claim Verification Agent. The services that this agent should provide include the verification of supplier authenticity by obtaining and checking references. Small contracts in non-critical areas may be recommended (to the negotiator in the order request broker) by this agent in order to build trust in a newly registered vendor.

Quality and Performance Tracker. This agent provides services that include: monitoring the performance of suppliers on their extant contracts, assessing the end-to-end performance of the supply chain, and updating the knowledge stored in the meta-database. Information is collected on user preferences, vendor performance, and services rendered. Data mining techniques can be used to extract relationships between the orders, suppliers, clients and information obtained from the order request broker. This information can be used to provide evaluations of vendor performance, which, in turn, can be used by the negotiator in the order request broker when selecting a vendor for a new contract.

4.5 INFORMATION FACILITATION AGENCY

This agency handles access to data in enterprise-wide information sources, facilitates searching these databases, and accesses external Web-based data sources. Requests for special transportation services and vendor information are handled by this agency. In addition, information integration, fusion and filtering are provided by this agency.

4.6 INFORMATION INTERCHANGE AMONG AGENTS

Information required by the different agents in order for them to perform correctly comes from various sources. The information exchange takes place by sending mes-

sages from one agent to another. The flow of information is critical to the agent performance. In table 1 we summarize the information flow among agents.

Table 1: Information Interchange among Agents

Source of Information	Information Description	Receiver of Information
Customer, Strategic planner	Order info (requisitions, priorities, etc.)	Marketspace Executive Negotiator, Supplier Claim Verification Agent
Customer	Order ratification	Negotiator
Supplier Claim Verification Agent, Vendor Database – Established contracts	Supply Sources	Readiness Assessor, Strategic Planner, User
Readiness Assessor	Recommended alternative sources of supply prioritized to meet user order criteria.	Strategic Planner
Vendor Database, Vendor Meta-Database	Historic contract and vendor performance information	Strategic planner, Demand Forecaster
Order Request Broker	Contract and vendor performance information	Strategic Planner, Quality and Performance Tracker, Vendor database
Vendor Database	Historic vendor performance measures and risks. Vendor's expected contract performance parameters.	Readiness Assessor
Quality & Performance Tracker	Dynamic, real-time vendor performance estimates as well as uncertainty /confidence levels, Tradeoff relationships between vendor cost/price and performance	Readiness Assessor
Demand Forecaster	Demand data over the course of the planning scenario, Demand priorities established	Strategic planner
Strategic Planner Negotiator Readiness Assessor	Customer ordering criteria Real-time requisitions Subset of established business relationships that meet legal and regulatory conditions	Negotiator

Negotiator	Running contracts	Order Tracking Manager, Quality and Performance Tracker, Automatic Invoicing
Automatic Invoicing	Partial fulfillment of contract	Payment Agent Quality and Performance Tracker, Order tracking Manager
Demand Forecaster	Estimated risk of completing the transaction	Strategic Planner
Readiness Assessor	Best supply support alternative for demand, Prioritized list of alternative supply support, Measure of fit with customer ordering criteria	Strategic Planner
Order Tracking Manager	Deviation from schedule, Delays in delivery, Incomplete fulfillment of contract	Quality & Performance Tracker
Payment Agent	Record of funds transferred	Auditor
Vendor	Electronic Catalog Payment methods	Vendor Registration Facilitator
Supplier Claim Verification Agent	New supplier information	Readiness Assessor Vendor Database
Readiness Assessor Strategic Planner	Information about unconventional requirements	Information Facilitation Agency

5. Conclusions and Future Research

This paper has presented our ongoing research on the role of cooperating information agents in E-business. We have presented some of the impacts and benefits of moving the enterprise into an E-Business framework. The issues of agent coordination, control and evolution have been discussed within a multiagent systems framework. Much research needs to be done in the areas of agent communication, workflow, condition monitoring via sentinels, and agent negotiation to make such an architecture a reality.

We have presented an agency-based architecture for an Electronic Marketspace which is appropriate for the class of logistics and supply chain systems. By identifying the components, or agencies, of such systems, we hope to focus on the types of agents needed within each agency, and then examine negotiation, coordination and control mecha-

nisms, within and among agencies, needed to realize enterprise-wide goals and objectives. We are presently beginning to construct a software prototype of the agency concept.

Acknowledgements

This research was sponsored in part by the Defense Advanced Research Projects Agency, Advanced Logistics Program, under contract number N00600-96-D-3202.

References

1. Bertino, E., Ferrari, E., Atluri, V.: A flexible model for the specification and enforcement of role-based authorizations in workflow management systems. 2nd ACM Workshop on Role-Based Access Control. Fairfax, VA (1997)
2. Biddle, B. J.: Role theory: concepts and research. Kreiger Publishing (1979)
3. Boehm, B., Bose, P., Horowitz, E., Le, M.J.: Software requirements negotiation and renegotiation aids: a theory-W based spiral approach. International Conference on Software Engineering (1995)
4. Brodsky, A., Kerschberg, L., Varas, S.: On Optimal Constraint Decomposition in Distributed Databases. Center for Information Systems Integration and Evolution, George Mason University, Fairfax, VA, Technical Report October (1997)
5. Brown, A. W.: Control Integration Through Message Passing in a Software Development Environment. Carnegie Mellon University, Pittsburgh, CMU/SEI-92-TR-35, December (1992)
6. Brown, D. C., Lander, S. E., Petrie, C. J.: Special Issue on The Application of Multiagent Systems to Concurrent Engineering. Concurrent Engineering: Research and Applications. vol. 4 (1996)
7. Bussmann, S., Miller, J.: A Negotiation Framework for Cooperation. Proceedings 1992 of the Special Interest Group on Cooperating Knowledge Based Systems (CKBS-SIG 1992). Dake Center, University of Keele (1992) 1-17
8. Chaibdraa, B.: Interaction between Agents in Routine, Familiar and Unfamiliar Situations. International Journal of Intelligent & Cooperative Information Systems. vol. 5 (1996) 1-25
9. Chaibdraa, B., Desharnais, J., Lizotte, S.: A Cognitive Map Formulation for Relationships between Agents. University of Laval, SteFoy, PQ, Canada, DIUL-RR- 9505, (1995)
10. Cohen, P. R.: Pulling Together or Pulling Apart? A Survey of the Eighth National Conference on Artificial Intelligence. AI Magazine, vol. 12, (1991) 16-41
11. Cohen, P. R., Levesque, H. J.: Persistence, Intention and Commitment.: Morgan Kaufman (1987)
12. Cohen, P. R., Levesque, H. J.: Intention is Choice with Commitment. Artificial Intelligence (1990)
13. Cohen, P. R., Levesque, H. J.: Communicative Actions for Artificial Agents. ICMAS '95 (1995) 65-72
14. Cohen, P. R., Perrault, C. R.: Elements of a Plan Based Theory of Speech Acts. Cognitive Science. vol. 3 (1979)

References

15. Crowston, K. G.: Towards a Coordination Cookbook: Recipes for Multi-Agent Action. MIT, Cambridge, MA, CCS TR#128, Sloan WP# 3416-91, (1991)
16. Davis, R., Smith, R.: Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence*. vol. 20 (1983) 63-109
17. Decker, K. S.: Environment Centered Analysis and Design of Coordination Mechanisms. Ph.D. Thesis, Department of Computer Science. Amherst, MA,: University of Massachusetts, (1995)
18. Decker, K. S., Sycara, K.: Intelligent Adaptive Information Agents. *Journal of Intelligent Information Systems*. vol. 9 (1997) 239-260
19. Dix, A., Finlay, J., Hassell, J.: Environments for Cooperating Agents: Designing the Interface as Medium. HCI Group, University of York, June (1992)
20. Durfee, E.H.: Planning in Distributed Artificial Intelligence. In: O'Hare, G. M. P., Jennings, N. R., Eds.: *Foundations of Distributed Artificial Intelligence*. Wiley InterScience (1996)
21. Ephrati, E., Pollack, M.E., Sigalit, V.: Deriving Multi-agent Coordination through Filtering Strategies. Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95). Montreal, Canada (1995) 679-685
22. Gaines, B.R.: Knowledge Management in Societies of Intelligent Adaptive Agents. *Journal of Intelligent Information Systems*. vol. 9 (1997) 277-298
23. Genesereth, M., Ketchpel, S.P.: Software Agents. *Communications of the ACM*. vol. 37 (1994) 48-53
24. Guha, R.V., Lenat, D.B.: Enabling Agents to Work Together. *Communications of the ACM*. vol. 37 (1994) 126-142
25. Hewitt, C.: Viewing Control Structures as Patterns of Passing Messages. *Artificial Intelligence*. vol. 8 (1977) 323-364
26. Imam, I., Kerschberg, L.: Adaptive Intelligent Agents. *Journal of Intelligent Information Systems*. vol. 9 (1997) 211-214
27. Jajodia, S., Kerschberg, L.: *Advanced Transaction Models and Architectures*. Norwall, MA: Kluwer Academic Publishers, (1997)
28. Jennings, N. R.: Coordination Techniques for Distributed Artificial Intelligence. In: Georgeff, M. P., Jennings, N. R., Eds.: *Foundations of Distributed Artificial Intelligence*. Wiley InterScience (1996)
29. Kamel, M., Ghenniwa, H.: A Quantitative Analysis of Coordination Complexity for DAI Systems. Canadian Workshop on Distributed Artificial Intelligence. Banoe, Alberta, Canada, (1994)
30. Kerschberg, L.: Knowledge Rovers: Cooperative Intelligent Agent Support for Enterprise Information Architectures. In: Kandzia, P., Klusch, M., Eds.: *Cooperative Information Agents*, vol. 1202, *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin (1997) 79-100
31. Kerschberg, L.: The Role of Intelligent Agents in Advanced Information Systems. In: Small, C., Douglas, P., Johnson, R., King, P., Martin, N., Eds.: *Advanced in Databases*, vol. 1271, *Lecture Notes in Computer Science*. Springer-Verlag, London (1997) 1-22
32. Klein, M.: Computer Supported Conflict in Concurrent Engineering: Introduction to Special Issue. *Concurrent Engineering: Research and Applications*. vol. 2 (1993) 145-147
33. Klusch, M.: Using a Cooperative Agent System FCSI for a Context-Based Recognition of Interdatabase Dependencies. Workshop on Intelligent Information Agents, CKIM Conference. Gaithersburg, MD (1994)
34. Klusch, M.: Utilitarian Coalition Formation Between Information Agents. In: Kirn, S., O'Hare, G., Eds.: *Cooperative Knowledge Processing*. Springer-Verlag, London (1996)

References

35. Klusch, M., Shehory, O.: Coalition Formation Among Rational Information Agents. Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-96). Eindhoven, Netherlands (1996) 204-217
36. Klusch, M., Shehory, O.: A Polynomial Kernel-Oriented Coalition Algorithm for Rational Information Agents. Second International Conference on Multi-Agent Systems. Kyoto, Japan (1996)
37. Kraus, S., Wilensky, J., Zlotkin, G.: Multiagent Negotiation Under Time Constraints. *Artificial Intelligence* (1995) 297-345
38. Kue-Chu, L., Mansfield Jr., W.,H, Sheth, A.P.: A Framework for Controlling Cooperating Agents. *IEEE Computer* (1993) 8-16
39. Lai, K.-Y., Malone, T. W.: Object Lens: Letting End-Users Create Cooperative Work Applications. *Proceedings of CHI'91* (1991) 425-426
40. Laird, J.E., Pearson, D.J., Huffman, S.B.: Knowledge-Directed Adaptation in Multi-Level Agents. *Journal of Intelligent Information Systems*. vol. 9 (1997) 261-276
41. Lux, A., Bomarius, F., Stiener, D.: A Model for Supporting Human Computer Cooperation. *AAAI Workshop on Cooperation among Heterogeneous Intelligent Systems* (1992)
42. Maes, P.: Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back. *Special Issues of Robotics and Autonomous Systems*. Cambridge, MA, London, England: The MIT Press, (1990) 194
43. Maes, P., Darrell, T., Blumberg, B., Pentland, A.: The ALIVE System: Wireless, Full-Body Interaction with Autonomous Agents. *ACM Multimedia Systems* (1996)
44. Malone, T. W., Crowston, K. G.: The Interdisciplinary Study of Coordination. *ACM Computing Surveys*. vol. 26 (1994) 87-119
45. Prasad, B.: Concurrent Engineering Fundamentals. *Integrated Product Development* (1996)
46. Prasad, B.: Concurrent Engineering Fundamentals, II. *Integrated Product and Process Organization* (1996)
47. Rosenschein, J. S., Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation among Computers. Cambridge, MA,: MIT Press (1994)
48. Rosenschein, S., Kaelbling, L.P.: The Synthesis of Digital Machines with Provable Epistemic Properties. *Conference on Theoretical Aspects of Reasoning about Knowledge* (1986) 83-98
49. Rus, D., Gray, R., Kotz, D.: Transportable Information Agents. *Journal of Intelligent Information Systems*. vol. 9 (1997) 215-238
50. Sandhu, R.: Role hierarchies and constraints for lattice-based access controls. In: Bertino, E., Ed.: *Proc. Fourth European Symposium on Research in Computer Security, Lecture Notes in Computer Science*. Springer-Verlag, Rome, Italy (1996)
51. Sandhu, R., Bhamidipati, V.: The URA97 model for role-based administration of user-role assignment. In: Lin, T.Y., Qian, X., Eds.: *Database Security XI: Status and Prospects*. North Holland (1997)
52. Sandhu, R., Coyne, E., Feinstein, H., Youman, C.: Role-based access control models. *IEEE Computer*, vol. 29, (1996) 38-47
53. Seligman, L., Kerschberg, L.: Federated Knowledge and Database Systems: A New Architecture for Integrating of AI and Database Systems. In: Delcambre, L., Petry, F., Eds.: *Advances in Databases and Artificial Intelligence, Vol. 1: The Landscape of Intelligence in Database and Information Systems*, vol. 1. JAI Press (1995)
54. Seligman, L., Kerschberg, L.: A Mediator for Approximate Consistency: Supporting "Good Enough" Materialized Views. *Journal of Intelligent Information Systems*. vol. 8 (1997) 203 - 225
55. Shoham, Y., Tennenholtz, M.: On the Synthesis of Useful Laws for Artificial Agents Societies. *Tenth National Conference on Artificial Intelligence*. San Jose, CA, (1992) 276-281

References

- 56. Sichman, J.S., Demazeau, Y.: Exploiting Social Reasoning to Deal with Agency Level Inconsistency. First International Conference on Multi-Agent Systems. San Francisco, CA (1995)
- 57. Varas, S.: On Optimal Constraint Decomposition, Monitoring, and Management in Distributed Heterogeneous Environments. Doctoral Dissertation in Information Technology. Fairfax: George Mason University, (1998)
- 58. Wang, X.S., Bettini, C., Brodsky, A., Jajodia, S.: Logical Design for Temporal Databases with Multiple Temporal Types. *ACM Transactions on Database Systems* (1997)
- 59. Wang, X.S., Jajodia, S., Subrahmanian, V.S.: Temporal Modules: An Approach toward Federated Temporal Databases. *Information Sciences*. vol. 82 (1995) 103-128
- 60. Wellman, M.: A Market Oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems. *JAIR*. vol. 1 (1993) 1-23
- 61. Widom, J., Ceri, S.: *Active Database Systems: Triggers and Rules for Advanced Database Processing*. : Morgan Kaufmann Publishers, Inc., (1995)
- 62. Wiederhold, G.: Foreword to Special Issue on the Intelligent Integration of Information. *Journal of Intelligent Information Systems*. vol. 6, 2/3 (1996) 93-97
- 63. Wiederhold, G., Jajodia, S., Litwin, W.: Dealing with Granularity of Time in Temporal Databases. In: Anderson, R., others, Eds.: *Lecture Notes in Computer Science*, vol. 498. Springer-Verlag (1991) 124-140
- 64. Wiederhold, G., Jajodia, S., Litwin, W.: Integrating Temporal Data in a Heterogeneous Environment. In: Tansel, A.U., Jajodia, S., others, Eds.: *Temporal Databases: Theory, Design, and Implementation*. Benjamin/Cummings (1993) 563-579.

Appendix 1: A Survey of Agent Frameworks

Legend: The Type column denotes C for commercial product, R for research, and P for Prototype, respectively.

Agent System/ URL	Type	Orga- niza- tion	Lan- guage	Description
Agent Build- ing Environ- ment http:// www.network- ing.ibm.com/ iag/iag- soft.htm	C	IBM	C++, Java	IBM's Agent Building Environment (ABE) is a toolkit for building applications based on intelligent agents. The intelligent agent watches for a certain condition, decides what to do based on a set of rules, and triggers an action as a result. The architecture for the agent is based on reasoning engine and adapters. "Adapters" or interfaces allow the agent to interact with the rest of the world.
Agent Build- ing Shell (ABS) http:// www.ie.utor- onto.ca/EIL/ ABS-page/ ABS-over- view.html	R	U. of Tor- onto	COOrdina- tion Lan- guage (COOL)	This provides several reusable layers of languages and services for building agent systems: coordination and communication languages, description logic-based knowledge management, cooperative information distribution, organizational modeling and conflict management. The approach is being used to develop multiagent applications in the area of manufacturing enterprise supply chain integration.
Agent Devel- opment Envi- ronment (ADE) http:// www.gen- sym.com/	C	Gen- sym		ADE provides a predefined class hierarchy of agents and agent parts, an agent communications "middle-ware" infrastructure, and a graphical language for designing and developing agent behavior based on the Grafset standard. It takes advantage of the powerful capabilities of Gensym G2's application development environment on which it is built.
Agent TCL http:// www.cs.dart- mouth.edu/ ~agent/	R	Dart- mout h Uni- ver- sity	Tcl (Tool Com- mand Lan- guage)	Agent Tcl is a tool for developing transportable agent systems. Agent Tcl supports message passing. Agents can clone themselves and the system provides rudimentary security features. The agents migrate from machine to machine using the jump command. Each agent on a particular machine has a unique integer ID and a unique symbolic name. Migrating agents are encrypted and authenticated using Pretty Good Privacy (PGP). Access restrictions are imposed on the agent based on its authenticated identity. Safe Tcl enforces the access restrictions.

AgentTalk	C	NTT & Ishida Lab, Kyoto University.	LISP	AgentTalk is a coordination protocol description language for multi-agent systems. It allows coordination protocols to be defined incrementally and to be customized to suit application domains by incorporating an inheritance mechanism.
AgentBuilder http://www.agent-builder.com/	C	Reticular Systems, Inc.	Java	AgentBuilder is an integrated tool suite for constructing intelligent software agents. AgentBuilder consists of two major components - the Toolkit and the Run-Time System. The AgentBuilder Toolkit includes tools for managing the agent-based software development process, analyzing the domain of agent operations, designing and developing networks of communicating agents, defining behaviors of individual agents, debugging and testing agent software. Agents constructed using AgentBuilder communicate using the Knowledge Query and Manipulation Language (KQML) and support the performatives defined for KQML.
Agentx http://www.iks.com/	C	International Knowledge Systems	Java	Agentx is a set of lightweight, high performance, and scaleable distributed computing libraries for the Java programming environment. The libraries were designed to provide object request broker facilities that were easier to use, faster, more compact and highly functional, than the RMI libraries bundled with the Sun JDK or generally available Java based CORBA implementations. Objects can move around on the network, attach themselves to a host, and begin independent execution of their program code.
Aglets http://www.trl.ibm.co.jp/aglets/	C	IBM Japan	Java	An aglet is a Java object that can move from one host on the Internet to another. That is, an aglet that executes on one host can suddenly halt execution, dispatch to a remote host, and resume execution there. When the aglet moves, it takes along its program code as well as its state (data). A built-in security mechanism makes it safe for a computer to host untrusted aglets.
Architecture type-based Development Environment (ADE) http://samuel.cs.uni-potsdam.de/soft/taxt/	R	U. of Potsdam Dept. of Computer Science	Java	Software agents and agent systems are modeled using object-oriented design thus allowing explicit modeling of agent interactions and a large number of property classes. The software architecture acts as a system of components whose interactions are realized via connectors. Theoretical fundamentals for the implementation of computer-independent platforms for agent programming have been found. The advantages of a substantiated methodology of an architecture type-based approach are lost while transforming them onto existing platforms.

Bee-gent http://www2.toshiba.co.jp/beegent/index.htm	R	Toshiba Corporation	Java	Bee-gent is purely an agent system. The applications are agents and all messages are carried by agents. "Agent Wrappers" are used to agentify existing applications, while "Mediation Agents" support inter-application coordination by handling all communications. The mediation agents move from the site of an application to another where they interact with the agent wrappers. The agent wrappers themselves manage the states of the applications they are wrapped around, invoking them when necessary.
Bond Distributed Object System http://bond.cs.purdue.edu/	R	Purdue University	Java	The Bond distributed object system provides a message oriented middleware environment for developing distributed applications. Bond uses the KQML language for object communication. The agent framework of the Bond system simplifies the task of developing agents by allowing the programmer to concentrate on the specific strategies of a new agent. Bond agents have the intrinsic capability to be controlled remotely and to cooperate with each other. The task of an application programmer is limited to specifying the agenda, the finite state machine of the agent, and the strategies associated with each state.
Cable http://public.logica.com/~grace/Architecture/Cable/public/	R	Logica Corporation	Agent Definition Language, C++	Cable can be used to develop and execute distributed applications that are based on the metaphor of multiple, cooperating intelligent agents. Cable provides the user with an Agent Definition Language (ADL), for defining agents, and a parser known as the Scribe, for compiling agent definitions written in ADL into agent applications. Agents are developed using ADL and C++. ADL allows developers to use Cable without worrying about underlying detail, providing a language with a level of abstraction close to that of the agents with which an application is designed. Inter-agent communication over a local area network is handled using ORBIX, an implementation of the CORBA 2.0 standard.
Concordia http://www.meitca.com/HSL/Projects/Concordia	R	Mitsubishi Electric	Java	CONCORDIA is a framework for development and management of network-efficient mobile agent applications for accessing information on any device supporting Java. With Concordia, applications can: process data at the data source, process data even if the user is disconnected from the network, access and deliver information across multiple networks (LANs, Intranets and Internet), use wire-line or wireless communication, support multiple client devices, such as Desktop Computers, PDAs, Notebook Computers, and Smart Phones.
Cybele http://www.i-a-i.com/projects/cybele/index.html	R	Intelligent Automation, Inc.	unknown	This supports rapid development by providing infrastructure for agent-based applications: (i) Agent creation and deployment over a network of varied platforms, (ii) a message addressing scheme for agent communication which is independent of the location of a sending or receiving agent, (iii) the accumulation of messages intended for a currently busy recipient agent, (iv) the proper conversion of message data across platforms, (v) multicasting, broadcasting, and peer-to-peer messaging, and (vi) the migration of agents across processors for performance optimization and/or fault tolerance.

DECAF Agent Framework http://www.cis.udel.edu/~graham/DECAF/	R,P	University of Delaware	Java	DECAF (Distributed Environment Centered Agent Framework) allows rapid development of agents by building an operating environment that provides an interface, internal agent scheduling and monitoring in a fashion similar to operating system primitives. The agent developer does not need knowledge of any of this structure and can thus focus on development of the agent itself. The basic DECAF architecture has been built using the Java programming language.
dMARS http://www.aaii.oz.au/proj/dMARS-product-brief.html	R	Australian Artificial Intelligence Institute Ltd.	C, C++	dMARSTM designed for rapid configuration and ease of integration with the issues of robustness, efficiency and user-extensibility in mind, is an agent-oriented development and implementation environment for building complex, distributed, time-critical systems. This product is based on Procedural Reasoning System (PRS) developed by SRI International (California). It provides a sophisticated suite of graphical tools for development and debugging and takes advantage of the latest research into multiagent, real-time reasoning.
Grasshopper http://www.ikv.de/products/grasshopper/	C	IKV+	Java	Grasshopper built on CORBA, complies to the Mobile Agent System Interoperability Facility (MASIF) standard and provides the integration of traditional client/server paradigm and mobile agent technology. The agent applications are open towards other agent environments and allows one to build agent-enabled distributed applications which take advantage of local high-speed communication and local high-speed data access.
Gypsy http://www.info-sys.tuwien.ac.at/Staff/lux/Gypsy/	R	Technical University of Vienna	Java	Gypsy is a flexible environment for mobile agent programming intended for application in Internet information retrieval, Internet commerce, mobile computing, and networks network management.
iGENTM http://www.chi-inc.com/gina-home.htm	C	CHI Systems	C/C++	iGENTM is a toolkit and workbench based on a programmable model of human expertise. It includes high-level agent-building tools for the development of intelligent applications.
Infospiders http://www-cse.ucsd.edu/users/fil/agents/agents.html	R	UC San Diego - Computer Science Dept.	unknown	InfoSpiders (aka ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery) features an artificial life inspired model using endogenous fitness for information retrieval in large, dynamic, distributed, heterogeneous databases, such as the WWW. A population of agents is evolved under density dependent selection for the task of locating information for the user.

Intelligent Agent Factory http://www.bit-pix.com/iaf/iafintro/iafintro.htm	C	Bits & Pixels	Java	The Intelligent Agent Factory has agents that are controlled with rules written in Jess (CLIPS), a forward-chaining system. Agents and rules are generated from simple specifications of workflows.
Intelligent Agent Library http://www.bit-pix.com/business/main/bitpix.htm	C	Bits & Pixels	Java	The Intelligent Agent Library provides extensive facilities for agent communications for building large agent assemblies. There is a KQML-based agent framework and many examples illustrating agents that perform activities for web-enabled applications. The library also supports mobile agents.
JACK Intelligent Agents http://www.agent-software.com.au/jack.html	C	Agent Oriented Software, Ltd.	JACK Agent Language	JACK provides the architecture and capability for developing and running software agents in distributed applications. JACK Agent Language is an extension to Java and provides all the features of the Java language.
JAFMAS http://www.eecs.uc.edu/~abaker/JAFMAS	R	University of Cincinnati	Java	JAFMAS provides a framework to structure ideas into multiagent systems with a set of classes for agent deployment. It directs development from a speech-act perspective -supports multicast and directed communication, KQML or other speech-act performatives; gives an analysis of multiagent system coherency and consistency; and provides a good comparison of agent tools with particular emphasis on mobile agents.
JATLite http://java.stanford.edu/java_agent/html/	R	Stanford University	Java	JATLite provides a basic infrastructure in which agents register with an Agent Message Router facilitator using a name and password, connect/disconnect from the Internet, send and receive messages, transfer files, and invoke other programs or actions on the various computers where they are running. JATLite facilitates construction of agents that send and receive messages using KQML.
Jumping Beans http://www.JumpingBeans.com/	C	Ad Astra Engineering, Inc.	Java	Jumping Beans allows a Java application software to move from one host to another along with the executable, data, state, resources and other essential information so that can execute on hosts that did not have the application previously installed.

Kafka http://www.fujitsu.co.jp/hypertext/free/kafka/	C	Fujitsu	Java	Kafka is an extendable, flexible Java class library based on Java's RMI and designed for constructing multiagent based distributed applications.
Kasbah http://ecommerce.media.mit.edu/Kasbah/index.html	R	MIT	unknown	Kasbah is an agent-mediated electronic commerce system. A user wanting to buy or sell a product or service creates an agent, give it some strategic direction, and send it off into the agent marketplace. Kasbah agents pro-actively seek out potential buyers or sellers and negotiate with them on their creator's behalf. Each agent's goal is to make the "best deal" possible, subject to a set of user-specified constraints, such as a desired price, a highest (or lowest) acceptable price, and a date to complete the transaction.
Knowbot System Software http://www.cnri.reston.va.us/home/koe/	R	CNRI	Python	Knowbot is a mobile agent system intended for use in widely distributed systems such as the Internet. The design of the Knowbot architecture allows agents of several programming languages to interoperate.
LALO http://www.CRIM.CA/sbc/english/lalo/	R	CRIM	LALO	LALO is an Agent Oriented Programming (AOP) language and environment for the defining and creating extensible multiagent systems including reactive agents and deliberate agents. The inter-agent communication language used is KQML. A program written in LALO is translated into C++ source code, and then compiled with a C++ compiler.
LiveAgent http://www.agent-soft.com/	C	AgentSoft Ltd.	Java	LiveAgent Pro automates Web activity by creating Internet/Intranet scripts using a recording environment (like a high-level macro recorder or automated testing tool). A developer performs a sequence of Web operations in the browser, and those actions are automatically saved by LiveAgent Pro as a script or "agent". The completed script can then be run by the user or scheduled for automatic launching.
Microsoft Agent http://www.microsoft.com/workshop/imedia/agent/default.asp	C	Microsoft	Active X	Microsoft Agent uses animated characters as interactive assistants to introduce, guide, entertain, or otherwise enhance their Web pages or applications in addition to the conventional use of windows, menus, and controls. This enables the incorporation of speech recognition so applications can respond to voice commands. Characters can respond using synthesized speech, recorded audio, or text in a cartoon word balloon.

Mobiware Middleware Toolkit http:// comet.colum- bia.edu/mobi- ware/	R	Colum- bia Uni- ver- sity	Java	Mobiware is a set of open programmable interfaces and algorithms for adaptive mobile networking, built on CORBA and Java distributed object technology. It runs on mobile devices, wireless access points and mobile-capable switch/routers.
MOLE http:// www.informa- tik.uni-stut- tgart.de/ipvr/ vs/projekte/ mole.html	R	Uni- ver- sity of Stut- tgart	Java	This is a mobile agent with mechanisms for migration and communication.
Multi-Agent Modeling Lan- guage (MAML) http:// www.sys- lab.ceu.hu/ maml/	R	Central Euro- pean Uni- ver- sity	MAML	MAML can be used to develop models, run simulations, search in parameter space, and analyze results. It has a graphical user interface, support work through the web, and is aimed to support scientists who are not experts in programming.
MultiAgent Systems Tool (MAST) http:// www.gsi.dit.u pm.es/~mast/	R	Tech- nical Uni- ver- sity of Madr id	C++	MultiAgent Systems Tool is a general purpose distributed framework for the cooperation of multiple heterogeneous agents. The MAST architecture consists of two basic entities: the agents and the network through which they interact.
Odyssey http:// www.genmagi c.com/technol- ogy/odys- sey.html	C	Gen- eral Magi c	Java	Odyssey is a set of Java class libraries to support development of distributed, mobile applications. Odyssey agents are Java threads. They are created by subclassing the Odyssey agent class or the Odyssey worker class.
Open Agent Architecture TM http:// www.ai.sri.co m/~oaa/	R	SRI Inter- nation- al		Open Agent ArchitectureTM (OAA) is a framework for integrating a community of heterogeneous software agents in a distributed environment. In this framework, an agent is defined as a software process that registers its services in an acceptable form, speaks the Interagent Communication Language (ICL), and shares functionality common to all OAA agents (e.g. the ability to install triggers, manage data in certain ways).

ProcessLink http://cdr.stanford.edu/ProcessLink/	R	Stanford University		This consists of generic agents and a message protocol. It allows the "wrapping" of legacy software to provide the coordination functions and send messages conforming to a defined interaction semantics.
RETSINA http://www.cs.cmu.edu/~softagents	R	Carnegie Mellon University		RETSINA is a reusable agent. Each RETSINA agent has four reusable modules for communicating, planning, scheduling and monitoring the execution of tasks and requests from other agents. A RETSINA agent is distinguished according to the kind of task it performs (i.e. interface, task, information and middle agents).
Sodabot http://www.ai.mit.edu/people/sodabot/sodabot.html	R	MIT Artificial Intelligence Lab	unknown	Sodabot is essentially an agent operating system. A new programming language was designed around human-level descriptions of agent activity. Using this language, users can implement a wide-range of typical software agent applications, e.g. personal on-line assistants and meeting scheduling agents.
Swarm http://www.santafe.edu/projects/swarm/	R	Sante Fe Institute	C	Swarm is the simulation of collections of concurrently interacting agents.
Via: Versatile Intelligent Agents http://www.kinetscope.com/via/default.htm	R	Kinetscope	Java	Via contains tools for creating agent tasks, logic, and user interfaces into existing applications, Web sites and intranets. It also includes advanced features for supporting system-wide distribution, persistence, and scalability

Voyager http://www.objectspace.com/	C	Object Space	Java	Voyager combines the power of mobile autonomous agents and remote method invocation with complete CORBA support and distributed services such as directory, persistence, and publish subscribe multicast.. Agents are modeled as autonomous objects and Voyager allows construction of remote objects, sends messages and moves objects between applications so that agents can move themselves and continue executing as they move. In this way, agents can act independently on the behalf of a client, even if the client is disconnected or unavailable.
Zeus http://www.labs.bt.com/projects/agents/research/collaborative.htm	R	British Telecommunications Labs	Java	Zeus is a 'collaborative' agent building environment and component. Each ZEUS agent consists of a definition layer, an organizational layer and a co-ordination layer. The definition layer has a programming interface and represents the agent's reasoning and learning abilities, its goals, resources, skills, beliefs and preferences. The organization layer describes the agent's relationships with other agents. The co-ordination layer describes the co-ordination and negotiation techniques the agent possesses. Communication protocols are built into the co-ordination layer and enable inter-agent communication.