Chapter III

Intelligent Enterprise Integration: eMarketplace Model

Hamada H. Ghenniwa
University of Western Ontario, Canada

Michael N. Huhns
University of South Carolina, USA

ABSTRACT

This chapter describes an architecture for the eMarketplace *that integrates the interests of autonomous enterprises in a single open-market environment. The environment encompasses several systems and business issues, such as the many-to-many relationships between customers and suppliers, systems, and business-related services. The architecture for this integrated environment is business-centric and knowledge-oriented. In this architecture, the* eMarketplace *exists as a collection of economically motivated software agents. The architecture enables and supports common economic services, such as brokering, pricing, and negotiation, as well as cross-enterprise integration and cooperation in an electronic supply-chain. We demonstrate the* eMarketplace *with two prototype systems.*

INTRODUCTION

It is said that there are only two types of enterprises: those that change and those that disappear. Businesses today must be fast and flexible, responsive to customers, and cost-effective in their operations. They must collaborate more frequently with partners to build virtual organizations and supply-chains that reduce time-to-market and costs. More challenges loom as companies, organizations, and other business entities try to reorient their internal capabilities to exploit electronic business (*eBusiness*) techniques. These are difficult and expensive things to do in a fast-paced world where change drives business. *eBusiness* is the use of the Internet along with other electronic means and technologies to conduct within-business, business-to-consumer, business-to-business, and business-to-government interactions.

Traditional models of *eBusiness*, such as those based on EDI (Electronic Data Interchange), ERP (Enterprise Resource Planning), and enterprise-centric views, are useful for businesses with well-defined trading relationships, but unsuitable for the rapidly growing and changing global marketplace. In these models, point-to-point interfaces are created to support transactions involving replenishment orders for direct production goods of a previously negotiated contract. For example, the sell-side model requires that either a single distributor is responsible for aggregating all the suppliers, or the customer is responsible for comparison-shopping between suppliers. This makes it inefficient and expensive for both customers and suppliers. Another example is the buy-side model, which does not enable dynamic trading and requires the buying organizations to set up and maintain catalogs of their suppliers, and hence is costly and technically demanding.

An electronic marketplace (*eMarketplace*) model appears to be the most promising forum for reshaping *eBusiness* relationships, and will soon affect all businesses in one way or another. In this work, we view *eMarketplace* as a cooperative distributed system that integrates participating business entities, including consumers, suppliers, and other intermediaries. This architecture enables and facilitates common economic services and commerce transactions between the buyers and sellers, such as brokering, pricing, and negotiation, as well as cross-enterprise integration and cooperation in an electronic supply-chain. In this architecture, the *eMarketplace* exists as a collection of economically motivated software agents. We envision that *eMarketplaces* will become viable businesses, and the revenue for these marketplaces could come from several, possibly combined, avenues, including registration fees, advertising fees, commission fees on transactions, and revenue from bid/ask spreads in high volume markets.

The *eMarketplace* will enable one-stop shopping for products by consumers, who depend on a variety of other products and services that can spread across several marketplaces. Likewise, suppliers can reach, discover, and develop new customers across various *eMarketplaces* quickly with low cost. In general, *eMarketplaces* offer businesses the chance to develop and enhance their most important relationships — those with customers and suppliers.

To develop a successful engineering foundation for an open-market *eMarketplace* that supports many-to-many relationships between different business entities, several business and design issues need to be analyzed and addressed. The rest of the chapter is organized as follows. First, it reviews some of the business models related to *eBusiness* applications with a brief analysis of the main architectural design issues for *eMarketplaces*. After that, it briefly describes an architecture for a cooperative distributed system, namely, the Business-Centric Knowledge-Oriented architecture (BCKOA) for *eMarketplace* integration. This is followed by a description of a layered BCKOA implementation for an *eMarketplace*. Then the main components of an agent-oriented BCKOA for an *eMarketplace* are presented, including a supply-chain automation system for integration and management using a group of cooperating software agents. A short description of an ongoing implementation of the proposed model for virtual enterprise *eMarketplace* is described next. Then it discusses some of the related work in both the academic and industrial communities. Finally, it summarizes the main contributions of this chapter.

eBUSINESS MODELS

Surveys of small and large companies have shown that one of the most frequently mentioned barriers to successful *eBusiness* projects is the lack of an appropriate business model. It is certainly one of the most important aspects in *eBusiness* applications. A business model, in simple words, is "an architecture for the product, services, and information flows, including a description of the various business actors and their roles; and a description of the potential benefits for the various business actors; and a description of the sources of revenues" (Timmers, 1999). In an *eBusiness* environment, a business model can be viewed in terms of four principal components (Bartelt & Lamersdorf, 2001): (1) the products and services offered by the business entity, (2) the customer relationships that the business entity creates and maintains in order to generate revenues, (3) the financial aspects of the business, such as cost and revenue structures, and (4) the infrastructure and the network of partners. Possible architectures for business models are constructed by combining interaction patterns with value-chain integration (Timmers, 1999; Dubosson, Osterwalder, & Pigneur, 2002; Bartelt & Lamersdorf, 2001) for the possible creation of marketplaces. These can be fully open, with an arbitrary number of customers and suppliers, or semi-open, with one customer and multiple suppliers or vice-versa. In principle, a large number of architectures can be conceived for *eBusiness* applications. In practice, however, only a limited number can be realized (Timmers, 1999). The following are the most widely realized models (Timmers, 1999).

A basic model of an *eBusiness* is the *eShop* model. It is based on providing a self-service storefront to a customer by displaying the company catalogs and product offers on a web site. The business objective is to lower the sales cost. A major concern in this architecture is the assumption that the customer should be responsible for comparison-shopping between products of different suppliers. While an *eShop* model is based on the selling aspect of the business, an *eProcurement* model focuses on the buying aspect of the business. A typical architecture for *eProcurement* consists of a browser-based self-service front end to the corporate purchasing system or its ERP. The supplier catalogs are presented to end-users through a single unified catalog, thereby facilitating a corporate-wide standard procurement process. In addition, *eProcurement* might support calls for tender through the Web, which might be accompanied by an electronic submission of bids. Nonetheless, an *eProcurement* model does not support dynamic trading. The business objective of this model is cost savings on purchasing operations. Recently, online auction models have received much attention for automating dynamic trading. The prime business objective is to increase efficiency, reduce waste, and minimize overall cost. Other models are based on creating value-chain businesses. One model describes service provisioning of specific functions for the value chain, such as electronic payments or logistics. New approaches are also emerging in production and stock management, where new intermediary service providers are formed to provide specialized expertise to analyze and fine-tune production. The business objective is to generate revenue based on fee or percentage.

Although each of the above models attempts to provide an *eBusiness* solution, none of them addresses the challenge of how to create and leverage services and supply operations in a way that seamlessly integrates business entities (customers, suppliers, partners, and competitors) in a dynamic trading community. A very important and promising business model is the *eMarketplace*. This model supports value-chain integration and provisioning in its structure and services. The objective is to develop an *eBusiness* solution that relieves participating business entities of much of the burden to participate effectively in the *eBusiness* domain. This model combines the advantages of the sell-side, the buy-side, and the value-chain models. The business objective of the *eMarketplace* model can be based on a combination of subscription fees, transaction fees, and service fees.

The feasibility of implementing a business model depends upon the state-of-the-art of the technology, whether for realizing individual functions, for supporting interaction patterns, or for integrating components. The rest of this chapter lays the engineering foundation for developing an architectural framework for *eBusiness,* with special attention on an *eMarketplace* model. The specification of an *eMarketplace* as a cooperative distributed system describes the architecture of an ontology-driven *eBusiness* environment that deals with several technological and business issues.

eMARKETPLACES: REQUIREMENTS ANALYSIS AND DESIGN ISSUES

To develop a successful engineering foundation for an *eMarketplace*, we need a fully realized solution that accommodates the needs of *eBusiness* participants and allows them to extend advanced services to the trading community. As an *eBusiness* grows and becomes viable in the real world, its corresponding *eMarketplaces* must expand to support a broader base of services ranging from baseline interaction and directory services to specialty services, such as dynamic trading, cooperative supply-chain integration, and management. In addition, an *eMarketplace* should enable and facilitate tightening the relationship between suppliers and customers. To this end, a fundamental aspect that an *eMarketplace* architecture supports is the many-to-many relationship between customers and suppliers. This enables both customers and suppliers to leverage economies of scale in their trading relationships and access a more liquid marketplace. This in turn allows the use of dynamic pricing models, such as auctions, which improve the economic efficiency of the market where uncertainty about prices and demand are common. To provide smooth and effective integration at the business level, an *eMarketplace* model accommodates and supports interfaces to the existing business models of the participant entities through cooperative supply-chain integration and management.

Another key factor for the foundation of an *eMarketplace* is the ability to operate in an open environment. This is driven by the fact that in many cases a customer's needs may go beyond the specialist capabilities of any single *eMarketplace*. The ability of *eMarketplace*s to interact extends the idea of liquidity and network effect without sacrificing the ability to be highly specific to the supply-chain node or target the customer groups they serve.

It is also important that the architecture of an *eMarketplace* supports the incorporation and leveraging of the participants' legacy environments with minimum overhead. The support can take place, as is described later, over a cooperative distributed system that is technology-independent and scalable in the sense of supporting a large number of users in a dynamic open environment.

In this new *eMarketplace* environment there are significant interactions between the systems deployed by the participating business units of an enterprise, their customers, and other businesses. Therefore, designing *eMarketplaces* requires embodying greater levels of business knowledge within the *eMarketplace* transactions, activities, and service definitions. Additionally, it requires a greater degree of communication, coordination, and cooperation within and among the business entities and their systems in the *eMarketplace*. In other words, the *eMarketplace* architecture represents an integrated body of people, systems, information, processes, services, and products. Several attempts in business-process re-engineering addressed structural integration by reorganizing enterprises along critical business processes, such as the supply chain and the product life cycle (Hammer & Champy, 1993; Davenport, 1993). However, in this chapter, by integrated we mean the structural, behavioral, and informational integration of the participant business entities. To enable this, we develop an architecture that supports the communication of information and knowledge, the making of decisions, and the coordination of actions. The following subsections address these aspects in more detail in order to set the foundation for the proposed architecture.

Enterprise Model and Ontologies

At the heart of our integration architecture for an *eMarketplace* is a model of the enterprise. An enterprise model is an abstract representation of the structure, activities, processes, information, resources, people, behavior, goals, rules, and constraints of the *eMarketplace*. It can be used to support effective enterprise design, analysis, and operation. From an operational perspective, the enterprise model captures what is planned, what might happen, and what has happened. Therefore, it supplies the information and knowledge necessary to support the operations of an *eMarketplace*.

The information systems of participating business entities are usually built by different people, at different times, to fulfil different business requirements. Consequently, in the absence of an architectural framework for an *eMarketplace* geared toward enterprise integration, there are widely varying viewpoints and assumptions regarding what is essentially the same subject. Therefore, communication among the components supporting a business-to-business application is not possible without at least some translation. This problem, however, is much more than a simple agreement on XML tags or mappings between roughly equivalent sets of tags in related standards. Industry-wide *eBusiness* initiatives and academic studies have shown that complex representation issues can arise. To deal with these issues, an appropriate *eMarketplace* architecture should support enterprise-modeling ontologies. An ontology is a vocabulary along with some specification of the meaning or semantics of the terminology within the vocabulary. Ontologies can vary based on the degree of formality in the specification of meaning. The objective is to provide a shared and common understanding of a domain that can be communicated to people, application systems, and businesses. In an *eMarketplace* model, ontologies are integrated or related to support reasoning among the elements of the model.

Many of the foundation concepts of an ontology have already been established in work on intelligent agents and knowledge sharing, such as the Knowledge Interchange Format (KIF) and Ontolingua languages (Genesereth & Fikes, 1992). With the wide acceptance of XML by the Web and Internet communities, XML gained tremendous potential to be the standard syntax for data interchange on the web. It is also becoming desirable to exchange ontologies using XML. This motivated the development of XML-based ontology languages, such as SHOE (Heflin, Hendler, & Luke, 1999), Ontology Exchange Language (XOL) (Karp, Chaudhri & Thomere, 1999), and the Resource Description Framework Schema (RDFS) (Lassila & Swick, 1999). Other proposals, such as OIL (Ontology Interchange Language) (Fensel, Horrocks, van Harmelen, Decker, Erdmann, & Klein, 2000) and its successors DAML+OIL (Fensel, van Harmelen, & Horrocks, 2002) and more recently OWL (Ontology Web Language), attempt to extend RDF and RDFS for ontology representations.

Other approaches, like ebXML (Eisenberg & Nickull, 2001), attempt to develop an open XML-based infrastructure specification to enable the global use of *eBusiness* information to exchange business messages, conduct trading relationships, communicate data in common terms, and define and register business processes. Simply, ebXML is an attempt to develop and to promote shared ontologies. The W3C Semantic Web (The Semantic Web Community Portal) initiative takes a similar approach. Also, there are various attempts to achieve standardized content descriptions, such as the Common Business Library (CBL) of Commerce Net, Commerce XML (cXML or Commerce eXtensible Markup Language), Dublin core, and RossettaNet.

Market Structure and Economy Model

An important aspect of the *eMarketplace* is the economic model of its structure. A market structure governs the trading process and defines the formal rules for market access, traders' interactions, price determination, and trade generations. Its behavior restricts the set of message sequences that traders may exchange and determines the trading outcome. Therefore, a market institution (McCabe, Rassenti & Smith, 1992) is the specification of the set of admissible messages (i.e., traders' actions, usually price and/or quantity offers), and the final commodity allocation given any combination of messages chosen by the participants and any initial allocation. In classical economic theory there are several market models for specific trading situations and structural behaviors. Here we review some economic models for *eMarketplace* structures, such as a commodity market, auctions, and bargaining, with an objective to realizing them. The focus on realizing different economic models is kept as generic as possible. Special emphasis will be placed on components and heuristics that an *eMarketplace* can support for the participating business entities to establish an appropriate trading approach.

In the commodity market model, various suppliers and consumers participate to trade goods/services (commodity) of the same type. The market price is publicly agreed upon for each commodity independent of a particular supplier. All consumers and suppliers decide whether to buy and how much to buy or sell at each agreed-upon price. The challenge in this market structure is to deploy a pricing methodology that produces price adjustments that bring about market equilibrium (i.e., equalize supply and demand).

In an auction-based market, each participant (consumer and supplier) acts independently and contracts to buy or sell at a price agreed upon privately. An auction-based *eMarketplace* is a form of centralized facility, or clearinghouse, by which costumers and suppliers execute trades in an open and competitive bidding process. All auctions can be classified as open or closed (sealed) auctions. In open auctions, bidders can know the bid value of the others and will have an opportunity to offer competitive bids. In sealed auctions, the participants' bids are not disclosed to others.

The two market structures above are not appropriate for bargaining situations where few participants try to reach an agreement that will leave them at least as well off as they could be if they reached no agreement. Most of these situations cannot be entirely determined by the market forces. In bargaining, both customers and suppliers have their own objective functions and they negotiate with each other as long as their objectives are met. The participants can engage in direct negotiations with each other using their respective bargaining strategies to arrive at a "fair" price for a particular item. This market structure does not support a specific negotiation protocol, rather the participants will use an unrestricted bidding protocol. A major challenge in this structure is how to enable any participant to determine the "fair" price.

Supply-Chain Integration and Management

An *eMarketplace* can be treated as a physically and logically distributed system of interacting autonomous business entities. Yet, there is a need for well-accepted interoperability standards, which must be meshed for supply-chain integration to meet business demands. Conceptually, a supply-chain manages coordinated information and material flows, production operations, and logistics of the *eMarketplace*. It provides the *eMarketplace* with flexibility and agility in responding to customer demand shifts without conflicts in resource utilization. The fundamental objective is to improve coordination within and between various participant business entities in the supply-chain. The increased coordination can lead to reduction in lead times and costs, alignment of interdependent decision-making processes, and improvement in the overall performance of each participant in the chain, as well as the supply chain itself.

In an *eMarketplace* setting, supply-chain management can be viewed as a cooperative, distributed problem-solving activity among a society or group formed by autonomous business entities that work together to solve a common problem (Smirnov & Chandra, 2000). With their collective and collaborative efforts, they sustain the progress of each member as well as the group. The group is responsible for coordination throughout the supply chain, whereas each member provides specialized expert knowledge and product and process technology to the supply chain. The decision-making process is centralized for the group, but decentralized for the local decisions of each member. Therefore, the problem of supply-chain design in an *eMarketplace*, as discussed later, can be solved by the design of a structure and mechanism for coordination in a cooperative, distributed system.

Foundation Architecture for Integration

The architecture of the *eMarketplace* provides the foundation to integrate and leverage the participants' resources, such as applications and databases. Traditionally, the foundation technology that enables enterprises to connect resources together is known as *middleware*. Mainstream middleware solutions focus on integration at the data-level. There are several commercial middleware products and standards, such as OMG CORBA™ (Object Management Group, Inc., 1995), J2EE™ (Java™ 2 Platform, Enterprise Edition), and DCOM (Distributed Component Object Model), that focus on providing infrastructure tools and frameworks of integration. This approach provides a communication framework for the integration of data resources with minor or no support for integration at the business level. Enterprise application integration (EAI) is a trend has recently emerged in designing middleware technology with an objective to ease the burden and lower the costs of application integration. However, different EAI tools are developed to accommodate different levels of integration requirements. Object-level integration provides synchronization of data between different applications or databases. Business process-level integration extends the object-level by supporting multiple, distributed, and heterogeneous applications. Finally, cross-enterprise, process-level integration involves multiple, distributed, heterogeneous business-process applications across different enterprises cooperating in a supply-chain. There are currently very few EAI solutions specifically designed for this type of integration scenario.

While EAI tools focus on technology-centered integration, other complementary approaches focus on integration as an architectural aspect. In this direction, the focus is on the components' ability to perform their functions in a larger system context and not on their precise implementation. One approach is a mediator-based architecture (Wiederhold, 1992), which comprises a layer of "intelligent" middleware services to link data resources and applications, such as integrating data from multiple sources in a way that is effective for the client application. In addition, the mediator architecture partitions system resources and services into two dimensions, horizontal and vertical (Wiederhold &

Genesereth, 1997). The vertical partition includes specific domain services. The horizontal partition includes servers, clients, and mediator layers. Another approach is the facilitator (Genesereth, 1992), in which integration is based on the principle that any system (software or hardware) can inter-operate with any other system without the intervention of human users or their developers. This level of automation depends on supporting ontologies to describe the resources. Facilitators use meta-level information in converting, translating, or routing data and information. Along this direction, several integration tools have been developed for specific application domains. For example, TSIMMIS (Chen, Yerneni, Vassalos, Garcia-Molina, Papakonstantinou, & Ullman, 1998) focused on developing mediator-based tools to access and combine information from heterogeneous data sources. Infomaster (Genesereth, Keller & Duschka, 1997) is an information integration system that utilizes a facilitator as integration infrastructure and dynamically determines an appropriate way to answer a user's query and harmonizes the heterogeneities among different sources. InfoSleuth (Nodine, Fowler, Ksiezyk, Perry, Taylor ,& Unruh, 2000) extends Carnot (Huhns, Jacobs, Ksiezyk, Shen, Singh, & Cannata, 1993) as a mediator-based, agent-oriented infrastructure for semantic integration of information sources in open and dynamic environments. The C3DS project (Shrivastava, Bellissard, & Lacourte, 2001) focused on developing distributed object technologies to create a framework for complex service provisioning.

BUSINESS-CENTRIC KNOWLEDGE-ORIENTED ARCHITECTURE

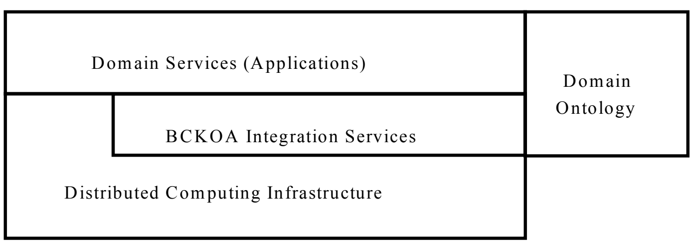
Now, it is clear that the development of an architecture for an *eMarketplace* requires a new design paradigm, improved integration architectures, and services. The architecture must be semantically rich and describe the organization and the interconnection among the software components, business services, and business ontologies of the *eMarketplace*. These should not be viewed as alternatives to existing technologies, but rather as advanced features that are implemented at a higher level of abstraction. In this architecture, the *eMarketplace* is a cooperative, distributed system composed of economically motivated software agents that interact cooperatively or competitively, find and process information, and disseminate it to humans and to other agents. The architecture also supports common economic services and commercial transactions, such as pricing, negotiation, and automated supply chains, as well as cross-enterprise integration and cooperation. In this work, we deal with both the *fundamental* and the *practical* issues of integration.

Fundamentally, we view integration as an abstraction level at which a distributed system environment can be described as collective, coherent universe of cooperative entities. In a cooperative, distributed system, integration is captured at the foundation architecture that supports all the entities' individual architectures and, therefore, the complete computing environment. Here we describe a business-centric, knowledge-oriented architecture (BCKOA) for cooperative distributed systems. The BCKOA specifications provide the abstraction to support the domain entities and applications independent of any specific technology. The main elements of BCKOA include domain services, integration services, and domain ontology. A key to BCKOA is a service-oriented model in which the overall connectivity of the system supports a "virtual" point-to-point integration mechanism. Therefore, unlike classical tier-oriented architectures, BCKOA supports an abstraction for an *ad hoc* layered architecture without an explicit distinction between application layers, as depicted in Figure 1.

To support heterogeneity and technology-independent properties at the system level, the boundaries between the layers correspond to standardized interfaces (e.g., OMG CORBA). However, industrial standards for interfaces do not support or supply common underlying semantics. To deal with this issue, BCKOA includes a domain ontology. The domain ontology captures and implements the conceptualization of an application domain at the knowledge level. BCKOA provides three families of integration services:

1. Ontology and semantic integration services support the semantic manipulations needed when integrating and transforming information or knowledge to satisfy a BCKOA task, as well as the capabilities required to reuse components.
2. Coordination and cooperation services support ad hoc and automated BCKOA configurations. This includes locating and discovering domain and/or BCKOA services that are potentially relevant to a domain or BCKOA service.
3. Wrapping services make different applications, components, objects, or modules comply with internal or external standards. Such standards may involve the interface to the software system or its behavior.

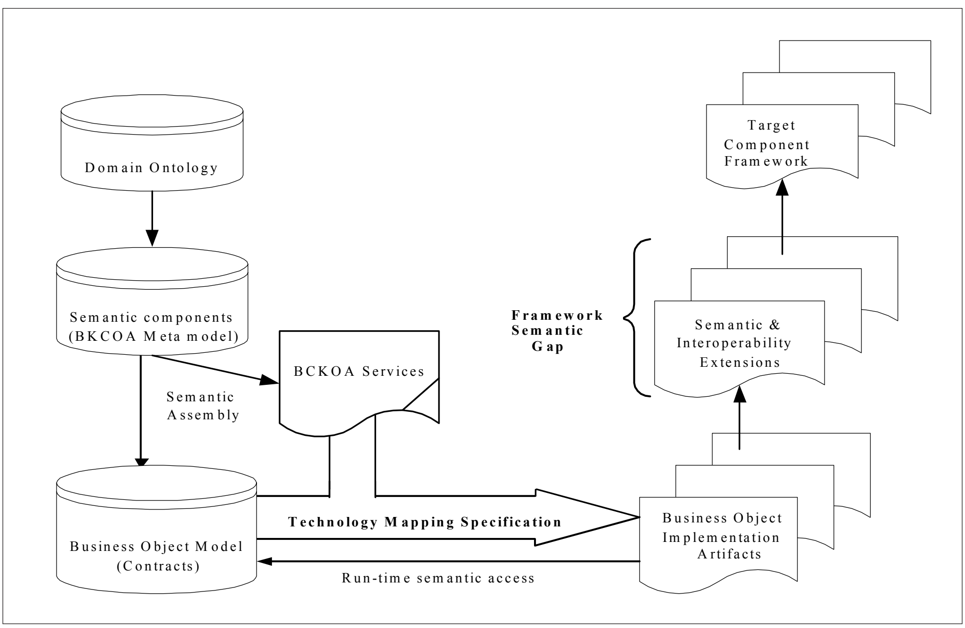
Figure 1: BCKOA as a Service-Oriented Architecture



The specifications of BCKOA services are independent of any component framework, but their implementation can be based on the services provided by the target framework, as is described below.

A key challenge in putting BCKOA into a practical context is the transformation or the mapping of its abstract description to the specification of the target component framework. In fact, most distributed computing environments, especially those supporting cross-enterprise integration like *eMarketplaces*, will likely include several different component technologies, such as CORBA, J2EE, and COM+. To deal with this issue, BCKOA requires that the business object implementations, regardless of their component technology, be obligated to conform to the domain ontology. The business object specification itself in the domain ontology becomes the reusable component that can be configured and assembled into multiple solutions (business objects), independent of technology implementation. Therefore, the domain ontology in BCKOA governs the structural and the behavioral semantics of the business objects in a way that is consistent across all implementations, and is accessible from any implementation. The BCKOA framework, shown in Figure 2, provides an integrated execution environment for integrated business object implementations. Mapping a BCKOA description to an implementation framework is driven by three specifications: domain ontology description, maps, and a profile. Technology mapping specifications include a map to specify a transformation from the BCKOA domain ontology and services to the implementation components and

Figure 2: BCKOA Framework



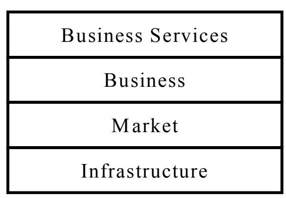
service extensions for the target component framework. The mapping of each business concept representation to its implementation is managed by a *profile*, which is a set of properties that defines the environment for a mapping. This mechanism enables an automated transformation from a relatively stable domain ontology and service description to different component technologies.

BCKOA-BASED eMARKETPLACE

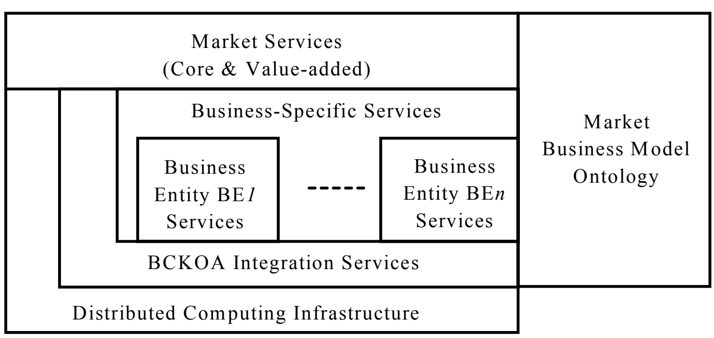
This section describes the use of BCKOA for achieving business integration in an *eMarketplace* with a specific focus on demonstrating its feasibility as a design paradigm for a cooperative, distributed system that is independent of any specific technology. Based on the assumption that an *eMarketplace* is a coherent, service-oriented universe, the BCKOA-based *eMarketplace* is shown in Figure 3(b), which builds upon the abstraction architecture of the *eMarketplace* in Figure 3(a) (Ghenniwa, 2001).

The lower layer of the *eMarketplace* architecture in Figure 3(a) is the infrastructure that might represent one or more physical network-based environments in which *eBusiness* systems can exist. The BCKOA representation, in Figure 3(b), supports the *eMarketplace* infrastructure using two layers: the

Figure 3: Use of BCKOA for the Architecture of an eMarketplace



(a) Abstraction layers for *eMarketplace*



(b) BCKOA-based *eMarketplace*

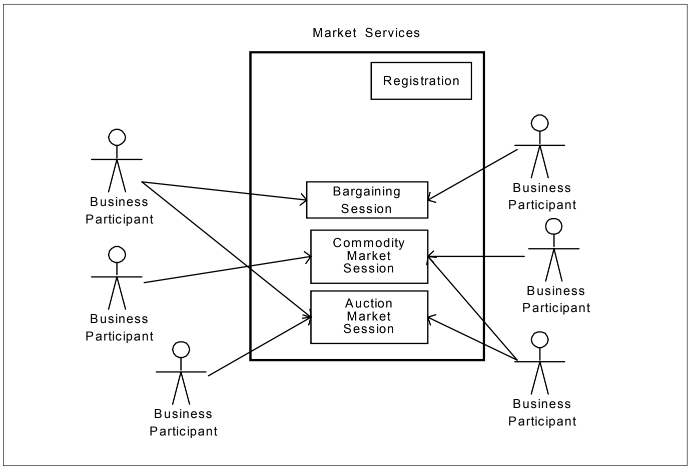
distributed-computing layer and the integration-services layer. The assumption is that this infrastructure might support various markets for providing or obtaining specific goods and services. Yet, each *eMarketplace* may be independent and may support its own rules, procedures, and protocols as described by the market layer.

The market layer may support several business domains as described in the business layer. BCKOA, in Figure 3(b), provides the integration between the business context of the market, which is described by the market business ontology, and the services provided by the participant entities in the business layer. Note that a business entity may participate in multiple *eMarketplace*s. A bank, for example, could participate in several markets, such as an investment management market, mutual fund management market, and financial advisory market. In this setting, any activities or functions within a specific business entity could participate in an *eMarketplace* dedicated to satisfying its needs as described by the business ontologies of the market.

In BCKOA, the business-service layer is supported by three types of services, depicted in Figure 3(b): (1) business-specific services, (2) business-entity services, which represent the implementation of the business services by the specific business entities, and (3) market services, which are categorized further into core, such as dynamic trading and supply-chain services, and value-added, such as procurement process services. Here we focus on the core services. Ideally, the market services of the *eMarketplace* should be able to offer a wide variety of coordinating and trade mechanisms to fit with multiple business models. Because of the shortcomings of each type of market structure, there is no "one-size-fits-all" structure. Based on the success of applying economic theories in the real world as a sustainable model for exchanging and regulating resources, goods and services, we propose to apply a flexible computational economy framework for the market services layer. Therefore, a BCKOA-based *eMarketplace* incorporates mechanisms for different types of market structures, which are viewed as a set of sessions, as shown in Figure 4, such as commodity and auction markets.

Various suppliers and consumers can register in the commodity market session. Given a system of prices, each participant decides upon a course of action, which may consist of the sale of some commodities and the purchase of others. Thus supply and demand functions for each commodity can be defined as the aggregate behavior of all the participants as described in the following section. These are determined by the set of market prices for the various commodities. Equilibrium for the economy is established when supply is equal to demand (i.e., the excess demand function has a zero value). Practically, it will be sufficient to find approximate equilibrium in the sense of finding a price that makes the values of the excess demands close to zero. One approach is the tâtonnement ("groping") process (Walras, 1874). With tâtonnement, each

Figure 4: BCKOA Market Sessions



individual price is raised or lowered according to whether that commodity's excess demand is positive or negative. Then, new excess demands are measured, and the process is iterated. The tâtonnement process does bring about convergence to an equilibrium price under the hypothesis of "gross substitutes," i.e., increasing the price of a commodity (say C_j) while holding the others constant will bring about an increase in excess demand in all commodities other than the C_j. In fact, this approach is naturally appealing for automation and direct implementation (Wellman, 1993).

In the auction-based market session, each participant (consumer and supplier) acts independently and contracts to buy or sell at a price agreed upon privately. Here we focus on private-value auctions, such as the Vickrey mechanism (Vickery, 1961), because it provides a market mechanism that is simpler, but more efficient and more stable than open auction mechanisms and classical sealed bid auctions (Varian, 1995). Furthermore, it promotes truthful bidding (or revealing reservation prices as the dominant strategy) among self-interested agents and thus avoids the need for counter speculation. Hence, it combines several advantages. It does not require iterative negotiation strategies, dynamic strategic behavior, or a high degree of security, since truth telling is the dominant strategy and the agents do not need to hide their reservation prices. While it is a simple yet powerful mechanism, the Vickrey mechanism may not be appropriate in all domains. For example, truthful bidding is not necessarily the dominant strategy for domains where an agent's marginal costs (and thus its reservation price) are determined by other agents' valuations, such as the case with *public-value auctions* in the stock market (Sandholm & Lesser, 1995).

In a BCKOA *eMarketplace*, supply-chain integration and management is treated as a coordination methodology that manages information and material flows, production operations, and logistics. The objective is to provide an automated coordination mechanism for the participants in a supply chain. The adopted development framework makes use of the coordination methodologies reported in Van Dyke and Parunak (1996) and Singh (2000). In this work, we particularly extended Singh's application to supply-chain management and B2B interactions (Ivezic, Barbacci, Libes, Potok, & Robert 2000). The methodologies, as described later, promote the interchange of standard business documents and compensate for exceptions that might occur during execution. This methodology requires that the participant business entities in a cooperative supply chain only describe their supply processes using Open Applications Group (OAG) standard business documents and UML interaction diagrams. These are converted automatically into roles and specifications for software agents. These sets of agents then cooperate in automating the resultant supply chain.

A combination of the market services and the business-entity services can be used to generate different business models of an *eMarketplace* as desired by the participating business entities. This structure enables a business entity to integrate and describe the types of business services offered and the information needed to use a particular service offering within the *eMarketplace*. The details of each service type and the required information might vary among business entities, although the definition of the service type is based on some common conventions described for an *eMarketplace*.

BCKOA recognizes the integration services as separate functionalities. Yet, they can be ubiquitously integrated in an *ad hoc* structure to fulfill a complex business service or a market structure. The interaction mechanisms supported by the integration layer describe both the pattern and protocol of exchanging messages between the services.

AGENT-ORIENTED BCKOA FRAMEWORK FOR eMARKETPLACE

All services (business, market, and integration) in a BCKOA-based *eMarketplace* usually involve complex and nondeterministic interactions, often producing results that are ambiguous and incomplete. Auctions and *ad hoc* service integrations are some examples. In addition, the dynamic nature of the environment requires that the components of the system be able to change their configuration to participate in different, often simultaneous roles in *eMarketplaces*. These requirements could not be accomplished using traditional ways of manually configuring software. For this domain, we strongly believe that agent-orientation is a very promising design paradigm for integration. In fact, such a paradigm is essential to model an open environment such as an *eMarketplace*, especially considering the multiple dynamic and simultaneous roles a single business entity may need to participate in given *eMarketplace* sessions (a financial services organization may have representatives acting on its behalf simultaneously within the context of brokering, service provisioning, and marketing).

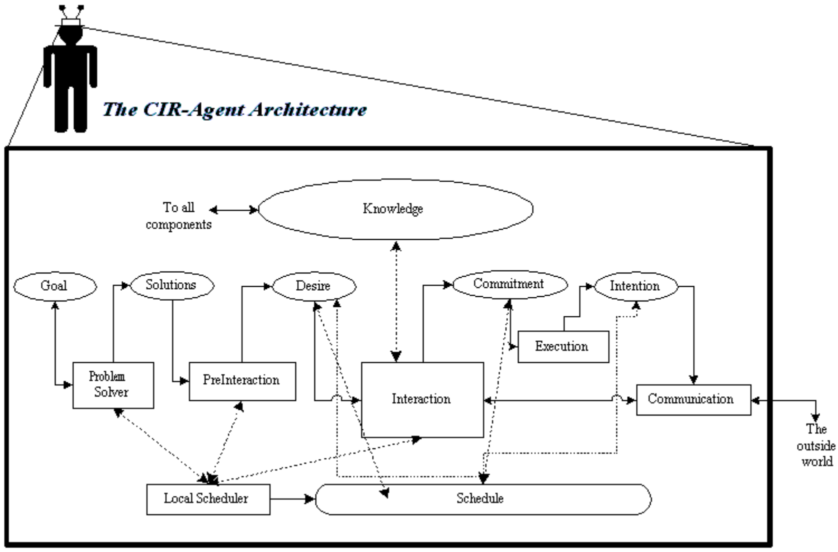
Software agent technology provides the next step in the evolution of computational modeling, programming methodologies, and software engineering paradigms. The first principle of agenthood is that an agent should be able to operate as a part of a community of cooperative, distributed systems environment, including human users. In our view, an agent can be described as an individual collection of primitive components that provide a focused and cohesive set of capabilities. Figure 5 depicts the Coordinated Intelligent and Rational, Agent (CIR-Agent) model (Ghenniwa & Kamel, 2000). The basic components include a problem solver, interactions, and communication, as shown in Figure 5(b). A particular arrangement or interconnection of the agent's components is required to constitute an agent, as shown in Figure 5(a). This arrangement reflects the pattern of an agent's mental state as related to its reasoning to achieve a goal. However, no specific assumption is made on the detailed design of the agent components. Therefore, the internal structure of the components can be designed and implemented using object-oriented or any other technology, provided that a developer conceptualizes the specified architecture of the agent as described in Figure 5(b).

A CIR-Agent model provides software engineers with features at a higher level of abstraction that are useful for cooperative environments. It supports flexibility at different levels of the design: system architecture, agent architecture, and agent component architecture. These degrees of flexibility allow information systems to adapt to changes with minimum requirements for redesign. Based on this view, an agent within the context of a BCKOA-based *eMarketplace* might play roles as described by the functionality of its problem-solving component and be able to coordinate, cooperatively or competitively, with the other agents, including humans. Therefore, an agent's role can be categorized as user-interface, business-specific service, business-entity service, market service, or integration service.

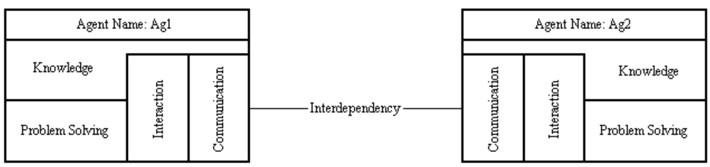
User interface agents play an important and interesting role in many applications. The main functionality of user interface agents is to support and collaborate with users in the same work environment to achieve the users' goals.

Business-specific service agents are specialists that provide a collection of business services available in the *eMarketplace*. Performing the functionality of a business service is typically the cooperative integration of several agents, including business-entity agents and market service agents. A business-entity service agent may be a representative in the *eMarketplace* for some function-

Figure 5: The CIR-Agent Architecture



(a) Detailed Architecture of CIR-Agent



(b) Logical Architecture of CIR-Agent

ality that is based on legacy applications or libraries, such as a product catalogue web site.

Market service agents are specialists that provide a collection of functions for the generic *eBusiness* in the *eMarketplace* environment in which a single entity (usually an agent) can perform its tasks in the *eMarketplace*. Market services, both value-added and core services, are horizontal, i.e., services that are used in several business domains by several business entities. Here the focus is on core services, particularly dynamic trading services, commodity market and Vickery auctions, and supply-chain integration and management.

The commodity market service governs the trading behavior of the participant business entities in the session as described in Figure 6. This service recognizes three types of agents, namely, market-mediators, consumers, and suppliers. Consumers and suppliers are roles assigned to agents or types of a business-entity service or user interface. These roles will be assigned upon registration with the market session. Each market session is assigned to a mediator to coordinate the actions taken by consumers and suppliers in a way that will eventually clear its respective market. There is a one-to-one correspondence between market mediators and commodities. Initially, a mediator agent is assigned to a specific commodity market and broadcasts a randomly chosen initial price vector to all registered participants in its market. Then, each participant computes the demand function for each of its commodities of interest. Each demand function specifies the net quantity demanded of a commodity (which for a net supply is negative) as a function of its price, assuming that the prices for the remaining commodities are constant. The participants then send these demand functions — the *bids* — to the respective mediator for each commodity. Each mediator, upon receiving the bids from all participants, computes the clearing price, for which the aggregate excess demand is zero. The mediator then notifies the participants of the new price. Upon seeing new prices, the consumers and suppliers compute revised demand functions as necessary based on these new prices, and send these updated bids to the mediator. This process continues until the pricing changes are within a specified threshold. Then the process terminates and the mediator reports the final state of the price vector as the equilibrium.

The **auction market session** also recognizes three types of agents representing suppliers, auctioneer, and buyers. However, the trading process mainly involves the auctioneer and buyers (or bidders). Each bidder agent declares its valuation function to the auctioneer. Under the general Vickery mechanism, it is in the interest (the dominant strategy) of the bidder to report its true valuation function. Then, the auctioneer agent:

- Calculates the allocation (x^*_i) that maximizes the sum of the bids subject to the items constraint.
- Calculates the allocation ($x^*_{\sim i}$) that maximizes the sum of the bids other than that of bidder agent *i* such that it excludes all items allocated to agent *i*.
- Announces the winners and their payment given by $p_i = \Sigma_{j \neq i} v_j(x^*_{\sim i}) - \Sigma_{j \neq i} v_j(x^*)$.

Under the assumption of quasi-linear preferences, each bidder agent calculates its utility. For bidder agent *i* the utility will be $u_i(x^*) = v_i(x^*) - p_i = v_i(x^*) - \Sigma_{j \neq i} v_j(x^*_{\sim i}) - \Sigma_{j \neq i} v_j(x^*)$.

Agent-based, supply-chain integration and management in an *eMarketplace* is a cooperative, distributed problem-solving activity. Here we briefly describe the methodology and an agent-based system for supply-chain integration. Our methodology is supported by a development environment that automatically generates specifications for the agents in the supply chain. Business participants need only describe their supply processes using OAG standard business documents and UML interaction diagrams. The methodology

Figure 6: Tâtonnement Model of the Commodity Market

Trading behavior of the Mediator Agent

Initialize price to $p_g = 1, \ \forall g \in [1, \dots, n-1]$

Set λ_g to a positive number $\forall g \in [1, \dots, n-1]$

Repeat

> Broadcast **p** to **participants** (consumers and suppliers)
>
> Receive a production plan y_j from each supplier *j*
>
> Broadcast the plans y_j to consumers
>
> Receive a consumption plan x_i from each consumer *i*
>
> *For g* =1 to *n – 1* (adjust price as follows)
>
> $p_g = p_g + \lambda_g (\sum_i (x_{ig} - e_{ig}) - \sum_j y_{jg})$

Until $|\sum_i (x_{ig} - e_{ig}) - \sum_j y_{jg}| < \varepsilon \ \forall g \in [1, \dots, n-1]$

Broadcast the *equilibrium message*

Trading behavior of Consumer Agent ^{i}

Repeat

> Receive **p** from the Mediator
>
> Receive a production plan y_j for each *j* from the Mediator
>
> Announce to the Mediator the consumption plan $x_i \in \Re^n_+$ that
>
> *Maximizes* $u(x_i)$ s.t. $\mathbf{p}.x_j \le \mathbf{p}.\mathbf{e}_i + \sum \theta_{ij} .\mathbf{p}. y_j$

Until *equilibrium message* is received

Perform the Exchange

Trading behavior of Supplier Agent ^{j}

Repeat

> Receive **p** from the Mediator
>
> Announce to the Mediator the production plan $y_j \in Y_j$ that
>
> *Maximizes* $\mathbf{p}.y_j$

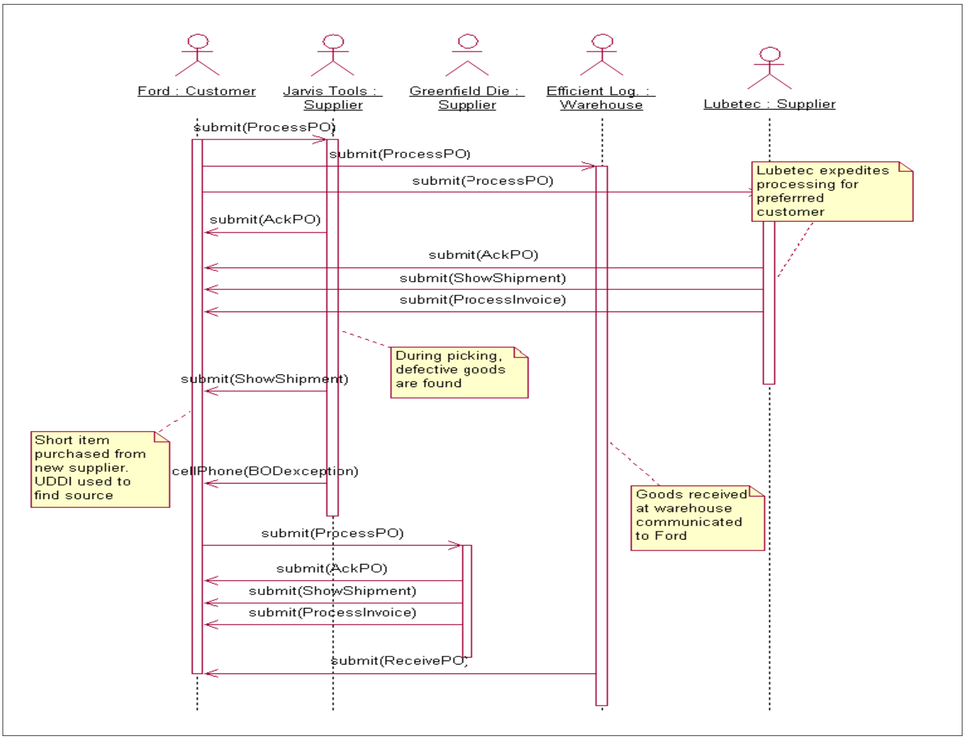
Until *equilibrium message* is received

Perform the Exchange

begins with capturing a supply-chain scenario and its associated UML interaction diagrams, exemplified in Figure 7. The interactions, in Figure 7, consist of the exchange of structured documents, such as the OAG business object documents (BODs).

For B2B interactions, a ProcessPO BOD is a *directive* that carries the composite semantics of *request* and *inform*, in which the sender requests that the recipient evaluate a purchase order and inform the sender of the results. The

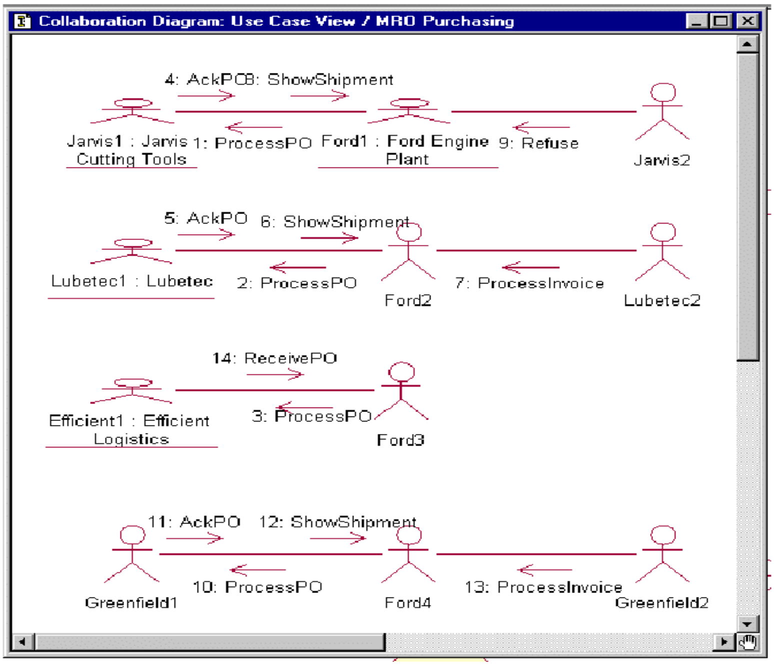
Figure 7: Interaction Diagram for the OAG Scenario Involving Ford and its Suppliers



informal semantics is that ProcessPO will be followed by a response from the recipient and that the response will be either an AckPO or a DeclinePO. Using the semantics of each document, the messages in the interaction diagram are converted into a bi-partite conversation graph (not shown here), which delineates each participant's conversations. A bi-partite conversation graph is used to identify the roles of the participants in B2B transactions. This graph is the basis for constructing Dooley graphs, shown in Figure 8, as collaboration diagrams. Note that collaboration participants can fill different roles at different times, and thus can be involved in many conversations simultaneously. Each of the roles identified in the collaboration diagram can be assigned to an agent in the supply chain. Moreover, the diagram for each role is converted directly into a state-machine description for the agent's behavior, enabling automatic agent generation. Agents are then allocated for the corresponding role and business entity, and then collectively they manage the supply-chain process. Figure 9 shows several of the state-machine behavioral descriptions.

The methodology, summarized in Figure 10, uses — and begins to formalize — the BODs that OAG and RosettaNet are standardizing. It provides a basis for

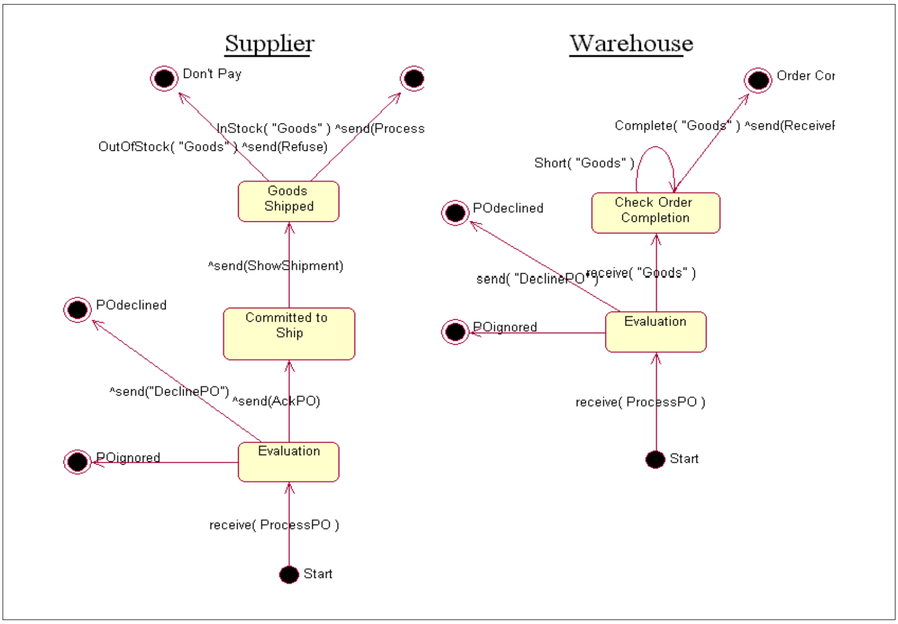
Figure 8: Collaboration Diagram with Participant Roles for Ford Interoperability Scenario



the convergence of multiple standards for supply-chain management, which could become ready-to-use technology for different participant business entities in the *eMarketplace*.

Integration agents are specialists that provide a collection of integration functions for a cooperative distributed system in which a single entity (agent, component, object, etc.) can perform its tasks. Integration services are used by several distributed entities. For example, a brokering agent provides a capability-based integration service in the *eMarketplace*. The brokering agent allows agents (for integration, market, or business services) to describe the properties of a requested service. Then, on behalf of the requester, it establishes interactions with service providers to fulfill the requests. The brokering agent is responsible for identifying and interacting with other integration services, such as resource discovery services and ontology manager services to accomplish its tasks. Another type of integration agent provides view integration, which is a service to merge and map the description of business objects (e.g., source schemas) in the *eMarketplace* supported by the business ontology into an integrated view or schema. For instance, a catalogue service might require information provided by several business entities supporting different product schema. A view-integration service provides the integration into a common

Figure 9: State-Machine Diagrams for Enacting Agents that Implement Supply-Chain Processes

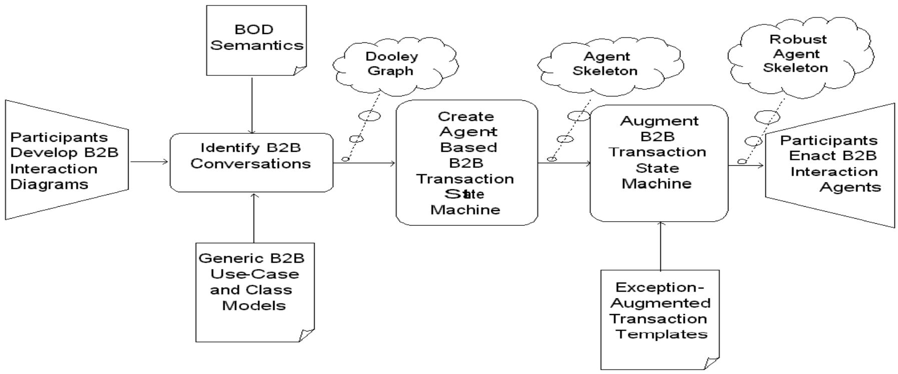


definition language (e.g., XML-based), which is in turn mapped into a target representation language by a specialized language mapping service. View integration is responsible for identifying and interacting with several services to fulfill its functionality, including brokering, source-schema, ontology, and language-mapping services.

PROTOTYPE IMPLEMENTATION OF BCKOA

To validate and experiment with our theoretical analysis and foundations described in the previous sections, we have developed a prototype of an agent-oriented BCKOA for an *eMarketplace,* as shown in Figure 11. *ABC* Corp and *XYZ,* Inc. are virtual business entities registered with the *eMarketplace* for both purchasing and sales services. Both organizations use a BCKOA-based computation environment. As illustrated in Figure 11, individual customers or business-entity personnel in the *eMarketplace* can participate in the market through their user interface agent. Similarly, each business-entity service is represented by an agent in the *eMarketplace*. These agents provide thin, intelligent, highly autonomous interfaces for the business-entity services that might be based on

Figure 10: Agent-Based Coordination Methodology for B2B Automation



legacy applications. For example, in Figure 11, the ABC purchasing-service agent represents the implementation of the business-specific purchases by ABC in the *eMarketplace*. Each user interface and business-entity service agent is registered in the *eMarketplace*. Thus, a user interface agent can benefit from the market, business-specific, and business-entity services by interacting with their representative agents. Each business-entity service must also be registered with a registry agent for the corresponding business-specific service. Each layer, and its registry services, are intended to provide some aspect of information

Figure 11: Prototype of BCKOA-Based eMarketplace *Using JADE*

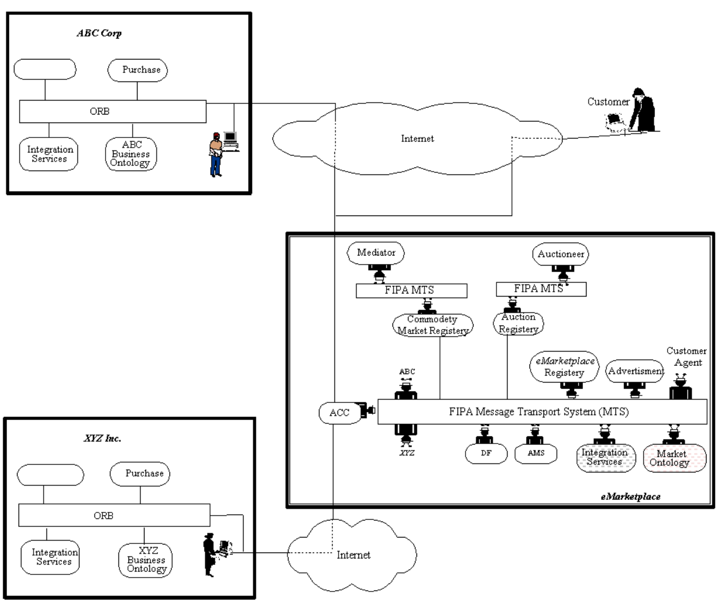
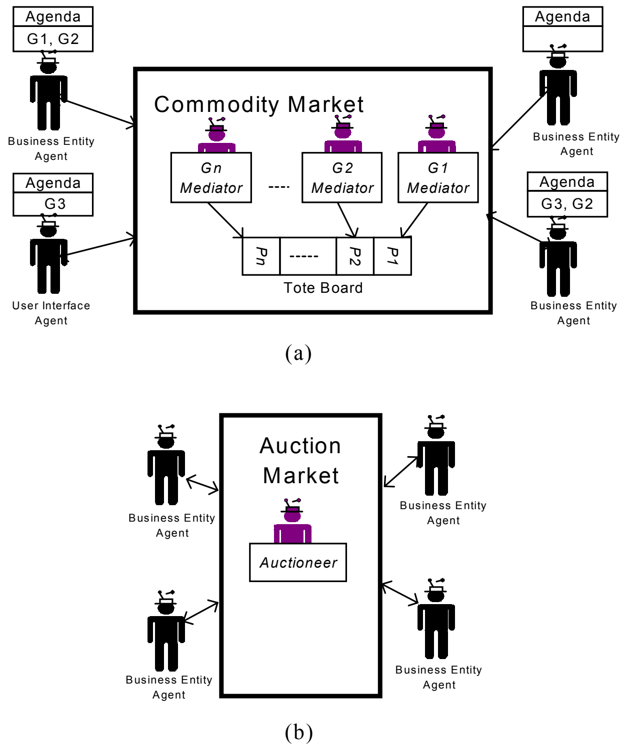


Figure 12: Logical Design of (a) Commodity Market and (b) Auction Market Sessions



about the *eBusiness* environment and enable an interested party to obtain information to potentially use offered services, or to join the *eMarketplace* and either provide new services or inter-operate as a trading partner with other business entities in that *eMarketplace*.

In the current prototype, we experiment with commodity and auction market structures, as shown in Figure 12, where customers and suppliers are brought together to trade with each other and prices are set by the selected market structure. The trading behavior of the participant agents is governed by the selected market structure as described previously. An individual customer is able to participate in the market through a dedicated user interface agent possibly assigned by the *eMarketplace*. Similarly, each participating business entity is assigned to a team of CIR-Agents for the registered services and representative personnel who might have a direct contact with the market, as well as with the customers. The market ontology provides a conceptualization of the domain at the knowledge level.

The implementation utilizes the JADE platform (Bellifemine et al., 1999), which is a software framework to develop agent applications in compliance with the FIPA specifications (Foundation for Intelligent Physical Agents, 1998) for multi-agent systems. It deals with all aspects external to agents that are independent of their applications, such as message transport, encoding and parsing, and agent lifecycle. JADE supports a distributed environment of agent containers, which provide a run-time environment optimized to allow several agents to execute concurrently. This feature has been utilized to create several concurrent market sessions, such as commodity and auction sessions. A complete agent platform may be composed of several agent containers. Communication in JADE, whether internal to the platform or externally between platforms, is performed transparently to agents. Internal communication is realized using Java Remote Method Invocation to facilitate communication across the *eMarketplace* and its market sessions. External non-Java-based communication between an *eMarketplace* and its participating organizations is realized through the Internet InterOrb Interoperability Protocol mechanism or http. JADE provides support for standard FIPA ontologies and user-defined ontologies. Although our implementation takes advantage of the JADE platform and its supporting agents, such as a nameserver and a directory facilitator, the architecture of the application agent is based on the CIR-Agent model (shown in Figure 5). Java features, such as portability, dynamic loading, multi-threading, and synchronization support make it appropriate to implement the inherent complexity and concurrency in an *eMarketplace*. These features were also instrumental for executing the CIR-Agents in parallel. The design of each agent is described in terms of its knowledge and capabilities. The agent's knowledge includes the agent's self-model, goals, and local history of the world, as well as a model of its acquaintances. The agent's knowledge also includes its desires, commitments, and intentions as related to its goals.

The main capabilities of the CIR-Agent include communication, reasoning, and domain actions. Implementation of the communication component takes advantage of JADE messaging capabilities. It is equipped with an incoming message inbox, whereby message polling can be both blocking and non-blocking, and with an optional time-out mechanism. Messages between agents are based on the FIPA ACL. The agent's reasoning capabilities include problem-solving and interaction devices. The problem solving of an agent is implemented through the use of complex behaviors. Behaviors can be considered as logical execution threads that can be suspended and spawned. The agent keeps a task list, containing active behaviors. The problem-solving component varies from one agent to another. The agent behaviors can be classified as follows: behaviors that are concerned with market services, such as a market-registry service, advertisement service, mediator and auction service, and behaviors that are concerned with providing business-specific services, such as selling and purchasing.

RELATED WORK AND DISCUSSION

There have been several recent attempts to promote *eMarketplace* models by the academic and industrial communities. For example, the electronic marketplace (EMP) (Boll, Gruner, Haa,f & Klas, 1999) is an attempt to develop a business-to-business system architecture. It is viewed as a DBMS solution to support many-to-many relationships between customers and suppliers. The Global Electronic Market (GEM) system (Rachlevsky-Reich, 1999; Ben-Shaul et al., 1999) attempted to develop a logical market framework and infrastructure. A main objective was to separate system-related and market-related design issues. In GEM, the market provides trading mechanisms that include bids and offers. A more complex architecture for *eMarketplace* is MAGMA (Tsvetovatyy, Gini, Mobasher, & Wieckowski, 1997), with its special focus on the infrastructure required for conducting commerce on the Internet. In MAGMA, the *eMarketplace* has been viewed in terms of three main functionalities, namely, traders, advertising, and banking. Alternatively, OFFER (Bichler, Beam, & Segev, 1998) proposed a brokering-based architecture marketplace. In OFFER the *eMarketplace* was viewed as a collection of suppliers, customers, and brokers. A customer can search for a service either directly in the e-catalogue of the supplier or use the e-broker to search all the e-catalogues of the suppliers that are registered with this broker. E-brokers employ a simple auction mechanism. In a different approach, MOPPET (Arpinar, Dogac, & Tatbul, 2000) proposed an *eMarketplace* system as agent-oriented workflows. MOPPET viewed the market as a workflow management system carried out by several types of agents: task, scheduling, facilitator, and recovery agents.

Another approach was driven by the bottom-up modeling of market processes with self-organizing capabilities (Arthur, Holland, LeBaron, Palmer, & Tayler, 1997). The objective was to develop a computational study of economies modeled as evolving systems of autonomous interacting agents, and known as agent-based computational economics (ACE) (LeBaron, 2000; Timmers, 1999). The ACE researchers relied on computational laboratories (McFadzean, Stewart, & Tesfatsion, 2001) to study the evolution of decentralized market economies under controlled experimental conditions. The goal was to develop analysis tools that enable an economist to test economic theories developed using standard modeling approaches.

Several companies have emerged to automate logistics and resupply within specific industrial segments. For example, Ariba (2000) developed a marketplace based on procurement portals and dynamic exchanges for horizontal marketplaces. Ariba Dynamic Trade, for instance, attempts to provide dynamic trade mechanisms, such as auctions, reverse auctions, and bid/ask exchanges and negotiations. SAP Service Marketplace (SAP Services Marketplace, SAP AG) is an Internet portal for the SAP community. It provides basic online services such as catalogue browsing, matchmaking, and ordering from SAP and its partners. Other approaches were directed to support vertical marketplaces, such as PaperExchange (Paperexchange Marketplace), that enables customers and suppliers to negotiate pricing and transact directly with one another. PaperExchange also attempts to provide several supporting services, such as logistics and clearing services, industry-specific job listings, industry events, news headlines, and a resource directory. VerticalNet (VerticalNet® Marketplaces) also built a set of Web-based marketplaces for specific industrial segments, such as financial services, healthcare, and energy. Each web site forms a community of vendors and customers in a specific area. Vertical trade communities are introduced in segments with a substantial number of customers and suppliers, fragmentation on both the supply and demand sides, and significant online access.

Another direction adopted by many major software vendors is to develop Internet-based commerce platforms. Examples are IBM CommercePOINT (IBM Corporation, CommercePOINT Payment.), Microsoft Site Server Commerce Edition (Microsoft Corporation, Internet Commerce 1998), Oracle Internet Commerce Server INTERSHOP (Intershop Communications, Inc., 1998), and Sun JavaSoft JECF (Java Electronic Commerce Framework, Sun Microsystems). These proprietary attempts focus on providing infrastructure services such as security payment directories and catalogues to be integrated with existing systems and the Web.

The proposed agent-oriented BCKOA *eMarketplace*, however, provides a framework of enterprise integration that deals with several systems and business issues. Unlike the above-mentioned attempts, it is fundamentally based on business integration rather than systems integration. The objective is to develop an architecture that is semantically rich in describing an organization and the interconnection among all elements of the *eMarketplace*, including people, business services, software components, and business ontologies. Technologically, BCKOA is service-oriented in the sense that it enables business entities and their supporting systems to join the market at the highest abstraction level (service level) with minimum overhead and independent of specific technology. Utilizing BCKOA, the business object implementations of the participating services can be integrated in the execution environment. In addition, BCKOA provides an appropriate architecture for an *eMarketplace*. The form of the architecture supports *eMarketplace* functions that are inherited from real-world marketplaces. They are complex and non-deterministic, yet they characterize a real business environment. A BCKOA *eMarketplace* provides an integration environment for the broad-based services that are required for interaction and directory services, dynamic trading, cooperative supply-chain integration and management. Therefore, we believe that the BCKOA-based *eMarketplace* is appropriate for integrating horizontal business services, vertical business services, specific business functionalities, and the leveraging of legacy systems in a way that supports end-to-end integration. Furthermore, a BCKOA *eMarketplace* provides a wide variety of coordinating and trade mechanisms to fit multiple business models. In our research we have applied a flexible computational economy model for the market-services layer. Therefore, a BCKOA-based *eMarketplace* incorporates mechanisms for different types of market structures.

We believe that agent orientation is an adequate paradigm for producing the information architecture of next-generation *eBusiness* systems, especially *eMarketplaces*. Agent technology richly enables and supports the automation of complex tasks and yields systems that are reliable and able to assume the responsibilities of the *eMarketplace* in which they compete. The components of agent-based BCKOA, namely, *eMarketplaces*, business entities (products, suppliers, customers, etc.), and the foundation (integration) architecture and services that glue them together, are essential to building robust many-to-many value chains in emerging *eBusiness*.

CONCLUSION AND FUTURE WORK

This chapter presented ongoing research on developing an agent-oriented architecture for an *eMarketplace* that provides intelligent enterprise integration. The objective has been to develop a successful engineering foundation for the *eMarketplace*. To this end, several business and design issues have been identified for an *eMarketplace,* most importantly: to capture and enable many-to-many relationships between customers and suppliers, to provide services ranging from baseline interaction and directory services to specialty business-related services, and to integrate business organizations in an open market environment.

This chapter has described the agent-based Business-Centric Knowledge-Oriented Architecture (BCKOA) based on an abstraction-layered architecture for an *eMarketplace*. BCKOA is a service-oriented architecture for cooperative, distributed systems. BCKOA provides an abstraction of domain entities and applications independent of any specific technology. In putting the proposed solution into practical use, we described a methodology to map a BCKOA abstract description into the specification of target components.

A CIR-Agent model is used as the underpinning technology for the BCKOA based *eMarketplace* in which several types of agent roles were identified, namely, user-interface, business-specific service, market services, and integration services. A prototype of BCKOA using the FIPA-compliant platform, JADE is currently being developed for mutual fund management with redemption and purchasing functionalities.

The current focus is on demonstrating the feasibility and the effectiveness of our architecture as a design model for *eMarketplace*, with special attention to its foundation components, services within *eBusiness* applications, fundamental active representatives for specific business services, and leveraged legacy systems. In continuing our research, the computational effectiveness of the architecture will be our main concern. Also, we will expand the application and the implementation of our prototype *eMarketplace* to investigate the most appropriate techniques to support secure, reliable, and effective transactions.

REFERENCES

Ariba. (2000, October). *B2B marketplaces in the New Economy.* Retrieved from the World Wide Web: http://www.commerce.net/other/research/ebusiness-strategies/2000/00_07_r.html.

Arpinar, S., Dogac, S. A. & Tatbul, N. (2000, July). An open electronic marketplace through agent-based workflows: MOPPET. *International Journal on Digital Library,* 3(1).

Arthur, W. B., Holland, J., LeBaron, B., Palmer, R. & Tayler, P. (1997). Asset pricing under endogenous expectations in an artificial stock market model. In W. B. Arthur, S. N. Durlauf & D. A. Lane (Eds.), *Proceedings on the Economy as an Evolving Complex System*.

Bartelt, A. & Lamersdorf, W. (2001). A multi-criteria taxonomy of business models in electronic commerce. *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms.*

Bellifemine, F., Poggi, A. & Rimassa, G. (1999, April). JADE – A FIPA-compliant agent framework. *Proceedings of PAAM'99*, *London,* (pp. 97-108).

Bichler, M., Beam, C. & Segev, A. (1998). OFFER: A broker-centered object framework for electronic requisitioning. *IFIP Conference Trends in Electronic Commerce.*

Boll, S., Gruner, A., Haaf, A. & Klas, W. (1999, April). EMP-a database driven electronic marketplace for business-to-business commerce on the Internet. *Journal of Distributed and Parallel Databases, Special Issue on Internet Electronic Commerce*, 7(2).

Chen, L., Yerneni, R., Vassalos, V., Garcia-Molina, H., Papakonstantinou, Y. & Ullman, J. (1998). Capability based mediation in TSIMMIS. *Proceedings on SIGMOD 98 Demo*, Seattle, Washington, USA.

Commerce eXtensible Markup Language, cXML 1.2.007. (n.d.). Retrieved from the World Wide Web: http://www.cxml.org.

Common Business Library (CBL) of Commerce Net. (n.d). Retrieved from the World Wide Web: http://www.commerce.net.

Davenport, T. H. (1993). *Process Innovation: Reengineering Work through Information Technology*. Harvard, MA: Business School Press.

Distributed Component Object Model (DCOM). (n.d). Retrieved from the World Wide Web: http://www.microsoft.com/com/tech/DCOM.asp.

Dooley, R. A. (1976). Repartee as a graph. In R. E Longacre (Ed.), *An Anatomy of Speech Notions* (pp. 348-358). Lisse: Peter de Ridder Press.

Dublin core. (n.d). Retrieved from the World Wide Web: http://dublincore.org.

Dubosson, M., Osterwalder, A. & Pigneur, Y. (2002). eBusiness model design, classification and measurements. *Thunderbird International Business Review*, 44(1).

Eisenberg, B. & Nickull, D. (eds.). (2001, February). *ebXML Technical Architecture Specification v1.04.*

Fensel, D., Horrocks, I., van Harmelen, F., Decker, S., Erdmann, M. & Klein, M. (2000, October). OIL in a nutshell. *Proceedings of the 12th European Workshop Knowledge Acquisition Modeling, and Management* (pp. 1-16). Springer-Verlag.

Fensel, D., van Harmelen, F. & Horrocks, I. (2002). OIL & DAML+OIL: Ontology languages for the Semantic Web. In J. Davis, D. Fensel & F. van Harmelen (Eds.), *Towards the Semantic Web: Ontology-Driven Knowledge Management.* John Wiley & Sons.

The Foundation for Intelligent Physical Agents: *FIPA '97 version 2.0 specifications*. (1998).

Genesereth, M. (1992). An agent-based approach to software interoperability. In *Proceedings of the DARPA Software Technology Conference.*

Genesereth, M. & Fikes, R. (eds.). (1992, June). *Knowledge Interchange Format, Version 3.0 Reference Manual* (Technical Report Logic-92-1). Computer Science Department, Stanford University, USA.

Genesereth, M., Keller, A. M. & Duschka, O. (1997). Infomaster: An information integration system. *Proceedings of 1997 ACM SIGMOD Conference*.

Ghenniwa, H. (2001, November). eMarketplace: Cooperative distributed systems architecture. *Fourth International Conference on Electronic Commerce Research*, Dallas, Texas, USA.

Ghenniwa, H. & Kamel, M. (2000). Interaction devices for coordinating cooperative distributed. *Intelligent Automation and Soft Computing,* 6(2), 173-184.

Hammer, M. & Champy, J. (1993). *Reengineering the Corporation*. Harper Collins.

Heflin, J., Hendler, J. & Luke, S. (1999). *SHOE: A Knowledge Representation Language for Internet Applications* (Technical Report CS-TR-4078). Department of Computer Science, University of Maryland at College Park, USA.

Huhns, M., Jacobs, N., Ksiezyk, T., Shen, W., Singh, M. & Cannata, P. (1993, June). Integrating enterprise information models in Carnot. *International*

Conference on Intelligent and Cooperative Information Systems (ICICIS), Rotterdam.

IBM Corporation CommercePOINT Payment. (n.d). Retrieved from the World Wide Web: http://www.internet.ibm.com.commercepoint.payment.

Intershop Communications, Inc. (1998). Intershop 3. Retrieved from the World Wide Web: http://www.intershop.com.

Ivezic, N., Barbacci, M., Libes, D., Potok, T. & Robort (2000, July). An analysis of a supply-chain management agent architecture. *Proceedings of the Fourth International Conference on Multiagent Systems*. *IEEE Computer Society Press*, Los Alamitos, CA (pp. 401-402).

Java™ 2 Platform, Enterprise Edition (J2EE™). (n.d). Retrieved from the World Wide Web: http://java.sun.com/j2ee/.

Karp, P., Chaudhri, V. & Thomere, J. (1999, July). *XOL: An XML-Based Ontology Exchange Language* (XOL version 0.3).

Lassila, O. & Swick, R. (1999, February). *Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation*.

LeBaron, B. (2000). Agent-based computational finance: Suggested readings and early research. *Journal of Economic Dynamics and Control, 24*, 679-702.

McCabe, K., Rassenti, S. & Smith, V. (1992). Institutional Design for Electronic Trading. *Conference on Global Equity Markets*, *N.Y. University*, Salomon Center.

McFadzean, D., Stewart, D. & Tesfatsion, L. (2001). A computational laboratory for evolutionary trade networks. *IEEE Transactions on Evolutionary Computation, 5*, 546-560.

Microsoft Corporation. Internet Commerce. (1998). Retrieved from the World Wide Web: http://www.microsoft.com.

Nodine, M., Fowler, J., Ksiezyk, T., Perry, B., Taylor, M. & Unruh, A. (2000). Active information gathering in Infosleuth. *International Journal of Cooperative Information Systems*, 9, 3-27.

Oracle Corporation. Oracle Internet Commerce Server. Retrieved from the World Wide Web: http://www.oracle.com/products/asd/ics/ics.html.

Parunak, H. & Van Dyke. (1996). Visualizing agent conversations: Using enhanced dooley graphs for agent design and analysis. *Proceedings of the Second International Conference on Multiagent Systems*, *AAAI Press*, Menlo Park, CA (pp. 275-282).

Paperexchange Marketplace. (n.d). Retrieved from the World Wide Web: http://www.paperexchange.com.

Rachlevsky-Reich, B. et al. (1999). GEM: A global electronic market system. *Information Systems Journal, Special Issue on Electronic Commerce*, 24(6).

RossettaNet. (n.d). Retrieved from the World Wide Web: http://www.rosettanet.org.

Sandholm, T. & Lesser, V. (1995). On Automated Contracting In Multi-Enterprise Manufacturing. Distributed Enterprise: Advanced Systems and Tools, Edinburgh, Scotland, 33-4.

SAP Services Marketplace, SAP AG. (n.d). Retrieved from the World Wide Web: http://www.sap.com.

The Semantic Web Community Portal. (n.d). Retrieved from the World Wide Web: http://www.semanticweb.org/.

Shrivastava, S., Bellissard, L. & Lacourte, S. (2001). *Assessment of the C3DS Service Provisioning Framework.* Public Technical Report, No. 36.

Singh, M. P. (2000). Synthesizing coordination requirements for heterogeneous autonomous agents. *Journal of Autonomous Agents and Multi-Agent Systems, 3*(2), 107-132.

Smirnov, A. & Chandra, C. (2000, March). Ontology-based knowledge management for cooperative supply chain configuration. *Proceedings of AAAI Spring Symposium Bringing Knowledge to Business Processes.* Stanford, CA: AAAI Press.

Sun Microsystems. (n.d). Java Electronic Commerce Framework (JECF). Retrieved from the World Wide Web: http://www.javasoft.com/products/commerce.

Object Management Group, Inc. (1995, July). *The Common Object Request Broker Architecture and Specification (Revision 2.0).* Framingham, MA.

Tesfatsion, L. (ed.). (2001). Special double issue on agent-based computational economics. *Journal of Economic Dynamics and Control*, 25(3-4).

Timmers, P. (1999). *Electronic Commerce: Strategies and Models for Business-to-Business Trading*. John Wiley & Sons.

Tsvetovatyy, M., Gini, M., Mobasher, B. & Wieckowski, Z. (1997, September). MAGMA: An Agent-Based Virtual Market for Electronic Commerce. *Journal of Applied Artificial Intelligence, special issue on Intelligent Agents,* 11(6), 501-523.

Varian, H. R. (1995, July). Mechanism Design For Computerized Agents. The First USENIX Workshop on Electronic Commerce, New York, 11(19), 13-21.

VerticalNet® Marketplaces. (n.d). Retrieved from the World Wide Web: http://www.VerticalNet.com.

Vickery, V. (1961). Counter speculation, auctions, and competitive sealed tenders. *Journal of Finance,* 16(8), 37.

Walras, L. (1954), *Elements of Pure Economics*. English translation by William Jaffe. Allen and Unwin. (Originally work published 1874.)

Wellman, M. P. (1993). A Market-Oriented Programming Environment and Its Application To Distributed Multicommodity Flow Problems. *Journal of Artificial Intelligence Research,* 1, 1-22.

Wiederhold, G. (1992, March). Mediators in the architecture of future Information Systems. *IEEE Computer,* 25(3), 38-49.

Wiederhold G. & Genesereth, M. (1997, September/October). The conceptual basis for mediation services. *IEEE Expert, Intelligent Systems and their Applications,* 12(5), 38-47.