NRC Publications Archive Archives des publications du CNRC

Privacy-based computation model in e-business

Aburukba, R.; Massaud-Wahaishi, A.; Ghenniwa, H.; Shen, W.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. / La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

For the publisher's version, please access the DOI link below. / Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

Publisher's version / Version de l'éditeur:

<https://doi.org/10.1080/00207540902847421>

International Journal of Production Research, 47, 17, pp. 4885-4906, 2009-09-01

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=28002381-03a4-4c46-82ac-5e8c04af7511>

<https://publications-cnrc.canada.ca/fra/voir/objet/?id=28002381-03a4-4c46-82ac-5e8c04af7511>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

<http://www.nrc-cnrc.gc.ca/irc>

Privacy-based computation model in e-business

NRCC-51146

Aburukba, R.; Massaud-Wahaishi, A.;
Ghenniwa, H.; Shen, W.

September 2009

A version of this document is published in / Une version de ce document se trouve dans:
International Journal of Production Research, 47, (17), pp. 4885-4906,
September 01, 2009, DOI: [10.1080/00207540902847421](https://doi.org/10.1080/00207540902847421)

The material in this document is covered by the provisions of the Copyright Act, by Canadian laws, policies, regulations and international agreements. Such provisions serve to identify the information source and, in specific instances, to prohibit reproduction of materials without written permission. For more information visit <http://laws.justice.gc.ca/en/showtdm/cs/C-42>

Les renseignements dans ce document sont protégés par la Loi sur le droit d'auteur, par les lois, les politiques et les règlements du Canada et des accords internationaux. Ces dispositions permettent d'identifier la source de l'information et, dans certains cas, d'interdire la copie de documents sans permission écrite. Pour obtenir de plus amples renseignements : <http://lois.justice.gc.ca/fr/showtdm/cs/C-42>

National Research
Council Canada

Conseil national
de recherches Canada

Canada

Privacy-Based Computation Model in eBusiness

Raafat Aburukba^a, AbdulMotalib Masaud-Wahaishi^{a,b}, Hamada Ghenniwa^a, Weiming Shen^{a,c}

^aDepartment of Electrical and Computer Engineering, University of Western Ontario, London Ontario, Canada; ^bDepartment of Information Technology, United Arab Emirates University, Al-Ain, United Arab Emirates; ^cNational Research Council, London Ontario, Canada

Abstract

In the rapidly growing electronic business domain, organizations in the open environment are required to cooperate to achieve certain goals related to their business model. Given such an open environment, maintaining the privacy of organizations becomes a critical challenge. We believe the electronic marketplace (eMarketplace) is a promising architecture model for achieving cooperation between organizations. This is because of the nature of the market, where each entity selfishly attempts to achieve its own goals. The primary objective of this work is to model cooperation between different entities in the eMarketplace where the entities' privacy requirement is inherited. The proposed architecture, the eMarketplace, exists as a collection of software agents in the cooperative distributed systems. It enables and supports scheduling services between market participants. These services take into consideration any privacy desires that may be required from various participants in an open, dynamic, and heterogeneous environment.

Keywords: *cooperation, decentralized scheduling, privacy, eMarketplace, eBusiness.*

1. Introduction

Information technology has enabled, and in some cases has forced organizations to redefine their business models and to reorient their internal capabilities to exploit electronic business (eBusiness) techniques. They are finding it necessary to collaborate to build more efficient operations and supply chains that reduce time-to-market and costs. Electronic business is the use of the Internet along with other electronic means and technologies to conduct such collaborations within businesses, from businesses to consumers, among businesses, and from businesses to government (Ghenniwa et al., 2006).

An electronic marketplace (eMarketplace), however, appears to be a promising integration model for eBusiness. eMarketplaces provide an integrated environment for consumers, who depend on a variety of products and services that can spread across several suppliers or marketplaces. Likewise, they provide suppliers with the ability to reach, discover, and develop new customers within a single eMarketplace or across various eMarketplaces quickly with low cost. In general, eMarketplaces offer businesses the chance to develop and enhance their most

* Email: roaburuk@uwo.ca

important relationships—those with customers and suppliers. It enables the creation and leveraging of services and supply operations in a way that seamlessly integrates business entities (customers, suppliers, partners, and competitors) in a dynamic trading community.

In such an open environment, privacy is becoming a critical issue. Consequently, decentralized systems architects, developers and administrators are faced with the challenge of securing the privacy of the consumer (requester) as well as that of the supplier (provider). The terms consumer and supplier are used interchangeably with requester and provider, respectively. In general, requesters and providers are concerned about their privacy from different perspectives. For example, they may wish to protect their identities from being used, or decide by whom it will be revealed, and for what purposes, or retain the choice about whether or not to reveal their personal interests or capabilities. This work presents a cooperation analysis and model in the eMarketplace environment where privacy is inherited as part of the proposed model. The architecture supports ad hoc configurations among decentralized, autonomous and heterogeneous entities with various degrees of privacy requirements in terms of three attributes: entity's identity, capability and preferences.

In this work, we view an eMarketplace as a cooperative distributed system that integrates participating business entities, including consumers, suppliers, and other intermediaries by taking the privacy concerns within these entities. The proposed work enables different entities to cooperate in the eMarketplace and to consider their privacy requirements. In this architecture, the eMarketplace exists as a collection of economically motivated software agents.

2. Related Work

In traditional optimization, a central decision maker would be equipped with all the relevant data of the problem, and would be asked to derive a solution that fulfills all the necessary side constraints, optimizing some kind of global performance criterion. However, if the decisions are taken by several independent units, it might be the case that these units would be aiming at optimizing their own objectives rather than the performance of the system as a whole. Such situations call for models and techniques that take the strategic behavior of individual units into account, and simultaneously keep an eye on the global performance of the system. Strategic situations are traditionally analyzed in Economic Theory. In classical economic theory, there are several market models for specific trading situations and structural behaviors. 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. 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) (Parkin and Bade, 2006). On the other hand, 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 customers and suppliers execute trades in an open and competitive bidding process. In open auctions, bidders can know the bid value of the others and will iteratively have an opportunity to offer competitive bids. However, in open distributed environments, where an auction can be distributed over space and/or time, an iterative mechanism might not be feasible (Ghenniwa et al., 2006). A standard form of one-shot auctions is the first-price sealed bid auction. It avoids iterations but introduces another computational problem of counter-speculation of the other entities' valuations and might not achieve the highest price. The Vickrey auction eliminates the computational cost of both the iterative valuations and

counter-speculations overhead. The two market structures 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.

Different architectures were created to enable the collaboration in an eMarketplace including brokering via middle agents (Decker et al., 1999). The work in (Decker et al., 1998) proposed an agent-based mediation approach, in which privacy has been treated as a base for classifying the various mediation architectures only for the initial state of the system. In another approach, agents’ capabilities and preferences are assumed to be common knowledge, which might violate the privacy requirements of the participants. Other approaches such as in (Li and Horrocks, 2003) and (Motta et. al, 2003) have proposed framework structures to facilitate coordination between web services by providing semantic based discovery and mediation services that utilize semantic description languages. The work in (Ellman and Eschenbaecher, 2005) reviewed and classified the functional requirements that are needed for collaboration in networked virtual organizations. Another recent approach distinguishes a resource brokering architecture that manages and schedules different tasks on various distributed resources on the large-scale Grid (Brook And Fellows, 2005). Other initiatives proposed the use of privacy policies along with physical access means (such as smartcards), in which the access of private information is granted through the presence of another trusted authority that mediates between information requesters and information providers in virtual organizations (Amin and Keng, 2002). An attempt to develop a business-to-business system architecture is proposed in (Boll et al., 1999). It is viewed as a DBMS solution to support many-to-many relationships between customers and suppliers. Another attempt in (Rachlevsky-Reich and Ben-Shaul, 1999) was made to develop a logical market framework and infrastructure. The main thrust in (Rachlevsky-Reich and Ben-Shaul, 1999) was to separate system- related and market-related design issues. The proposed system provided trading mechanisms to include bids and offers. A more complex architecture for the eMarketplace is proposed in (Tsvetovatyy et. al, 1997). Its special focus is on the infrastructure required for conducting commerce on the Internet. The work viewed the eMarketplace in terms of traders, advertising, and banking. Alternatively, (Bichler et al., 1998) proposed a brokering-based marketplace. The eMarketplace in (Bichler et al., 1998) is viewed as a collection of suppliers, customers, and brokers. A customer can search for a service either directly in the e-catalog of the supplier or use the ebroker to search all the e-catalogs of the suppliers that are registered with the broker. E-brokers employ a simple auction mechanism. In a different approach in (Arpinar et al., 2000) an eMarketplace system as agent-oriented workflows was proposed. The work viewed the market as a workflow management system carried out by several types of agents: task, scheduling, facilitator, and recovery agents.

A Web-based marketplace for specific industrial segments such as financial services, healthcare, and energy was proposed in (VerticalNet®). 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 on-line access.

None of the mentioned approaches considered cooperation as a model of the interaction behavior where privacy is inherited from the cooperation model.

3. Cooperation Background

To visualize the abstract perspective of cooperation, consider the following example: an environment contains two entities, A and B. Each entity has its own resources that can be used to achieve a goal based on a private objective and knowledge of the specific entity. In some cases, for entity A to achieve a specific goal, it needs to use resources from entity B. In such a distributed environment, entity B must coordinate with entity A to reach an agreement. This agreement is what signifies cooperation. From this example, we can observe the interdependency between entity A and entity B to achieve a specific goal. Interdependency is viewed as a goal relevant interrelationship among actions performed by various entities. For instance, interdependency may exist between two or more entities when each has a specific knowledge or data acquisition that can only be achieved through the use of a shared resource. Another interdependency that may exist is when an entity attempts to acquire specific knowledge or data that is beyond its capability, but it can be achieved with the help of another entity. The solution to this interdependency problem is known as coordination (Ghenniwa 1996). Coordination between entities is a class of solutions that provides structure and mechanism to the system to deal with the interdependency problem. Structure refers to the entities' pattern of communication and decision-making that are related to coordination. Mechanism is a composition of decision points coordinated control and interaction devices directed to resolve problems associated with interdependencies. Cooperation, on the other hand, is a model for the interaction behaviour and is defined as the characteristic of entities having the will to share their capabilities with other entities on trading knowledge as well as on achieving their goals, whenever it is possible and beneficial (Ghenniwa 1996).

From the cooperation definition, the "will" implies that entities are autonomous. This means that entities have authority to perform actions without any outside decisions forced upon an entity. The "share" is the behavior of the entities, as in trading knowledge and the capability to achieve goals. The nature of cooperation behavior is the ability to share knowledge and capabilities. This means when an entity is asked for specific knowledge acquisition or the use of specific capabilities, the entity is willing to cooperate if possible to do so. Knowledge is a set of prepositions that is present explicitly or implicitly in the entity's memory. Capabilities are the entity's ability to execute actions either mentally or physically, such as move-action and send-information, respectively. "Possible" implies the feasibility of achieving the goal. "Beneficial" presents that the entities are economically rational. Economics rationality is based on finding the "best" strategy that is more beneficial and feasible to achieve. The strategy is decided upon the entity's goal and constraints within an environment geared towards the cooperative behavior. The selected strategy governs the behavior of the interaction with other entities.

Two fundamental components in cooperation: the protocols that govern the interactions, and the corresponding strategy. The protocol and strategies can be designed in a way that the desired cooperative outcomes are achieved.

4. Market Structure

The 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 et al., 1992) is the specification of the set of admissible messages (i.e., traders' actions, usually price and/or quantity offers), and the final goods/services allocation given any combination of messages chosen by the participants and any initial allocation. Hence, the market structure includes service requesters and service providers. Figure 1 depicts the interaction between the service requester and providers within a distributed environment. A requester is an entity with goals that are either beyond its capability or believes that its goal might be better achieved by other entities to maximize its benefits. A provider is an entity that is able to provide some services in the domain. Formally, given a set of r requesters denoted by $Q = \{Q_1, Q_2, \dots, Q_r\}$ where $Q_q \in Q$ ($q=1 \dots r$) has a set of n tasks, denoted by $T = \{T_1^1, T_1^2, \dots, T_q^n\}$, where task $T_q^j \in T$ ($j=1 \dots n$) is formulated as a set of operations $T_q^j = \{o_j^1, o_j^2, \dots, o_j^{n_j}\}$, where n_j is the number of operations belonging to T_q^j .

Service providers is formulated as a set of p service providers denoted by $P = \{P_1, P_2, \dots, P_p\}$ where $P_t \in P$ ($t=1 \dots p$) own a set of m resources denoted by $R = \{R_1^1, R_1^2, \dots, R_t^m\}$ where $R_i^i \in R$ ($i=1 \dots m$). A resource is a physical/logical device that is capable of executing tasks. The resource can be defined by a set of capabilities $R_i^i = \{c_i^1, c_i^2, \dots, c_i^{m_i}\}$, where m_i is the number of capabilities that describe R_i^i .

Insert Figure 1 about here.

It is clear that the eMarketplace provides a structure to allocate requesters' tasks to capable resources owned by service providers. This is traditionally known as the scheduling problem. The eMarketplace scheduling problem however, is a class of scheduling problem where entities are decentralized. Compared with classical scheduling problems, the decentralized scheduling problem differentiates itself in the distribution of both scheduling knowledge and control. This means that decisions are taken by several independent entities. It might be the case that these individual entities aim at optimizing their own objectives rather than the performance of the system as a whole. Such situations call for models and techniques that take the strategic behavior of individual entities into account, and simultaneously keep an eye on the global performance of the system. Strategic situations are traditionally analyzed in economic theory. In this work, we will model cooperation between entities in the decentralized environment considering the requesters' and providers' privacy requirements.

5. Cooperation: Proposed Model

It is important that the architecture of an eMarketplace supports and leverages the participants' legacy environments with minimum overhead. The support can take over technology-independent cooperative distributed system architecture. 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 architecture of the eMarketplace provides the foundation to integrate and leverage the participants' resources. The brokering service provides a capability-based integration and scheduling service in the eMarketplace. The brokering allows entities (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 eMarketplace environment involves complex and nondeterministic interactions, often producing results that are ambiguous and incomplete which require that the components of these eMarketplaces be able to change their configuration to participate in different, often simultaneous roles. These requirements could not be accomplished using traditional ways of manually configuring software. We strongly believe that agent technology is a very promising design for brokering in cooperative distributed systems. Here we view agent-orientation as a metaphorical conceptualization tool at a high level of abstraction (knowledge level) that captures, supports and implements features that are useful for distributed computation in open environments. These features include cooperation, coordination, interaction, as well as intelligence, adaptability, and economic rationality. An agent is an individual collection of primitive components that provide a focused and cohesive set of capabilities (Ghenniwa and Huhns, 2004).

Within this context, a distributed group of intelligent agents are goal oriented, autonomous, and rational. They act in a cooperative and interactive manner to provide coordination activities and enable collaboration between various entities. Architecturally, the brokering is viewed as a layer of services where a brokering service is modeled as an agent with a specific architecture and interaction protocol that are appropriate to serve various requests. The architecture allows various entities in the domain to join and benefit from the functionalities provided by the brokering service. A registration and naming service permits the addition or removal of any agents, service and sources at runtime. The brokering service uses the registration and naming service to build up a knowledge base of the environment that will facilitate locating and identifying the relevant existing providers that are capable of serving a specific request.

5.1 Privacy Requirements

Beside the essential brokering services within cooperative distributed systems to facilitate integration, entities within the eMarketplace may wish to have an interest in sustaining a “*personal space*”, free from interference by others. In other words, this interest can be viewed as the ability of those entities (agents) to control the privacy aspects of any self-related attributes or personnel features. A system that respects their privacy will allow them to select which information will be revealed and to whom.

From the privacy standpoint, the brokering layer’s services can be categorized into different roles and functionalities. These roles are identified according to the participants’ (providers and requesters) desired degree of privacy, and will vary from one specific scenario to another. This section presents a privacy-based model, and illustrates the different interactions patterns between the brokering layer’s agents and other agents in the environment. The model will allow various participants (requesters and providers) to have the ability to protect their identity, capabilities, or any goals to be achieved while benefiting from the services provided by the brokering layer. Formally, an agent can be represented as a 2-tuple $Ag = \langle (RA : Id, G); (PA : Id, Cap) \rangle$

Where RA and PA refer to the agent role as requester and provider, respectively, Id, G, and Cap refer to the agent’s identity, goals and capabilities, respectively, which might have a null value. For example, an agent might participate with a privacy degree that enables the hiding of its identity as a requester by setting the value of (Id) to null. The challenge in this context is how to architect the brokering layer with the appropriate set of services that enable cooperation across

the different degrees of privacy and reach a desirable solution related to the requesters and providers goals. The interaction protocols represent both message communication and the corresponding constraints on the content of messages.

5.2 Privacy-Based Cooperation Model

The agent's rationality can be modeled as a mathematical optimization problem. The optimization problem is made up of three basic elements:

- A set of independent variables $\mathbf{X}$ – measures the domain issues. This affects the value of the objective function (F).
- A set of constraints $\mathbf{C}$ – are features (characteristics) of the admissible values of $\mathbf{X}$. In other words, constraints allow the variables to take on certain values and exclude others.
- An objective function $\mathbf{F}$ – is the structure of the independent variables that yield the optimal solution.

The objective function in the optimization model formulates the agent's goal. The agents' goals can be based on its role as a requester or as a provider. The agent's components shown in Figure 2 depict the classical way of solving the optimization problem by taking in the three elements of the optimization model, and through the problem solver, will provide a solution.

In the cooperative distributed systems environment, however, agents must cooperate to resolve the interdependency problem amongst themselves. A high-level interaction between agents is illustrated in Figure 2. Each agent has its knowledge and control to find solutions, based on its own objective. The agent's knowledge is based on its local knowledge as well as on the surrounding environment. Hence, the agent's constraints are related to the local agent constraints, as well as constraints related to the cooperating agents. The overall solution S^* can be modeled as $f(S_1^*, S_2^*, \dots, S_n^*)$. The function f is the cooperation computation model that allows agents to share knowledge as well as capabilities to achieve goals. The cooperation computation model in this work includes the privacy requirements related to the agents. Hence, the decentralized scheduling problem in the eMarketplace involves the assignment of the set of service providers' resources $R = \{R_1^1, R_1^2, \dots, R_1^m\}$ to the set of requesters' tasks $T = \{T_1^1, T_1^2, \dots, T_q^n\}$ over time, such that the schedule S^* does not violate the agent's local constraints as well as other constraints related to the cooperating agents. S^* produces the best measure of the cooperation function $f(S_1^*, S_2^*, \dots, S_n^*)$.

Insert Figure 2 about here.

In many cases, scheduling methods are developed under the assumption of centrally available information, or distributed information with cooperative behavior. When a centralized approach is possible, it will, in general, yield results far superior to any purely decentralized method (Reeves et al., 2005). This is the case in our privacy model when the goals of the agents are revealed to the Broker.

Nevertheless, centralized methods are not directly applicable when agents have distinct interests and privately held information about the requirements and goals. We can not always assume agents will reliably communicate such information to the centre, as the centre's use of the information to determine an allocation will typically create incentives for the entities to misrepresent their situations in order to obtain more advantageous results. Hence, we adopt economic based approaches to solve/compute a solution under the requirement that agents must hide their goals.

Cooperation between agents defines a structured process that determines which agent gets which resources, based on interaction. The field of mechanism design considers how to organize such mechanisms, taking into account the information and incentives facing the participating agents. Typically, carefully designed mechanisms induce agents to reveal essential information via monetary transfers tied to the messages and resources allocated. Markets comprise a class of resource allocation mechanisms in which agents exchange resources for money, at prices determined through interaction or bids. When the price-determination process is mediated, and follows explicit rules mapping bids into allocations, the mechanism constitutes an auction (Ghenniwa et al, 2006).

5.3 Auction: Computation Model

Auctions are dynamic and often efficient mechanisms for selling items in complex eMarketplaces. The proposed auction market in this work incorporates a Vickrey auction mechanism. In private-value Vickrey auctions, an agent's valuation is determined locally and independent from other agents' valuations. It provides incentives to promote truthful bidding among self-interested agents and avoids the computational cost of counter-speculations. Hence, there is no need for iterative negotiation strategies, dynamic strategic behavior to reveal other bidders beliefs, or a high degree of security. Furthermore, it has been shown that this mechanism can achieve the same utility for the participants as other less direct mechanisms where bids might not be the true valuation (Varian 1995).

The broker provides a common registry service for the auction market structure, through which business-entities, suppliers and consumers register with the eMarketplace. The entries in the agent's registry database would include the identity of business-entity agent, its role as a requester or provider, its list of preferences, such as product, price range, and specific supplier brand. Potential providers advertise their services with the broker agent.

The auction can be a seller centric "forward" auction or buyer (requester) centric "reverse" auction. A seller-centric auction helps a provider agent to maximize its revenue at equilibrium, whereas a buyer-centric auction helps a requester agent to identify the lowest price at equilibrium. In bilateral markets, a combination of double auction (both forward and reverse auctions) and supply-demand profile integration help a market-based supply chain determine efficient allocation. Any potential provider or requester (buyer) that wishes to join an auction session is required to register once again with the broker registry service a priori to be able to participate in the bidding process. Once an auction event is decided upon, the broker agent is able to extract the auction parameters such as start bid, ask price, and increment bid.

In this work, we focus on the forward auction protocol. In a single item forward auction, we have a set of agents $A = \{A_1, A_2, \dots, A_g\}$ $A_y \in A$ ($y=1 \dots g$). Each agent A_y , has a "maximum willingness to pay" or "value" for the item that is assumed to be private information known only by the agent A_y , denoted by v_y . Then, the objective is to award the item to A_y^* with the highest bid value.

Although the Vickrey auction ensures that the item is awarded to the highest bidder agent, it does not maximize the provider's revenue unless "reserve prices" are imposed. Given a set of combinatorial bids, the provider agent (through broker agent) then decides how best to allocate individual services to those bundles for which bids were placed, with the objective being to maximize revenue. Groves' mechanism in combinatorial auctions is a combinatorial allocation problem (CAP). It involves allocating bundles of items that may overlap to n bidders. CAP is an optimization problem equivalent to the weighted set packing problem (Parkes 1999). The proposed auction market is comprised mainly of three agents: the broker agent, requester agent and provider agent. Under the general Vickrey mechanism, it is in the interest (the dominant strategy) of the bidder to report its true valuation function. Then, the broker agent:

- calculates the allocation (x_i^*) that maximizes the sum of the bids, subject to the items constraint;
- calculates the allocation (x_{-i}^*) that maximizes the sum of the bids, other than that of bidder agent y such that it excludes all items allocated to agent y ;
- announces the winners and their payment given by $p_y = \sum_{z \neq y} v_z(x_{-y}^*) - \sum_{z \neq y} v_z(x_y^*)$

Under the assumption of quasi-linear preferences, each requester agent calculates its utility. For requester agent y the utility will be $u_y(x^*) = v_y(x^*) - p_y$.

In our case, once all agents have placed their bids in the OR* bidding language for any arbitrary set of bids, and if these bids conform linear-order bids, nested-hierarchical bids or a combination of both structures, then the linear program guarantees an exact optimal solution. This linear formulation can utilize standard algorithms and hence can be run directly on standard commercially available optimization software, such as CPLEX (ILOG). In other cases, the linear programming relaxation may not yield non-integral-solution. Thereby, heuristics can be used to somehow find an approximation to the optimal solution. A greedy algorithm that provably runs in polynomial time, however, does not guarantee to always produce an optimal solution. Alternatively, a recursive branch-and-bound algorithm based on an upper bound LP solution can be used. This technique trims the search space in exponential searches and can provably produce an optimal solution, however, it does not guarantee to always run in polynomial time.

5.4 Entities Interaction Patterns

This section presents the interaction patterns between entities to cooperate within the decentralized environment considering the different privacy concerns from the service requester and provider. Table 1 summarizes the different brokering privacy scenarios categorized by the possible combination of privacy attributes of the Requester and Provider agents. Brokers within the layer might represent a set of services in which providers can advertise their service capability. More details related to the interaction and the privacy attributes are presented in the subsections.

Insert Table 1 about here.

5.4.1 Requesters Revealing identities and goals

Agents playing the role of requestors for this privacy case are required to reveal their identities and goals to the relevant broker within the layer. The interaction pattern, as shown in Figure 3, is as follows:

- The Requester formulates service's requests, which includes the requester identity, and goals with respect to the tasks that needs to be scheduled
- The Broker contacts a set of potential providers
- The service provider submits valuation results to the Broker
- The Broker finds a schedule solution related to the requester goal
- The Broker assigns the appropriate schedules that achieve the Requester's goal
- The broker returns the result of the services to the Requester agent

Insert Figure 3 about here.

5.4.2 Requesters hiding identities

Requesters may wish to access services or seek further assistance without revealing their identities. Note that the interaction imposes a significant effort on the performance and efficiency. The interaction pattern, as shown in Figure 4, is as follows:

- The Requester formulates service's requests, which includes the requester goals with respect to the tasks that need to be scheduled. The requester identity is not revealed in this case.
- The request is inserted into a requests repository in the Broker
- The Broker get the request from the requests repository
- The Broker contacts a set of potential providers
- The service provider returns the result to the Broker
- The Broker finds a schedule solution related to the requester goal
- The Broker stores the results in the Request Repository
- Requesters are responsible of checking the availability of the service's result.

Insert Figure 4 about here.

5.4.3 Requesters hiding goals

Requesters may wish to access services or seek further assistance without revealing their goals. The Broker permits those requestors to submit valuations/bids based on their goals and allocates schedules based on the auction mechanism mentioned in the previous section. The interaction pattern, as shown in Figure 5, is as follows:

- The Requester registers with the broker with the parameters of hiding the goal
- The Broker advertise the services that exists in the environment
- The Requester submits bids to the Broker based on its goals
- The Broker starts the auction and allocates schedules based on the auction mechanism
- The Broker contacts a set of potential providers
- The service provider returns the result to the Broker
- The Broker delivers the result to the requester

Insert Figure 5 about here.

5.4.4 Requesters hiding identities and goals

Requesters would have the possibility of hiding their identities and goals from the entire environment. The interaction pattern as shown in Figure 6 is as follows:

- The Requester registers with the broker with the parameters of hiding identity and goal
- The Requester checks service offering repository from the Broker
- The Requester selects specific services and submits bids to the Broker based on its goals
- The Broker starts the auction and allocates schedules based on the auction mechanism
- The Broker contacts a set of potential providers
- The service provider returns the result to the Broker
- The Broker stores the results in the “result location” repository
- The Requester retrieves the scheduling result from the “result location” located in the Broker. Note: Requesters are responsible for checking for the availability of the service’s result

Insert Figure 6 about here.

5.4.5 Providers Revealing identities and capabilities

Providers have the possibility to reveal their identities and capabilities. The interaction pattern, as shown in Figure 7 is as follows:

- Service providers register with the Broker with its identity and capabilities
- The brokering layer issues a call-for-proposals (CFP) to available providers, based on their capabilities, informing them of the problem's specifications.
- Each potential contractor (provider) determines the valuation parameters (such as goal quality, goal expiration time, and cost) and accordingly submits a bid to the brokering agent, or might reject the proposal.
- The Broker receives the bids, and selects the most appropriate bid that satisfies the request's parameters.

Insert Figure 7 about here.

5.4.6 Providers hiding capabilities

Providers have the possibility to hide their capabilities. The interaction pattern, as shown in Figure 8, is as follows:

- Service providers register with the Broker with its identity and hiding capabilities
- When the Broker is confronted with a request, the Broker will form requests to every registered provider with unknown capability.

It is noteworthy that, for every advertised request, providers need to determine whether the request is within its capabilities and/or of an interest, which implies that a considerable elapsed time will be spent on evaluating every single request. Therefore, providers would be deluged by a variety of service requests, which significantly impact performance and efficiency.

Insert Figure 8 about here.

5.4.7 Providers hiding identities and capabilities

Providers have the possibility to hide their identity and capabilities. The brokering layer's functionality is mainly seen as a directory service, in which a brokering agent maintains a repository of service's requests along with any required preferences. Providers will have the ability to browse this repository and then determine the relevant requests that might be of an interest and within their capabilities. The interaction pattern, as shown in Figure 9, is as follows:

- Service providers register with the Broker while hiding its identity and capabilities.
- When the Broker is confronted with a request, the Broker stores requests in the "Requests Repository".
- The Service Provider checks the "Requests Repository" for available requests that can provide services.
- The Service Provider will evaluate the request and sends its bids to the broker in the "Result Location" repository.
- The Broker retrieves the service provider's bids from the "Result Location" repository, and through its reasoning, will provide a solution.
- The results are communicated to the service requester.

Insert Figure 9 about here.

6. Implementation

To validate and experiment with our theoretical work and foundations, described in the previous sections, we have developed a prototype of an agent-oriented *eMarketplace*, as shown in Figure 10. ABC Corp and XYZ Inc. are virtual business entities registered with the *eMarketplace* for both requester and provider services. Both organizations use cooperation services in the *eMarketplace* environment. As illustrated in Figure 10, individual business-entities in the *eMarketplace* can participate in the market. Similarly, each business-entity service is represented by an agent in the *eMarketplace*. These agents provide intelligent, highly autonomous interfaces for the business-entity services that might be based on legacy applications. For example, in Figure 10, the ABC purchasing-service agent represents the implementation of the business-specific purchases by ABC in the *eMarketplace*. Each business entity service agent is registered in the *eMarketplace* through the Brokering agent with their privacy requirement. In the current prototype, we experiment with auction market structures, as shown in the privacy interaction diagrams mentioned in the previous section, where service requesters and providers are brought together to trade with each other and prices are set by the market structure. The trading and interaction behavior of the participant agents is governed by the market structure and the privacy requirement by the business-entity, as described previously.

The implementation utilizes the Java Agent Development (JADE) platform as shown in Figure 10. JADE is a software framework to develop agent applications in compliance with the Foundation of Intelligent Physical Agents (FIPA) specifications for multi-agent systems. JADE deals with all aspects that are external to agents and independent of their applications. These include message transport, encoding, parsing and agent lifecycles. JADE supports a distributed

environment of agent containers, which provide a run-time optimized environment to allow several agents to execute concurrently. This feature has been utilized to create several concurrent market 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 the directory facilitator (DF), agent management service (AMS), and agent communication center (ACC), the architecture of the application agents is based on the CIR-Agent model (Ghenniwa 1996). Java features, such as portability, dynamic loading, multithreading, and synchronization support make it appropriate to implement the inherent complexity and concurrency in an *eMarketplace*. These features were also instrumental in 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 timeout 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 and brokering services; and behaviors that are concerned with providing business-specific services, such as selling and purchasing.

Insert Figure 10 about here.

The brokering agent created a scheduling decision based on the requester and provider (XYZ Inc, ABC Corp.) agents privacy requirement, as proposed in section 5. The broker-agent interaction and problem solver components are implemented as follows:

- Interaction – it describes the interaction protocol used by the broker-agent to cooperate with the requester and provider-agents in the environment. The interaction component of the broker-agent is implemented by making use of the existing JADE behavior classes: *FipaContractNetInitiatorBehavior*. In that protocol, the broker-agent can solicit proposals from the provider by sending a *CFP* (*call for proposal*) message that checks whether or not the provider accepts the assignment, which might contain conditions or

constraints required upon executing that goal, such as the requester's deadline. The provider agents can reply by sending a *PROPOSE* message, including the acceptance of the request and the time when the provider agent can finish processing the request, or alternatively the provider may refuse a certain proposal by sending a *REFUSE* message. The *PROPOSE* message, sent by the resource, is taken into the broker-agent problem solver and an *OFFER* message is created to the winner resource to schedule the request. The interaction with the winner resource and the provider agents is done based on the defined requester or provider privacy concerns, as mentioned in section 5.

- Problem Solver – it is the decision making of the broker agent to schedule the request into the providers, based on its self-knowledge, and the knowledge of the provider and requester agents. As mentioned in the proposed architecture in section 5, the broker agent provides a schedule, based on the proposed interaction related to the requester and providers privacy requirements. The broker-agent problem solver derives the scheduling based on the modeled auction problem using OPL 5.0 and CPLEX 10.1 from (ILOG) which finds the exact optimal value of the objective under the linear inequality constraint.

The role of the requester-agent is to express its preferences in OR* bid format and sends “bid” messages out to the broker-agent. The interaction of the requester-agent interaction and problem solver is as follows:

- Interaction – it describes the interaction protocol used by requester-agents to cooperate with the broker-agent in the environment. It contains a class that extends the JADE behavior class *ContractNetReponder* Behavior through which the requester-agent prepares the *PROPOSE* message that is later followed by the formulated OR* Bids or the *REFUSE* message to refuse the proposal or the *NOT-UNDERSTAND* message to request resending the information of the announcement again. The interaction with the broker is based on the required privacy concerns by the requester-agent as mentioned in section 5.
- The problem-solver component contains a Bid class that implements a cyclic behaviour in order to respond to incoming messages from the broker-agent that requests bids. This class implements all the requester-agent's tasks such as registration with the DF service as well as a method that formulates preferences and bid valuations in OR* language. The bids valuations are implemented using OPL 5.0 and CPLEX 10.1 from (ILOG) which finds the exact optimal value of the objective under the linear inequality constraint.

The provider-agent's role is to formulate the sell-order and assigns it to the broker-agent. The provider-agent's reasoning component consists of the following:

- Interaction – it describes the interaction protocol used by provider-agents to cooperate with the broker-agent in the environment. The interaction device in the provider-agent is based on incoming and outgoing processes. The resource's interaction makes use of the existing JADE class *FipaContractNetResponderBehavior* when interacting with the broker-agent. The pattern of the interaction is based on the privacy concern set by the provider-agent as mention in section 5.
- Problem Solver – it is the decision making of the provider-agent, based on its objective. The provider-agent problem solver performs local scheduling and is implemented using OPL 5.0 and CPLEX 10.1 from (ILOG) which finds the exact optimal value of the objective under the linear inequality constraint.

7. Discussion and Conclusion

In this paper we have developed an agent-based cooperation architecture that provides a computational framework and considers the privacy requirements for different entities within a decentralized environment. Unlike the traditional market structures, it is further classified into several sub-roles categorized by the desired privacy attributes of service requesters and providers.

The proposed approach is innovative, in the sense that privacy is an inherited component from the cooperation approach. Different application domains can benefit from the proposed model, such as intelligent cooperative information systems, and agent-based electronic business. The opportunities for exploiting the proposed architecture in building a collaborative eMarketplace are enormous. Based on the level and the amount of information that can be released, eMarketplace organizations can privately collaborate while translating their privacy concerns to an applicable related privacy that suits their desires and objectives.

We validated the proposed approach with a prototype implementation. Certainly, there are complexities that are inherited from the nature of the eMarketplace environment being decentralized. If we look at a simpler environment where we have centralized decision making for the scheduling solution, this class of problem has been proved to be complex and NP-hard. Adding the challenge of the distribution of control and knowledge, such as in the eMarketplace environment, increases the complexity of the problem. This challenge derives from the self-interested nature of entities in a decentralized environment.

The theory of computational complexity provides the relative computational merits for measuring the complexity of scheduling problems. However, the measuring is not based on the characteristics of the problems. It is done by means of time complexity functions of different algorithms which have been found for the problems. The time complexity function $f(v)$ of an algorithm gives the maximum number of operations that would be required to solve a problem instance of size v . If an algorithm has polynomial time complexity, its time complexity function $f(v)$ is $O(p(v))$ for some polynomial $p(v)$. Otherwise, an algorithm has exponential time complexity. All problems for which algorithms with polynomial time behavior have been found belong to class P. The set of problems for which algorithms with exponential behavior have been found belong to class NP.

The time complexity function formulated in the complexity theory is suitable for the analysis of classical scheduling problems (centralized scheduling problems). However, it may not be equally suitable for decentralized scheduling problems because it is a function with a single input (size of a problem instance) and does not cover other aspects of the problems, such as the distribution of knowledge and control, which can also be sources of computational complexities. To measure the complexities of decentralized scheduling problems, we expand the classical time complexity function by adding the distribution of knowledge and control as inputs. The extended time complexity function is denoted as $g(v, k, c)$, where v is the size of the problem; $k \in \{distributed, common\}$ indicates if the problem has the property of the distribution of knowledge; $c \in \{decentralized, centralized\}$ indicates if the problem has the property of the distribution of control. If the time complexity function of a scheduling problem is $g(v, common, centralized)$, this problem is a centralized one. The complexity of the problem is purely derived from the scheduling domain

components (tasks, processors, constraints, and objectives), which we refer to as scheduling domain complexity. Accordingly, the complexities derived from the distribution of knowledge and control is referred to as knowledge distribution complexity and control distribution complexity. These complexities actually form a hierarchy with control distribution complexity at the top and the scheduling domain complexity at the bottom, for all scheduling problems have the scheduling domain complexity knowledge and only decentralized scheduling problems have control distribution complexity.

In general, if the knowledge of a centralized scheduling problem is distributed across agents, the problem complexity increases. This is because in addition to the scheduling domain complexity, the algorithm has to deal with the complexity of interaction among agents, such as managing message passing, reaching closure, and determining the final schedule. Furthermore, if control is also decentralized, extra computation needs to be spent on aligning different (in many cases conflict) objectives of self-interested agents to a system wide goal.

References

- Amin, T. and Keng P. H., 2002. Inter-organizational Workflow Management System for Virtual Healthcare Enterprise. *Proceedings of 3rd IFIP Working Conference on Infrastructures for Virtual Enterprises - PRO-VE'02*, Portugal, pp. 182-190.
- Arpinar, S. Dogac, A., and Tatbul, N., 2000. An Open Electronic Marketplace through Agent-based Workflows: MOPPET. *International Journal on Digital Library*, 3 (1), 36-59.
- Bichler, M., Beam, C., and Segev A., 1998. OFFER: A Broker-centered Object Framework for Electronic Requisitioning. *IFIP Conference Trends in Electronic Commerce*. LNCS 1402, pp. 154-165.
- Boll, S., Gruner, A., Haaf, A., and Klas, W., 1999. EMP-a database driven electronic marketplace for business-to-business commerce on the Internet. *Journal of Distributed and Parallel Databases*, 7 (2), 149-177.
- Brook, J. and Fellows, D., 2005. An Architecture for Distributed Brokering on the Grid. *Proceedings of 11th International Euro-Par Parallel Processing*, Portugal.
- Clarke R., 1996. Identification, Anonymity and Pseudonymity in Consumer Transactions: A Vital System Design and Public Policy Issue [Online]. Available from: <http://www.anu.edu.au/people/Roger.Clarke/DV/AnonPsPol.html> [Accessed 01 October 2008]
- Decker, K., Sycara, K. , and Williamson, M., 1999. Middle-agents for the Internet. *Proceedings of IJCAI97 International Joint Conference on Artificial Intelligence*, Nagoya, Japan.
- Decker, K., Lesser, V., Prasad, M.V., and Wagner, T., 1995. MACRON: An Architecture for Multi-agent Cooperative Information Gathering. *Proceedings of the CIKM '95 Workshop on Intelligent Information Agents*.
- Ellman, S. and Eschenbaecher, J., 2005. Collaborative Network Models: Overview and Functional Requirements. In: G.D. Putnik and M.M. Cunha, eds. *Virtual Enterprise Integration: Technological and Organizational Perspectives*, Idea Group Publishing, PA, pp. 102-123.
- Ghenniwa, H. and Huhns, M., 2004. Intelligent Enterprise Integration: eMarketplace Model. *Creating Knowledge Based Organizations*, Idea Group Publishing, Pennsylvania, USA, pp. 46-79.
- Ghenniwa, H., Dang, J., Huhns, M. and Shen, W., 2006. eMarketplace Model: An Architecture for Collaborative Supply Chain Management and Integration. In: B. Chaib-draa and J. Müller, eds. *Multiagent-based Supply Chain Management*, Springer Berlin Heidelberg, pp. 29-62.
- Ghenniwa, H., 1996. *Coordination in Cooperative Distributed Systems*. Ph.D. Thesis, University of Waterloo.
- ILOG CPLEX Suite, (n.d.). Available from: <http://www.cplex.com> [Accessed 01 October 2008]
- Java Agent Development Framework: JADE. Available from: <http://jade.cselt.it/> [Accessed 01 October 2008]
- Li, L. and Horrocks, I., 2003. A Software Framework for Matchmaking Based on Semantic Web Technology. *Proceedings of the 12th International Conference on WWW*, Hungary.
- McCabe, K., Rassenti, S., and Smith, V., 1992. Institutional Design for Electronic Trading. *Conference on Global Equity Markets*, N.Y. University, Salomon Center.
- Motta, E., Domingue, J., Cabral, L., and Gaspari, M., 2003. IRS-II: A Framework and Infrastructure for Semantic Web Services. *Proceedings of the International Semantic Web Conference*, LNCS 2870, pp. 306-318.
- Parkin, M. and Bade, R., *Economics: Canada in the Global Environment*. 6th Edition, Pearson Addison-Wesley, Toronto.
- Parkes D., 1999. iBundle: An Efficient Ascending Price Bundle Auction. *Proceedings of the First International Conference on Electronic Commerce*, pp. 148-157.
- Rachlevsky-Reich, B. and Ben-Shaul, I., 1999. GEM: A Global Electronic Market System. *Information Systems*, 24 (6), 495-518.

- Reeves, D. M., Wellman, M. P., Mackie-Mason, J.K., and Osepayshvili, A., 2005. Exploring bidding strategies for market-based scheduling. *Decision Support Systems*, 39, 67–85.
- Tsvetovatyy, M., Gini, M., Mobasher, B., Wieckowski, Z., 1997. MAGMA: An Agent-Based Virtual Market for Electronic Commerce. *Journal of Applied Artificial Intelligence*, 11 (6), 501-523.
- Varian, H.R., 1995. Economic Mechanism Design for Computerized Agents. *Proceedings of the First USENIX Workshop on Electronic Commerce*, New York.
- VerticalNet® Marketplaces. (n.d). Available from: <http://www.VerticalNet.com> [Accessed 01 October 2008]

Tables

Table 1 Brokering Layer Interaction Patterns with Service Requestors

Privacy Attributes					Interaction Patterns	
Case	g(RA)	Id(RA)	Id(PA)	Cap(PA)	With Requesters	With Providers
1	Revealed	Revealed	Revealed	Revealed	Receive request Deliver result	Search for relevant agents, Negotiate and Obtain result
2	Revealed	Revealed	Revealed	Hidden	Receive request Deliver result	Advertise request to known PA
3	Revealed	Revealed	Hidden	Revealed	Receive request Deliver result	PA check for requests PA to reply with results
4	Revealed	Hidden	Provider's privacy attributes are either one of the status shown in (1,2, or 3)		Receive request RA retrieve result	same interaction protocol depicted in any selected case shown in (1,2, or 3)
5	Hidden	Known			Advertise services to RA	
6	Hidden	Hidden			RA to check for services.	

List of Figures Captions

Figure 1 Entity-to-Entity Interaction

Figure 2 Decentralized Optimization Problem

Figure 3 Interaction Pattern for Requesters Revealing Privacy Attributes

Figure 4 Interaction Pattern for Requester Hiding Identity

Figure 5 Interaction Pattern for Requester Hiding Goals

Figure 6 Interaction Pattern for Requester Hiding Privacy Attributes

Figure 7 Interaction Pattern for Provider Revealing Privacy Attributes

Figure 8 Interaction Pattern for Provider Hiding Capability

Figure 9 Interaction Pattern for Provider Hiding Privacy Attributes

Figure 10 Implementation Environment

Figure 11 Entity-to-Entity Interaction

Figure 12 Decentralized Optimization Problem

Figure 13 Interaction Pattern for Requesters Revealing Privacy Attributes

Figure 14 Interaction Pattern for Requester Hiding Identity

Figure 15 Interaction Pattern for Requester Hiding Goals

Figure 16 Interaction Pattern for Requester Hiding Privacy Attributes

Figure 17 Interaction Pattern for Provider Revealing Privacy Attributes

Figure 18 Interaction Pattern for Provider Hiding Capability

Figure 19 Interaction Pattern for Provider Hiding Privacy Attributes

Figure 20 Implementation Environment