한국지능정보시스템학회
02 춘계학술대회 논문집, pp. 222~229

멀티에이전트 기반 가치넷 설계

김 태 운

Multi-agent based value net design

Taewoon Kim

A value net is a business design that uses digital supply chain concepts to achieve both superior customer satisfaction and company profitability. In order to implement the value net model, information processing and distribution needs to occur in real time. Software agent technology is becoming popular due to the inherent characteristics of autonomy, distributedness and modularity. In this paper, we adopt agent technology to handle all real time decision process, making the value net model a complex multi-agent network of decision makers. For the agents to properly coordinate their respective activities we develop MAVN model, a Web-based multi-agent language grounded in the XML and Java.

Key words: Value net, supply chain, multi-agent, BDI, XML, SOAP

1. Introduction

The business environment is changing rapidly. Change requires businesses to adapt and to reinvent themselves. During the current digital age, companies are expected to be agile, flexible, and customer oriented. They're required to provide cost quotes, delivery times, delivery tracking, and production updates to the customer in real time. In response, organizations are reengineering and streamlining various processes and business practices to improve customer relations and productivity. One process that businesses in all sectors of industry are finding to improve customer relations, productivity, and efficiency is the supply chain.

The supply chain is a process that spans the entire enterprise integrating processes based on flawless delivery of basic and customized services. Therefore the objective of Supply Chain Management (SCM) is to optimize information and product flow from the receipt of the order to the purchase of raw materials, to delivery and consumption of the finished product [Fox et al, 1993]. For the supply chain to operate efficiently during the digital era, all the functions and processes must be integrated within a network providing a rapid and quality response to any demand or informational request from the customer. Therefore the supply chain becomes a digitalized and dynamic, high performance network of customer/supplier partnerships and information flows. This new model of the supply chain is known as a value net [Bovet and Martha, 2000].

A value net is a business design that uses digital supply chain concepts to achieve both superior customer satisfaction and company profitability. The value net model creates an informational infrastructure that facilitates the distribution of information from a customer to the various manufacturing sites and suppliers. Therefore the problem with any model is to fulfill the customer order with high quality products meeting delivery requirements while minimizing the cost of the parts, manufacturing, and transportation. To make these decision-making processes efficient and hence make the value net reliable, it is imperative that we integrate and automate all the decision-making processes to solve the various different value net problems. Within the old supply chain model the automation of some decision processes led to *Enterprise Resource Planning* (ERP). However the solution to the value net problem lies within a network of autonomous and intelligent software agents that react to the environment and to the different nodes of the net. Each agent would represent a part of the decision making process within the value net. Therefore creating an agile network of decision makers that react to the customer in real time as opposed to a rigid process flow that is decided before the customer places an order.

Intelligent software must be utilized to create real time decisions distributed across a worldwide network to make the value net instantaneous. Thus intelligent agents by virtue of their adaptability, autonomy, and social ability make them a viable technology in the implementation of a value net. Each intelligent agent within the network would represent a part of the decision making process and would contain information specific to their process.

We consider the value net of a manufacturing enterprise as a world-wide network of suppliers, factories, warehouses, distribution centers, and retailers through which information is processed and raw materials are acquired, transformed into products, and then distributed to customers. Therefore in this paper we will model the value net with heterogeneous systems and intelligent agents distributed across the Internet. The focus of this work is more on modeling the communication in the value net.

경성대학교 산업공학과

2. Multi-agent value net architecture

A value net is not just about supply only, but it's about creating value for customers, the company, and its suppliers. It is a digitalized, dynamic and efficient collaboration of partnerships and information flows. The driving changes in today's world of business are more demanding customers, the Internet and digital technology, growing competitive pressures and globalization. The characteristics of value net are defined as customer-aligned, collaborative and systemic, agile and scalable, fast flow and digital.

Multi-agent systems are the best way to characterize or design distributed computing systems [Weiss, 1999]. Information processing is ubiquitous. The Multi-Agent Value Net (MAVN) proposed in this paper consists of multiple agents that communicate across the Internet and Intranets of each business enterprise involved in the network. Each business within the network is considered as an agent society, a group of agents working together to achieve a goal.

Utilizing the hierarchical architecture and the message performatives, the communication complexity of the multi-agent value net will grow controllably as the system increases in size. Typically each departmental agent has a set of non-redundant tasks, however as the degree of coordination decreases, departmental agents will have to share tasks as to reduce resource contention and increase agent response time. Since the coordinators are responsible for the group of departmental agents, the amount of work a coordinator oversees and distributes will grow with an increase of departmental agents. Hence the coordination problem shifts from the departmental level to the coordinator level. After the expansion of the departmental agents, the coordinator agents will begin to increase workload and decrease resource availability. To alleviate their workload the coordinator agents will need to allocate their required tasks to other agents. The two-tier multi-agent architecture will then expand into a three-tier architecture with two levels of coordinator agents, and an expanding departmental tier. The three-tier architecture generates a departmental coordinator for each department that will be responsible for coordinating all activity within the respective department. The department coordinator will then respond to the coordinator of the entire society of agents. The three-tier architecture and responsibility distribution is illustrated in [Fig. 1].

The hierarchical architecture of MAVN is centered around the roles and responsibilities of the individual agents. In this specific design we are utilizing a variation of the master-slave organizational structure and we have developed three types of agents, the master agents (coordinator agents), the slave agents (departmental agents), and agents which will have duties as both a master and a slave (service agents). The departmental agents will be responsible to the coordinator agents and the coordinator agents are responsible to the customer. Service agents will have a unique dual role; they will be responsible to a coordinator but will also have coordinator responsibilities.

The master-slave-service architecture defines the specific roles each agent has within the dialogue of a conversation. Depending on agents' responsibilities, within the value net architecture, an agent will either have a passive or an active role in a conversation. The passive agent accepts a query from an external source and then reacts toward solving the given problem from their decision algorithms, beliefs, and / or environmental variables. After analyzing the problem or request and generating a solution, the passive agent will inform the speaker of a recommended solution. The active agent will issue the queries and also make the needed assertions to departmental and service agents based on customer requests and environmental conditions. The active agent does not have the ability to perform queries but can dictate and distribute them to departmental and service agents. Therefore within MAVN, the coordinators will have an active role, the departmental agents will have a passive role, and the service agents will have both a passive and an active role depending on the request.

3. Agent communication protocol

Agents communicates in order to achieve better the goals of themselves or of the society / system in which they exist. Communication can enable the agents to coordinate their actions and behavior, resulting in systems that are more coherent. Coordination is a property of a system of agents performing some activity in a shared circumstances. Cooperation is coordination among nonantagonistic agents, while negotiation is coordination among competitive or simply self-interested agents. Coordination is the process of managing dependencies between activities. Within MAVN the activities are managed through messages and interactions. Interactions occur when two or more agents are brought into a dynamic relationship through a set of activities, which have a strong influence on the future of the environment. In this research we have designed and implemented a communication language and protocol, which generates successful social interactions between agents and coordinates all activities within the multi-agent system. This language is called the Multi-Agent Value Net Communication Protocol (MVCP). MVCP will allow individual agents to understand the goals and activities of other agents through social interactions and therefore manage the relationships and communications passed within the system. Finally, MVCP will define a protocol needed to achieve the global goals within the multi-agent system.

MVCP is a message based communication language. The messages will be sent across the Internet or Intranets, depending on the location of both the sender and the receiver, using TCP/IP supported transport services. Since an agent is considered a programmable object, it will be able to exchange messages within a structured set of conversations between other agents or humans across the Internet.

3.1 Information/workflow within the multi-agent value net

The workflow and information flow of an order through the multi-agent value net can be split into two parts: information / workflow for an order proposal and information / workflow for an order. The information workflow for a proposal is shown in [Fig. 2].

After the proposal has been agreed upon by both the manufacturing facility and the customer, the following steps are processed.

1) After the customer has agreed upon an order, the proposal, constraints, activities, and final order are all passed to the Sourcing Agent.
2) The Sourcing Agent will store and distribute all the information regarding a final order throughout the life cycle of that order.
3) Each departmental agent will determine if the necessary activities are still feasible. If the activities are no longer feasible, a message is returned to the Sourcing Agent describing the problems.
4) If the Sourcing Agent receives a reject message, the Sourcing Agent will decide if an alternative plan can be developed. If a plan cannot be developed the customer is notified of the problems.
5) If the Sourcing Agent receives an Accept message from all the departmental agents, then the order information is sent to the Accounting Agent, and the Accounting Agent takes care of all the billings.

3.2 BDI model

The BDI (belief, desire, and intention) logic model is used to describe an agents' cognitive state. The BDI model proposed by Georgeff [Kinny and Georgeff, 1996] contains a library of plans and selected plans are added to the intention set through a plan recognition process. Belief of an agent is the set of facts about the environment or external world, that the agent believes to be true with different degrees of truthness. These facts could be either directly perceived and observed from the world or could have been deduced from other belief. An agent is continually updating its beliefs based on perceptions, using its beliefs to reason about possible plans, committing to certain intentions based on its beliefs and desires, and realizing these intentions by acting.

Let f denote a ground formula as in first-order logic i.e. a formula containing no variable, free or bounded. Belief of an agent (BEL) is a set of ground formula that the agent believes to be true in its world. If f is a ground formula that the agent p believes to be true then $BEL_p(f)$ (simply f) is a ground formula contained in its belief set. We will use the predicate BEL_s to describe the beliefs of the speaker, BEL_L to describe the beliefs of the listener, and f to denote a ground formula as in first order logic. The belief set (BEL) of an agent is a set of ground formulas that the agent believes to be true. Due to an application dependent set of rules we are able to illustrate the revision of an agents' beliefs as in equation (1) [Satapathy and Kumara, 1999].

$$BEL_L + BEL_S(f) \rightarrow BEL_S' \text{ (Belief Addition)}$$
$$Bel_L - BEL_S(f) \rightarrow BEL_S' \text{ (Belief Extraction)} \quad (1)$$

We also need to introduce another notation BEL tautologically implies f,

$$BEL \models f \text{ (meaning } BEL \text{ models } f) \textit{ iff } f \in (AD) \quad (2)$$

if and only if every truth about BEL also satisfies f and if and only if f exists within the algorithmic decisions of the agent. In other words the belief set of an agent contains the results of algorithmic decision created by the same agent. The algorithmic decision is generated from the agents' current belief set and the current set of constraints given to the agent by a speaker.

Desires are goals that an agent is trying to achieve. We will use the predicate $GOAL$ to describe the set of goals for a specific agent. Each goal within the set is considered as 'g' and $g \in GOAL$. Typically, 'g' is a ground formula that the agent is trying to achieve and therefore 'g' does not exist in the agents' current set of beliefs. However the agent believes that 'g' will exist in a future set of beliefs [Rao and Georgeff, 1995]. When the goal becomes a belief then 'g' will not exist in the set of goals but in the agents set of beliefs. We can add and extract goals within a set just as we had done with the belief set.

$$GOAL + g \rightarrow GOAL' \text{ (Goal Addition)}$$
$$GOAL - g \rightarrow GOAL' \text{ (Goal Extraction)} \quad (3)$$

Intentions are the set of plans or actions arranged in some order to accomplish the goals or desires of the agent. The resources agent will intend to use all of one product before ordering more of the same product. This will help the resources agent achieve the desire of low inventory for all raw material and subcomponents in the future by doing inventory distribution in the current state.

Let $INTENT$ denote a set of plans and each plan has to contain a set of actions ACT that when performed results in achieving a goal. Each action, designated by a within the set of actions ACT generates a set of expected results. Therefore we can designate a plan as a collection of actions on a time line by $PLAN(BEL, BEL')$. $PLAN(BEL, BEL')$ indicates that if an agent believes BEL then the actions in the plan can be executed and if the actions are successfully executed then the agent believes in BEL'. Assuming that a plan is an element of the intentions of an agent, we can denote an action a to be an intended action by $INTENT \models a$, stated as follows [Satapathy and Kumara, 1999]:

$INTENT \models a$ iff $(\exists P(BEL, BEL') \in INTENT)$ s.t. $(a \in P(BEL, BEL'))$ (4)

Similar to the addition and retraction of beliefs and goals, we can also define the addition and extraction of intentions as follows:

$INTENT + PLAN(BEL, BEL') \rightarrow INTENT'$ (Intent Addition)

$INTENT - PLAN(BEL, BEL') \rightarrow INTENT'$ (Intent Extraction) (5)

3.3 Semantics of the MVCP Performatives

The semantics of the performatives used in the agent based value net communication protocols are important for the customer-coordinator, coordinator-coordinator, service-coordinator, service-departmental, and coordinator-departmental interactions. The first groupings of performatives are used to interact with the knowledge base of another agent. Performatives in this group are ask, inform, retract, subscribe and unsubscribe. The second set of performatives is used to bind other agents not for the distribution of information but for the performance of a future action. Commit, uncommit, request and unrequest corresponds to this group. The third set of performatives describes the acceptance or rejection by the listener of a previously sent performative from a speaker. The performatives, accept and reject, corresponds to third group. Three group of performatives are summarized from Table 1,2 and Table 3 respectively.

The interactions within MAVN are defined by the messages sent between any of the agents or between the customer and the agent. They can be categorized into three types; coordinator / customer – coordinator, coordinator – departmental, departmental – external coordinator. As an example of the first case, performative **ask** can be considered. For example, a customer will issue a proposal for an order to the order processing agent over the Internet. The customer wants to know if the manufacturing organization will be able to fulfill the order within an appropriate amount of time, and therefore the message is defined as **ask**. The message protocol for the ask performative is defined in Table 4. The order processing agent will pass this information to the sourcing agent and the message will contain **ask** performative.

<Table 4> Message protocol for **ask** performative

```
<body>
   <contents> ? Proposal </contents>
   <ProposalNum> 1 </ProposalNum>
   <data>
      <ProductInfo>"        "</ ProductInfo>
      <ShippingInfo>"       "<ShippingInfo>
      <BillingInfo> "         "</BillingInfo>
</data>
```

4. Implementation of Multi-agents for value nets (04) (4)

An object-oriented Java and XML approach was employed in the development of MVCP, and thus can be used within many multi-agent systems currently deployed or under development and can handle data intensive messaging. Simple Object Access Protocol (SOAP) was adopted as a XML parser. SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses. SOAP can potentially be used in combination with a variety of other protocols. SOAP was designed in 1999 to use valid XML as a wire protocol, or packet layout language, that can be transported to a remote system, typically using HTTP. There are other software packages necessary for the development of MVCP. These packages are not as crucial, however are necessary for development. The three other software packages utilized were the Apache web server, Xerces Java Parser 1.3.1, and Xalan. The protocol stack adopted to implement MVCP is shown in [Fig. 3].

MVCP is a message based agent communication language, where XML is being passed as the content through the Internet. Within the MVCP package, there are many operations and programs that have been developed in order to receive a message, parse the message, place the message into a queue, and then send a message to another agent. In addition MVCP has been designed such that messages can be sent to an agent from an HTML based web page. Therefore MVCP performs header processing, error and exception handling, routing, invocation, and transformation.

The ProcessOrderForm class is designed specifically for HTML access to the multi-agent value net. This is a servlet that accepts HTTP POST messages from an HTML form. The servlet extracts information from the HTML form, and then places the pertinent information into an XML document.

The Client class is the central class for sending messages. This class creates the header and an envelope that wraps around the message content the agent would like to send. The header contains the information concerning the receiving agent, the sending agent, and the performative of the message. The body of the message varies depending on the sending and receiving agent and the performative, however the body of the message is encapsulated by XML.

Once the Client class creates the header and the body, the MVCP message is wrapped by SOAP. SOAP then creates its respective header and body around the MVCP message. After the MVCP message is wrapped by SOAP, the entire package is sent to the respective agents via HTTP, which uses TCP and IP as the transport layer. When a message is sent from one agent to another, the Exchange class initially receives the message. Currently Exchange.java accepts the HTTP transport protocol; however can be easily altered to support various transport protocols, such as SMTP, JMS, FTP, etc. A majority of the operations performed on a message prior to reaching an agent are done in the Broker class. After the message is considered authentic, the Broker class determines where the message should be forwarded based on the performative and content. The Archive class is designed to illustrate an agent, and the reception of a message.

5. Conclusions

With the availability of advanced information technology, considerable research has been conducted on generating a flow of information between suppliers, manufacturing facilities, and distribution centers in real time. However this solution only enhances information distribution and not information processing. Therefore in this research, we wanted to create a network of intelligent real time decision makers that are customer centric. In this paper, we described the solution to this new business problem as a multi-agent value net model. We described the multi-agent architecture, and communication language necessary for a customer centric value net implementation.

Traditionally, the communication amongst distributed agents is handled by variations of KQML. However KQML has many limitations when applied to heterogeneous systems that are distributed across the Internet. This problem can be avoided by creating a communication language that utilizes Internet technologies with KQML logic structure. Therefore we need to solve the agent communication problem within the value net model. This language must be able to integrate with customers, other agents, legacy systems, and databases anywhere in the world. The MVCP language is one such language that was motivated by this specific problem. MVCP was developed with Java, XML, and HTTP technologies and KQML logic structure can be incorporated into the message system.

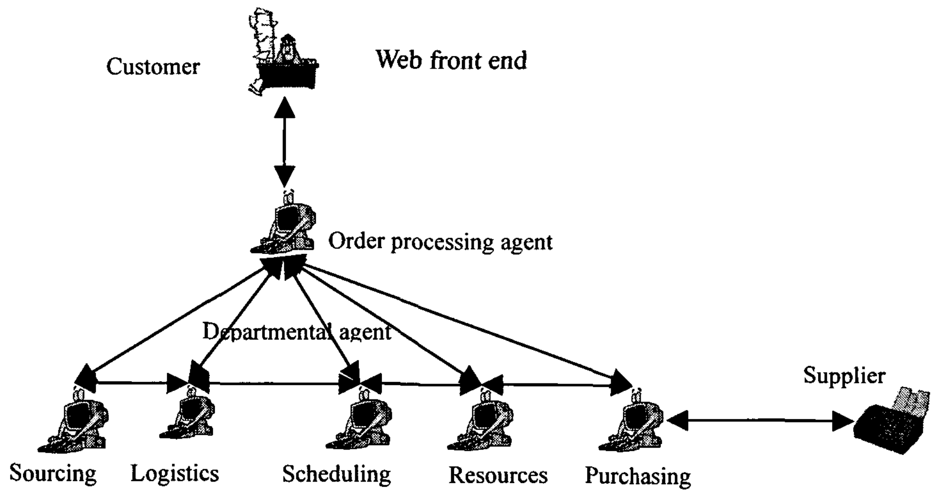
A full-scale implementation of a multi-agent value net requires future research and development. This research can be split into three areas: agent development, agent communication and general implementation issues. With regard to the agent development, all agents should have the knowledge base, the ability to learn, and the algorithms necessary to make educated decisions. For communication between agents, the language should contain a directory service, network security and message delivery. The third issue involves solving time constraint and wireless capabilities.

Acknowledgements:

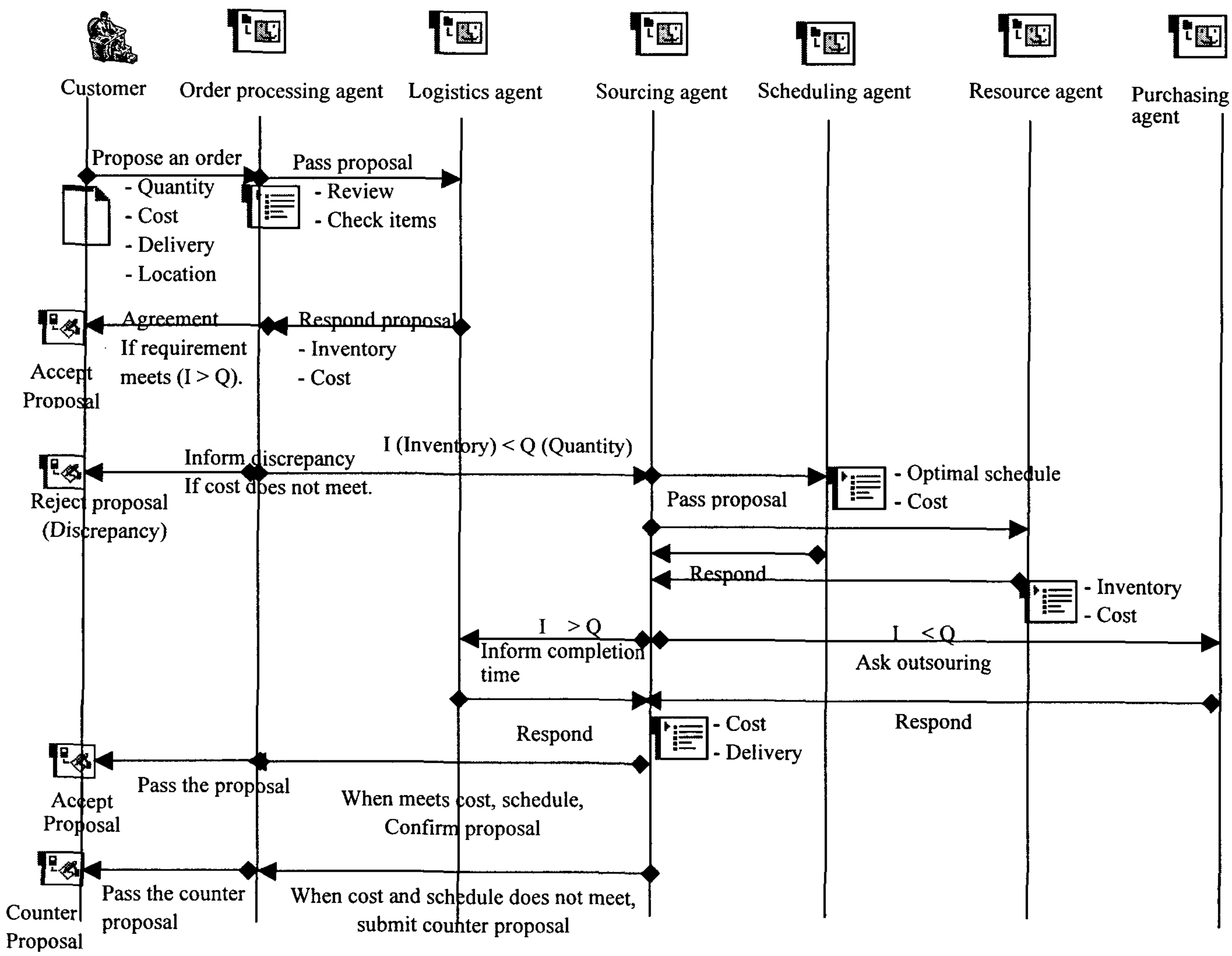
This work was supported by grant No. 2001-31500-004-1 from the Basic Research Program of the Korea Science & Engineering Foundation.

References

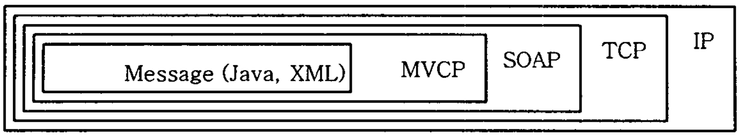
1. Bovet, D., and Martha, J., *Value Nets: Breaking the supply chain to unlock hidden profits.* John Wiley and Sons, Inc: New York, New York, 2000.
2. Fox, M.S., Chionglo, J.F., and Barbuceanu, M., "The integrated supply chain management system," http://www.eil.utoronto.ca/ Department of Industrial Engineering, University of Toronto, Toronto, Ontario, Canada, 1993.
3. Kinny, D. and Georgeff, M., "Methodology and modeling technique for systems of BDI agents," In *Intelligent Agents III: Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages (ATAL-96), Budapest, Hungary (Lecture Notes in Artificial Intelligence, Volume 1193)*, page 56, August 12-13, 1996.
4. Rao, A.S. and Georgeff, M., "BDI Agents: from theory to practice," In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95).* 312-319, San Francisco, CA, June 1995.
5. Satapathy, G., and Kumara, S.R.T., "Semantics of KQML Messages based on illocutionary acts and BDI model of an agent," *Working Paper Series, Department of Industrial and Manufacturing Engineering, Pennsylvania State University,* University Park, PA, 1999.
6. Weiss, G., *Multiagent systems: A modern approach to distributed artificial intelligence,* The MIT press, 1999.



[Fig. 1] Higher order Multi-Agent value Net (MAVN) Architecture



[Fig. 2] Information workflow process for the processing of a proposal



[Fig. 3] Layers of communication protocol

<Table 1> Performatives used to interact with the knowledge base of another agent

Performa-ives	Issued for the cases	Expected actions of listeners
Ask	The listener will assert the truth based on their knowledge base and / or algorithmic decisions.	$[BEL_L \models f] \Rightarrow$ [mess (inform(content, data))*]
Inform	Giving the listener information based on the experiences or algorithmic expression generated by the speaker.	$BEL_L + BEL_S(f) \rightarrow BEL_L'$
Retract	Which information (f), previously given by the speaker, should be disregarded.	$BEL_L - BEL_S(f) \rightarrow BEL_L'$
Subscribe	A form of the ask performative, which is only initiated once and is continued until the unsubscribe performative has been issued.	$[[BEL_L^{t_0} \models f_{t_0}] \Rightarrow$ [mess (inform(content, data))]] $\wedge$ $[BEL_L^{t_0} + BELs(f) \rightarrow BEL_L^{t_0}] \wedge$ $[[\forall t > t_0, BEL_L^t \models BELs\ (f_t)] \wedge \Rightarrow$ [mess (inform(content, data))]
Unsubscribe	To retract a current subscribe message.	$BEL_L - BEL_S(f) \rightarrow Bel_L'$

* mess(k(contents,data)) means a message to be sent by the listener with performative k, containing the ontological expression content, and message content data.

<Table 2> Performatives used to bind other agents

Performa-ives	Issued for the cases	Expected actions of listeners
Commit	To commit the listener to perform a future activity (f).	$[GOAL_L + g_S(f)] \wedge [INTENT_L \models ACT] \Rightarrow$ [mess(Accept(content, data))[$[GOAL_L + \neg g_S(f)]$ $[INTENT_L \models \neg ACT] \Rightarrow$ [mess(Reject(content, data))]
Uncommit	To retract a previously sent commit message.	$[GOAL_L - g_S(f)] \wedge [INTENT_L - PLAN(BEL(f), BEL') \rightarrow INTENT_L'] \Rightarrow$ [mess(Accept(content, data))]
Request	When the speaker believes that a future activity (f) will be needed to successfully achieve a future goal.	$TRUST_L \Rightarrow [GOAL_L + g_S(f)] \wedge [INTENT_L \models ACT]$
Unrequest	To retract a current request	$TRUST_L \Rightarrow [GOAL_L - g_S(f] \wedge [INTENT_L - PLAN(BEL(f), BEL') \rightarrow INTENT_L']$

<Table 3> Performatives regarding acceptance or rejection

Performa-ives	Issued for the cases	Expected actions of listeners
Accept	After receiving a subscribe, unsubscribe, commit, and uncommit performative from the speaker.	$BEL_L + BEL_S(f) \rightarrow Bel_L'$
Reject	From the listener of a previously sent subscribe, unsubscribe, commit, and uncommit performative.	$[[BEL_L - BEL_S(f) \rightarrow Bel_L] \wedge [BEL_L \models f] \Rightarrow$ [mess(Inform(content, data))]

국문요약:

기존의 공급망이 디지털화되고 고객과 공급자간의 파트너쉽과 정보흐름에 있어서 고성능의 네트웍을 형성하는 새로운 개념의 네트웍을 가치넷 (value net)으로 정의한다. 이러한 가치넷은 고객이 그 중심에 위치하며 연관되는 제조 사이트와 공급자에 이르는 정보의 배분을 원활하게 해 준다. 본 연구의 목적은 복잡한 공급사슬의 흐름과정에서 발생하는 의사결정과정을 자동화하고 효율화하기 위해서 멀티에이전트를 이용한 프레임웍을 설계하고 구매업무와 관련된 영역에 대하여 본 시스템을 구현해 보고자 하는 것이다. 에이전트의 지능적 판단을 위해서는 BDI (Belief, Desire, Intension) 모델을 이용하였다. 본 연구의 결과는 B2B 및 e-Business에서 에이전트를 이용한 조달 및 획득과 관련된 업무에 적용이 될 수 있다.