가상기업 지원을 위한 컨텍스트 기반 멀티에이전트 시스템

A Context-based Multi-Agent System for Enacting Virtual Enterprises

이경휘(Kyung-Huy Lee)*, 김덕현(Duk-Hyun Kim)**

초 록

가상기업은 다양한 역할의 에이전트들이 상호 작용함으로써 공동의 목표를 달성하는 멀티 에이전트 시스템(Multi-Agent System)으로 구현될 수 있다. 다만, 가상기업을 지원하기 위한 MAS는 멤버들의 이질성(heterogeneity), 복합구조(complex structure) 등의 특성을 고려한 가운데 자율성(autonomy)과 역동성(dynamism)을 보장할 수 있어야 한다는 점에서 보다 고도화 된 기능(예: 컨텍스트)을 필요로 한다.

본 논문은 가상기업의 구성과 운영을 지원하기 위한 플랫폼으로 컨텍스트(context) 기반 MAS를 제안하기 위한 것이다. 컨텍스트 기반 MAS는 가상기업의 구성요소를 액터(Actor)와 액터 간의 인터랙션(Interaction), 그리고 액터 컨텍스트 (Actor Context)와 인터랙션 컨텍스트(Interaction Context) 등으로 정의한다. 컨텍스트 기반 MAS는 액터들에 대한 단순한 상태 정보뿐만 아니라 목표, 역할, 작업, 작업자의 시간, 장소, 사용 기기 등 상황 정보 즉, 컨텍스트를 활용함으로써 의사결정이나 실행의 신속성, 정확성과 자동화 수준을 높일 수 있다.

컨텍스트 기반 MAS는 다양한 컨텍스트를 제어하기 위한 컨텍스트 온톨로지, 컨텍스트 모델, 컨텍스트 분석기와 추론기 등으로 구현될 수 있다. 하나의 가상기업이 공동 기술개발 파트너를 찾는 예제를 통해 본 연구의 타당성을 검토하였다.

ABSTRACT

A virtual enterprise (VE) can be mapped into a multi-agent system (MAS) that consists of various agents with specific role(s), communicating with each other to accomplish common goal(s). However, a MAS for enacting VE requires more advanced mechanism such as context that can guarantee autonomy and dynamism of VE members considering heterogeneity and complex structure of them. This paper is to suggest a context-based MAS as a platform for constructing and managing virtual enterprises. In the Context-based MAS a VE is a collection of Actor, Interaction (among Actors), Actor Context, and Interaction Context. It can raise the speed and correctness of decision-making and operation of VE enactment using context, i.e., information about the situation (e.g., goal, role, task, time, location, media) of Actors and Interactions, as well as simple data of their properties. The Context-based MAS for VE we proposed('VECoM')may consists of Context Ontology, Context Model, Context Analyzer, and Context Reasoner. The suggested approach and system is validated through an example where a VE tries to find a partner that could join co-development of new technology.

* 대전대학교 IT경영공학과 교수

** 세종사이버대학교 정보경영전공 교수

1. Introduction

Owing to the technological advancement of computer network, digitization of conventional business processes is getting faster than ever. E-business on the Internet and mobile or ubiquitous business on Broad-band composite Networks are transforming traditional companies into extended enterprises and virtual enterprises.

A virtual enterprise (VE) is a temporary alliance of enterprises or VE members that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks [1]. Business Process Outsourcing is one popular practice of VE. There have already been many success stories of VE in small and medium businesses as well as large companies [3, 19].

A VE has several lifecycles stages: construction of business network, identification of business opportunities, formation of VE teams and projects, design and implementation of solutions, and dissolution of VE teams [12]. For a VE doing business is usually to search proper members, resources, competencies etc., and bind them to solve defined problems.

The effectiveness of a VE heavily relies on the efficiency of a VE enactment platform that can support the entire lifecycle activities of VEs. A VE enactment platform requires more advanced mechanism than traditional IT systems because it requires handling the following features of VEs:

- *Autonomy*: VE members behave independently, constrained only by their contracts and environments;
- *Heterogeneity*: VE members are independently designed and constructed, constrained only by the applicable interface descriptions;
- *Dynamism*: VE members join or leave with minimal constraints, thus, the configuration of VE can change at runtime;
- *Structure*: VEs have complex internal structures, reflected in the relationships among their members [18].

A VE enactment platform actually requires dynamic searching, binding, and invocation of proper business processes or virtual businesses (VBs). Only with static properties of VE members those virtual businesses could not be properly resolved because there should be many undefined conditions. Occasional human intervention or some mechanisms are necessary to help resolve such situation-dependent decision-makings or operations.

In this paper, we propose a context-driven Multi-Agent Approach for VEs to support the above mentioned properties of VEs, i.e., autonomy, heterogeneity, dynamism, and complex structure. Context generally means information that can be used to characterize the situation of entity [1]. Context, whether proactive or reactive, is used here to represent situations of a VE and VE constituents, such as individuals, organizations, VEs, person, place, and objects. VE members can create, operate, change, and destroy its own context as well as group context where they interact with others for some VBs. In this sense, contexts are modeled as a network of situations. Agents are persistent computations representing independent VBs. Multi-Agents that consist of interacting agents can be mapped into VE constituents. So, a VE is mirrored in a context-based Multi-Agent System, where collaboration among agents is structured in accordance with situational commitments.

A VE consists of multiple participants interacting each other throughout its lifecycle and the roles of participants usually change as time goes. So, an actor of a VE can be naturally mapped into an agent in a MAS. Meanwhile, creation and operation of a VE usually done under a situation of actors as well as the interactions among actors, which also changes as time goes. In our approach a context is created for a situation of problem solving for the VE. A VE would resolve many problems with different contexts, sequentially or in parallel. Due to the heterogeneity in communicating each other among agents, context should be integrated or merged in an understandable manner [7]. Collaboration among agents is structured in accordance with situational commitments for problem solving, each of which is represented as a finite set of actor sub-contexts and interaction sub-contexts.

The organization of this paper is as follows: In section 2, mapping VE into a Multi-Agent System is described via a formal definition of VE. Context-based modeling of VE is followed in section 3. Section 4 describes the components of the Context-based Multi-Agent System we proposed. In section 5 related works on context are briefly discussed. Section 6 draws conclusion.

2. Virtual Enterprise as a Multi-Agent System

2.1 Formal Definition of a VE

A VE can be formalized in a hierarchical manner without loss of generality. A VE includes at least one or more VEs as its business partners. It is quite a lot straightforward to model that each business partner, whether a user, an organization, or a VE, must be an agent.

A VE ν is defined as the following tuple:

$VE = \{A, G, R, T, C, f\}$, where

1) $A = \{A_1, A_2, \cdots, A_n\}$: a finite set of **actors**. For $\forall i$, $A_i \in \{INDIV$ (*individual actors*) $\subseteq A$ XOR $GROUP$ (*collective actors*) $\subseteq A\}$, that is, an actor A_i can be either an individual or a group, exclusively.
2) $G = \{G_1, G_2, \cdots, G_m\}$: a finite set of ***goals*** and subgoals, each of which is assigned to actors, $A \cup \nu$.

3) $R = \{R_1, R_2, \cdots, R_l\}$: a finite set of ***roles***, each of which participates in VE actors, $A \cup V$.
4) $T = \{T_1, T_2, \cdots, T_k\}$: a finite set of ***tasks*** (or services), each of which is a set of actions and interactions committed by VE actors with roles, for $\forall(i, j)$, $\{A_i \cap R_j\}$.
5) $C = \{C_1, C_2, \cdots, C_p\}$: a finite set of relevant ***contexts***, each of which supports situations comprising a set of contextual tasks $T^C \subseteq T$. It is noteworthy that a global *VE* context is implicit in a *VE* environment.
6) $f = \{f_1, f_2, \cdots, f_q\}$: a finite set of ***configuration rules***, each of which maps or transforms from one concept to another concept, for example, context mapping $f_i: C \rightarrow A \times C$, role assignment $f_j: R \times C \rightarrow A \times R \times C$ or task allocation $f_k: T \times C \rightarrow A \times R \times C$, etc.

Based on the VE definition, we use four-level hierarchy of virtual businesses, i.e., (1) VB project level, (2) (collaborative) problem solving level, (3) interaction level, and (4) actor level. From the viewpoints of VE and virtual businesses, VE performs VB project which consists of one or more problems to be collaboratively solved by actors and their interactions.

2.2 Mapping VE to Multi-Agent System

Generally, an agent is a piece of autonomous software created by and acting on behalf of an actor or actors' group. It is set up to achieve VE goal, with the characteristics of autonomy, interaction, reactivity as well as pro-activeness. VE formation is gone through agent setting with consideration of goals, roles, and tasks based on agent and interaction contexts in a given situation. VE business processes are also done by agent collaboration in accordance with agent interaction protocols such as communicative acts. As a base case, an agent may be an individual such as a person, a business partner, a resource, or VE. That is, a VE is an agent that comprises other agents, including other VEs.

Table 1 shows various types of agents which come from a number of concepts VE domain. We model the following types of agents for VE:

- *VE* Organizer agent (*VOA*): who plays organizing and facilitating problems for VE formation and operation. It includes creating, updating, and canceling contextual objects.
- *VE* Aggregator agent (*VAA*): who manages group context information and member agents. These may be further

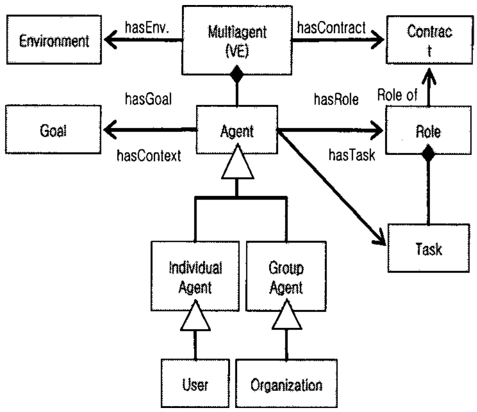
specialized to Customers, Providers, and Supporters VEAs.

- *VE* Member agent (*VMA*): who represents each member of business partners of VE, including individuals, organizations, and VEs.
- *Problem Solver agent* (*VSA*): who solves the problem issued by VPA. It can comprise various kinds of problem agents, for example, business partners selection, contracting, quality level determination, etc.
- *Service Repository agent* (*VRA*): who manages repositories of services or possible breeding environment.

⟨Fig. 1⟩ shows a conceptual model of a Multi-Agent System proposed here. As we see

Table 1. Comparison between VE domain and Multi-Agent System (MAS) domain

VE domain	MAS domain
• VE coordination	• VE organizer agent
• VE members (individual, group)	• VE member agent
• VE team (more than two different individuals and/or organizations)	• VE aggregator agent
• VE relation, networking and interaction	• VE agent interaction
• VE business processes (with decisions)	• VE problem solver agent
• VE repository	• VE repository agent



⟨Fig. 1⟩ Conceptual Model of a Multi-Agent System

in ⟨Fig. 1⟩ each agent has its role, task, and goal in a VE, for example, VSA_i = {*Goal(Gi): Find_Trustable_Business_Partner_for_Ti, Role(Ri): Problem_Solver_for_Ti, Task(Ti): Requirements (Skills(RequiredSkills), Experience(RequiredExp), PerformanceRating(RequiredPerf))*}.

3. Context-based Modeling of a Virtual Enterprise

Contexts have been used as various meanings in several areas in that it can capture many of the interesting aspects such as *relativity, partiality, locality, and independence* [5]. There have been made so many definitions in each of context based applications. In this research, we adopt one of the most widely used definitions of context characteristics as proposed by Dey and Abowd. They define context as *any information that can be used to* *characterize the situation of an entity.* An entity may be described as person, place, or object that is considered relevant to the interaction between a user and an application [5].

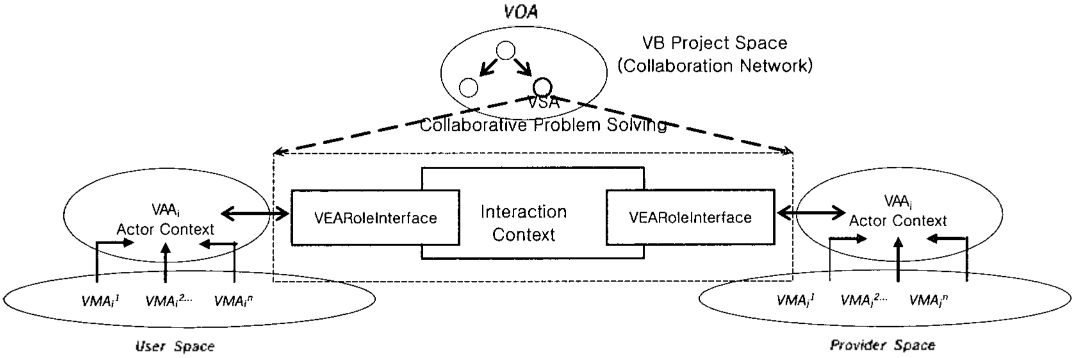
Recently, many research efforts on context have been focusing on user's preference in ubiquitous computing environment. Context and context awareness are necessary concept for relevant services and information to users based on situational conditions. Creating better context models which represent situations based on specific context information has always been a major goal in context aware computing. In modeling user-oriented or organization-oriented applications, modeling of context plays a critical role to get context information about current situations of users or organizations because different situation needs different information, decisions, processes, or services.

In the domain of VE, as we know the VE definition, VEs must cope with a lot of structural and/or behavioral things to be solved with collaboration of several actors and problem solvers, i.e., a set of interactions and interaction patterns of behavior. In other words, operation of VE requires situation-dependent interaction among various actors. This is why we try to combine context-driven approach and MAS in this research.

An entity in VE application refers to a VE itself, or a participant actor with a decision maker as an individual, or a group. Situation may be described in contextual information states including roles, tasks (a set of activities), and actions and interactions between VE internal actors and/or between VE and external actors. According to the context definition, therefore, the most important contexts are actor context and interaction context, the states of which determine VE situation.

3.1 Basic Context Model

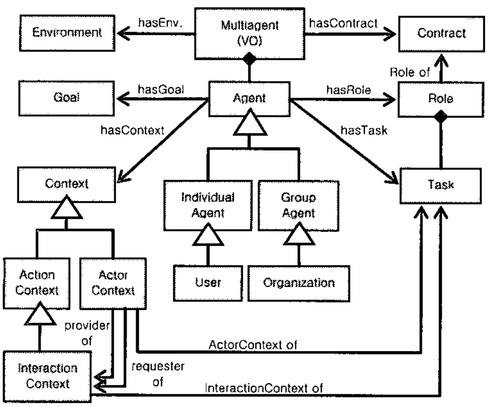
In our research, we use three kinds of contexts: *VE context* (VE Organizer agent, *VE* Problem solver agent), *actor context* (VE



〈Fig. 2〉 Basic Context Framework

Organizer, Aggregator, Member agent), and *interaction context* (Problem Solver agent). ⟨Fig. 2⟩ shows the basic context framework in VE. For example, consider the problem of business partner selection, in which VE Organizer agent (*VOA*) formulates the selection problem and then pass it to the Problem Solver Agent (*VSA*), based on VE context, that is, VE goal, role, tasks as well as time, location, etc. The *VSA* decompose the

problem into a set of sub-problems, that is, advertising, subscribing, evaluating, and selecting, etc. In order to solve each sub-problem, possible partners' identification and interaction (including communication) need to be proactively known in advance by means of context information including actor context and interaction context. The message of advertising task is sent to VE Aggregator agent (*VAA*), which in turn delivers to VE Member agent (*VMA*). In this problem



⟨Fig. 3⟩ A Conceptual Model of Context-based Multi Agent System

solving process, context information including actor and interaction contexts plays a critical role of situation awareness.

⟨Fig. 3⟩ shows a conceptual model of context-based model, which is an extension of ⟨Fig. 1⟩.

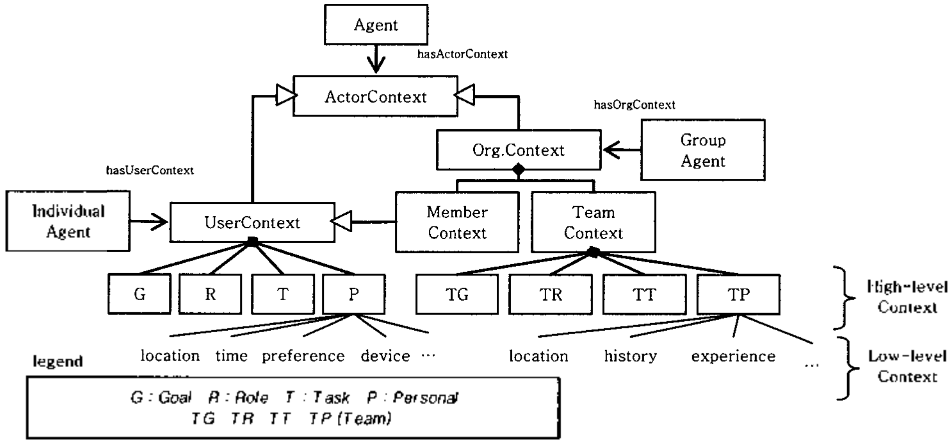
In the following, each of the contexts is further explained.

3.2 Actor Context

In *VE*, an actor means an individual or an organization and a group actor represents organized groups of interacting individuals with specific purposes such as teams or departments. A collective group actor may be further refined by nested relations. It includes agents which are software systems such as web services and other subsystems as well. Actor context plays a specific role within a given interaction context, that is, actor context specifies which roles are to be assigned and launches the necessary role assignment processes, in which a role of an actor may be described as an abstraction of the behavior of an actor.

⟨Fig. 4⟩ shows a VE actor context model proposed in this paper. *Actor context* is created when an actor participates in a VE with a specific role and destroyed when an actor exits from the VE. As seen in the figure, It may be figured out in two levels of context information:

1) high-level context, such as goal, role, task

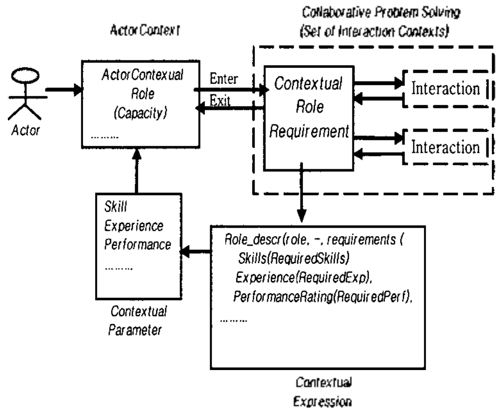


〈Fig. 4〉 VE Actor Context Model

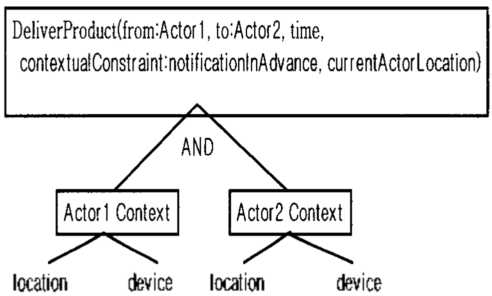
and environment that is associated with structural aspects;

2) low-level context, such as time, location, device, preference, history, experience, etc. that is associated with behavioral or operational aspects.

In high-level context, one of the most important use of context is to describe various roles taking part in problem solving processes, select business partners, and configure tasks (business processes) in a given goal. Actor context is necessarily considered in making such situational decisions. Fig. 5. shows an illustrative example to illustrate the use of such context variables and information on requirements basis, especially in role description and business partner selection. In specifying business processes and activities including interaction low-level context provides critical information such as



〈Fig. 5〉 Context Role



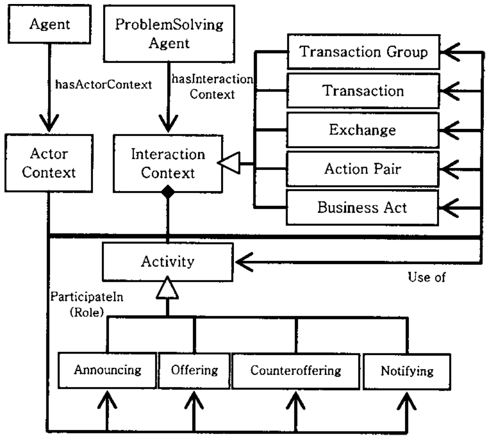
〈Fig. 6〉 Use of Actor Context

time, location, device, etc. of relevant actors in doing them. ⟨Fig. 6⟩ shows such an illustrative example to specify *DeliverProductProcess with considering location and device of actor1 and actor2 contexts.*

3.3 Interaction Context

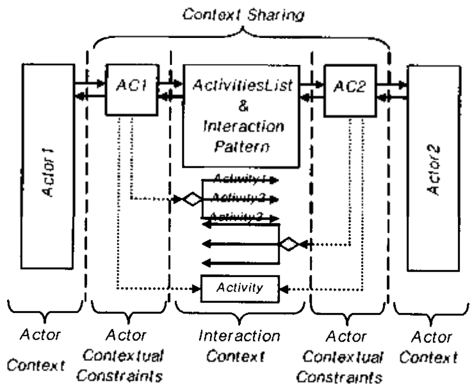
Business interaction represents the external manifestation of internal business processes. Lind and Goldkuhl classified business interaction as Transaction Group, Transaction, Exchange, Action Pair, and Business Act [14]. Interaction logic is regulated by specific business rules. The body of rules governing the information exchange shapes the interaction in accordance with specific patterns. On the other hand, different parts of an organization use different processes. Different rules are applied in different contexts. For example, the notification process used by the finance department can be quite different from the one used by the sales department.Formal specification and enforcement mechanisms can be quite different, if at all available.

Interaction context may be considered as procedural context which situates actor interaction behaviors, that is a list of activities and interaction pattern. Actors can share contextual elements of their individual context to build collaboratively a procedural context in



⟨Fig. 7⟩ VE Interaction Context Model

the interaction context. ⟨Fig. 7⟩ shows a VE interaction context model presented here. Interaction context consists of a set of activities such as announcing, offering, counter-offering, notifying, etc. and has several subtypes such as collaboration, negotiation, transaction, communicative act, and so on. Each type of business transactions consists of set of activities and an



⟨Fig. 8⟩ Context Interaction

interaction pattern based on context information. Therefore it is noteworthy that context determines interaction protocol specification by reasoning with interaction rules. 〈Fig. 8〉 shows the concept of contextual interaction which means how to build context sharing in a procedural interaction context.

3.4 Contextual Ontology and Reasoning

For each domain specific situation (a set of context instances), contextual ontology plays a critical role of giving contextual classification and process instance specification. Contextual ontology describes conceptual knowledge about the given domain, which is represented by classification and process ontology in the specific domain. The context model is linked to contextual ontology that enables high-level structural semantics as well as low-level operational semantics of the given situation. Once a set of context model objects have been created, situation can be assessed and according to the assessment results, contextual ontology can be selected appropriately.

〈Fig. 5〉 shows a partial list of VE contextual ontology. In the following, each ontology and ontology reasoning are further explained.

1) *Classification Ontology*: High-level contextual ontology deals with situation-aware classification problems of VE agents such as aggregator and problem solving agents. In a given domain, consumers groups are classified in three different kinds of subgroups, i.e., users, organizations, other VEs, respectively. Supporters group are comprised of delivery, payment, etc. In this context level, context is called abstract context, which does not specify how-to-do in order to achieve it.

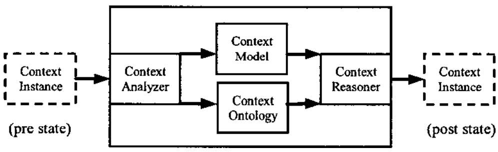
2) *Process Ontology*: Low-level contextual ontology is a kind of process ontology, which deals with operational processes in a given situation configured by means of low-level context information. That is, according to low level context data, how to take the operational processes varies a lot in all cases. For example, Technology Provision 〈*Technology Development*〉 process can be figured quite a lot differently out according to due date, cost, risks, etc. In this level of contextual ontology, abstract context is concretized accordingly.

3) *Context Reasoning*: Whether high-level or low-level contextual ontology, it is done by context reasoning, more specifically speaking, ontology reasoning. Ontology reasoning comprises a set of rules activating tree-search process in a contextual ontology.

4. Context-based Multi-Agent System

4.1 VECoM Architecture

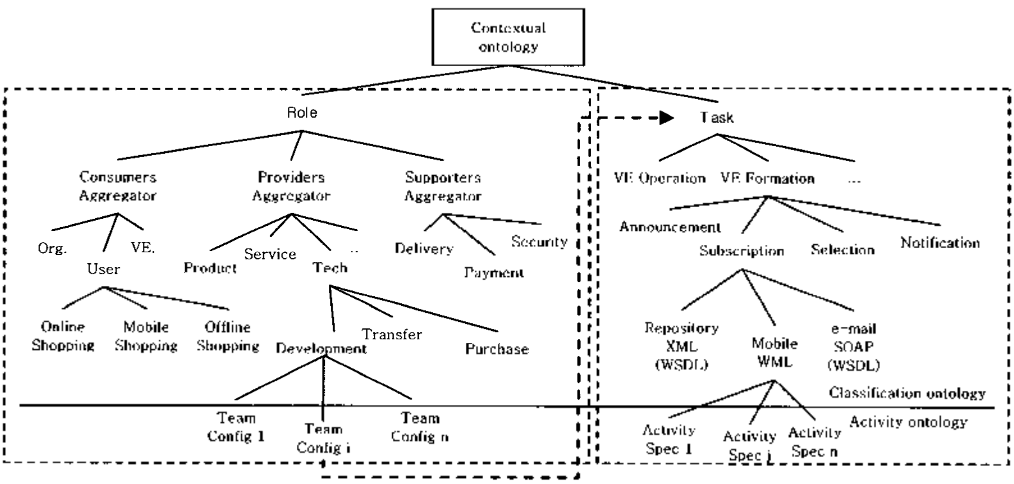
As noted before, an agent plays a specified role in performing VE tasks based on actor contexts and interaction contexts autonomously. ⟨Fig. 9⟩ shows conceptual architecture of the Context-based Multi-Agent System (VECoM), which may consist of four major components: context analyzer, context model, context ontology, and context reasoner.



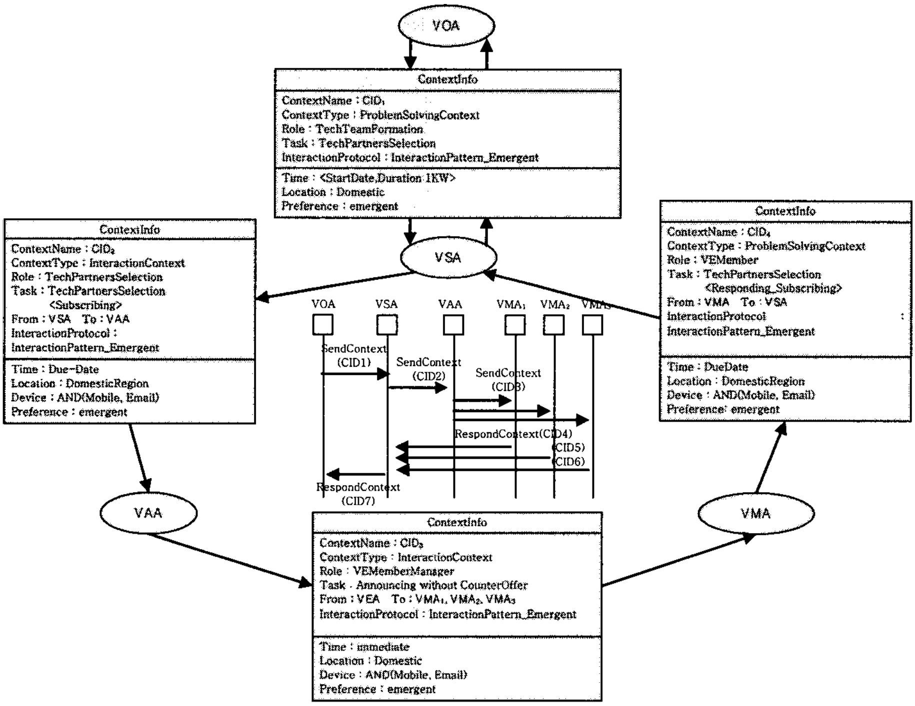
⟨Fig. 9⟩ Agent Architecture

Each of them is explained in a brief manner as follows:

1) *Context Instance (pre state)*: Agent performs tasks based on a role in a VE taking input of context instance from other agent. Context instance is a pre state of a given situation, which contains a lot of relevant context information about actors and/or problems.
2) *Context Analyzer*: Once an agent has taken a context instance, it interprets and decomposes the situation to be delivered in the context analyzer. Based on context information being extracted, context model and context ontology are configured with considering the situation awareness.
3) *Context Model*: Context model plays



⟨Fig. 10⟩ VE Contextual Ontology



<Fig. 11> An Illustrative Example

a role of instantiating a basic skeleton of new context for problem solving or tasks to be performed based on commonality of situations such as roles, tasks, and so on.

4) *Context Ontology*: The basic context skeleton is further refined with context ontology which has been properly selected in the domain and situation. That is, context ontology comprises multiple set of context ontology, each of which represents classification and processes in a specific domain, e.g., technology development, technology transfer, and so on, as shown in <Fig. 10>

5) *Context Reasoner*: With context model instance and context ontology, context reasoner creates new context which is a post state of context instance. A lot of rules, cases, and context operators such as conjunction, disjunction, etc. may be used in performing this task. In an emergent situation, for example, a rule selects an emergent specification of an interaction protocol.

6) *Context Instance (post state)*: All context information is transformed and then contained in a context instance for doing further processes in other agents.

4.2 An Illustrative Example

Let's consider a simple VE example, in which organization A (*VOA* agent) is looking for capable partners collaborating tasks: ***TechDevelopment***. The contextual requirements for doing this are listed as follows:

- *time*, within one week, a proper partner must be selected because of *emergent* situation,
- *location*, a partner *VAA* is going for must be located in a domestic region,
- *device*, all possible communication devices including a mobile phone of contact point must be used for doing this,
- *interaction protocol (pattern)*, an interaction protocol must be figured out in a fast response manner, which directly responded to VSA without routing *VOA*.

⟨Fig. 11⟩ shows a partial list for satisfying the above contextual requirements. Context information is a kind of messages, but differs in including an explicit situation as well as doing ontology reasoning procedure for creating it. In the Figure, *VOA* deliver the context: CID_1 to *VSA*, and *VSA* sends the context: CID_2 to *VOA* after interpreting by means of context analysis and context reasoning. *VOA* who manages to keep the up-to-date context information of all members VMA, send the context: CID_3 to all members: VMA_1, VMA_2, VMA_3. Each of them responds his/her response to *VSA* in the specified time and devices. After analyzing the responses from the members and selecting the proper member with an emergent model of business partner selection, *VSP* notifies the result to *VOA*.

5. Related Work

Much research on context modeling has been done in context-aware computing. Context models usually represented on the basis of ER model or Object-oriented model[11]. Some research work such as Comprehensive Structured Context Profiles (CSCP)[10], Context Broker Architecture (CoBrA)[4], or Context Ontology Language[17] adopt ontology for context in a specific domain to reach the goals of knowledge sharing across distributed systems. We only adopt existing research on context and context models to VE modeling.

In business domain, lots of research on business contexts has been done based on their own point of views till now. Research trends on business context moves from the view of giving simple situational information (e.g., time, location), to the view of making complementary approach of critical constituents of business information systems (e.g., first-class objects, components, middleware, etc.). Table. 2 shows principal results on business context research. Although those noteworthy results exist, research on context and situation is still in its infancy stage.

Table 2. Related work on explicit approach of business contexts

Where	Criteria	Taxonomy
ebXML *http://ebxml.org/specs/ccDRIV.pdf*	Standard classification scheme for each of the context areas	- Business Process context - Geopolitical context - Product/Service context - Legislative/Conventional context - Industry/Sector context - Role context
Object Management Group[16]	Each context to capture the collaboration of endpoints required to complete a business transaction	- Payment context - Identity context - Settlement context - Orders context - Risk context - Product Orders context - Market Orders context
Web Services Context (WS-Context)[20]	Information related to the services associated with the framework (e.g., security)	- Context Service Context (contextidentifier:URI, activity-service, optional identifier, optional list of the services currently participating in the activity, optional list of child activities, timeout value)
Zacarias et al. 2005[21]	Engineering approach: Enhancing automated actors or reasoning mechanisms	- Action context (Task, Role, Individual, Time) - Activity context - Interaction context

6. Conclusion

In this paper we proposed a context-based Multi-Agent System for supporting Virtual Enterprises. In our research, we view context as models of elements and instances as well as contextual ontology enhancing situation awareness with ontology reasoning capability. We also view context as actors and problem solving interactions between different problem spaces.

The contribution of this paper is modeling of VE using two complementarily concepts, i.e., context and Multi-Agent System. MAS is for modeling various actors in a VE with different and evolving roles. Context is for modeling the situation of a problem solving, which consists of actor context and interaction context. Finally, actors in a VE are mapped into agents in a MAS; and a problem solving dynamism is mapped into a context. A VE as a finite set of problems to be resolved by multiple actors is modeled as a MAS with multi-level contexts. This will guarantee the autonomy of actors in a VE while effectively manage heterogeneity and dynamism in a VE.

In the next research stage, we will try to implement the 'VECoM' in Java middleware environment. We are considering JADE that is one of the most widely deployed open source agent systems. A few efforts such as WSIG [8] are trying to integrate Web Services on JADE. It will support the interoperability of various constituents in a VE by incorporating OWL, WSDL, BPELWS, and SOAP, etc. WSDL can be used for describing agent services, SOAP for messaging of FIPA ACL messages, OWL-S for mapping WSDL elements onto ontology elements, and BPELWS for representing business process collaboration.

References

[1] Camarinha-Matos, L.M. Multi-Agent Systems in Virtual Enterprises, *Proceedings of AIS'2002-International Conference on AI, Simulation and Planning in High Autonomy Systems*, Lisbon, 2002.

[2] Cardoso, H.L. et al. Supporting Virtual Organizations through Electronic Institutions and Normative Multi-Agent Systems, Center for Organizational Engineering, INSEC, 2005.

[3] CE-NET, web site (www.ce-net.org).

[4] Chen, H. and Finin, T. An Ontology for a Context Aware Pervasive Computing Environment, *IJCAI workshop on Ontology and Distributed Systems*, Acapulco MX, August 2003.

[5] Dey, A.K. and Abowd, G. Towards a Better Understanding of Context and Context-Dependent Awareness, Georgia Institute of Technology, 2001.

[6] Do, V.T. et al. A Context-based Approach to Support Virtual Enterprises, *Proceedings of the 33rd International Conference on Systems Sciences*, 2000.

[7] Firat, A. Information Integration using Contextual Knowledge and Ontology Merging, Ph.D. Dissertation, MIT, 2003.

[8] Greenwood, D. JADE Web Service Integration Gateway (WSIG), JADE tutorial AAMAS 2005.

[9] Gutenrag, E. and Greory, D. XML-based Rules: Automating Business Context Categorization to Produce Semantic Interoperability: The rules-based ebXML approach, *Proceedings of Extreme Markup Languages 2001* (Montreal, Quebec), 2001.

[10] Held, A., Buchholz, S., and Schill, A. Modeling of Context Information for Pervasive Computing, *Proceedings of the 6th World Multiconference on Systems, Cybernetics and Informatics (SCI)*, Orlando, FL, July 2002.

[11] Henricksen, K., Indulska, J. and Rakotonirainy, A. Modeling Context Information in Pervasive Computing Systems, *Proceedings of Pervasive Computing*, Zurich, August 2002.

[12] Kim, Duk-Hyun. A Generic Architecture of a Virtual-Enterprise Broker, *Int'l J. of Industrial Engineering? Theory, Applications, and Practice*, Vol. 10, No. 4, Dec 2003, pp. 384-390.

[13] Kim, T.Y. et al. Metamodelling approach for Virtual Enterprises, *Computers In Industry*, 2006.

[14] Lind, M. and Goldkuhl, G. The Constituents of Business Interaction-Generic Layered Patterns, Data & Knowledge Engineering,Vol. 47, Vol. 3, 2003.

[15] Norman, T. J. et al. Agent-Based Formation of Virtual Organizations, Knowledge-Based Systems, No. 17, 2004.

[16] OMG, Model Driven Payment Gateways Initiative, 2003.

[17] Strang, T. Linnhoff-Popien, C. and Frank, K. CoOL: A Context Ontology Language to enable Contextual Interoperability, In LNCS 2893: *Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems*. 2003.

[18] Udupi, Y. B. and Singh, M.P. Contract Enactment in Virtual Organizations: A Commitment-Based Approach, *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, July 2006.

[19] VENet, web site (www.virtual-organization.net).

[20] W3C, Web Services Context (WS-Context), 2003.

[21] Zacarias, M. et al., Enhancing Collaboration Services with Business Context Models, 2005.

저 자 소 개

이경휘 (E-mail : khlee@dju.ac.kr)

1983.	서울대학교 산업공학과 (학사)
1985.	KAIST에서 산업공학 (석사)
1985 ~ 1990.	국방과학연구소 연구원, 한국생산성본부 경영컨설턴트
1991 ~ 1995.	POSTECH 산업공학과 (박사)
1997 ~ 현재	대전대학교 IT경영공학과 교수
관심분야	Business Information System, e-Business/ Virtual Enterprises, Business Intelligence, Web Service

김덕현 (E-mail: dhkim@sjcu.ac.kr)

1976.	서울대학교 산업공학과
1988 ~ 1993.	KAIST 경영과학과 (석사) (박사)
	국방과학연구소에서 MIS 개발
	CALS 기술개발
2001 ~ 2002.	㈜핸디소프트 연구개발본부장
	KAIST와 아주대의 대우교수
2003 ~ 현재	세종사이버대학교 교수
관심분야	가상기업 아키텍처, e-비즈니스 전반, BPM, SCM