협업 네트워크 조직의 아키텍처 모델링 언어

An Architecture Modeling Language for Collaborative Networked Organizations

김덕현(Duk-Hyun Kim)*

초 록

'협업 네트워크'는 가상기업, 동태적 공급망, 전문가 가상 커뮤니티 등을 포함하는 새로운 연구 분야로서 협업 네트워크 조직(CNO)의 생성과 운영에 초점을 둔 모델링 언어가 필요한 상황이다. 본 논문은 모델 자체의 표현력과 정보시스템의 구현 용이성을 모두 높인, CNO 대상의 아키텍처 모델링 언어(CAML)를 제안하기 위한 것이다. CAML은 모델 기반 아키텍처(MDA)에 입각해서 메타 모델과 도메인 모델을 통합하고 자크만 프레임워크에 입각해서 데이터, 프로세스, 링크, 참여자, 사건, 목적 등의 여러 초점을 통합할 수 있게 한 것이다. 공급망 문제의 모델링과 모델 변경 영향 분석을 통해 CAML의 유효성을 검토하였다.

ABSTRACT

Reference modeling for *Collaborative Networked Organizations*(CNOs) has just started, and Enterprise Architecture Modeling Languages(EAMLs) for CNOs are very few. Lack of reference models makes it difficult for people to communicate with each other and lack of EAMLs also makes it difficult to implement information systems for CNOs. We propose an EAML for CNO called CAML. It supports (1) *multi-level modeling* based on Model-Driven Architecture of OMG's for expressive power and efficiency of implementations, and (2) *multi-focus modeling* based on Zachman Framework for completeness of modeling. The effectiveness of the CAML is investigated through modeling of a supply chain and execution of change impact analysis.

키워드 : 협업 네트워크 조직, 아키텍처 모델링 언어, 변경 영향 분석
Collaborative Networked Organizations, Architecture Modeling Language, Change Impact Analysis

* 세종사이버대학교 경영학부 정보경영 교수
2008년 08월 19일 접수, 2008년 10월 29일 심사완료 후 2008년 11월 03일 게재확정.

1. INTRODUCTION

A ***collaborative network*** (CN) is 'an alliance constituted by a variety of entities (e.g. organizations and people) that are largely autonomous, geographically distributed, and heterogeneous, but that collaborate to better achieve common or compatible goals, and whose interactions are supported by computer network'[16]. A ***(collaborative) networked organization*** (CNO) is a manifestation of CN, which includes extended enterprise, virtual enterprise/organization (VE/VO), dynamic supply chain, value web or constellation, professional virtual community (PVC), virtual laboratory, Virtual organization Breeding Environment (VBE), etc[3, 5, 19]. VBE is a pool of organizations to support creation and operation of VOs, and it usually exists longer than any member organizations[1].

Recently, lots of CNOs appear in industry, government, academia, and society[3, 19], which arises from changes of both business environment and technology infrastructure. In business sense, CNOs are a result of segmentation and/or distribution, outsourcing, and globalization of business functions against increasing competition. In technology sense, they are a result of rapid development and diffusion of ICT including the Internet and web technology, which makes it easy for an organization to digitize business processes and integrate them. With the advancement of emerging technologies including ubiquitous computing and networking CNOs will be more popular in all around the world soon.

Enterprise architecture (EA) is a practice of applying an *EA framework* for describing strategy, business, applications, information or data, technology, and outcomes of an enterprise such as a CNO. It helps develop, manage, and use EA that in turn increases (1) agility for business and/or technological changes, (2) ROI of ICT investment thru checking interoperability between new and legacy applications, and (3) collaboration between managers/users and IT professionals. *EA frameworks*, e.g., Zachman Framework, GERAM, RM-ODP, FEAF, DoDAF, and TOGAF, have been developed and applied to many enterprise applications[5, 22]. A ***modeling language of enterprise architecture*** (EAML) is conceptual/logical representation of EA, which is requisite for developing and maintaining information systems. EAMLs also have been developed accompanying with EA frameworks, and they could be classified into two groups : one (e.g. IDEF, NEML, ARIS) focuses on organization and processes, the other (e.g. UML, ACME) focuses on technology and applications[4, 16].

Information systems for CNO are more complicated than traditional ones for intra- or inter-organizational functions because of the distribution, heterogeneity, autono-

my, and dynamism of components. With effective EA framework and EAML for CNO, we could exactly define structural and behavioral characteristics of CNO and raise efficiency of development and maintenance of information systems for CNO. In CN research, reference modeling for CNO has just started and a rough model called ARCON has developed in the ECOLEAD project[3, 4, 5]. However, research on EAML for CNO, such as NEML[18] and ArchiMate [9, 10], is very few. Besides, those research results show limitations in expressive power or efficiency of implementation as a language. For example, as ARCON focuses on conceptual-level modeling there may be semantic loss while transforming models based on it into logical-level models, e.g., UML. Meanwhile, as ArchiMate and NEML support logical-level modeling there may be lack of modeling concepts for expressing unique features of CNO, e.g., location and goals of CNO and members.

The primary goal of this research is to develop an EAML for CNO (hereafter we'll call it ***CAML***) that can support multi-level and multi-focus modeling. '***Multi-level***' means that it includes meta-level models above domain-specific models, which is similar to Model-Driven Architecture (MDA) of OMG's. '***Multi-focus***s' means that it supports the six focuses in Zachman Framework, i.e., data (or *what*), function (or *how*), people (or *who*), location (or *where*), time (or *when*), and ends-means (or *why*). *Meta-modeling* refers to the way of representing meta-data and meta- knowledge[2]. It can raise intelligence of information systems in development, maintenance, integration, evolution, reuse of resources, and analysis of change impact[15, 20]. The resultant CAML has rich modeling constructs that can raise expressive power of modeling, and it makes a model self-reflective through meta-knowledge in meta-level models. In this paper we focus on defining modeling concepts rather than implementation issues of a modeling language.

The remainder of this paper is organized as follows. In Section 2 we firstly review fundamental characteristics of CNO and conventional EA frameworks and EAMLs for CNO. Section 3 describes modeling concepts of the CAML and section 4 describes application of the CAML to modeling of supply chain and execution of change impact analysis. Section 5 draws conclusion with further research and comparison of the CAML with other research results.

2. THEORETICAL BACKGROUNDS

2.1 Fundamental Characteristics of CNO

CNOs are much different in topology

(e.g., star, chain, or network), levels of co-operation (e.g., information exchange, transaction, and collaboration), stability (i.e., transient or persistent), interdependency (e.g., of resources, cost, IT), and mechanism for coordination (e.g., hierarchy vs. market debate)[18]. CNO is different from simple networked organizations (e.g., B2B e-Marketplace for buying or selling products) in the sense that members of it have common goals and there are rather complex procedures of doing collaboration. CNO requires sharing of resources/information and integration of functions or processes among members throughout entire life-cycle.

CNOs can be classified into supply chain, hub-and-spoke (or star), and peer-to-peer (P2P)[11] depending on topology of members. ***Supply chain*** usually consists of supplier, manufacturer, distributor, and further consumer of products and/or services, and this type of horizontal CNO is widely spread in most industries. Major concern of supply chain is horizontal integration or synchronization of procurement, production, and distribution functions to minimize inventory. ***Hub-and-spoke*** usually has one or more lead contractors that control several sub-contractors, e.g., 1st-tier and 2nd-tier suppliers, and this type of hierarchical CNO is popular in auto, telecom, and construction industry. Major concern of hub-and-spoke is command-and-control or vertical integration of sub-contractors. In ***Peer-to-peer*** (P2P) there is no leader or centralized control, but exist mutual relationships among members as in academic societies and film/pharmaceutical industries. It is the most advanced form of CNO as mutual collaboration among members is dynamically established and performed. Major concern of P2P is to get asynchronous and loosely-coupled integration of various functions.

There is little consensus on the life-cycle phases of a CNO, yet. In ECOLEAD project, life-cycle phases of a single CNO and those of VBE are separately but similarly defined as (1) creation (initiation and foundation), (2) operation and evolution, (3) metamorphosis[1, 4]. We define here the life-cycle phases of a CNO and VBE in a single context : (1) construction of a VBE, (2) identification of business opportunities, (3) formation of VE team(s), (4) design and implementation of solutions, and (5) dissolution of VE team(s).

2.2 Requirements for Modeling of CNO

Referring to above description, a CNO has more complex structure and behavior than traditional single or extended enterprises. In general, CNO has key properties that distinguish it from traditional organizations, i.e., distribution, heterogeneity,

autonomy, and structural complexity[21]. Distribution means that business functions and ICT infrastructure are distributed because resources (e.g., personnel, equipment, and facility) are usually physically distributed. Heterogeneity means that culture, business (strategy and processes), ICT infrastructure, etc. Autonomy means that members want to maintain their own identity and business area although they may have common goals. Structural complexity means that a CNO may have many different participants and it itself may belong to many different CNOs at the same time with different roles. Besides, the components of CNOs, e.g., members, tasks, products, and rules, are varying as time goes, which refers to dynamism of CNO.

Steen et al. suggested requirements of an EAML for CNO : (1) appropriateness, (2) ease of use, and (3) general quality criteria. 'Appropriateness' means expressiveness of various concepts in a CNO (e.g., actors, roles, activities, data, systems, protocols) [18]. 'Ease of use' means intuitive and graphical support, multi-levels of abstraction, formalism, etc. 'Quality' comprises generality, economy, orthogonal, consistency, coherence, etc. In this research we consider three key requirements of CAML : (1) completeness of underlying framework, (2) expressive power of language constructs, and (3) efficiency of implementation, i.e., ease of transformation from business (or conceptual-level) model to system (or logical-level) model. A modeling language is in general required both expressive power and efficiency of implementation, which needs trade-off.

2.3 Conventional Frameworks and Languages

Early contributions to reference modeling for CNO can be classified into three groups : (1) enterprise modeling, (2) organizational/management school (e.g., SCOR), and (3) VE/VO ICT-based projects (e.g., PRODNET)[5]. We here classify conventional approaches of enterprise modeling into five categories : (1) EA frameworks for general enterprise, (2) EA frameworks for CNO, (3) EAML for general enterprise, (4) EAML for CNO, i.e., CAML, and (5) EA modeling of software.

<Table 1> shows a summary of some notable research results in each category. Categories '(1)' through '(4)' focus on conceptual and/or logical modeling for description, while category '(5)' focuses on logical and/or physical modeling for implementation. Two approaches need to be integrated in an EAML for expressive power and efficiency of implementation. EA frameworks for general enterprise cover three to five views and up to six focuses, whereas, EAMLs cover relatively few aspects. For example, NEML and ArchiMate cover three

〈Table 1〉 A summary of EA frameworks and EAMLs

Cat.	Models	Aspects or architectural domains
(1)	Zachman Framework	◦ *views* : scope, business, system, technology, details ◦ *focuses* : what(data), how(process), where(network), who(people), when(time), why(motivation)
(1)	FEA(US Government)	◦ business(strategy and processes), performance measure, application, information/data, technology
(2)	***ARCON***[4, 5]	◦ *life-cycle of CNO* : creation, operation, evolution, metamorphosis or dissolution ◦ *modeling intent* : general concepts, specific modeling, implementation modeling ◦ *environment characteristics*, i.e., In-CNO, About-CNO a. In-CNO ; structural/componential/functional/behavioral dimension b. About-CNO ; market/support/societal/constituency dimension
(3)	***EAML***[17]	◦ *views* : enterprise, computational, information, engineering, technology, plus software organization
(3)	***ArchiMate***[9, 10]	◦ *aspects(focus)* : information, behavior, structure ◦ *layers(views)* : business, application, technology
(4)	***NEML***[18]	◦ *views* : business, ICT ◦ *focus* : structure, behavior, artifacts ◦ functional, operational(i.e., platform-independent, platform-specific)
(5)	Model-Driven Architecture(OMG's)	◦ Computation-Independent Model(CIM), e.g., MOF ◦ Platform-Independent Model(PIM), e.g., CWM, UML ◦ Platform-Specific Model(PSM) : CORBA, EJB, EDOC
(5)	UML(OMG's)	◦ *views* : functional(use-case)/logical(structure and behavior)/ component/concurrency/deployment

views, i.e., business, ICT or technology, and application or system. Regarding the complexity and dynamism of CNOs more modeling aspects need to be introduced to an EAML. An EA framework for CNOs, i.e., ARCON covers too many perspectives and dimensions and has somewhat inter-mixed views and focuses in the sense of Zachman Framework, which will be a burden for software engineers to model logical constructs of CNOs using an EAML.

3. MODELING CONCEPTS OF THE CAML

3.1 Basic Features of the CAML

3.1.1 Zachman Framework as the EA framework for CNO

In our approach Zachman Framework is regarded as the EA framework for CNO. It may have some weaknesses : (1) being

a conceptual framework it does not support software engineering in itself, and (2) the distinction between different views or focuses is not so clear or orthogonal. However, it has been widely applied to various problem spaces because of its sufficient modeling perspectives. In this sense, five views and six focuses of Zachman Framework are adopted to the CAML, as follows.

- Views : scope, business, system, technology, detailed representations
- Focuses : data ('what'), process ('how'), link ('where'), participant ('who'), event ('when'), and goals ('why').

3.1.2 Multi-level modeling based on MDA

CAML supports meta-level model above domain-specific model. *Meta-level model* consists of meta-meta model and meta-model, and *domain (-specific) model* consists of business model, system model, and technology model. The CAML becomes self-reflective through meta-knowledge in meta-level model. In addition, the level of alignment between business and ICT could be raised through modeling of three views of domain models in a framework. The characteristics of modeling constructs at each level will be further explained in this section.

<Table 2> shows modeling concepts at each level of the CAML.

3.1.3 Multi-focus modeling based on Zachman Framework

As explained above CAML supports six focuses of a problem, which provides completeness of underlying framework and the

〈Table 2〉 Multi-level modeling concepts of the CAML

Level	Modeling concepts	Remarks
(L0) Meta-meta model	*Entity, Relationship, Property*	first class modeling constructs
(L1) Meta-model	*Meta-entity*, i.e., Data, Process, Link, Participant, Event, Goal ; *Meta-relationship*, e.g., Share-some, Precede, Base-of, Xor, And, etc.	*Scope or context-level, Ontology model*
(L2) Business model	*Entity*, e.g., CNO, project, contract ; *Relationship*, e.g., is-a, use, refer, etc.	CIM in MDA ('M3'), *Conceptual-level model*
(L3) System model	e.g., class, attribute, operation, rule, etc.	PIM in MDA ('M2'), *Logical-level model*
(L4) Technology model	e.g., server configuration, network protocol, etc.	PSM in MDA ('M1'), *Physical-level model*
(L5) Detailed representations	N/A	Code in MDA ('M0'), *Instance level*

CAML itself. For example, suppose a CNO is created as a joint venture to develop a new product and it consists of globally distributed suppliers, manufacturers, distributors, and customers. Modeling of common goals, links of participants and resources, and principal events of control is essential for the CNO in addition to data, process, and participants that conventional EAMLs usually support. In the similar sense, OMG also develops six basic modeling packages for inter- and intra- enterprise integration : *business domain* ('what'), *business process* ('how'), *location* ('where'), *business organization* ('who'), *event* ('when'), and *business motivation* ('why')[7, 8]. The characteristics of modeling constructs in the CAML will be further explained in this section.

3.2 Meta-level Modeling Concepts

Meta-meta model ('L0') represents the first modeling constructs of the CAML, i.e., Entity, Relationship, and Property, which corresponds to *MOF Class*, *MOF Association*, and *MOF Attribute*. Property type represents structural property (i.e., attribute), behavioral property (i.e., operation), and rules or constraints. Meta-model ('L1') represents domain-independent entity/relationship types, i.e., meta-entity type and meta-relationship type. These are instances of the types in meta-meta-model. The meta-entity type represents the six focuses of the CAML. The meta-relationship type represents semantic primitives of various relationships. *Semantic primitives* refer to sharing of properties, existential dependency, cardinality or multiplicity between two entities. In the following, the properties of modeling concepts, i.e., attributes and operations of meta-entity types and meta-relationship types are explained. Among three domain models, only modeling concepts in business model and system model are exemplified. Modeling concepts in technology model will be addressed in future papers.

3.2.1 Meta-entity types

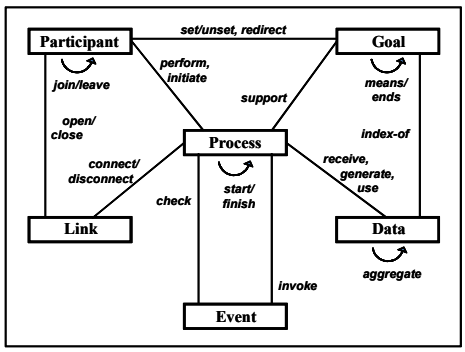
- ***Participant*** is for modeling the subject of activities occurring inside or outside of a CNO. Subtype of the Participant type will be described in business model as CNO itself, individual/group/company member, government, social organization, information system, etc. ; and it will be described in system model as actor of processes or agent software.
  - ◦ Attributes, e.g., role, responsibility, authority, access rights, duration
  - ◦ Operations, e.g., join or leave, perform, initiate, set, open or close
- ***Data*** is for modeling the object of life-cycle activities of CNO's. Subtype of the Data type, as input or output of

one or more processes, will be described in business model as information about product, service, project, contract, document, resource, etc. ; and it will be described in system model as database schema defining classes or types, attributes, operations, constraints, etc.

- ◦ Attributes, e.g., type, media, location, access path
- ◦ Operations, e.g., aggregate

- ***Process*** is for modeling a sequence of activities and control of flow. Subtypes of the Process type will be described in business model as function, procedure, transaction, or workflow ; and it will be described in system model as process.
  - ◦ Attributes, e.g., start/finish date or time, owner, input/output, flow (i.e., sequential or parallel)
  - ◦ Operations, e.g., start, finish, receive, generate, use
- ***Link*** is for modeling connection between two participants or processes. Subtypes of the Link type will be described in business model as communication/reporting channel, control, interface, collaboration path between two participants ; and it will be described in system model as interface between two processes or services.
  - ◦ Attributes, e.g., online/offline, connected nodes, network protocol, capacity, topology (e.g., hierarchical, horizontal, or network), type (e.g., direct/indirect)
  - ◦ Operations, e.g., connect/disconnect
- ***Event*** is for modeling specific time of control for participants, data, processes, and link. Subtypes of the Event type will be described in business model and system model as events.
  - ◦ Attributes, e.g., time and precondition of invocation, invoking processes
  - ◦ Operations, e.g., check, invoke
- ***Goal*** is for modeling motivation of participants. Subtypes of the Goal type will be described in business model as mission, purpose, strategy, means-ends of participants ; and it will be described in system model as rules with condition and action.
  - ◦ Attributes, e.g., assigned participants, level (e.g., strategic, tactical, operational), duration (e.g., years, months, days), measure of effectiveness
  - ◦ Operations, e.g., means/ends, redirect

<Figure 1> shows some relationship types between the above meta-entity types in the context of CNO. Although the name of relationship types may be different in each problem domain the semantics of them, we believe, could be consistently defined in



〈Figure 1〉 Entity-Relationship diagram of meta-modeling concepts

a meta-model independent of problem domains. This induces us to develop meta-relationship types, as in the following.

3.2.2 Meta-relationship types

The meta-relationship type refers to semantic primitives of relatively composite relationships in the real world, e.g., *IS-A*, *PART-OF*, *Use*, etc. Composite relationship is defined in the CAML as a combination of several semantic primitives. We adopt three categories of semantic primitives, i.e., sharing of properties, existential dependency, and cooperation from a semantic data model called FORM[15]. ***Sharing of properties*** means that two modeling concepts may have common properties or not. Three types of meta-relationships, i.e., ***Share-none***, ***Share-some***, and ***Share-all*** are defined in the CAML according to the degree of sharing of properties.

Existential dependency means that one entity (say, S) should be created/destroyed (or inserted/deleted) in a model before (or after) another entity (say, T) is done. Five types of meta-relationships, i.e., ***Isolate***, ***Alternate***, ***Precede***, ***Base-of***, ***Master-of*** are defined in the CAML. ***Isolate*** means that S can be created/destroyed independent of the existence of T. ***Alternate*** means that S should be created when T is destroyed, vice versa. ***Precede*** means that S should be in the problem space when T appears. ***Base-of***, as a specialization of ***Precede***, means that S should be in the problem space when T appears and deletion of S does not affect the existence of T. ***Master-of***, as a specialization of ***Base-of***, means that S should be in the problem space when T appears and deletion of S is allowed when there is no T.

Cooperation means the number and degree of participation of T for each S in a relationship. Five types of meta-relationships, i.e., ***None***, ***One***, ***Or***, ***Xor***, and ***And*** are defined in the CAML. ***None*** means that for each S none of T is participated in the relationship. ***One*** means that for each S only one of T is participated in the relationship. ***Or*** means that for each S several members of T are participated in the relationship. ***And*** means that for each S all members of T are participated in the relationship.

After all a composite relationship is represented as 3-tuple of meta-relationships. For example, *IS-A* relationship can be de-

〈Table 3〉 Rules in meta-relationship types(example)

Meta-Relation.	Definition	Rules of change (Condition ⇒ Action)
Share-none(S, T)	Property(S) ∩ Property(T) = 0	Insert(s) ⇒ (none) (where s ∈ S)
Share-some(S, T)	Property(S) ∩ Property(T) ≠0	Insert(s) ∨ Delete(s) ∨ Change(s) ⇒ Change(t)
Share-all(S, T)	Property(S) = Property(T)	Insert(s) ∨ Delete(s) ∨ Change(s) ⇒ Change(t)
Isolate(S, T)	Existence of S is independent of existence of T.	Insert(s) ∨ Delete(s) ∨ Change(s) ⇒ (none)
Alternate(S, T)	Either S or T, but not both, should be in a model.	Insert(s) ⇒ Delete(t), Delete(s) ⇒ Insert(t)
Precede(S, T), Follow(T, S)	S should be in a model before T is introduced into the model.	Num(S)>0 ⇒ Insertable(t)
Base-of(S, T), Branch-of(T, S)	T may be in a model if S is already in the model, and S may be removed from the model if there is none in T.	(Num(S) > 0 ⇒ Insertable(t)) ∨ (Num(T) = 0 ⇒ Deletable(s))
Master-of(S, T), Slave-of(T, S)	T may be in a model if S is already in the model, and T should be removed from the model if S is removed.	(Num(S) > 0 ⇒ Insertable(t)) ∨ (Num(T) = 0 ⇒ Deletable(s)) ∨ (Num(S) = 0 ⇒ Delete(T))
None(S, T)	For each s in S no t in T is participated in, i.e., Multi(S, T) = 0	Not available
One(S, T)	For each s in S only one t is participated in, i.e., Multi(S, T) = 1	Not available
Or(S, T)	For each s in S more than one t in T is participated in, i.e., 1≤Multi(S, T)≤Num(T)	Not available
Xor(S, T)	For each s in S at most one t in T is participated in, i.e., (Multi(S, T) = 1)∧(1≤Num(T)	Not available
And(S, T)	For each s in S all t in T is participated in, i.e., Multi(S, T) = Num(T)	Not available

Note) Num(T) refers to the number of members in T.
Multi(S, T) refers to multiplicity or the number of association between S and T.

fined as <*Share-some, Base-of, Or*>, *PART-OF* as <*Share-none, Master-of, And*>, *Refer-to* as <*Share-none, Isolate, Or*>, etc.

<Table 3> shows some rules of change that may be defined in each meta-relationship type.

3.3 Domain-specific Modeling Concepts

Domain model consists of conceptual-level business model ('L2'), logical-level System model ('L3'), and physical-level

technology model ('L4'). Each model of domain model supports software engineering phases of requirements analysis and definition, preliminary design, and detailed design. Business model and system model belong to PIM of OMG's, whereas technology model belongs to PSM.

Entity types and relationship types in domain models are subtypes of meta-entity types and meta-relationship types, respectively, so that properties are inherited to subtypes.

4. APPLICATION OF THE CAML

4.1 Meta-modeling of CNO

<Figure 2> shows a representation of a meta-model of general CNO. Note that we represent the model using XML-like syntax without formal definition as the focus of this paper is not on implementation of the CAML as a modeling language, yet.

```
< Meta-model name = 'CNO' >
< Meta-entity type = Participants >
   < Attribute name = String
               role = String
               responsibility = String
               access-rights = List
               performs = List(Process)/ >
   < Operation join = {} … / >
< /Meta-entity >
< Meta-entity type = Process >
   < Attribute name = String
               start-time = Time
               finish-time =Time
              performed-by = List(
                              Participants)
              started-by = List(Process)
              input = List(Data)
              output = List(Data)
              on-event = List(Event) .. / >
   < Operation start = {} … / >
< /Meta-entity >
…
< Meta-relationship type = Share-some >
   < Attribute source = Entity
               target = Entity … / >
   < Rule On-insert(s) OR On-delete(s) OR
         On-change(s) THEN Change(t)/ >
< /Meta-relationship >

< Meta-relationship type = Master-of >
   < Subtype-of Base-of / >
   < Attribute source = Entity
               target = Entity … / >
   < Rule On-delete(S) THEN delete(T)/ >
   < Rule Num(S) > 0 THEN insertable(t)/ >
< /Meta-relationship >
…
< /Meta-model >
```

〈Figure 2〉 A meta-model of CAML (example)

4.2 Domain-Modeling of a Supply Chain

In this section we show an example of modeling a typical supply chain, i.e., one of CNO that consists of many companies with different roles, i.e., part supply, manufacturing, distribution, transportation, etc. using the CAML. Suppose a simplified scenario of supply chain execution, as follows.

- A manufacturer (XYZ) performs de-

sign of a supply chain for its main product. The supply chain consists of three distributors (D1, D2, D3), two suppliers (S1, S2), and two transportation companies (T1, T2). XYZ sets goal of the supply chain to minimize the total inventory and minimize the total lead time from procurement of supplies to delivery of products to consumers. XYZ opens online and offline communication channels (i.e., Link) between XYZ and other members.

- D1 performs sale of goods to consumers. Sale of goods (a process) starts check inventory and request for replenishment of goods to XYZ when inventory reaches reorder point.
- ERP system of XYZ (a process) receives order (data) from D1, order receipt starts production scheduling, determination of material requirements, and request for raw materials and supplies to S1 and S2.
- ERP system of S1 and S2 receives order from XYZ, order receipt starts production of supplies, delivery of supplies to XYZ just on time, and request for payment.
- XYZ initiates production of goods that starts planning of delivery and request for transportation to T1.
- ERP system of T1 receives shipping order from XYZ, order receipt starts transportation scheduling and delivery of goods to D1.

Each process in the above scenario is started or stopped by other processes when

```
<Business-model name = 'A Supply Chain'>

<Entity type Company>
  <Subtype-of Participants>
  <Attribute CEO-name = String … />
</Entity>
<Entity type Person>
  <Subtype-of Participants />
  <Attribute competency = List … />
</Entity>
<Entity type Manual>
  <Subtype-of Process />
  <Attribute form-number = String … />
</Entity>
<Entity type Automatic>
  <Subtype-of Process />
  <Attribute software-name = String … />
  <Attribute hardware-name = String … />
</Entity>

<Relationship type = Perform>
   <Subtype-of (Share-none, Precede, Or)
     />
   <Attribute source = Participant
               target = Process … />
</Relationship>
<Relationship type = Generate>
   <Subtype-of (Share-none, Master-of,
     Or)>
   <Attribute source = Process
               target = Data … />
</Relationship>
<Relationship type = Start>
   <Subtype-of (Share-some, Precede,
     And)>
   <Attribute source = Process
               target = Process … />
</Relationship>

…
</Business-model>
```

〈Figure 3〉 A business model of a supply chain

```
< Company name = 'XYZ'
    role = 'Manufacturer'
    responsibility = 'Leader'
    access rights = (Create, Manage, Use)
    CEO-name = 'Kim' …
</Company >
< Company name = 'D1'
    role = 'Distributor'
    responsibility = 'sales thru e-Market'
    access rights = (Use)
    performs = (sale of goods,
      receive goods, …) …
</Company >

< Process name = 'update-inventory'
    performed-by = (XYZ, D1, D2, D3, S1,
        S2)
    on-event = on-goods-receipt or
        manager-request
    input = (Inventory M/F,
            Production Schedule) …
</Process >

< Data name = 'inventory M/F'
        generated-by = update-inventory …
</Data >
```

〈Figure 4〉 Instance description of a supply chain

some events occur and predefined conditions are met; and it also receives, generates, and uses proper data. <Figure 3> and <Figure 4> respectively show a part of business model and instance description of the supply chain. We find the structural and behavioral properties of the supply chain can be described without much loss of semantics in the real world using the CAML. In addition modeling itself is more efficient than modeling using other languages that do not support meta-level modeling because meaningful properties of the types in the business model inherit from the meta-entity types (e.g., Participant, Process, Link, etc.) and the meta-relationship types (e.g., Perform, Receive, Start, etc.) in the meta-model.

4.3 Execution of Change Impact Analysis

One of traditional issues in enterprise engineering is change impact analysis[6, 15, 20] the goal of which is to see what would happen if a change occurs before the change really takes place[6]. To show the expressiveness and self-reflexive feature of the CAML change impact analysis is given to the above business model of a supply chain.

Suppose a change occur to the supply chain, i.e., D1 (a distributor) leave the supply chain.

<Step 1> Identification of entities that may be affected by a change.

Analyzing the business model a system can identify the following entities and relationships among them that may be affected by deletion of D1 from the supply chain.

- D1 performs sale of goods to consumers.
- Sale of goods starts check inventory, check inventory starts request for replenishment.

- Check inventory generates inventory M/F.
- XYZ open communication channel between XYZ and D1.

<Step 2> Identification of the semantics of relationships among entities.

Next, the system can identify the semantics of the three relationships in the business model, i.e., *perform* = <*Share-none*, *Precede*, *Or*>, *start* = <*Share-some*, *Precede*, *And*>, *generate* = <*Share-none*, *Master-of*, *Or*>, and *open* = <*Share-some*, *Isolate*, *One*>. The system can also identify rules in some meta-relationships in the meta-model, i.e., for *Share-some*(S, T) a rule of 'Insert(s) ∨ Delete(s) ∨ Change(t) ⇒ Change(t)' and for *Precede*(S, T) a rule of 'Insert(s) ∨ Delete(s) ∨ Change(t) ⇒ (none)' are identified.

<Step 3> Applying rules of change to obtain the scope of change impact.

Applying rules to all entities connected thru relationships recursively the system can get all entities that need to be changed. After all, it is identified that 'deletion of D1' needs change of processes, e.g., check inventory and request for replenishment and change of data, e.g., inventory M/F.

5. CONCLUSION

On the basis of Zachman Framework and OMG's Model-Driven Architecture (MDA), we propose various modeling concepts for modeling languages of Enterprise Architecture (EAML) for CNO called CAML. It's an extension of existing research of EA frameworks and EAMLs. After all, the CAML supports five levels of views and six focuses of modeling, as follows :

- 5 views : meta-model, business model, system model, technology model, and detailed representations,
- 6 focuses : data ('what'), process ('how'), link ('where'), participant ('who'), event ('when'), and goals ('why').

Comparing with existing EAMLs the resultant CAML is self-reflective and has rich modeling constructs, which raises flexibility and expressive power of modeling as well as implementations. Modeling constructs in the CAML could be applied to existing EAMLs and/or EA frameworks. For example, adding link, event, and goal to structure ('who'), behavior ('how'), and information/artifacts ('what') in NEML and ArchiMate could enhance modeling power of them. Ten generic dimensions in ArchiMate, i.e., action, process, function, interaction, service, transaction, actor/component, role/interface, collaboration/connector, data object, could be redefined by the

six focuses and get more orthogonal perspectives.

As for ARCON we found (1) 'life-cycle of CNO' can be mapped into process and event in the CAML, (2) 'modeling intent' can be mapped into three levels of views, (3) differentiation of inside and outside of a CNO is not so critical because of ever changing roles of participants, and (4) 'In-CNO' and 'About-CNO', i.e., the 'environmental characteristics' can be mapped into various focuses. For example, 'structural dimension' could be mapped into participants (e.g., node) and link (e.g., relationships), 'market dimension' also could be mapped into participants (e.g., customer, competitor, contract), link (e.g., interaction with participants), and goal (e.g., strategy, mission).

To make the CAML more complete we plan to further investigate the following issues : (1) formal definition of modeling concepts in the CAML, (2) prototyping of the CAML as a modeling language, and verification of modeling concepts through applying to various types of CNOs.

References

[1] Afsarmanesh, H. (ed), Characterization of Key Components, Features, and Operating Principles of the Virtual Breeding Environment, ECOLEAD Deliverables D21.1, March 2005.

[2] Brodie, M. L., D. Bobrow, V. Lesser, S. Madnick, D. Tsichritzis, and C. Hewitt, "Future AI Requirements for Intelligent Database Systems," (ed.) L. Kerschberg, Expert Database Systems, Benjamin Cummings, 1989, pp. 45-62.

[3] Camarinha-Matos, L. M. and H. Afsarmanesh, "The Emerging Discipline of Collaborative Networks," Virtual Enterprises and Collaborative Networks, (ed.) L. M. Camarinha-Matos, IFIP, Vol. 149, Kluwer Academic Publishers, August 2004, pp. 3-16.

[4] Camarinha-Matos, L. M. and H. Afsarmanesh, "Towards a Reference Model for Collaborative Networked Organizations," Proceedings of Information Technology for Balanced Manufacturing Systems(BASYS06), Niagara Falls, Canada, September 2006, pp. 193-201.

[5] Camarinha-Matos, L. M., H. Afsarmanesh, F. Ferrada, A. Klen, and E. Ermilova, Rough Reference Model for Collaborative Networks, ECOLEAD Deliverables D52.2, March 2006.

[6] De Boer, F. S., M. M. Bonsangue, L. P. J. Groenewegen, A. W. Stam, S. Stevens, and L. van der Torre, Change "Impact Analysis of Enterprise Archi-

tecture," Proceedings of IEEE Information Reuse and Integration Conference, 2005, pp. 177-181.

[7] Frankel, D. S., P. Harmon, J. Mukerji, J. Odell, M. Owen, P. Rivitt, and M. Rosen, "The Zachman Framework and the OMG's Model Driven Architecture," Business Process Trends, White Paper, September 2003.

[8] Hendryx, S., Architecture of Business Modeling, OMG Document BR/2003-11-01, November 2003.

[9] Jonkers, H. et al., "Towards a Language for Coherent Enterprise Architecture Descriptions," Proceedings of the 7th IEEE International Enterprise Distributed Object Computing Conference (EDOC), 2003, pp. 28-39.

[10] Jonkers, H., M. Lankhorst, R. Buuren, S. Hoppenbrouwers, M. Bonsangue, and L. Torre, "Concepts for Modeling Enterprise Architectures," International Journal of Cooperative Information Systems, Vol. 13, No. 3, 2004, pp. 257-287.

[11] Katzy, B., C. Zhang, and H. Loeh, Reference Models for Virtual Organizations, Center for Technology and Innovation Management(CeTIM) Working Paper, Vol. 2704, March 2006.

[15] Kim, D. H. and S. J. Park, "FORM : A Flexible Data Model for Integrated CASE Environments," Data and Knowledge Engineering, Vol. 22, 1997, pp. 133-158.

[16] Kim, D. H., "Towards an Architecture Modeling Language for Networked Organization," Proceedings of the 8th IFIP Working Conference on Virtual Enterprise(PRO-VE'07), IFIP WG 5.5, Guimaraes, Portugal, September 2007, pp. 309-316.

[17] Sarkar, S. and S. Thonse, "EAML-Architecture Modeling Language for Enterprise Applications," Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business(CEC-East'04), 2004, pp. 40-47.

[18] Steen, M. W. A., M. M. Lankhorst, and R. G. Wetering, "Modeling Networked Enterprise," Proceedings of the 6th International Enterprise Distributed Object Computing Conference(EDOC), 2002, pp. 109-119.

[19] Tapia, R. S., What is Networked Enterprise, Technical Report TR-CTIT-06-23, Center for Telematics and Information Technology(CTIT), University of Twente, The Netherlands, May 2006.

[20] Thangarathinam, T., G. Wyant, J. Gibson, and J. Simpson, "Metadata management: the Foundation for Enterprise Information Integration," Intel Technology Journal, Vol. 8, No. 4, 2004, pp.

337-344.

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

[22] The MITRE Corporation, Guide to the Enterprise Architecture Body of Knowledge(EABOK), February 2004.

저 자 소 개

김덕현	(E-mail : dhkim@sjcu.ac.kr)
1976년	서울대학교 공과대학 산업공학과 (학사)
1988년	KAIST 경영과학과 (석사)
1993년	KAIST 경영과학과 (박사)
1976년~2000년	국방과학연구소 책임연구원/실장
2001년	(주)핸디소프트 연구개발본부장
2002년	KAIST 초빙교수, 아주대 대우교수
2003년~현재	세종사이버대학교 교수
관심분야	네트워크 기업, SCM, ITA/EA, u-러닝