TOWARDS AN ARCHITECTURE MODELING LANGUAGE FOR NETWORKED ORGANIZATIONS

Duk-Hyun Kim
Sejong Cyber University, KOREA
dhkim@sjcu.ac.kr

Reference models for Collaborative Networked Organizations (CNOs) is under development, but 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 supporting CNOs. For enrichment of reference models and EAMLs for CNO we develop modeling constructs of an EAML. It supports (1) multi-level modeling based on OMG's Model-Driven Architecture for expressive power and ease of implementation, and (2) multi-focus modeling based on Zachman Framework for completeness of modeling.

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' (Camarinha-Matos et al., 2006). ***Collaborative networked organizations*** (CNOs) (e.g. virtual organizations, dynamic supply chains) are manifestations of CN.

Recently, lots of CNOs, networked organizations, or joint ventures appear in industry, government, academia, and society (Camarinha-Matos and Afsarmanesh, 2004; Tapia, 2006). In business sense, it comes from digitization or virtualization of business and organizations for agility against increasing competition in (global) markets. In technology sense, it owes to the rapid development and diffusion of ICT including the Internet and web technology. With the advancement of emerging technologies including ubiquitous computing and networking CNOs will be more popular in all around the world soon.

'A ***reference model*** is an abstract representation of the *entities* and *relationships* involved in a problem space, and it forms the conceptual basis for the development of more concrete models of the space and ultimately implementations, in a computing context' (refer. www.wikipedia.org). ***Enterprise architecture*** (EA) is the practice of applying a reference model for describing strategy, business, applications, information/data, technology, and outcomes of an enterprise (e.g. CN or CNO). A ***modeling language of enterprise architecture*** (EAML) is a conceptual or logical representation of EA. To build an information system for a CNO, an EA of the CNO needs to be developed based on a reference model or an EA framework; and the EA needs to be implemented into information systems through an EAML.

EA frameworks have been developed and applied to many enterprise applications. Examples of EA frameworks include Zachman Framework, GERAM, RM-ODP, Federal Enterprise Architecture, DODAF, TOGAF, Model-Driven Architecture (MDA), etc. EAMLs also have been developed accompanying with EA frameworks. Existing EAMLs can be classified into two groups: one focused on *organization and processes* (e.g. IDEF, NEML, ARIS), the other focused on *technology and applications* (e.g. UML, ACME) (www.telin.nl).

In CN research, reference modeling for CNO has started and a rough model called ARCON is developed in the ECOLEAD project (Camarinha-Matos et al., 2006; Camarinha-Matos and Afsarmanesh, 2006). However, research on EAML for CNO such as NEML (Steen et al., 2002) is very few. Even such research results as ArchiMate (Jonkers et al., 2004) and ARCON show limitations in expressive power and ease of implementation as a language, or in completeness of underlying framework. Lack of reference models for CNO makes it difficult for specialists and non-specialists to understand the problem space, and to communicate with each other. Lack of proper EAML makes it difficult to implement information systems supporting CNO.

The ultimate goal of our research is to develop *an EAML for CNO* (hereafter we'll call it a ***CAML***), but in this paper we firstly define and suggest modeling constructs of a CAML. The CAML supports ***multi-level modeling*** based on OMG's MDA and ***multi-focus modeling*** based on Zachman Framework. 'Multi-level' means that it includes meta-level models above domain-specific models. 'Multi-focus' means that it includes all six focuses in Zachman Framework. Comparing with existing EAMLs the resultant CAML is self-reflective and has rich modeling constructs, which may raise flexibility and expressive power of modeling as well as implementations. Although this paper is an introductory work, it would help develop information systems as well as reference models for CNO.

The remainder of this paper is organized as follows. In Section 2 we review various modeling aspects in EA modeling research comparing with requirements of a CAML. Section 3 describes modeling constructs of the CAML. In section 4 this paper is concluded with comparison of the CAML with existing research results, and further research.

2 MODELING ASPECTS FOR CNOS

2.1 Requirements of a CAML

CNOs usually have different topology (e.g., star, chain, or network), different levels of cooperation (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) (Steen et al., 2002). A CNO has much more complex structure and behavior than traditional single or extended enterprises. This comes from the characteristics of CNOs: (1) *distribution* of participants and resources (2) *autonomy* of members, (3) *heterogeneity* of culture, business (strategy and processes), ICT infrastructure, etc. Besides, the elements of CNOs such as members, tasks, products, and rules are varying as time goes. Proper modeling of a CNO can help reduce the complexity and dynamism.

Steen et al. (2002) suggested requirements of a CAML, as follows: (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). 'Ease of use' means intuitive and graphical support, multi-levels of abstraction, formalism, etc. 'Quality' comprises generality, economy, orthogonality, consistency, coherence, etc. In this research we consider three key requirements: (1) completeness (of the underlying framework), (2) expressive power (of language constructs), and (3) ease of implementation. 'Ease of implementation' means ease of transformation from business (i.e., conceptual-level) model to system (i.e., logical-level) model. A model is in general required both expressive power and ease of implementation, which needs trade-off.

2.2 Modeling Aspects in Existing Research

Camarinha-Matos and Afsarmanesh (2006) classified early contributions to the reference model for CNO into three groups: (1) enterprise modeling, (2) organizational/ management school (e.g., SCOR), and (3) VE/VO ICT-based projects (e.g., PRODNET). We classify existing 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.

Table 1 – Aspects in EA frameworks for general enterprise

	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 & processes), performance measure, application, information/data, technology
(2)	***ARCON*** (Camarinha-Matos et al., 2006)	• *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'*** (Sarkar & Thonse, 2004)	• views: enterprise, computational, information, engineering, technology (which are aspects in IEEE 1471 & RM-ODP), plus software organization
(3)	***ArchiMate*** (Jonkers et al., 2004)	• aspects: information, behavior, structure • layers: business, application, technology
(4)	***NEML*** (Steen et al., 2002)	• business, ICT • structure, behavior, artifacts • functional, operational (i.e., platform-independent, platform-specific)
(5)	Model-Driven Architecture (OMG)	• Computation-Independent Model (CIM), e.g., MOF • Platform-Independent Model (PIM), e.g., CWM, UML • Platform-Specific Model (PSM): CORBA, EJB, EDOC • Code
(5)	UML (OMG)	• views: functional (use-case)/ logical (structure & behavior)/ component/ concurrency/ deployment

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 a CAML for expressive power and ease 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 views, i.e., business, ICT or technology, and application or system. Regarding the complexity and dynamism of CNOs more modeling aspects need to be adopted to a CAML. An EA framework for CNOs, i.e., ARCON covers too many perspectives and dimensions and has somewhat intermixed *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 CONSTRUCTS OF THE CAML

3.1 Design Rationale 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 that: (1) it is a conceptual framework not to 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 considered in 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 meta-modeling and MDA

Meta-modeling is a way of representing meta-data and meta-knowledge (Brodie et al., 1989). It helps make software self-reflective for development, maintenance, integration, evolution, reuse of components or resources, and analysis of change impact (Kim and Park, 1997; Thangarathinam, 2004). OMG's MDA is to create an EA modeling capability that analysts and developers can use to describe a company's business and software assets, so it is naturally related with Zachman Framework (Frankel et al., 2003).

In the above sense, the CAML supports meta-level models above domain-specific model. Meta-level models consist of *meta-meta model* and *meta-model*, and domain (-specific) model consists of *business model*, *system model*, and *technology model*. Multi-level models of the CAML provide expressive power and ease of implementation or transformation between different models. The characteristics of modeling constructs at each level will be further explained in this section.

3.1.3 Multi-focus modeling based on Zachman Framework

According to Zachman Framework the CAML supports six *focuses* or dimensions of modeling, i.e., data, process, link, participant, event, and goal, which provides completeness of underlying framework and the 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, link of participants and resources, and principal events of control is essential for the CNO. In the similar sense, OMG also develops six basic modeling packages for inter- and intra-enterprise integration and collaboration: *business domain* ('what'), *business process* ('how'), *location* ('where'), *business organization* ('who'), *event* ('when'), and *business motivation* ('why') (Hendryx, 2003). Note that most EAMLs usually support *data* ('what'), *process* ('how'), and *participant* ('who'). The characteristics of modeling constructs in the CAML will be further explained in this section.

3.2 Multi-level Modeling Concepts

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

Table 2 – Multi-level modeling concepts of the CAML

Level	Modeling concepts	Remarks
(L0) Meta-meta model	*Entity, Relationship, Property*	CIM in MDA ('M3'), first class constructs
(L1) Meta-model	*Meta-entity*, e.g., data, process, link, participant, event, goal; *Meta-relationship* e.g., composition, generalization, association	PIM in MDA ('M2'), *Scope or context level*, *Ontology* model
(L2) Business model	*Entity*, e.g., CNO, project, contract; *Relationship*, e.g., is-a, use, etc.	PIM in MDA ('M2'), *Conceptual level*
(L3) System model	e.g., class, attribute, operation, rule, etc.	PIM in MDA ('M2'), *Logical level*
(L4) Technology model	e.g., server configuration, network protocol, etc.	PSM in MDA ('M1'), *Physical level*
(L5) Detailed representations	N/A	Code in MDA ('M0'), *Instance level*

Meta-meta model ('L0') represents the first modeling constructs of the CAML, i.e., Entity, Relationship, and Property, which is compatible with MOF Class, MOF Association, and MOF Attribute in Computation Independent Model (CIM) of MDA. Property type represents structural property (i.e., attribute), behavioral property (i.e., operation), or rule. Meta-model ('L1') represents domain-independent entity types or relationship types, i.e., Meta-entity type and Meta-relationship type that are instances of the types in meta-meta-model. Meta-model is compatible with Platform-Independent Model (PIM) of MDA. The six focuses are subtypes of the *Meta-entity* type. The *Meta-relationship* type represents semantic primitives, e.g., property sharing, existential dependency, cooperation level, multiplicity, etc. (Kim and Park, 1997), detailed explanation of which is beyond the focus of this paper.

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, preliminary design, and detailed design. Business model and system model belong to PIM, whereas technology model belongs to Platform-Specific Model (PSM).

3.3 Multi-focus Modeling Concepts

In the following, the characteristics of modeling constructs including attributes and operations in six focuses are explained. Among three domain models modeling constructs in business model and system model are exemplified. Modeling constructs in technology model will be addressed in future papers.

• *Participant* ('who') is for modeling the subject of activities occurring inside or outside of a CNO. In *business model* it represents CNO itself, individual/ group/ company member, government, social organization, information system, etc. In *system model* it represents *actor* of processes or agent software.

- Attributes, e.g., role, responsibility, authority, access right, duration
- Operations, e.g., join or leave, perform, request or serve

• *Data* ('what') is for modeling the object of life-cycle activities of CNO. In *business model* it represents *information* about product, service, project, contract, document, resource, etc., that are input or output of one or more functions. In *system model* it represents *database schema* defining classes or types, attributes, operations, constraints, etc.

- Attributes, e.g., type, media, location, access path
- Operations, e.g., create, manage, use

• *Process* ('how') is for modeling a set of activities. In *business model* it represents function, procedure, transaction, or workflow to accomplish CNO goals. In *system model* it represents *process* that has a sequence of activities and control of flow.

- Attributes, e.g., start/finish date or time, owner, input/output, flow (i.e., sequential or parallel)
- Operations, e.g., start, finish, terminate, resume

• *Link* ('where') is for modeling connection between two participants or processes. In *business model* it represents communication/reporting channel, control, interface, collaboration path between two participants. In *system model* it represents 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., open/close, connect/disconnect

• *Event* ('when') is for modeling specific time of control for participants, data, processes, and link. In *business model* and *system model* it represents events.

- Attributes, e.g., time and precondition of invocation, invoking processes
- Operations, e.g., check, invoke

• *Goal* ('why') is for modeling motivation of participants. In *business model* it represents mission, purpose, strategy, means-ends of participants. In *system model* it represents rules with condition and action.

- Attributes, e.g., assigned participants, type (e.g., strategic, tactical, operational), duration (e.g., years, months, days), measure of effectiveness
- Operations, e.g., set/unset, redirect

4 CONCLUSION

On the basis of Zachman Framework and OMG's Model-Driven Architecture (MDA), we suggest a Modeling Language of Enterprise Architecture (EAML) for CNO called CAML. It's an extension of existing research of EA frameworks and EAMLs. As far as we know, inclusion of meta-level models based on MDA is unique in CN discipline. 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 (Jonkers et al., 2003), 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 that (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).

In this paper we only suggest modeling constructs of the CAML. To make the CAML sound and complete we plan to further investigate the following issues: (1) modeling of relationships between suggested modeling constructs in meta-models and domain models, (2) verification of modeling constructs through applying practical examples of CNOs, (3) formal definition of the modeling constructs, and (4) prototyping of the CAML as a modeling language.

5 REFERENCES

[1] Brodie ML, Bobrow D, Lesser V, Madnick S, and Tsichritzis D, Hewitt C. "Future AI requirements for intelligent database systems", In L. Kerschberg (ed), Expert Database Systems, Benjamin Cummings, pp.45-62.

[2] Camarinha-Matos LM and Afsarmanesh H. "The emerging discipline of collaborative networks", In L. M. Camarinha-Matos (ed), Virtual Enterprises and Collaborative Networks, Kluwer Academic Publishers, IFIP V.149, August 2004, pp. 3-16.

[3] Camarinha-Matos LM, Afsarmanesh H, Ferrada F, Klen A, and Ermilova E. Rough reference model for collaborative networks, D52.2, ECOLEAD, March 2006.

[4] Camarinha-Matos LM and Afsarmanesh H. “Towards a reference model for collaborative networked organizations”, In Proceedings of Information Technology for Balanced Manufacturing Systems (BASYS06), Niagara Falls, Canada, September 2006, pp. 193-201.

[5] Frankel DS, Harmon P, Mukerji J, Odell J, Owen M, Rivitt P, and Rosen M. The Zachman Framework and the OMG’s Model Driven Architecture, Business Process Trends, White Paper, September 2003.

[6] Hendryx S. Architecture of Business Modeling, OMG document br/2003-11-01, November 2003.

[7] Jonkers et al. “Towards a language for coherent enterprise architecture descriptions”, In Proceedings of the 7th IEEE International Enterprise Distributed Object Computing Conference (EDOC), 2003, pp. 28-39.

[8] Jonkers H, Lankhorst M, Buuren R, Hoppenbrouwers S, Bonsangue M, and Torre L. “Concepts for modeling enterprise architectures”, International Journal of Cooperative Information Systems, 2004, 13(3), pp.257-287.

[9] Kim DH and Park SJ. “FORM: A flexible data model for integrated CASE environments”, Data & Knowledge Engineering, 22, 1997, pp.133-158.

[10] Sarkar S and Thonse S. “EAML – Architecture modeling language for enterprise applications”, In Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East’04), 2004, pp.40-47.

[11] Steen MWA, Lankhorst MM, Wetering RG. “Modeling networked enterprise”, In Proceeding of the 6th International Enterprise Distributed Object Computing Conference (EDOC), 2002, pp.109-119.

[12] Tapia RS. “What is networked enterprise”, CTIT Technical Report TR-CTIT-06-23, Center for Telematics and Information Technology, University of Twente, The Netherlands, May 2006.

[13] Thangarathinam T, Wyant G, Gibson J, Simpson J. “Metadata management: the foundation for enterprise information integration”, *Intel Technology Journal*, 2004, 8(4), pp.337-344.