Proposed Integration Framework for Viewpoint-based Enterprise Architecture

Akira Tanaka[†], and Osamu Takahashi [‡]

[†]view5 LLC, Yokohama, Japan
[‡] Department of Media Architecture, Future University Hakodate, Japan
atanaka@view5.co.jp, osamu@fun.ac.jp

Abstract – Enterprise Architecture needs to cover various aspects of the target enterprise, starting from business strategy down to communication protocols. However, business environment changes daily, and new technologies are constantly introduced. There is a need to define technology integration framework for system architecture such as Enterprise Architecture. This paper presents a proposed mechanism for integrating technology elements, which did not exist when the architecture was introduced, such as mobile computing, cloud computing, and social network, into Viewpoint-based Enterprise Architecture. Specifically, the mechanism is targeting for RM-ODP viewpoint language based Enterprise Architecture, which includes modification approach to meta-model and UML Profile. Findings and discussion covers integration approaches, relationship with model driven software development, and openness or interoperability of Enterprise Architecture.

Keywords: Enterprise Architecture, RM-ODP, Mobile Device, Cloud Computing, Social Networks, UML

1 INTRODUCTION

1.1 Enterprise Architecture

Enterprise Architecture is a widely recognized term for designing enterprise's software and system architecture. Examples include Zachman Framework [1][2], TOGAF [3], and Federal Enterprise Architecture (FEA) [4]. In standards domain, RM-ODP [5][6][7] is an ISO/IEC/ITU-T standard for specifying Open Distributed Processing. It provides foundational concepts and standard viewpoints including languages and structuring rules for specifying enterprise systems. The standard viewpoints in RM-ODP are Enterprise, Information, Computational, Engineering, and Technology. Other frameworks use different classification: Perspectives in Zachman Framework, Architecture Domains in TOGAF, and sub-architecture domains in Federal Enterprise Architecture. Because of its neutrality, openness, availability of tools and documents, we chose RM-ODP to represent Enterprise Architecture for the research described in this paper.

One of the issues with Enterprise Architecture Framework is they are designed to be stable, which is good but at the same time that may imply hard to modify architectural elements or hard to introduce new architectural elements. Therefore, there is a need to study how Enterprise Architecture can be extended in consistent way. The proposed mechnism involves an approach to consistently modify Enterprise Architecture's or RM-ODP's meta-model and corresponding UML Profile.

1.2 Modeling Software Architecture

Widely used modeling language for specifying software systems is UML or Unified Modeling Language [12]. It provides a means for defining structural and behavioral aspects of the target system, using its modeling elements such as Class, Component, Activity, State Machine, and Use Case. Although it is a general purpose modeling language, with its profiling mechanism, users can define customized model elements for specific domains such as embedded systems, real-time systems, and Enterprise Architectures. Example such profiles covering Enterprise Architecture domain include UML Profile for DoDAF and MODAF [20] and Use of UML for ODP system specifications [8][9]. It usually starts with defining meta-model for the Enterprise Architecture concepts and relationships among them, followed by defining its UML profile for use with UML tools. This approach is also applicable to our case, which is to introduce meta-model elements for recently adopted technologies, define UML Profiles for it, and integrate them into the base Enterprise Architecture Framework.

2 RECENT TECHNOLOGIES' IMPACT ON ENTERPRISE ARCHITECTURE

Growing popularity of mobile devices such as smart phones and tablets, cloud computing, and social networks are examples of the changes we have witnessed recently. Although those are just examples, those technologies were not really considered in Enterprise Architecture ten years ago, but we need to integrate them today. We will first examine kinds of impact those technologies bring to Enterprise Architecture.

2.1 Mobile Devices

Mobile devices at early days were mainly portable laptop PCs and communication means were very limited.

Reason for success may include Moore's law, people's welcome of mobile devices (actually computer systems), and acceptance of living in cyber world (emails, social networks etc.).

Impact on Enterprise Architecture includes addition and processing of new data elements like moving object's location, position, direction and time-stamp. Additional work of adjusting UI for mobile devices and increased security concerns e.g. by stolen devices or less secure network is also part of the impact. Mobility may be applied to people and things, such as software elements and hardware elements, specified in Enterprise Architecture. The impact would be to all the viewpoints.

2.2 Cloud Computing

Application Service Provider or ASP might be conceptually close to today's SaaS (Software as a Service). PaaS (Platform as a Service) equivalent did not exist, and rental server services or hosting services of the time could be considered as services similar to IaaS (Infrastructure as a Service) with limited capability.

Reason for success may be that cloud computing usually provides better ROI of IT resources than internal investment in hardware, software, system administration, training, and system development.

Impact on Enterprise Architecture includes increased use of external application services (SaaS), and external platforms to run the applications (IaaS and PaaS). There are various types of clouds: public cloud, private or internal cloud, hybrid cloud, and personal cloud. NIST's definition of cloud computing [10] is widely accepted. However, the use of external resources requires strong governance over security and regulatory issues. The impact would be to enterprise viewpoint for external applications and to computational/engineering/technology viewpoints for integration with this technology.

2.3 Social Network

Except for forums provided by e.g. online service providers, there was nothing comparable with today's social networks in popularity and scale.

Reason for social networks' success may be that owners of enterprise systems have looked at success cases in consumer oriented social networks and realized the importance of people aspects, involving e.g. managers, developers and operators, who may not be communicating with each other within the enterprise system. It is, therefore, logical to consider including this capability into Enterprise Architecture based systems and expect "social effects."

Impact on Enterprise Architecture includes adding and processing new data elements (social profile and people oriented network information), which leads to a new class of applications that enables posting, reading, reacting to messages to construct his/her social networks and analyzing social network [11], in the context of enterprise systems. The impact would be to all the viewpoints.

Those three elements, mobility, cloud computing, and social network, may have the following use relationships or dependencies (Fig. 1).

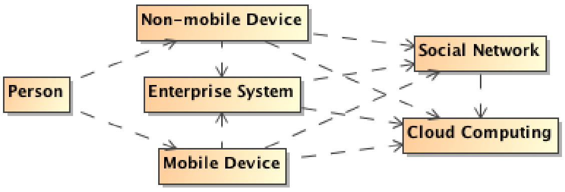


Figure 1: Relationship/Dependencies between mobile devices, cloud computing, and social networks

Mobility enhances access capability to a part of enterprise system running on cloud platform, cloud computing's scalability and reliability supports dynamic traffic changes of social networking, and mobile access to social network will accelerate the use of social network.

3 ANALYSIS AND PROPOSED MODIFICATIONS ON RM-ODP

We analyze these three technology areas and consider meta-models or UML Class diagrams representing MOF [13] model, consisting of their core concepts and its relationships among them. Those meta-models with RM-ODP's meta-models can be used to define integrated meta-models, which can be used for UML Profile development. Note that some concepts from three areas may already be defined in RM-ODP (or Enterprise Architecture), and we will use existing concepts with possible modifications when needed. Other elements are assumed to be independent and to be integrated into RM-ODP meta-model as domain-specific extensions.

3.1 Mobile Devices

In case of mobile devices, mobility elements were not really present in the most Enterprise Architectures. Some did have location concept, but it was not introduced to represent moving object's location. It was more for static location e.g. person's address or department's address such as "Stockholm, Sweden." In order to add mobility to an object, it needs to have a dedicated attribute for mobility or a link to mobility. Mobility attribute or mobility will consist of time-stamp and location. If necessary, velocity of moving object can be computed using this record ($v=\Delta location/\Delta t$).

Modifications to RM-ODP are the followings.

Enterprise/Computational/Engineering/Technology Languages: Optional link to Mobility added to Viewpoint Objects

Information Language: No change [since information models do not change depending on time and location]

As proposed changes to RM-ODP concepts, Mobility can be defined as a combination of LocationInTime and LocationInSpace (Fig. 2, fragment of the modified meta-model), both are defined terms in RM-ODP. Geographic Information standard could be used for LocationInSpaceType.

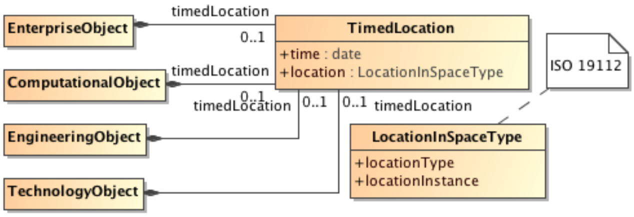


Figure 2: TimedLocation

3.2 Cloud Computing

In case of cloud computing, especially in business domain, the major concerns are actually on business execution, and underlying technologies such as computing platform are usually considered as engineering issues. However, use of cloud computing based external services can become an impact. When they are incorporated into business processes, they will work as action/activity or step implementations within certain business processes. Those external cloud applications usually provide services using web interfaces. Therefore, to incorporate external services, support of web interfaces or Service-Oriented Architecture (SOA) [14] will be needed in the architecture. In addition, integration with existing systems (legacy systems) could also be done in the same fashion. Interaction with external services, though, will require its policy to cover agreement for using external services and/or regulatory restrictions.

Modifications to RM-ODP are the followings.

Enterprise/Computational/Engineering/Technology Languages: Optional link to CloudService was added to Objects, and a new datatype "CloudServiceType" was introduced.

Information Language: No change [since information models do not change depending on where and how they are managed]

As proposed changes to RM-ODP concepts (Fig. 3, fragment of the modified meta-model), CloudNature can be defined as a combination of CloudSupport (Boolean) with CloudType ((Public, Private, Hybrid) & (SaaS, PaaS, IaaS)). Also in Enterprise Viewpoint, a Step may be labeled with CloudNature, showing that an Object with Cloud Support is supporting the Step.

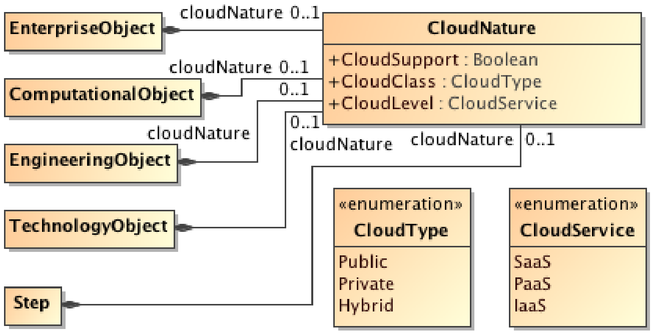


Figure 3: CloudNature

3.3 Social Network

When a person starts working for an enterprise, he/she will be given a title or role in an organization, which is connected to what the person is obligated to do, allowed to do, and prohibited to do, or job description. This model needs to be modified to incorporate sociality, which is described in the person's social profile (Fig. 4, fragment of the modified meta-model). A new data types such as person's interests, experience, and participating social communities with roles within need to be there. Person is usually modeled as Party, which can have a relationship with other Parties. RM-ODP's Community concept is a good fit to represent social community. The resulting architecture will include parties, services, processes, etc. within the enterprise just like normal business processes to make best use of people's capability.

Modifications to RM-ODP are the followings.

Enterprise Language: Optional link to SocialProfile with SocialRelationship were added to Party, and definition of SocialProfile was added.

Information/Computational/Engineering/Technology Language: No change

We use Party as defined in RM-ODP and introduced SocialRelationship, Social Profile, and Social Community, which is a subclass of Community. We can also use suitable Viewpoint Language elements.

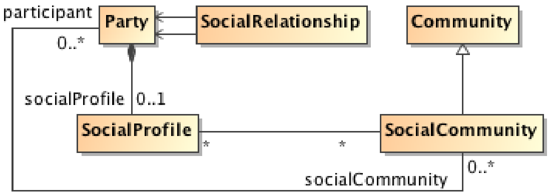


Figure 4: SocialProfile

4 UML PROFILES

The changes to conceptual model need to be reflected in the UML Profile. There is an ISO/IEC/ITU-T standard called "Use of UML for ODP system specifications" or UML4ODP for short, with which ODP models can be created with UML tools.

There are two ways for modifying UML Profile. First is to modify the existing stereotypes, and the second is to define new stereotypes such as MobileObject and CloudObject by customizing existing stereotypes. The latter will lead to a creation of many subclass stereotypes such as NV_MobileBinder and NV_CloudBinder. In order to avoid the confusing too-many stereotypes, we chose modifying existing stereotypes approach.

The following shows the mapping of modified meta-model to UML Profile.

4.1 Mobile Device and Cloud Extension as UML Profile

Each viewpoint object's stereotype (EV_Object, CV_Object, NV_Object, and TV_Object) is enhanced to include properties covering mobility and cloud-ness (Fig. 5). A property "mobility" is a Boolean with default value false, meaning if it is true the object is mobile object. Only in that case, time and location properties are set. In the same manner, a property “cloud-ness” is a Boolean with default value false, meaning if it is true the object is cloud-supported object. In this case, cloud type and cloud service type properties are set.

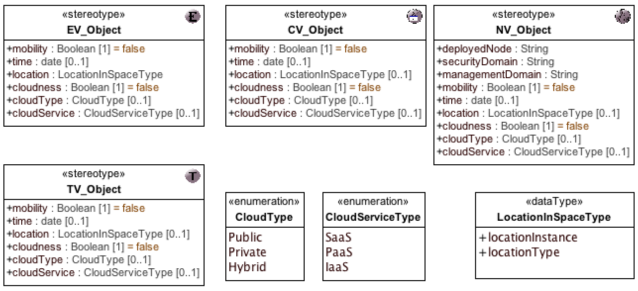

«stereotype»
EV_Object
+mobility : Boolean [1] = false
+time : date [0..1]
+location : LocationInSpaceType
+cloudness : Boolean [1] = false
+cloudType : CloudType [0..1]
+cloudService : CloudServiceType [0..1]
«stereotype»
CV_Object
+mobility : Boolean [1] = false
+time : date [0..1]
+location : LocationInSpaceType [0..1]
+cloudness : Boolean [1] = false
+cloudType : CloudType [0..1]
+cloudService : CloudServiceType [0..1]
«stereotype»
NV_Object
+deployedNode : String
+securityDomain : String
+managementDomain : String
+mobility : Boolean [1] = false
+time : date [0..1]
+location : LocationInSpaceType [0..1]
+cloudness : Boolean [1] = false
+cloudType : CloudType [0..1]
+cloudService : CloudServiceType [0..1]
«stereotype»
TV_Object
+mobility : Boolean [1] = false
+time : date [0..1]
+location : LocationInSpaceType [0..1]
+cloudness : Boolean [1] = false
+cloudType : CloudType [0..1]
+cloudService : CloudServiceType [0..1]
«enumeration»
CloudType
Public
Private
Hybrid
«enumeration»
CloudServiceType
SaaS
PaaS
IaaS
«dataType»
LocationInSpaceType
+locationInstance
+locationType

Figure 5: Stereotypes for Mobility and Cloud

4.2 Social Network Extension as UML Profile

EV_Party was enhanced to have sociality, social information, and social communities properties. EV_Community was enhanced to have sociality and participants properties A new stereotype SocialRelationship, which extends UML Association, was also introduced (Fig. 6). The details of SocialInformationType are not defined here.

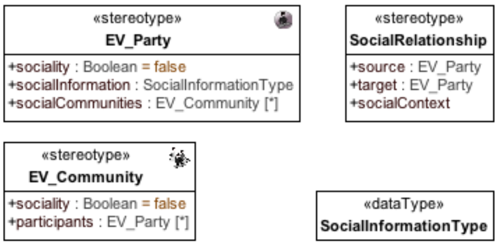

«stereotype»
EV_Party
+sociality : Boolean = false
+socialInformation : SocialInformationType
+socialCommunities : EV_Community [*]
«stereotype»
SocialRelationship
+source : EV_Party
+target : EV_Party
+socialContext
«stereotype»
EV_Community
+sociality : Boolean = false
+participants : EV_Party [*]
«dataType»
SocialInformationType

Figure 6: Stereotypes for Social Network

5 IMPACT ON ENTERPRISE ARCHITECTURE ELEMENTS

We have applied those stereotypes together to see what kinds of impact Enterprise Architectures receive.

5.1 Mobile Device

We observe the following impact.

1) Enterprise Viewpoint: The most part of this viewpoint model does not get impact of mobile object, since this viewpoint model mainly talks about why and what. However, a Role, an abstraction of behavior performed by Object, may get influence by mobile object. Especially, policy value will need update to include the cases where some Roles are performed by mobile object. For instance, if a mobile object performs a part of the process or interaction, e.g. new security policy X may apply.

2) Information Viewpoint: No impact

3) Computational Viewpoint: Although this viewpoint does not care about distribution, mobility is functionality, and a mobile object can still be introduced. That will give surrounding objects some impact. For instance, an object providing geographical map based on a mobile object’s TimedLocation information may be introduced to support mobile objects.

4) Engineering Viewpoint: A case of a mobile object moving from Node A to Node B becomes a possibility. Also, a mobile object may need multiple channels for communication, since available channel may be different from place to place and from time to time.

5) Technology Viewpoint: Technology objects representing software, hardware, and network will be categorized into mobile and non-mobile object.

5.2 Cloud Computing

We observe the following impact.

1) Enterprise Viewpoint: The same observations as above (5.1 1)) apply. Policy value will need update to include the cases where some Roles are performed by cloud object. For instance, if a part of the process (or a step) or interaction is performed by a cloud object, or an artifact used is fulfilled by a cloud object, e.g. new security policy Y may apply.

2) Information Viewpoint: No impact

3) Computational Viewpoint: Computational Viewpoint model specifies distributed transparency attributes as a whole (see UML4ODP). Depending on the cloud provider or service, a part of distribution transparency may be provided by the cloud, which means with cloud object computational model may become composite of ODP specified distribution transparency part and cloud provided distribution transparency part.

4) Engineering Viewpoint: In case of SaaS, Engineering Viewpoint model including cloud objects and channels to communicate with them is all we need to define. Other elements such as Node for SaaS may be created as virtual element. In case of PaaS, a cloud object providing specific application functionality and the platform are the main elements to be modeled. Other elements such as Node for PaaS may be created as virtual element as well. In case of IaaS, it is possible to model most of the engineering viewpoint except for Nucleus etc.

5) Technology Viewpoint: Technology objects representing software, hardware, and network will be categorized into cloud and non-cloud object.

5.3 Social Network

We observe the following impact.

1) Enterprise Viewpoint: SocialParty, SocialRelationship and SocialCommunity are additions to the viewpoint model, and those need to be defined. The behaviors defined in the model will need updates to reflect the new elements. A process to construct social profile, setup/execute social activity, and to achieve some social objective with the help of social relationships may be added.

2) Other Viewpoints: No impact except for normal viewpoint modeling of supporting viewpoint objects for communicating with social networks.

6 APPLYING PROFILES TO MAJOR ELEMENTS OF ENTERPRISE ARCHITECTURE

We have applied all the UML Profile elements described before into existing UML4ODP Profile definition. With this revised UML4ODP, we can create new kinds of models or diagrams as a step towards Flexible Enterprise Architecture.

6.1 Enterprise Viewpoint Model

There are various types of model or diagram in Enterprise Viewpoint when UML4ODP is used. The following covers only major diagrams.

Objective diagram: A diagram showing Objective decomposition

CommunityContract diagram: A package diagram showing Community and Objective, a package of EnterpriseObjectTypes, a package of Roles, a package of Policies, and a set of Processes.

EnterpriseSpec diagram: A package diagram showing included CommunityContract packages and associated FieldOfApplication.

EnterpriseObjectTypes diagram: A package showing included EnterpriseObjectTypes including ODPSystem, and the relationships among them

RolesInCommunity diagram: A Community and a list of Roles involved

RolesObjects diagram: A diagram showing a set of Roles, a set of Objects, and FulfilsRole relationships between them

Interaction diagram: One or more Interactions with associated Roles, referenced Artifacts, and Enterprise Objects fulfilling the artifact roles

Process diagram: An activity diagram showing Roles and their bahaviors (Steps, Artefact, etc.)

Policy diagram: A set of Policy Envelope, Policy Value, relevant controlling Process, and affected behaviors such as Interactions

A package of Enterprise Object Types can include Enterprise Object Types with new properties (Fig. 7 and 8).

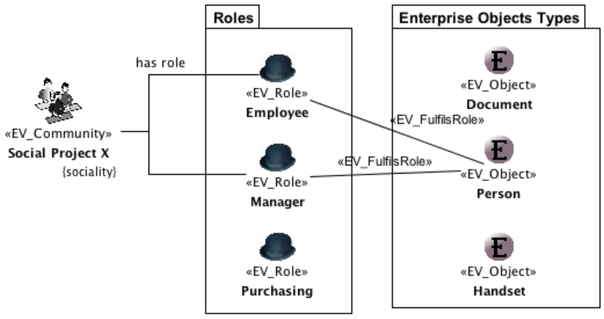
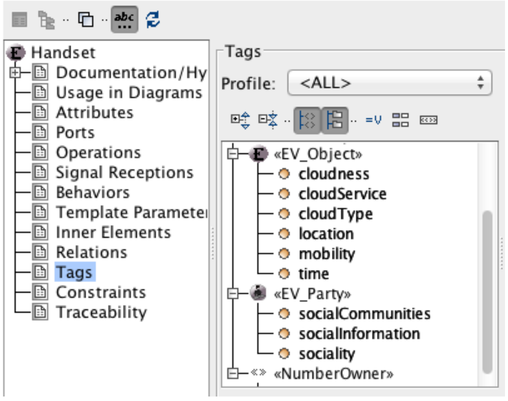


Figure 7: Example Relationships among Enterprise Object Types, Roles, and their participating Social Project



Profile: <ALL>

- «EV_Object»
  - cloudness
  - cloudService
  - cloudType
  - location
  - mobility
  - time
- «EV_Party»
  - socialCommunities
  - socialInformation
  - sociality
- «NumberOwner»

Figure 8: Enterprise Objects with properties and Roles

Enterprise Objects with new properties (see Fig. 8 for Handset) can also appear in Interaction Model (Fig. 9).

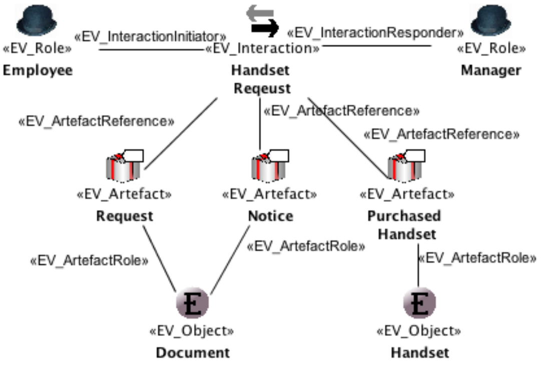


Figure 9: Sample Interaction Diagram

Defined Enterprise Objects can also appear in process diagram (Fig. 10).

Although those are RM-ODP specific diagrams, similar models can be found in other Enterprise Architecture Frameworks.

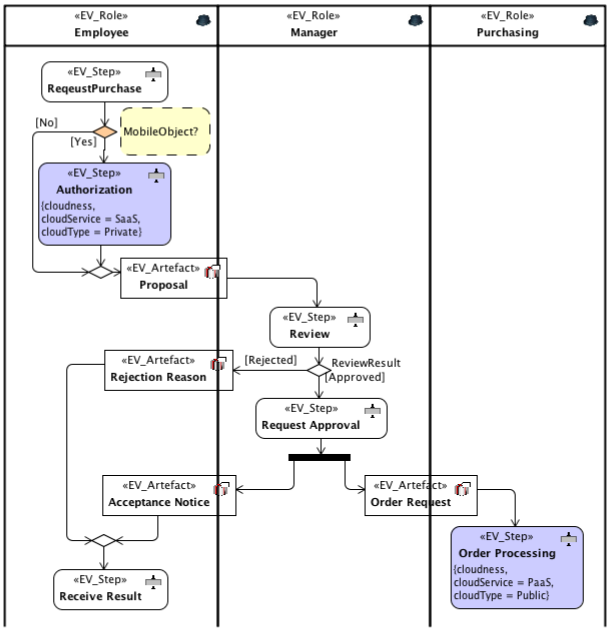


Figure 10: MobileObject in Sample Business Process

6.2 Information Viewpoint Model

We have defined no additional stereotypes for this viewpoint. We can, however, still define additional data types or Information Objects in Invariant Schema diagram using standard UML4ODP (Fig. 11).

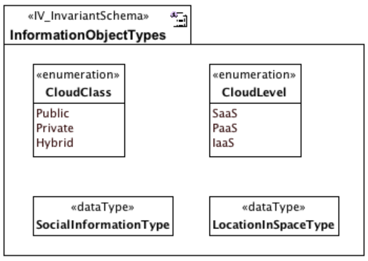


Figure 11: New Invariant Schema Elements

6.3 Computational Viewpoint Model

There are various types of model or diagram in Computational Viewpoint when UML4ODP is used. The followings are two of the examples.

Architecture diagram: A diagram showing logical grouping of architectural packages such as application objects package containing business functions package and ODP function package. Components definition can be a part of this diagram.

Interface/Signature diagram: A diagram showing a set of interface definition and signature definitions, with datatype definitions used.

A Computational Object can have TimedLocation and/or CloudNature as properties and be used in the architecture diagram (Fig. 12).

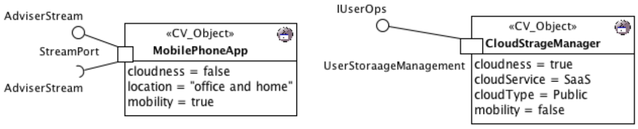


Figure 12: Sample Mobile and Cloud Components

6.4 Engineering Viewpoint Model

Types of diagrams in Engineering viewpoint are similar to those of Computational viewpoint, and the differences are in their distribution-awareness, or their mission to support distributed transparencies, and in their internal structure of Nodes and Channels. When we use e.g. SaaS, it would be the objects on remote node to access and use, and in general there is no need, or no way to describe internals of the target SaaS system. However for the purpose of this modeling, we used the same stereotypes with properties (Fig. 13) for describing SaaS object.

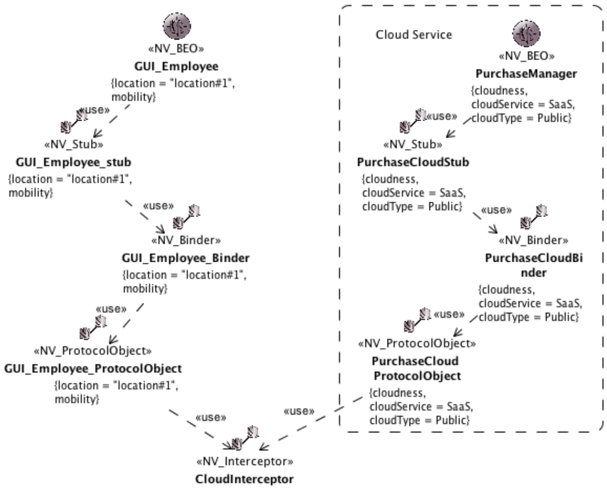

«NV_BEO»
GUI_Employee
{location = "location#1", mobility}
«use»
«NV_Stub»
GUI_Employee_stub
{location = "location#1", mobility}
«use»
«NV_Binder»
GUI_Employee_Binder
{location = "location#1", mobility}
«use»
«NV_ProtocolObject»
GUI_Employee_ProtocolObject
{location = "location#1", mobility}
«use»
«use»
«NV_Interceptor»
CloudInterceptor
Cloud Service
«NV_BEO»
PurchaseManager
{cloudness, cloudService = SaaS, cloudType = Public}
«use»
«NV_Stub»
PurchaseCloudStub
{cloudness, cloudService = SaaS, cloudType = Public}
«use»
«NV_Binder»
PurchaseCloudBinder
{cloudness, cloudService = SaaS, cloudType = Public}
«use»
«NV_ProtocolObject»
PurchaseCloud ProtocolObject
{cloudness, cloudService = SaaS, cloudType = Public}

Figure 13: Sample Channel from Mobile Device to Cloud Services

6.5 Technology Viewpoint Model

Technology Object with mobility or cloud-ness can be used to specify elements of hardware, software, or network. However, in case of cloud computing, and if it is a private cloud, there is not much difference with ordinary in house servers case (Fig. 14). However, if it is a public cloud, there is a limit for defining technological architectures, since internal of cloud services is not visible, and cannot be specified, from outside.

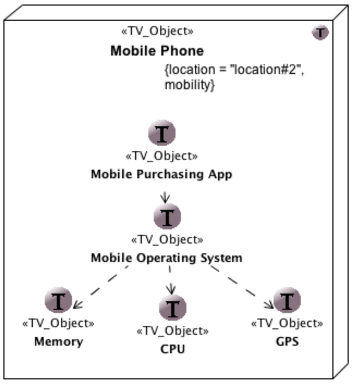


Figure 14: Sample Technology Object for Mobile Device

6.6 Viewpoint Correspondence Model

RM-ODP provides concept of Correspondence, with which we can specify a model element in one viewpoint is related to another model element or model elements in other viewpoint. This is implemented in UML4ODP, and without modifications, we can use this capability in our example as well (Fig. 15).

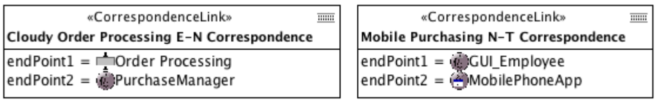

«CorrespondenceLink»
Cloudy Order Processing E-N Correspondence
endPoint1 = Order Processing
endPoint2 = PurchaseManager
«CorrespondenceLink»
Mobile Purchasing N-T Correspondence
endPoint1 = GUI_Employee
endPoint2 = MobilePhoneApp

Figure 15: Sample Correspondences

7 FINDINGS IN BRINGING FLEXIBILITY INTO ENTERPRISE ARCHITECTURE

7.1 Summary of Proposed Extension Mechanism

The mechanism we have employed can be summarized as follows. First, we need to have meta-models for Enterprise Architecture and new technologies in the context of Enterprise Architecture. Note it is necessary to understand the domain enough to define meta-models. Second, the meta-model for Enterprise Architecture, as a receiving package, package-merges with one or more of meta-models, as merged packages, for new technologies. When there is a conflict, a process to resolve the conflict described below should be followed. Third, based on those meta-models, we can define UML Profile definitions. UML Profile definitions can also use package merge by treating UML Profile for Enterprise Architecture as a receiving package and UML Profile for new technologies as merged packages.

7.2 Conflict resolutions process

When two concepts from two meta-models are considered similar, there may be several types of relationships between the two. The following diagram (Fig. 16) shows several possibilities.

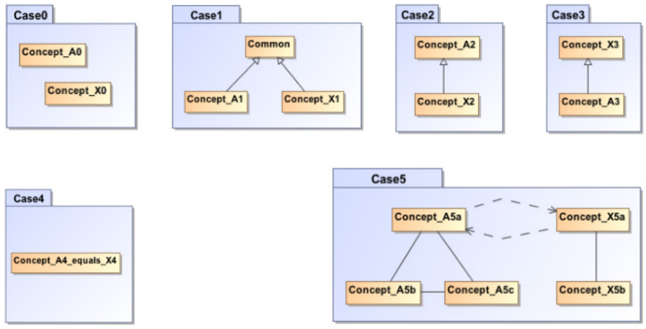


Figure 16: Possible relationships between two concepts

They could be different and independent (Case0), may share the common super concept (Case1), one of them may be the super class of the other (Case2 and Case3), they may be the same concept (Case4), or each of them has its structure and one of each element may be the same concept (Case5).

The proposed process is the following.

Step 1: Compare the definitions to understand the meaning of the concept in each context, and highlight the difference.

Step 2: Compare the surrounding concepts of the two to find differences between the two concepts.

Step 3: Compare the constraints (if existed) to highlight the similarity and differences.

Step 4: Identify the type of relationships between the two.

Step 5: Decompose two concepts into finer grained concepts and relationships, when they are composite concept, find similar concepts between the two sets, and apply Step1 to Step4.

Step 6: Apply direct integration or UML package merge to integrate two meta-models.

Once meta-model is integrated, we can proceed to define UML Profile for it.

7.3 Relatively Independent Cases

If target new technologies were relatively independent from existing Enterprise Architecture elements, there would be at least two enhancement strategies for integration. The first strategy is to add new technology elements at meta-model level (package merge of meta-models), and enhance existing elements at UML Profile level (package merge of UML Profile). The second strategy is to add new technology elements at meta-model level (package merge of meta-models), and at UML Profile level (package import of UML Profile). With the first strategy, users will need to understand the updates for each stereotypes, but will be able to use the same set of stereotypes. This can be used in the case that quick implementation is required. With the second strategy, users will need to learn new stereotypes. This can be used to get better modularity of UML Profile, but too many stereotypes may become an issue.

7.4 Conflicting Cases

If the target new technology meta-model has some conflict with existing Enterprise Architecture elements, the conflicting portions need to be resolved by following the resolution process explained in 7.2. One such example is SOA concepts against RM-ODP concepts. Key concepts are quite similar but they use different terminologies with the different scope. Additional resolution strategy for the conflict is 1) to choose primary one over the other and gradually migrate the second choice to the first one, or 2) to not change anything but treat these as independent and just add correspondence in formal way (e.g. using OCL) or even in informal way. If, for instance, we need to bring SoaML [15] into UML4ODP, the conflict may be around Interaction (RM-ODP) and Collaboration (SoaML). In this case, choose the one that is more suitable for defining behavior in integrated SOA/RM-ODP world. Another example is BMM (Business Motivation Model) [16] against RM-ODP, and the conflict is between Objective (RM-ODP) and Vision/Goal/Objective (BMM). In this case it would be less difficult to resolve, since BMM provides wider and richer goal/motivation model. It depends on how deep objective model you need.

8 DISCUSSIONS

8.1 Flexible Extension Mechanism

The question we started with was “is it possible to design flexible or extendable Enterprise Architecture, preparing for the time new technology emerges?” We developed an extension mechanism described in 7.1 “Summary of Proposed Extension Mechanism” and in 7.2 “Conflict resolution process." In the final step of merging UML Profiles, we need to add/modify stereotypes, properties, and constraints of UML4ODP. When doing this, use of modeling tool is helpful, especially when you already have meta-model data to be revised and profile data to be revised. Some modeling tools have validation capability. Even without tooling, it is still possible to take the same steps. However, if you did it by hand, it would be hard to avoid errors and inconsistencies.

8.2 Enterprise Architecture and Model based Software Development

Enterprise Architecture usually means high-level description of the entire enterprise system. That may not change, but if Enterprise Architecture is presented as e.g. a UML model, we can at least make consistent modifications to the model with the help of UML tools. When the idea of Model Driven Architecture (MDA) [17] was introduced, there was no MDA tooling available. Today, there are some commercial and open-source products for model transformations, such as eclipse modeling project. Once source model is prepared, it can be used as an input to model transformation tool chains. Enterprise Architecture in UML is a reasonable starting point for this process. Also, if it is UML model, whole or a part of the model may become candidate for reuse. The real challenge would be to achieve step-by-step model-to-model transformation chains. There are several UML tools that support RM-ODP, Zachman Framework, and TOGAF (e.g. MagicDraw). With those tools and with enhancement support mechanism in place, the tool chain can consume UML models, representing enhanced Enterprise Architecture. In this case, development of model transformation logic is needed and once it is done it would become a candidate for reuse, since it is built against standard UML models. Another possibility is use of Domain-Specific Language or DSL [18]. Some people prefer DSL to UML for its simplicity. If a set of DSLs are designed to describe Enterprise Architecture, then created models (XMI form) can be used as an input for the tool chains. In this case, however, development of model transformation logic is required for each DSL, and because they are not standard based, reuse may become an issue.

8.3 Interoperability among Enterprise Architecture Frameworks

As of today, there is no interoperability among different Enterprise Architecture Frameworks, which produced silos of Enterprise Architectures. The required actions are to make their meta-model open and encourage development of transformations between them, or to make one of the meta-models as standard and each framework provider to develop and provide transformation to/from the standard. The issue in doing this is the difference in scope or coverage of the concepts for Enterprise Architecture. The first thing to do, therefore, is to agree on common core set of concepts for Enterprise Architecture framework. At least among the three frameworks mentioned in this paper, RM-ODP based one is the most neutral and open, and this could be used as a base for the discussion on the common core.

9 RELATED WORK

There are a number of research papers and books on Enterprise Architecture and the modeling of it. After Zachman's first paper [2] in 1987, "Generalised Enterprise Reference Architecture and Methodology" [21] was developed by IFAC/IFIP Task Force on Architectures for Enterprise Integration in 1999, "Recommended Practice for Architecture Description of Software-Intensive Systems" or IEEE 1471-2000 [22] was standardized by IEEE in 2000, "A Method to Define an Enterprise Architecture using the Zachman Framework" [23] in 2004 proposed a practical method for utilizing Zachman Framework, and "A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform" [24] book discussed about meta-model for Enterprise Architecture in 2005. This stream of researches continues to reflect recent topics at e.g. EDOC Conference. In the area of UML modeling tool for Enterprise Architecture, there are activities within OMG to develop/enhance UML Profile for Enterprise Architectures such as UPDM. "A Tool for the Model-Based Specification of Open Distributed Systems" [25] in 2013 describes UML tool for ODP modeling, based on ISO's UML4ODP standard.

10 CONCLUSION

In order to integrate new technologies into Enterprise Architecture, we will need to take the following steps: analyze the domain that new technology is applied and follow the steps described in 7.1 "Summary of Proposed Extension Mechanism" and 7.2 "Conflict resolutions process." The use of "package merge" allows merging only necessary set of packages into Enterprise Architecture package. The steps handling UML Profile may be replaced with other steps, e.g. generating DSLs using eclipse-modeling project, for non-UML model case.

It is likely that new technology provides new capability to things or persons. In this case, new meta-model element related to the capability should be related to Object or Party (Person) so that the capability is explicitly visible to UML Profile designer.

If the Enterprise Architecture model is used in model driven environment as an input file, it should be interpretable by model transformation engines. This means input file should better be in the form of UML or XMI [19].

Regarding openness and interoperability of Enterprise Architectures, we will need a chance to discuss and find common core concepts. Until this is done, RM-ODP based Enterprise Architecture would be the most open one, since it is an ISO/IEC/ITU-T standard, and standard document, meta-model data, and UML Profile data are available on the Internet.

ACKNOWLEDGEMENTS

The authors thank to contributors in Enterprise Architecture domain, Modeling Language domain including

MOF, UML, UML Profile for DoDAF and MODAF and a variety of essential modeling standards by Object Management Group, and various Eclipse Modeling Projects by Eclipse Foundation including EMF, GMF, and Xtext. The authors also thank many people from various countries involved in ISO/IEC JTC1/SC21/WG7, SC7/WG19, and ITU-T SG19's continuing efforts to standardize and maintain RM-ODP family of standards including Use of UML for ODP system specifications. The authors finally thank many participants to IWIN 2103 who provided us with valuable feedbacks.

REFERENCES

[1] J.A. Zachman,``John Zachman's Concise Definition of The Zachman Framework™,'' http://www.zachman.com/about-the-zachman-framework (2013).

[2] J.A. Zachman, ``A Framework for Information Systems Architecture,'' IBM Systems Journal, Vol.26 No.3 (1987).

[3] The Open Group, ``TOGAF Version 9.1,'' http://pubs.opengroup.org/architecture/togaf9-doc/arch/.

[4] FEA Reference Models, ``Consolidated Reference Model Version 2.3,'' http://www.whitehouse.gov/sites/default/files/omb/assets/fea_docs/FEA_CRM_v23_Final_Oct_2007_Revised.pdf.

[5] ISO/IEC 10746-2:2009, Information technology -- Open distributed processing -- Reference model: Foundations.

[6] ISO/IEC 10746-3:2009, Information technology -- Open distributed processing -- Reference model: Architecture.

[7] ISO/IEC 15414:2006, Information technology -- Open distributed processing -- Reference model -- Enterprise language.

[8] ISO/IEC 19793:2008, Information technology -- Open Distributed Processing -- Use of UML for ODP system specifications.

[9] D. Hashimoto, A. Tanaka, and M. Yokoyama, ``Case study on RM-ODP and Enterprise Architecture,'' Proc. of 11th International IEEE EDOC Conference Workshop (EDOCW'07), pp.216-223 (2007).

[10] National Institute of Standards and Technology, ``The NIST Definition of Cloud Computing,'' Special Publication 800-145 (2011).

[11] D. Dreyfus, and B. Iyer, ``Enterprise Architecture: A Social Network Perspective,'' Proc. of 39th Hawaii International Conference on System Sciences, p.178a (2006).

[12] OMG, Unified Modeling Language™ (UML®), http://www.omg.org/spec/UML/

[13] OMG, Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, http://www.omg.org/spec/QVT/1.1/PDF/ (2011).

[14] OASIS, Reference Model for Service Oriented Architecture 1.0, http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf.

[15] OMG, Service oriented architecture Modeling Language (SoaML®).

[16] OMG, Business Motivation Model (BMM), http://www.omg.org/spec/BMM/.

[17] OMG, Model Driven Architecture ® (MDA ®), http://www.omg.org/mda/.

[18] M. Fowler, ``Domain-Specific Languages,'' Addison-Wesley (2011).

[19] OMG, XML Metadata Interchange (XMI®), http://www.omg.org/spec/XMI/.

[20] OMG, Unified Profile for DoDAF and MODAF (UPDM), http://www.omg.org/spec/UPDM/.

[21] IFAC/IFIP Task Force on Architectures for Enterprise Integration, ``Generalised Enterprise Reference Architecture and Methodology'' (1999).

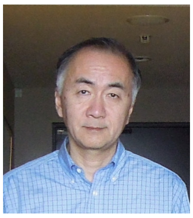
[22] IEEE 1471-2000, ``Recommended Practice for Architecture Description of Software-Intensive Systems''.

[23] C.M. Pereira, and P. Sousa, ``A method to define an Enterprise Architecture using the Zachman Framework,'' Proc. of ACM symposium on Applied computing (SAC '04), pp. 1366-1371 (2004).

[24] C. Braun, and R. Winter, ``A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodelling Platform,'' Desel & Frank (2005).

[25] J.R. Romero, J.I. Jaén, and A. Vallecillo, ``A Tool for the Model-Based Specification of Open Distributed Systems,'' The Computer Journal, doi: 10.1093/comjnl/bxs021 (2012).

(Received January 7, 2014)



Akira Tanaka received M.E. from University of Tokyo, Japan in 1978. From 1978 to 2008, he was with Hitachi Ltd. His current research area includes model-based software development. He is a member of ACM, IEEE Computer Society and IPSJ.

Osamu Takahashi received master's degree from Hokkaido University in 1975. He is currently a professor at the Department of System Information Science at Future University Hakodate. His research interest includes ad-hoc network, network security, and mobile computing. He is a member of IEEE, IEICE, and IPSJ.