Supporting Self-Development in Service Oriented Information Systems

MART ROOST, KARIN RAVA, TARMO VESKIOJA
Institute of Informatics
Tallinn University of Technology
Raja 15, 12618 Tallinn
ESTONIA

Abstract: - Information Society gives rise to dynamic networked organizations that form on the level of information system. This concept – an evolutionary information system – contains both the subject as well as the environment of development. We call the development process of such (information) system as IS self-development. An evolutionary information system must contain a development subsystem as its core that provides services for supporting the IS self-development. In this paper, the core of a model driven architecture for the development subsystem is proposed using UML based conceptual meta-model (class diagrams).

Key-Words: - Evolutionary Information System, Self-Development, Service Oriented Architecture, Business Architecture, Model Driven Development, Methodology, Eclipse Modeling Framework

1 Introduction

The context of the paper is Enterprise Information Systems and its Development. The Enterprise and Information Systems (IS) development situation has been totally changed in the recent years. New economic and business models (new digital or knowledge based economy, e- and m-businesses, etc.) have been discussed and applied. The digital economy presumes new organizational forms (like dynamic, virtual, learning/intelligent, networked, emerging, mobile, etc. organizations) [1]. These organizational forms are based on new technical and technological solutions (like Web Services and Service Oriented Architectures [2], Model Driven Development and Model Driven Architectures [3], Agent-Oriented Development/Architectures [6], etc.). We focus on Information Systems Development (ISD) in the context of dynamic *Networked Organisations and Enterprises*. Here, the term „Networked Organization" is a synonym for a „contemporary and/or future organisation" (that is, in more general, a subject of information society) and therefore presumes its abilities of *virtual operation* and *organizational learning* (a networked organization is normally a learning and virtual organization).

Virtual Organization (VO) is an organization of (virtual) subjects (actors, agents) that *forms on the level of IS*. VO forms on the basis of uniting (and/or separating) resources of independent work units called subjects or actors, where such a subject dynamically creates needed roles into multiple environments of different VO-s over the world, continuing at the same time its independent existence. VO forms, functions and develops as a result of *system work* that cooperative subjects (actors, agents) perform on the level of IS**.** The success of such organization depends on its ability to adapt to the environment and learn.

The business environment is changing quickly. A learning (intelligent) organization is an organization that is able to adapt to the environment by changing its business model. For such organization the key to success is a continuous process of its business model innovation. The organization and the IS must evolve dynamically (and partially automatically) with business according to changes in the business model and in the business environment. ISD ceases to be a „one-time effort" (or a project) with fixed final results; it becomes a continuous (business) process in the enterprise, that is an inevitable part of its organizational learning. [4]

In the process of organizational learning, firms are restructuring many relationships internally and externally to respond to the demands of a shifting market [5]. Internally: companies are disaggregating into smaller units/subject focused on well-defined market opportunities. Externally: they increasingly partner with other organizations in the context of their extended enterprises. The subjects of an enterprise obtain roles also in other (extended) enterprises. These roles are managed/developed on the level of the Enterprise IS (EIS). The extended enterprise is a NO that formes on the level of EIS. The continuos ISD can/should be modeled/managed in the context of extended enterprises, where the ISD service providers are included into this context.

IS is the main *organizational interface*, and immediate *functioning and learning environment* for NO and its subjects. Development of this environment *(ISD)* has to be a central/strategic role of the NO. To perform this role, the organization needs a *methodology* that enables subjects to develop the IS in their natural work environment, which is the IS under development. This concept, the *self-advancing (or evolutionary) IS* contains the subject as well as the environment of development. The development process of such an IS we call *IS self-development.* [8,9,10].

2 Problem Formulation

In this section the problem description is given using the following three subsections:

- Initial 'problem statement';
- Overview of our ISD approach (*IS Self-Development)*
- Overview of the architecture of the ISD Space that supports the self-development.

2.1 Problem Statement

As we saw in the introduction (section I), a new development situation has emerged, where traditional ISD models and paradigmas do not work. How do we develop IS for NO-s that evolve dynamically and automatically with business?

- What are the main requirements to IS and ISD in the context of NO?
- What is the ISD approach/model that is applicable in this context?
- What is IS self-development?
- What is the meta-model behind this concept?
- How can it work in practice?

2.2 Information Systems Self-Development. What is it?

IS self-development is an approach to ISD that is applicable in the context of a NO (and, in more general, a subject of information society). Within the (context of) traditional Customer-Developer relationship, this approach is Customer (or Subject) centric. ISD is handled here in the framework of *organizational roles* in the *customer organization* (and its extended enterprise), which form the basic (self-advancing) structure (architecture) for the *space of ISD*. In such framework, the whole system (NO IS) is not directly developed, but each autonomous part (subject) of the organization develops itself and its IS in mutual cooperation and in a common *space of development*. In the context of NO, this space is formed in the IS under development. This concept, the *self-advancing IS*, contains the subject as well as the environment of development.

Such IS must provide and mediate (in addition to the information and communication services) also the *development services* of the organization and its IS. Consequently, each IS must contain a *development subsystem* as its core, that provides (*pattern-based) high-level services for supporting IS self-development* by the subjects of the NO. In the role of the service provider potentially the whole global *community of ISD practice* is seen. The development subsystem should operate as two-directional (organizational) interface between the subjects of the NO (local ISD communities) and the global ISD community.

2.3 A General Role-Based Architecture of the ISD Space

To accomplish this concept (of *IS self-development)*, we need an adequate space of development, which has an architecture that is based on the *role* concept in a NO. Such architecture is outlined in Figure 1.

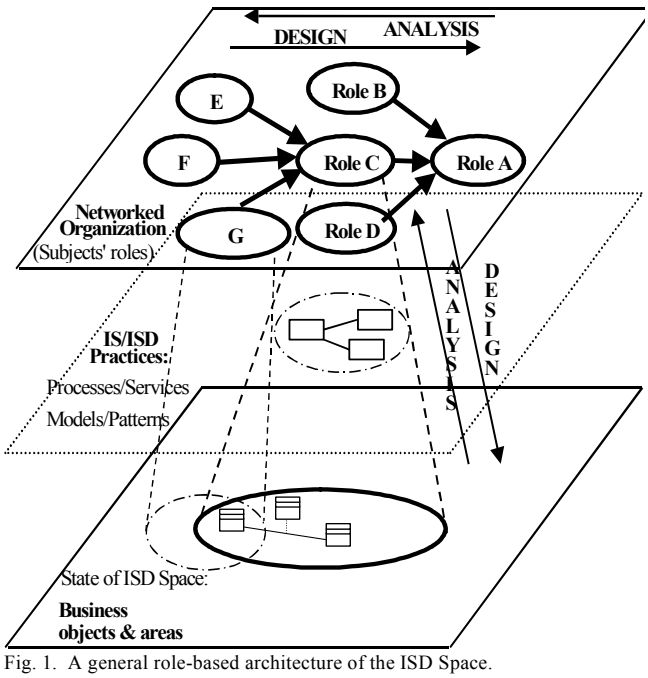

Fig. 1. A general role-based architecture of the ISD Space.

In this picture (Fig. 1) we see an *extended enterprise* that is interpreted here as a Space of ISD for this enterprise. We see 3 different levels. On the upper level there are Subjects of the Enterprise who may belong to the Internal or External context of the Enterprise. The Subjects are *closed* systems in the context of *this* Enterprise. They have one or more Roles in the context of this Enterprise, but they may have also roles in many other (extended) Enterprises. This means that for this Enterprise the Subjects are known/modeled only by their Roles in the context of this Enterprise. But each Subject has its own (local) IS, that knows / models all the Roles related to this Subject.

On the lower level we see the State of the ISD Space, that is handled as an Object of ISD. This global state is structured by Central Business Objects of the Enterprise that form areas of responsibilities for the Roles (the circles in Figure 1).

On the intermediate level we see IS/ISD practices (processes, services, models and patterns), that enable to change the State of the ISD Space. These practices are also (Business Objects and) Objects of system work and (IS) development, that belong to the State of the ISD Space, but only in the context of Subjects that serve in the role of ISD service providers.

Each Role is a „mini IS“, that can be modeled by the sentence “Subject develops/manages its IS”. It is possible (for a Subject) by applying (pattern-based) high-level ISD services, provided by the (members of the) community of ISD practice. The service providers are included into the (context of the) extended enterprise, that serves as a Space of ISD. From the viewpoint of each Subject, ISD occurs here through Synchronization of the business model of the Subject with business models of its Environment (that means other Subjects).

To implement this decentralized ISD model, the generic Analysis and Design processes have to be organized in this virtual Space of ISD. The Analysis is defined here as a (generic) process of coordination of Subjects’ Requirements, and the Design as a (generic) process of organizing (of) (the) Subjects’ Capabilities in order to satisfy the coordinated Requirements. (A supporting software is needed to implement this model.)

We also see a Supply Chain in this picture: the Customer side is on the right hand and the Supplier side on the left hand. (Potentially each Subject may be seen as a Customer and/or as a Supplier (=Service Provider). A Customer (view) is characterized by (its) Requirements and a Supplier is characterized by its Capabilities.).

The problem is how to translate this informally presented vision into a more formalized language that makes it possible to produce both software and process frameworks for supporting the (decentalized and evolutionary) IS/ISD model (described here and in [8,9,10]).

3 Problem Solution

In this section we describe the architecture of the ISD Space, which was introduced in the sub-section 2.3, in a more formal language (UML class diagrams). The aim of this is to generate a software tool that enables to prototype enterprise business architecture and through this to support evolutionary self-development of an enterprise information system.

3.1 A Meta-Model for the Architecture

In this sub-section we present a UML based meta-model that describes the architecture, introduced in the previous sub-section, in the form of UML class diagrams. From these class diagrams we generate the core of the software system (Development Subsystem of NO IS) for supporting our methodology (IS self-development) using EMF (Eclipse Modeling Framework [11]) MDA (Model Driven Architecture [3]) and GP (Generative Programming) tools.

By the above described decentralized IS/ISD model, the whole system (NO IS) is not directly developed. Each autonomous part (subject) of the organization develops itself and its IS in mutual cooperation. Such development occurs through the Role concept that forms a basic architectural unit for the (common) *ISD Space*. Each Role describes a *partial IS* where a Subject (on the upper level on the Fig. 1) interprets (analyses) and changes (designs) (the state of) the embracing world (a circle on the lower level of the Fig. 1). In the context of the MDA concept we can say that, for each Role, a Subject (a little circle on the upper level of the figure) describes or models (the state of its) Environment (a large circle on the lower level of the figure). In order to form a common space of development, all the (cooperative) Subjects in all their Roles should describe their (overlapping) Environments applying the same (meta)schema, that could be (on the conceptual level) the following:

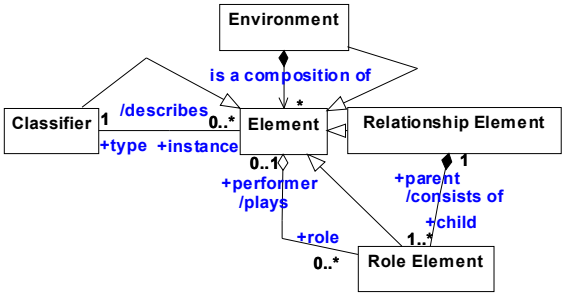

Fig. 2. A conceptual meta-model of an Environment of a Subject (UML Class diagram)

An Environment is a composition of related Elements (Fig. 2). (The Environment is an Element, too.) An Element may be a Relationship Element. A Relationship Element is a composition of Role Elements. A Role Element is an Element, that can be seen as a role played by another Element. The Role Element can point to the Element, that plays its role at the moment, as a 'performer'. Each Element must have a 'type' Classifier that represents/holds description(s) of that (instance) Element. Classifier is an Element that classifies and describes other Elements (that can be but must not be Classifiers).

An Element may (but must not) serve/act as a Relationship Element, a Role Element, and a Classifier at the same time. Therefore, we can express/describe the organization of the Environment/Elements more precisely using the following generic pattern (Fig. 3):

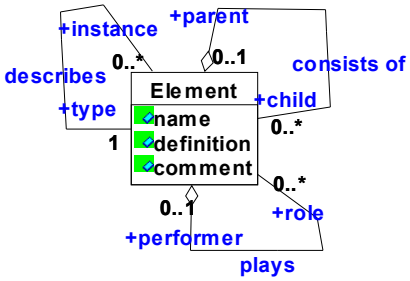

Fig. 3. A more generalized pattern for describing and relating Elements in decentralized continuous ISD environments.

An Environment (of a Subject) has three special views (Fig. 4):

- Organization
- Process
- State.

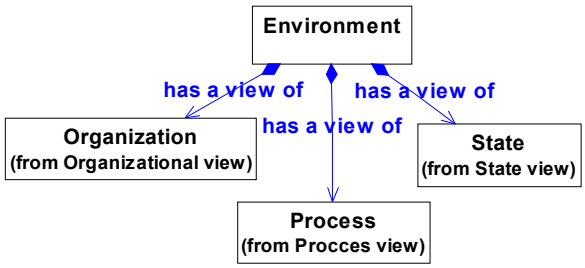

Fig. 4. An Environment has three main views: Organization, Process and State

In the organizational view, the Environment (of a Subject) is interpreted as an Organization (Fig. 5).

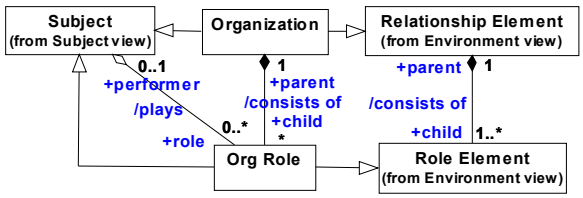

Fig. 5. An Organization is a composite Subject

An Organization is a composite Subject. This means that an Organization is, at the same time, a Subject and a Relationship Element that consists of organizational roles (Org Role) as Role Elements. An Org Role is, at the same time, a Subject and a Role Element that is played potentially by another Subject.

A Subject may (but must not) serve/act as an Organization, an Org Role, and a Classifier (Subject type) at the same time. Therefore, we can express/describe the organization of the Environment/Subjects more precisely using the following generic pattern (Fig. 6):

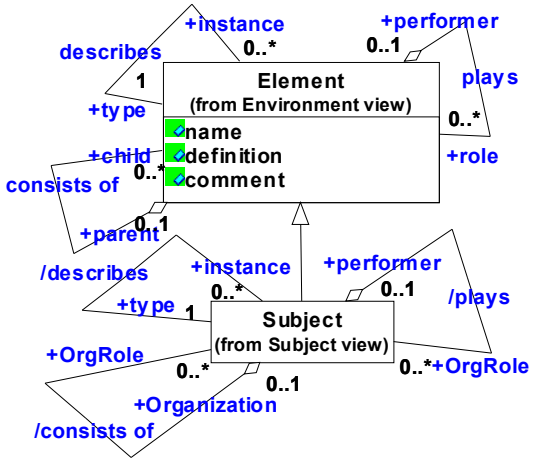

Fig. 6. A more generalized pattern for describing and relating Subjects and their Organizations in decentralized continuous ISD environments.

This Subject pattern applies and specializes the Element pattern presented in the Fig. 3. The same pattern (Element) is applied and specialized for Objects and Changes in the State and Process views, respectively (see Fig. 7).

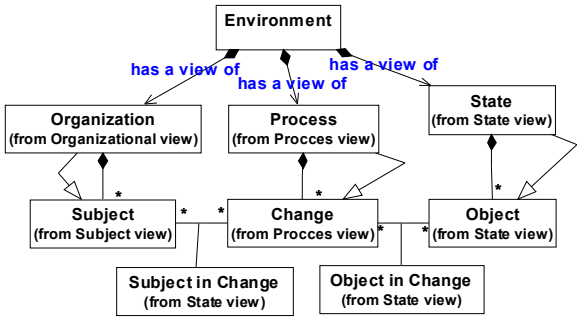

Fig. 7. An overview of the architecture of the Environment (of a Subject).

In the State view, the Environment (of a Subject) is interpreted as a composite Object or an organization of Objects. An Object is an Element in the context of the State view of the Environment (of a Subject). An Object (like an Element) may be seen as a (sub)State (that specializes the Relationship Element in the context of the State view) and/or as an Object's Role (that specializes the Role Element in the context of the State view).

In the Process view, the Environment (of a Subject) is interpreted as a composite Change or an organization of Changes of the State/Objects.

Both Subjects and Objects can participate in Changes (see the Subject in Change and the Object in Change in Fig. 7, respectively). Both classes, the Subject in Change and the Object in Change, are specializations of the Role Element in the context of the Change as a Relationship Element.

In addition to describing its Environment, a Subject should also describe itself and its Role in managing and development of this Environment. We model the Subject as a closed system that consists of Subject Interfaces (see Fig.8). A Subject Interface is a Role Element that points to an Element of the Subject's Environment (as the 'performer' of the 'role' of the Subject Interface at the moment). The Subject is a Relation Element in this context. Being a part (or a 'child') of a Subject, a Subject Interface may point also to another Subject or Organization that serves as the 'target group' of the Subject Interface.

.

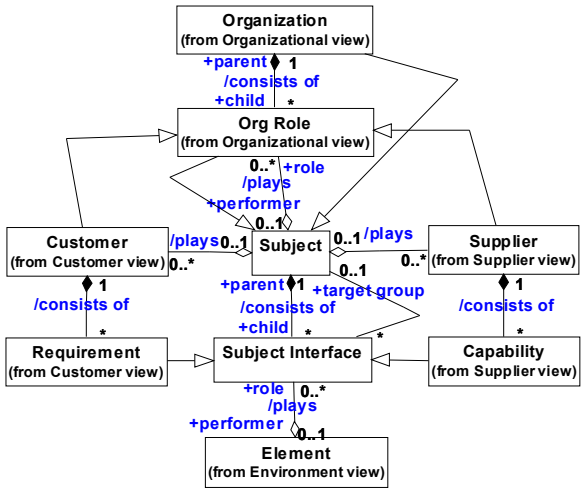

Fig. 8. The Subject is a closed system that is described by Subject Interfaces.

A Subject Interface can be interpreted from the viewpoint of the Subject (the owner of the interface) as a provided interface that we call a Capability, or as a required interface that we call a Requirement. The Capabilities and the Requirements point to (the Elements of) the internal and external environments of the Subject, respectively, and make it possible to integrate and/or cooperate Environments of different Subjects.

A Subject may play in the Organization of its Environment the role of the Customer and/or the role of the Supplier. The Customer role is characterized by Requirements and the Supplier role is characterized by Capabilities. The Customer and the Supplier are both Org Roles (i.e. Subjects as well).

In addition to describing itself (that means Subject Interfaces) and its Environment (in the framework of Organization, Process and State views), the Subject must be able to describe its Role in managing and development of the Environment. The Role is a composite Activity (see Fig.9). An Activity is formally a specialized Subject Interface, that is owned by a Subject and directed to (observing, analysing and/or changing/designing) an Element of the Environment of the Subject.

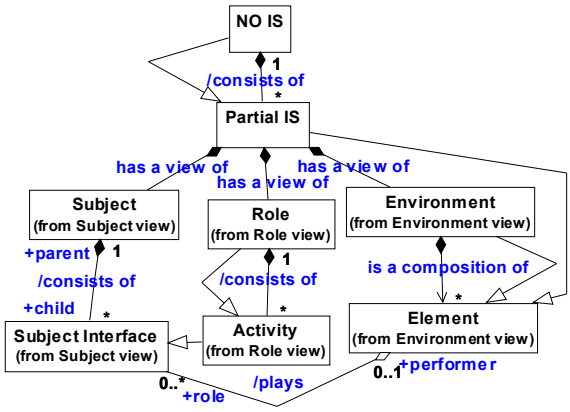

Fig. 9. An overview of the architecture of the Evolutionary IS (NO IS).

In the concluding class diagram (Fig.9) the whole system – an Evolutionary IS (or *NO IS*) is described. The whole system (*NO IS*) consists of partial systems (*Partial IS*) and is itself a specialized *Partial IS*. The *Partial IS* (and the *NO IS*, too) has three main views: Subject (the closed view), Environment (the open view), and Role (the integrating development view). Each of the main views are further divided into sub-views.

This was the role-based meta-model (introduced as *Role-Model* in [8,9,10]) for the ISD Space in the context of a NO and its IS. This virtual space is formed in the NO IS at the initial Strategic Analysis phase of the NO ISD. The meta-model is used as the basis for our (IS self-development) approach, methodology and its supporting software (the Development Subsystem of NO IS). On the basis of this meta-model we have generated, using Eclipse Modeling Framework (EMF) MDA platform tools [11], a prototype of the software that enables to prototype the business architecture of an enterprise at the initial Strategic Development phase of the Enterprise ISD, and to elaborate this architecture in evolutionary and/or decentralized way.

Our next goal on this way is to add to the described mainly product oriented view of the methodology also the process dimension by including the process engineering [7] and/or SOA [2] platform tools.

4 Conclusion

A general model of ISD for dynamic Networked Organizations (NO) was described here. This model reflects the new development situation emerged in the recent years. Then this vision was translated into a UML based conceptual meta-model, which we are prototyping and elaborating towards a working ISD methodology and its supporting software (the Development Subsystem of NO IS) using Eclipse EMF MDA platform.

References:

[1] N.L. Russo, Expanding the horizons of information systems development, *Organizational and social perspectives of information technology: Proceedings of the IFIP TC8/WG8.2 International Working Conference*, Aalborg, Denmark, 2000, pp. 103-112.

[2] J. McGovern et al, *Enterprise Service Oriented Architectures: Concepts, Challenges, Recommendations*, Springer, 2006

[3] D.S. Frankel, *Model Driven Architecture: Applying MDA to Enterprise Computing*, Wiley Publishing, Indianapolis, 2003

[4] P.R. da Cunha, A.D. de Figueiredo, Information Systems Development as Flowing Wholeness, *Proceedings of the IFIP TC8/WG8.2 Working Conference on Realining Research and Practice in Information Systems Development*, Boise, USA, 2001, pp. 29-48.

[5] E. Wenger et al, *A Guide to Manage Knowledge: Cultivating Communities of Practice*, Harvard Business School Press, Boston, USA, 2002

[6] B. Henderson-Sellers, P. Giorgini, *Agent-Oriented Methodologies*, Idea Group Publishing, USA, 2005

[7] OMG Software Process Engineering Metamodel (SPEM), Version 1.1 http://www.omg.org/technology/documents/formal/spem.htm [15.05.2007]

[8] M. Roost et al, A Role-Based Framework for Information System Self-Development, *Proceedings of the IFIP TC8/WG8.2 Working Conference on Realining Research and Practice in Information Systems Development*, Boise, USA, 2001, pp. 95-105.

[9] M. Roost, A Model of Self-Development of Information Systems, *Proceedings of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004)*, Orlando, USA, 2004, Vol I, pp. 126-131.

[10] M. Roost et al, A Model of Information Systems Development for Learning Virtual Organizations, *Information Systems Development: Advances in Theory, Practice and Education*, O. Vasilecas etc., Ed. Springer, 2005, pp. 315-325.

[11] F. Budinsky et al, *Eclipse Modeling Framework (The Eclipse Series)*, Addison-Wesley, 2007