

Towards Self-development of Evolutionary Information Systems: An Action Research of Business Architecture Development by Students in Socially Networked Groups

Mart Roost, Kuldar Taveter, Karin Rava, Jaak Tepandi, Gunnar Piho,
Rein Kuusik, and Enn Õunapuu

Dep. of Informatics, Tallinn Univ. of Technology, Akadeemia tee St.
15A, Tallinn 12618, Estonia

{Mart.Roost, Kuldar.Taveter, Karin.Rava, Jaak.Tepandi,
Gunnar.Piho, Rein.Kuusik, Enn.Ounapuu}@ttu.ee

Abstract. We present a case study of business architecture development by students working in socially networked groups. In this case study we emulated a self-development of an evolutionary information system. The “client system” in this emulated project was medical laboratory information system. In the role of the “change agent” were students of two different specialties: medical technology students (one group) and IT students (another group). We describe the process and results of the first (finished) phase of strategic analysis where the initial business architecture was developed. Later on this business architecture will be utilized as a platform for (social, self-) development of business processes and software. Medical technology students (knowing the problem) played the business process owner/analyst dual role. IT students (knowing IT-related solution patterns for the problem and processes) played the business designer role. The relationships between (and inside) the two groups/communities were managed using Google Sites (social) software.

Keywords: Enterprise Information Systems, Evolutionary Information Systems, Social Self-Development, Social Software, Action Research, Enterprise Architecture, Business Architecture, Methodology.

1 Introduction

The global information society has given rise to dynamic *networked organizations* (NWO) and enterprises. NWO is a term that is used to describe a variety of new emergent organizational structures such as *virtual and learning organizations* [1, 2, 3], which operate and evolve on the level of *information system* (IS). We can use the term of NWO as a synonym for a “contemporary or future organization”. The success of such organizations depends on their ability to adapt to the environment and learn.

In the process of organizational learning, organizations are restructuring many relationships internally and externally to respond to the demands of a shifting market [4]. Internally companies disaggregate into smaller units focused on well-defined market opportunities. Externally companies increasingly partner with other

organizations and form extended enterprises. *Extended enterprise* is a kind of NWO that forms on the level of *enterprise IS* (EIS) and embraces members from both its internal and external environments. From the above description follows that such NWO consist of relatively autonomous sub-units. Because of that, they can be adequately described by using the concept of agents. We define an agent as “an entity that performs a specific activity in an environment of which it is aware and that can respond to changes” [5]. Agents (in the same environment) can form multi-agent systems [5]. In the context of organizational learning, the agent may be termed as “change agent” and the environment as “client system” that can respond to organizational changes [6]. NWO is then a multi-agent system that forms (operates and evolves) in some IS environment as a result of the system- or development work [7] performed by agents. NWO forms on the basis of core competences and resources of the organization and is owned by independent agents. Each constituent agent of a NWO can play different roles and manages a set of business rules. Agents of a NWO form social networks [8], which can be managed with social software [9].

Social networks (in general) deal with agent relationship management. In the context of extended enterprise, social networks can be described using *business process management* (BPM) [10] lingua. We interpret a business process as a (specific) relationship/collaboration between two or more agents, playing the roles. In current case the roles are either business process owner/analyst role (medical students) or business designer role (IT students).

The main operational and learning environment for a networked organization and its business processes is its information system (IS). IS should be *evolutionary* [11]. The evolutionary information system is able to survive over time and has built-in support to handle evolutionary changes [12]. Therefore, in addition to traditional information and communication services of IS, the evolutionary IS should also provide and support services for the development of business processes and software.

Self-development of an IS is defined as a decentralized IS development model, where the whole system is not directly developed, but each agent develops its own part or role as an area of responsibilities in this whole [13, 14]. We define a *self-development subsystem* as a part of an evolutionary IS that is responsible for handling all aspects and services for the evolutionary self-development (change) of IS. Both, the evolutionary IS and its self-development subsystem, are socio-technical systems [15] with a social (business) and a technical (IT) parts. In the self-development subsystem we utilize the *End-User Development* (EUD) model in the context of networked organizations [16, 17].

Social self-development (SSD) is a community-supported self-development of a socio-technical system based on social networks. In this article we present a case study of enterprise information system self-development by students working in socially networked groups. In this case study we emulated a social self-development of an evolutionary information system. This case study has been conducted according to the *action research* (or *action learning*) methodology described in [6]. The “client system” in this emulated project was medical laboratory information system. In the role of the “change agent” were students of two different specialties: medical technology students forming one group and IT students making up another group. We describe the process and results of the first phase of strategic analysis where the initial business architecture was developed. Later on this business architecture will be

utilized as a platform for social self-development of evolutionary information systems. We focus on analyzing the relationships between our social self-development and the action research methodologies (processes). The main research problem is: how do we research and develop evolutionary IS in enterprises.

In section 2 we introduce our *action research framework*. In section 3 the main results and lessons learned are presented. Section 4 provides conclusions.

2 An Action Research Framework

According to Checkland [18], any piece of research may be thought of as entailing a particular framework of ideas F that is used in a methodology M to investigate some area of interest A . In the research reported in this article, A is *evolutionary enterprise information system*, F is *social self-development*, and M is *action research*. The initial research question or problem is: how these three topics are related and work together?

2.1 Area of Interest: Evolutionary Enterprise Information Systems

Our research area (A) is *evolutionary enterprise information systems*. An evolutionary IS is able to survive over time and has built-in support to handle evolutionary changes. Information system is a *socio-technical system* that has social (business) and technical (IT) parts. An evolutionary information system can be continuously changed by its constituent agents. Agents can be human agents (persons, organizational units, groups of persons) as well as “man-made” software agents. This means that evolutionary IS can be understood and described as a multi-agent system [5].

The evolution has two “sides”: continuous work improvement [19] and software evolution [20]. Business process evolution, which is a sub-domain of the *business process management* (BPM) [10] domain, bridges both evolution sides [21]. IS evolution is often considered as being only (or mainly) the evolution of the software [22]. This kind of software evolution is often considered at the later stages of software development (implementation, execution), mostly adopting pragmatic approaches [22, 23] only. Our approach is an architectural approach.

Our starting point is that evolution is a matter of architecture. System’s architecture consists “fundamental concepts or properties of a system in its environment embodied in its elements, their relationships, and in the principles of its design and evolution” [24]. Evolution of enterprise IS is a matter of *enterprise architecture* (EA). Enterprise architecture is the “[enterprise] conceptualization of the form, function, and fitness-for-purpose of a system in its environment, as embodied in the elements of the system, the relationships between those elements, the relationship of the system to its environment and the principles guiding the design and evolution of the system” [25].

In *model driven development* (MDD) approaches [26, 27], an EA is described by a meaningful and useful set of enterprise models [25, 28] for (and owned/evolved by) large sub-communities of the enterprise. The evolution of the models is guided by a meta-model that describes a more stable but in principle evolving meta-architecture

[28]. If this meta-architecture is implemented as a core subsystem of EIS, we have *evolving EA* [29].

The business architecture [30] is a part of an EA, which's describing set of models corresponds to the first two rows of the Zachman Framework [31, 32], representing respectively the Executive Perspective and Business Management Perspective.

To handle model-driven evolution, an evolutionary IS should also provide and support services for the development of business processes and software, in addition to traditional information and communication services. If the development is performed by agents of an IS in a social manner, we can talk about social self-development, which is addressed by next subsection.

2.2 Framework of Ideas: Social Self-development

Our framework of ideas (F, Checkland [18]) is *social self-development* that combines ideas from *end-user development* [33], *developmental work research* [19], *meta-design* [34], and *infrastructuring* [35]. Infrastructuring is defined as a bottom-up, participatory approach to EA development.

The solution of a *self-development subsystem* that was defined in section 1 can be based on the meta-design (MD) theory. According to MD, one of the main weaknesses of designing (architecting) is related to evolutionary character of the design (architecture) and the incapacity of fully anticipating at design-time the needs and tasks of users [36].

With the purpose of overcoming this limitation, MD aims at defining mechanisms that allow “owners of the problem” to become designers [37]. MD is defined as a conceptual framework that allows end-users to create contents by using socio-technological infrastructure, in which people can actively participate [38]. We define enterprise IS self-development subsystem as an enterprise-wide MD framework, which follows the concepts of *evolving enterprise architecture* and *infrastructuring*.

The MD has become a leading theory for end-user development (EUD) [34, 35]. The EUD is defined in [33] as “the set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create or modify a software artifact”. In the context of enterprise IS that we interpret as socio-technical systems, the EUD requires a complementary method for continuous work improvement. Syrjanen and Kuutti have proposed *developmental work research* [19] as a potential method to be coupled with EUD to deal with work improvement aspects needed to implement EUD in organizational environments.

Our architectural approach to *social self-development* (SSD) of *evolutionary information systems* requires proper methods and work products for continuous work improvement supported by EUD of software applications.

We see social self-development defined in section 1 as a method for both EUD and MD that is applicable in the context of networked organizations. An important goal of our research is to apply and extend MD theory in the context of extended enterprises and evolutionary information systems in order to describe and implement a platform for SSD of EIS.

2.3 Methodology: Action Research

Our research is conducted according to the Action Research methodology [39]. Action Research (AR) is known by many other names like participatory research, collaborative inquiry, emancipatory research, action learning, and contextual action research. AR can be seen as a kind of practical problem solving approach. The essence of the approach is “learning by doing” – a group of people identify a problem, do something to resolve it, see how successful their efforts were, and if not satisfied, try again [39]. AR aims to contribute both to the practical concerns of people in an immediate problematic situation and to further the goals of (social) science simultaneously [40]. According to Reason & Bradbury in [41], AR is an interactive inquiry process that balances problem solving actions implemented in a collaborative context with data-driven collaborative analysis or research to understand underlying causes enabling future predictions about personal and organizational change).

AR occurs as collaboration between a “client system” and a “change agent” (who is opposed to traditional “observer”). For AR performed by our research group, a “client system” is a concrete enterprise for which we perform problem domain analysis and the “change agent” is our university’s research group. It is important that in our SSD approach, not only an external research group, but each agent in the context of an (extended) enterprise is seen as a change agent who performs the needed changes in the context of its own roles and responsibilities defined by them. This is in harmony with the Developmental Work Research approach [19], which we also view as a special version of AR.

2.4 Case Study: Action Learning on Medical Laboratory

As an example case study for our AR, this article presents a real action learning process in the context of the subject “Teamwork in Information Systems’ Development” taught at our university. In the case study conducted, the ‘client system’ was Medical Laboratory with its Laboratory Information Management System (LIMS). In the role of the ‘change agent’ were students of two different specialties: students of medical technology forming one group and IT students making up another group. In this article we describe the process and work products of the first finished phase of problem domain analysis that produced initial business architecture. Within this process, the students of medical technology played the dual roles of core business process owners and business analysts knowledgeable about the problem. The IT students played the roles of business designers knowledgeable on possible IT-enabled solutions for the problem.

The relationships between and within the two groups and also work products created and changes introduced by them were managed by Google Sites social software. In the context of each agent’s role, which was defined in terms of its responsibilities, a diary was written, which reflected the process of playing and developing this role (analogously with the situation in a theatre, where each actor would write a web-based diary or blog between performances; an entry in the diary would usually reflect (from the viewpoint of the writer) a particular performance

(but sometimes might express more general knowledge). Such learning model is in good harmony with the Action Research methodology.

2.5 Limitations

An important limitation of the work is that we simulated the action research setup with students – thus it would be rather a kind of lab experiment because students “played” practitioners. That is close to but not the same like “really doing it”. Therefore we plan to repeat similar action research with a real enterprise.

3 Social Self-development within Enterprise Business Architecture: Our Methodology

In this section we present the main results of and lessons learned from the *action research* project introduced in the previous section. The results include the work products of the problem domain analysis of a medical laboratory and its IS (LIMS).

At the meta-level, our research results also include the problem domain analysis of our problem domain analysis methodology and its IS (as a part of LIMS Self-Development Subsystem, which includes a software factory described in [42]).

3.1 Strategic Analysis of the EIS (LIMS)

In this section we describe the process and work products of the first finished phase of the problem domain analysis that produced an initial version of the business architecture descriptions of the enterprise - Medical Laboratory. We will next view our methodology and its usage from three perspectives: Product Perspective, Process Perspective, and Self-Development Perspective.

The Product Perspective. According to our problem domain analysis methodology [43, 13], the overall work product termed as Enterprise Architecture (EA) is composed of the following three sub-architectures:

- Business Architecture;
- Technology Architecture;
- Development (Work) Architecture.

In the context of the student project, we focused on describing the Business Architecture. A snapshot of the description of the Business Architecture of the Medical Laboratory is depicted in Figure 1.

According to our methodology, a Business Architecture is described by the following three interrelated views:

- Organizational view;
- Functional view;
- Informational view.

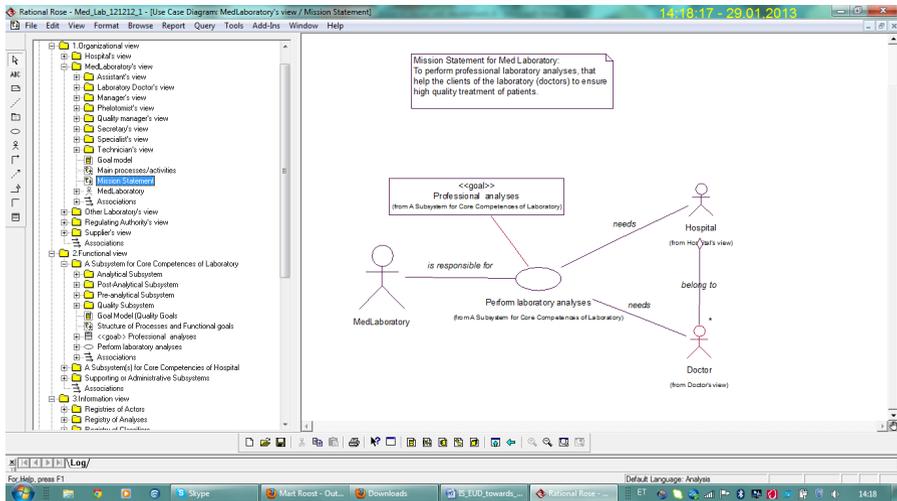


Fig. 1. An overview of the Business Architecture of the Medical Laboratory as described in the student project

The Organizational view reflects the organizational decomposition of a business, and consists of Organizational Subsystems. An Organizational Subsystem of EIS describes an Agent's Role in terms of responsibilities of an Agent playing the corresponding role in the context of the EIS as a socio-technical system. In the context of the Medical Laboratory, examples of Organizational Subsystems are Hospital Department, Laboratory Specialist, and Regulative Authority.

An Organizational Subsystem can be represented by a software agent [5], for example, by a Laboratory Specialist's Agent, which assists the respective human agent in fulfilling its professional responsibilities as well as the responsibilities that are concerned with the (self)-development of the EIS. An example of an Organizational Subsystem's description is given in Figure 2.

The Functional view reflects the process decomposition of an enterprise, and consists of Functional Subsystems. A Functional Subsystem of an EIS describes and implements a major Business Process as a potential service in the context of the EIS as a socio-technical system. In the context of the Medical Laboratory, examples of Functional Subsystems are Pre-Analytical Subsystem, Analytical Subsystem, and Quality Subsystem, which describe and implement the respective Pre-Analytical, Analytical, and Quality Management Business Processes. According to the theory of Meta-Design proposed in [34], such Functional Subsystems can be handled as Business Process Archetypes.

The Informational view reflects the structure of conceptual objects (informational architecture) of an enterprise, and consists of Informational Subsystems called Registries of EIS. A Registry describes and/or implements a major Business Object in the context of an EIS as a socio-technical system. In the context of the Medical Laboratory, examples of Informational Subsystems are Registry of Samples, Registry of Analyses, and Registry of Equipment. Such Registries are similar to Business Archetype Patterns proposed in [42].

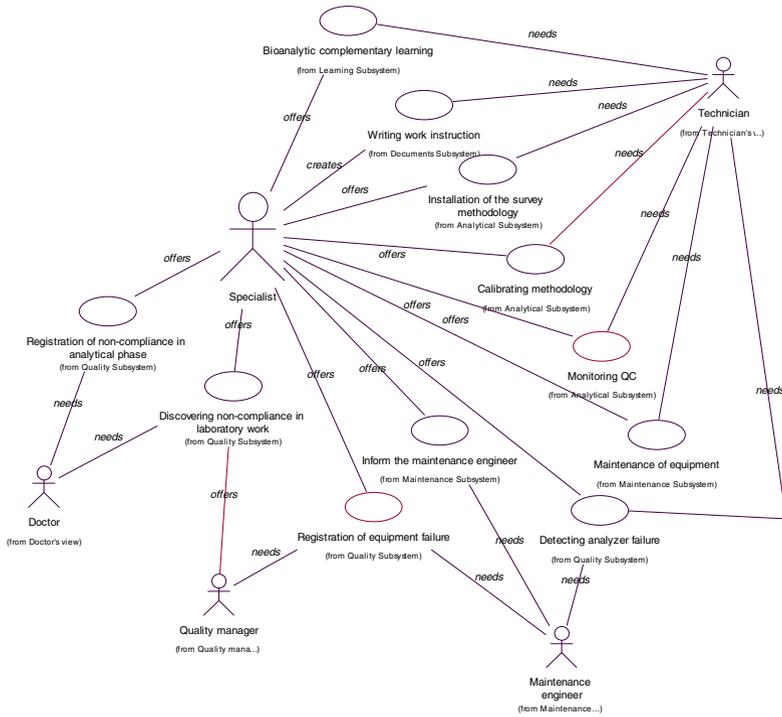


Fig. 2. An overview of an Organizational Subsystem of the Medical Laboratory described in the student project: the responsibilities of Laboratory Specialist and its relations to other roles

The Process Perspective. We next address the Process Perspective of Enterprise Business Architecture Development.

According to our methodology, the Enterprise Business Architecture Development process is owned by a player of the Enterprise Business Architect role, which is responsible for the overall Business Architecture of the Enterprise.

The Enterprise Business Architect works together with players of the Business Analyst and Business Designer roles.

A Business Analyst is responsible for models and descriptions of one or more Organizational Subsystems of an enterprise.

A Business Designer is responsible for models and descriptions of one or more Informational Subsystem(s) (Registries).

We decompose the process of Enterprise Business Architecture Development proposed by us (v. Figure 3) into the following three sub-processes:

- Strategic Business Analysis (performed by all Business Analysts)
- Strategic Business Design (performed by all Business Designers)
- Managing and Evolving the Whole (performed by the Business Architect).

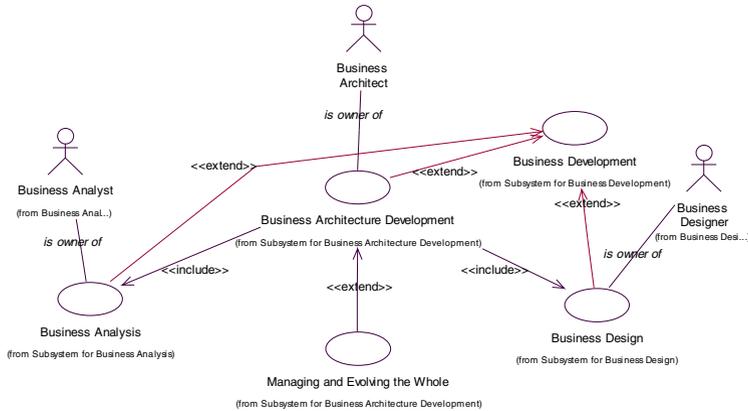


Fig. 3. An overview of Enterprise Business Architecture Development

Strategic Business Analysis is a part of the more general process of Business Analysis that is performed in the problem domain analysis phase of EIS development. Strategic Business Analysis is performed in the context of concrete Organizational Subsystems to be analyzed that are assigned to concrete Business Analysts. In the perspective of Self-Development, our goal is to educate and support business actors (agents) who are able and interested in acting as Business Analysts of their own Organizational Subsystems. For example, a Laboratory Specialist in a Medical Laboratory represented in Figure 2 should be able and interested in analyzing and modeling the Organizational Subsystem in the problem sub-domain of laboratory work *in concert with the responsibilities defined by his/her role*. In our student project, the students of medical technology played the dual roles that combine a laboratory-specific role with the Business Analyst's role. Each student was responsible for one or two Organizational Subsystems. Strategic Business Analysis of an Organizational Subsystem was divided into the following two major phases in a “bottom-up” manner:

- In the first phase of strategic business analysis, a student “playing” a laboratory-specific role described and modeled the vision of his/her Organizational Subsystem individually (in principle, such a part might alternatively be played and described by a team or community);
- In the second phase of strategic business analysis, the student coordinated her/his vision/descriptions with “players” of related laboratory-specific roles (this is a part of the “horizontal coordination” activity over the whole Organizational view), and established the necessary change requests in order to change concrete Functional and Informational Subsystems. For example, the student who played the Laboratory Specialist role, coordinated her work products (one of them is depicted in Figure 2) with the players of the roles of Technician, Maintenance Engineer, Quality Manager, and Doctor; and created requirements and change requests for the IT students, who served as Business Designers of the Quality, Maintenance, Analytical, Learning, and Documents Subsystems (v. Figure 2). During the second

phase, sub-communities of cooperating Business Analysts and Business Designers were formed around the Organizational Subsystems, with the help of the social software.

In the context of the Action Research (AR) methodology, which was briefly described in sub-section 2.4, the “bottom-up” sub-process of Strategic Business Analysis corresponds to the collaborative analysis and research activity with the purpose to understand one’s area of concern or problem sub-domain. For the latter, the “top-down” sub-process of Business Design creates an IT-enabled solution. Strategic Business Design is the part of the more general process of Business Design that is performed in the context of the Strategic Analysis phase of EIS development. Strategic Business Design is focused on major business objects that are assigned to concrete Business Designers. The business objects are primarily described as components of Informational Subsystems or Registries. The lifecycles of business objects are often managed by Functional Subsystems. For example, a central business object of the Laboratory is the biochemical Analysis (test), which is described by the Registry of Analyses and managed by the Analytical Subsystem. A Business Designer, who is responsible for specific business objects and Functional Subsystems related to them, can apply domain-specific or even more general solution patterns, which can be adapted differently in the contexts of different Organizational Subsystems, and even in different organizations (Laboratories). In the perspective of Self-Development, Business Designers belong to the context of the extended enterprise (Laboratory) as providers of respective development services supporting Social Self-Development.

In our student project, the role of Business Designer was played by IT students. Each IT student was responsible for two or three business objects and their respective Registries and Functional Subsystems. For example, one IT student was responsible for the Registry of Equipment, the Registry of Maintenance Works, and for the Maintenance Subsystem. This student worked in pair with a student of medical technology who represented the (Manager of the) Maintenance Department. The Business Design for these business objects and subsystems was also divided into two major phases, but in the following “top-down” manner:

- In the first phase of strategic business design, some general solution pattern for the business object and its Functional Subsystems was introduced by the Business Designer, and was integrated into the whole business architecture in collaboration with the Business Architect (played by one of the authors of the paper), and with the Business Designers of related objects and Functional Subsystems;
- In the second phase of strategic business design, which occurs in the context of concrete Organizational Subsystems, the respective elements of the Organizational Subsystems were described on the basis of the respective elements of the Registries and Functional Subsystems that describe and implement the business object. For example, the Laboratory Specialist’s view to his/her Organizational Subsystem was described on the basis of the elements of the Analytical, Quality, Documents, and Maintenance Subsystems, as is shown in Figure 2. Good collaboration within pairs of a Business Analyst and a Business Designer (for example, the Laboratory

Specialist and the Business Designer of the Documents), supported by the social software was regarded as a key to success of this phase.

The Managing and Evolving the Whole is the part of the Enterprise Business Architecture Development that is performed by the Enterprise Business Architect. This includes the tasks of building an initial structure of the architectural description that was introduced above as the Product Perspective of our methodology, deciding the division of labor according to Functional Subsystems between individual Business Analysts and Business Designers, supporting and coordinating the collaboration between individual Business Analysts and Business Designers, monitoring and changing the structure of the whole Enterprise Business Architecture, etc.

The Self-development Perspective. In the Self-Development perspective of the Strategic Analysis Methodology, we consider the methodology proposed by us as a part of the Meta-Design (MD) framework that includes the domain-independent Meta-Design layer, the domain-specific Design layer, and the agent-specific Usage layer. These layers form a separate “abstraction-concretization” dimension that is orthogonal to the Organizational, Functional, and Informational views of the Enterprise Business Architecture described above. We plan to implement this MD framework as the Self-Development Subsystem of an EIS supporting the Social Self-Development (SSD) of Business Processes and their underlying software applications in a similar fashion, but with a much higher quality compared to the student project described in this article. In the student project, the supporting infrastructure (software) included only the UML modeling software for describing and managing the business architecture, and the social software for relationship and change management in the form of blogs. The future Self-Development Subsystem will consist of subsystems for the following purposes: for Business Development (including Business Architecture Development, Strategic Business Analysis, and Strategic Business Design), for Technology Development (including Software Factories described in [42]), for Evolutionary Change Management (including the blogs similar to the ones used in the student project), and for Simulation and Training.

3.2 Social Self-Development (of Enterprise Business Architecture) as Action Research

In our methodology, the EIS evolution is handled according to the general learning model of Action Research, which was described in sub-section 2.3, as follows:

- Agents of an (extended) enterprise, who play the dual roles of some business concepts’ owners (v. the Product Perspective in section 3.1) and developers (Business Analyst, Business Designer, Business Architect), serve as ‘change agents’;
- The enterprise with its IS serves as the ‘client system’;
- The ‘bottom-up’ sub-process of Business Analysis serves as the ‘data-driven collaborative analysis or research’ (v. section 2.3);
- The ‘top-down’ sub-process of Business Design serves as the ‘collaborative problem solving or change’, in the context of AR (v. section 2.3).

The version of AR proposed by us follows the SSD approach and its meta-model, both of which have been introduced in [17].

3.3 Lessons Learned from the Action Research

The following major lessons were learned from the student project described in the current section:

- The approach (SSD) is applicable in the collaborative learning contexts that are similar to our student project, but requires much richer supporting infrastructure;
- Using the social software based on blogs is very useful in such contexts;
- Similarly to the pair programming, we can also perform strategic analysis and design effectively in pairs of a “business person” (like a student of medical technology) and an “IT-person” (like an IT student);
- Without continuous community-based modeling activity the SSD approach does not work.

4 Conclusion and Future Work

The research problem was: how do we research and develop evolutionary information systems of enterprises. In this paper, an approach to social self-development of evolutionary information systems was described. This approach was described by relying on the example of a real action research project that was performed in the context of the subject ‘Teamwork in information systems development’ taught at our university. The ‘client system’ was Medical Laboratory with its Laboratory Information Management System (LIMS). In the role of the ‘change agent’ were students of two different specialties: students of medical technology, forming one group and IT students, making up another group. The students of medical technology played the dual roles of core business process owners and business analysts, who are knowledgeable about the problem. The IT students played the roles of business designers knowledgeable on IT-enabled possible solutions for the problem. The relationships between and within the two groups or communities were managed by using Google Sites social software.

We described the process and work products of the first phase of problem domain analysis that produced descriptions and work products of the business architecture of the enterprise under discussion – Medical Laboratory. We interpret the enterprise business architecture development methodology proposed by us as a core part of the resulting business architecture. Such architecture follows the concepts of meta-design, evolving enterprise architecture, and infrastructuring, and serves as a platform for further social self-development of business processes of an enterprise and their underlying software applications. The social self-development was defined as an agent-centric and community-supported development of a socio-technical (multi-agent) system by means of social networks that are based on social software.

In the near future, we will design and develop a solution to self-development, where software agents representing organizational units or even individual roles engaged in self-development manage and coordinate the necessary collaboration and competition between them. Principles for this kind of solution are, for example, described in [44]. We will also elaborate the model-based description of our social self-development methodology, design its supporting toolset, and plan to use them in real enterprises engineering situations. Our methodology can be understood as a domain-specific version of the action research methodology, which is applicable for evolutionary information systems.

References

1. Kimble, C.: Management and Information Systems (March 30, 2007), <http://www.chris.kimble.com/Courses/mis> (retrieved October 4, 2009)
2. Strausak, N.: Resumee of VoTalk, Organizational Virtualness. In: Sieber, P., Griese, J. (eds.) Proceedings of the VoNet-Workshop, April 27-28, Simowa Verlag Bern (1998), <http://virtual-organization.net/news/proc-98.pdf> (retrieved May 11, 2001)
3. Senge, P.M.: A Guide to Manage Knowledge: The Fifth Discipline: The Art & Practice of the Learning Organization. Currency-Doubleday, New York (2006)
4. Wenger, E., et al.: A Guide to Manage Knowledge: Cultivating Communities of Practice. Harvard Business School Press, Boston (2002)
5. Sterling, L., Taveter, K.: The art of agent-oriented modeling. The MIT Press, Cambridge (2009)
6. O'Brien, R.: An Overview of the Methodological Approach of Action Research (1998), <http://www.web.ca/robrien/papers/arfinal.html> (retrieved July 9, 2012)
7. Rava, K., Mikli, T.: System Work in System Development. In: Proceedings of the IFORS Special Conference (SPC8) on Organizational Structures, Management, Simulation of Business Sectors and Systems, pp. 213–217. Technologija, Kaunas (1998)
8. Scott, W.R., Davis, G.F.: Networks in and around Organizations: Organizations and Organizing. Pearson Prentice Hall, New Jersey (2003)
9. Wikipedia Foundation - Wikipedia, Social software, http://en.wikipedia.org/wiki/Social_software (retrieved July 9, 2012)
10. Chang, J.F.: Business Process Management Systems: Strategy and Implementation. Auerbach Publications, New York (2006)
11. Patel, N.V.: Adaptive Evolutionary Information Systems. Idea Group Publishing, London (2003)
12. Bellahsène, Z., Léonard, M. (eds.): CAiSE 2008. LNCS, vol. 5074. Springer, Heidelberg (2008)
13. Roost, M., Kuusik, R., Veskiõja, T.: A Role-Based Framework for Information Systems Self-Development. In: Russo, N.L., Fitzgerald, B., DeGross, J.I. (eds.) IFIP TC8 WG8.2 International Working Conference on Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective, pp. 95–105. Kluwer Academic Publishers, Norwell (2001)
14. Roost, M., Rava, K., Veskiõja, T.: Supporting Self-Development in Service Oriented Information Systems. In: Le, M.H., Demiralp, M., Mladenov, V., Bojkovic, Z. (eds.) 7th WSEAS International Conference on Applied Informatics & Communications (AIC 2007), pp. 52–57. WSEAS Press, Athens (2007)

15. Ropohl, G.: Philosophy of socio-technical systems. *Society of Philosophy and Technology* 4(3) (Spring 1999), http://scholar.lib.vt.edu/ejournals/SPT/v4_n3html/ROPOHL.html (retrieved May 11, 2012)
16. Mehandjiev, N., De Angeli, A., Wajid, U., Namoun, A., Battocchi, A.: Empowering End-Users to Develop Service-Based Applications. In: Piccinno, A. (ed.) *IS-EUD 2011. LNCS*, vol. 6654, pp. 413–418. Springer, Heidelberg (2011)
17. Roost, M., Piho, G., Taveter, K.: Information Systems’ Self-development as a Model of End-User Development in Networked Organizations. In: *Proceedings of the Second International Workshop on EUD4Services Methodologies, Tools and Evaluations, EUD4Services* (June 6, 2011), <http://130.88.6.246/EUD4Services/Workshop2011/> (retrieved June 5, 2011)
18. Checkland, P., Holmes, S.: *Information, Systems and Information Systems*. John Wiley & Sons Ltd., London (1998)
19. Syrjänen, A.-L., Kuutti, K.: From System Development Toward Work Improvement: Developmental Work Research as a Potential Partner Method for EUD. In: Piccinno, A. (ed.) *IS-EUD 2011. LNCS*, vol. 6654, pp. 123–138. Springer, Heidelberg (2011)
20. Mens, T., Wermelinger, M., Ducasse, S., Demeyer, S., Hirschfeld, R.: Challenges in software evolution. In: *Proceedings of 8th International Workshop on Principles of Software Evolution*. IEEE Press (2005)
21. Netjes, M., Mansar, S.L., Reijers, H.A., van der Aalst, W.M.P.: An Evolutionary Approach for Business Process Redesign – Towards an Intelligent System. In: *Proceedings of the Ninth International Conference on Enterprise Information Systems, ICEIS 2007, Funchal, Madeira, Portugal, June 12-16, vol. EIS*, pp. 47–54 (2007)
22. Cimpan, S., Verjus, H., Alloui, I.: Dynamic Architecture Based Evolution of Enterprise Information Systems. In: *Proceedings of the Ninth International Conference on Enterprise Information Systems, ICEIS 2007, Funchal, Madeira, Portugal, June 12-16, vol. EIS*, pp. 221–229 (2007)
23. Demeyer, S., Ducasse, M., Nierstrasz, O.: *Object-Oriented Reengineering Patterns*. Morgan Kaufmann (2002)
24. International Organization for Standardization: *Systems and Software Engineering – Architecture Description (ISO/IEC CD1 42010)*, Draft version of January 2010, currently titled *Recommended Practice for Architecture Description of Software Intensive Systems*, <http://www.iso.ch> (retrieved December 30, 2012)
25. International Organization for Standardization: *Automation systems and integration - Framework for enterprise architectures and models (ISO WD 15704)*, draft version of January 2009, currently titled *Industrial automation systems – Requirements for enterprise architecture methodologies* (2009)
26. Krogstie, J.: Integrating Enterprise and IS Development using a Model Driven Approach. In: Vasilecas, O., et al. (eds.) *Information Systems Development: Advances in Theory, Practice and Education (13th International Conference on Information Systems Development, ISD 2004)*, pp. 43–54. Springer, New York (2005)
27. Frankel, D.S.: *Model Driven Architecture: Applying MDA to Enterprise Computing*. Wiley Publishing, Indianapolis (2003)
28. Frank, U.: *Enterprise Modeling* (2009), <http://www.wi-inf.uni-due.de/FGFrank/index.php?lang=en&groupId=1&contentType=ResearchInterest&topicId=14> (retrieved June 26, 2012)

29. Martin, R., Purao, S., Robertson, E.: *Evolving Enterprise Architectures* (2012), http://www.tinwisle.com/modeling/Martin_Purao_Robertson_preICEIMT_paper.pdf (retrieved June 18, 2012)
30. The Open Group, TOGAF Standard, Version 9.1 (2009), <http://www.opengroup.org/togaf/> (retrieved June 28, 2009)
31. Zachman, J.A.: *A Framework for Information Systems Architecture*. IBM Systems Journal 26(3) (1987)
32. Zachman International – The Official Home of The Zachman Framework, *The Zachman Framework for Enterprise Architecture: The Enterprise Ontology, Version 3.0* (2011), <http://www.zachman.com/> (retrieved June 26, 2012)
33. Lieberman, H., Paterno, F., Wulf, V. (eds.): *End-User Development*. Kluwer Academic Publishers, Dordrecht (2006)
34. Fischer, G., Giaccardi, E.: *Meta-Design: A Framework for the Future of End-User Development*. In: Lieberman, H., et al. (eds.) *End-User Development, Human-Computer Interaction Series*, pp. 427–457. Springer, Berlin (2006)
35. Bolmsten, J., Dittrich, Y.: *Infrastructuring When You Don't – End-User Development and Organizational Infrastructure*. In: Piccinno, A. (ed.) *IS-EUD 2011. LNCS*, vol. 6654, pp. 139–154. Springer, Heidelberg (2011)
36. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., Mehandjiev, N.: *Meta-Design: A Manifesto for End-User Development*. *Communications of the ACM* 47(9), 33–37 (2004)
37. Costabile, M.F., Fogli, D., Lanzilotti, R., Marcante, A., Mussio, P., Parasiliti Provenza, L., Piccinno, A.: *Meta-design to face co-evolution and communication gap between users and designers*. In: Stephanidis, C. (ed.) *HCI 2007. LNCS*, vol. 4554, pp. 46–55. Springer, Heidelberg (2007)
38. Díez, D., Díaz, P., Aedo, I.: *Meta-design Blueprints: Principles and Guidelines for Co-design in Virtual Environments*. In: Piccinno, A. (ed.) *IS-EUD 2011. LNCS*, vol. 6654, pp. 276–281. Springer, Heidelberg (2011)
39. O'Brien, R.: *An Overview of the Methodological Approach of Action Research* (1998), <http://www.web.ca/robrien/papers/arfinal.html> (retrieved July 9, 2012)
40. Gilmore, T., Krantz, J., Ramirez, R.: *Action Based Modes of Inquiry and the Host-Researcher Relationship*. *Consultation* 5(3), 161 (1986)
41. Reason, P., Bradbury, H.: *The SAGE Handbook. Participative Inquiry and Practice*, 1st edn. Sage, London (2001)
42. Piho, G., et al.: *From Archetypes Based Domain Model via Requirements to Software: Exemplified by LIMS Software Factory*. In: *Proceedings of the 34th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, Opatia, Horvatia, May 23-27* (2011)
43. Roost, M., Kuusik, R., Rava, K., Veskiöja, T.: *Enterprise Information System Strategic Analysis and Development: Forming Information System Development Space in an Enterprise*. In: Okatan, A. (ed.) *Proceedings of the International Conference on Computing Intelligence, ICCI 2004, Istanbul, Turkey*, pp. 215–219 (2004)
44. Huhns, M.N.: *From DPS to MAS to ...: Continuing the Trends*. In: Decker, Sichman, Sierra, Castelfranchi (eds.) *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS 2009, May 10-15. International Foundation for Autonomous Agents and Multiagent Systems, Budapest* (2009)