Towards a Contingency Framework for Engineering an Enterprise Architecture Planning Method

Mauri Leppänen
Computer Science and Information Systems
University of Jyväskylä
mauri@cs.jyu.fi

Katariina Valtonen
Information Technology Research Institute
University of Jyväskylä
katariina.valtonen@titu.jyu.fi

Mirja Pulkkinen
Information Technology Research Institute
University of Jyväskylä
mirja.pulkkinen@titu.jyu.fi

Abstract. Enterprise architecture (EA) has a key role in managing and relating business strategies and processes, information systems, and IC technologies. EA planning devises an EA plan with development steps to make necessary enhancements to the current state of the affairs in EA. Accomplishing EA planning in a systematic and efficient manner requires the availability of frameworks and methods suitable to the situation at hand. The selection and adaptation of these methodical artifacts for the use of EA planning is often quite challenging in practice. We argue that the contingency approach can help in these activities of method engineering (ME), and present a contingency framework (named here EACon, EA Contingencies) for engineering an EA planning method. The framework is composed of three layers: the topmost layer pertains to a method engineering context, the next lower layer corresponds to an EA planning context, and the lowest layer stands for an EA development context. Due to the largeness of the framework, this paper focuses on the topmost layer. The contingency framework has been derived from ISD contingency factors, conceptualizations of the ME domain, EA critical success factors, and experience obtained from a large national project of engineering an EA planning method for state administration.

Keywords. Contingency Framework, Enterprise Architecture (EA), EA Planning, EA Development, Method Engineering, EA Method Engineering.

Introduction

In the past few years enterprise architecture has become an increasingly important means to cope with the complexity faced in managing business strategies and processes, information processing and IC technologies. *Enterprise architecture* (EA) is seen as a collection of artifacts that define and describe the structure and processes of an enterprise (a private company or public organization), the information being stored, processed and communicated in this enterprise, the systems used for these activities, and the technologies and infrastructure that the systems are implemented with. These descriptions can take various forms, such as models, figures, tables, matrices and textual representations. *Enterprise architecture planning* stands for the activities that aim at improving the current state, in other words devising development steps to make enhancements to the present ICT infrastructure, systems, information management and also the organizational processes and structures. This emphasizes the simultaneous development of the business and organizational activities, on one hand, and the ICT systems and architectures, on the other hand (Applegate 1994). *EA development* means the realization of an EA plan, or part thereof, in some organizational unit(s). EA development is an effort limited in scope and time, typically conducted as projects either by the end-user organization IT function or by an ICT provider (Pulkkinen and Hirvonen 2005b).

Several frameworks, models and methods have been suggested for the EA planning and development (e.g. see reviews e.g. Whitman et al. 2001, Schekkerman 2003, Pulkkinen and Hirvonen 2005a). The Open Group Architecture Framework, TOGAF, version 8, “Enterprise Edition” together with the Architecture Development Method (ADM) collects the best practices from a large number of efforts, among the most influential ones being the US Federal Government Enterprise Architecture Framework (FEAF) and domain specific frameworks that were created for different government sectors (The Open Group 2003). Situations in EA planning differ greatly from one another, implying that there is a need to carefully consider which kinds of strategies, approaches, frameworks and methods should be applied in each of them. The EA field is evolving rapidly, and hence new methodical artifacts and new variants of the existing ones are needed. Due to the novelty of the EA field, engineering these methodical artifacts is quite challenging. To support the selection, construction and customization of methodical artifacts for EA planning, some kind of contingency framework would be of great help. There are some lists of success factors for EA planning (e.g., Ylimäki 2006), but, as far as we know, no contingency frameworks are available.

Method engineering (ME) means all kinds of actions by which a new method, or an improved method, is developed, customized and configured to fit the needs of a domain, an organization or a project (Leppänen 2005). The ME literature suggests a number of strategies, meta models, techniques and procedures to support this engineering work (e.g. Kumar et al. 1992, Harmsen 1997, Tolvanen 1998, Ralyte et al. 2003, Kelly et al. 1996, Leppänen 2005). Although these artifacts have been originally crafted for the use of engineering information systems development (ISD) methods, they are, to some extent, applicable to EA method engineering as well. Contingency frameworks have been commonly deployed in ISD (e.g. van Slooten et al. 1996, van Offenbeek et al. 1996, Kettinger et al. 1997, Hardgrave et al. 1999, Backlund 2002), but no frameworks are available for ME, and even less for EA ME.

This paper suggests a contingency framework, called the EACon framework, for engineering an EA planning method. This framework is composed of three layers: the topmost layer pertains to a method engineering context, the next lower layer corresponds to an EA planning context, and the lowest layer stands for an EA development context. Due to the largeness and complexity of the framework, this paper focuses on the topmost layer. The construction of the contingency framework has followed three strategies. First, based on a certain kind of analogy between EA planning and ISD, we have derived contingency factors from those suggested in ME and ISD literature. Second, we have utilized the literature on critical success factors of EA planning (e.g. Ylimäki 2006) to find out which of aspects in EA planning are crucial. Third, we have exploited our experience from a large national project of engineering an EA planning method for state administration.

The remainder of the paper is organized as follows. First, we shortly discuss EA, method engineering and contingency approach. Second, we describe the research framework and process. Third, we present contingency factors of the topmost layer for EA planning method engineering. The paper ends with the summary and conclusions.

Basic Concepts

Enterprise Architecture and its Planning

Enterprise architecture involves both enterprise management through the management of its ICT support, and the development and maintenance of the information systems in use in the enterprise. *Enterprise* (The Open Group 2003) means an organized group of people working towards a common goal such as production of products or services, and may mean either a commercial or public organization. *Architecture* (O'Rourke et al. 2003) is commonly understood as the description of a structure, its parts and their interdependencies. The dual reference to the enterprise and its information systems is captured in the definition:

> "Enterprise architecture (EA) identifies the main components of the organization, its information systems, the ways in which these components work together in order to achieve defined business objectives, and the way in which the information systems support the business processes of the organization. The components include staff, business processes, technology, information, financial and other resources, etc. Enterprise architecting is the set of processes, tools, and structures necessary to implement an enterprise-wide coherent and consistent IT architecture for supporting the enterprise's business operations. It takes a holistic view of the enterprise's IT resources rather than an application-by-application view." (Kaisler et al., 2005).

Comprehensive architectures can be described only taking different viewpoints (Zachman 1987) that enable the consideration of an aspect in detail by excluding other aspects. All viewpoints together constitute the architecture. The enterprise architecture viewpoints that are generally chosen for EA methods are business, information, systems and technology (NIST/Fong and Goldfine 1989, Armour et al. 1999, Hasselbring 2000, The Open Group 2003, Perks and Beveridge 2003). These viewpoints are also found in several commercial EA methods (e.g. META Group 2002), and usually they are called Business Architecture (BA), Information Architecture (IA), Systems or Applications Architecture (SA) and Technology Architecture (TA).

Business architecture means the descriptions of the organization of the business: the business processes, structures and services. *Information architecture* depicts the information model and flows of information within the enterprise at a high abstraction level. To detect business potential of information, sometimes the concept of information value chain is used. *Systems architecture* stands for the portfolio of applications in use in an enterprise and the interconnections between the applications. These can be concretized in information systems architecture diagrams or integration architecture descriptions. *Technology architecture* consists of technologies including software, hardware and data communication connections..

The set of four architectural viewpoints meets both the needs of EA management, and the planning and development projects targeting the modification or enhancement of the current EA structures. EA planning has its roots in the concept of information systems planning (Olle et al. 1988) meaning a phase of comprehensive planning of the enterprise systems prior to a development project of a system. With today’s integrated and interdependent information processing infrastructures, an architectural approach is needed instead of considering isolated systems and their development schedules. From early on, the idea of different abstraction levels was found necessary in EA planning (Zachman 1987). Based on studies of practical EA work, three levels were found necessary in EA planning cases for different decision making levels in the enterprise (Hirvonen and Pulkkinen 2004, Pulkkinen and Hirvonen 2005a). The business decisions and the overall information management decisions come from the top managerial level and the strategy process concerning the whole enterprise, thus called the enterprise level. However, for more concrete practical planning, considering different domains within the enterprise is undertaken at the domain level. Since the EA development in practice can only be done incrementally, the enterprise is divided for planning and development purposes into domains with closely related information processing needs. A business unit can be a domain, or a cross-organizational business process (The Open Group 2003). Whereas at the *enterprise level*, EA principles and guidelines are outlined for the whole enterprise, at the *domain level*, practicable plans are made in the forms of business process definitions, information models, divisions into information storages, information systems deployment plans to technology infrastructures, integration architectures etc. At the third and most concrete level, the *systems level*, the plans are translated to systems design descriptions and further refined with systems development guidelines (Hirvonen and Pulkkinen 2004).

An example illustrates the treatment of a viewpoint, business architecture: at the enterprise level, the core business processes and the support processes are outlined as a coarse grained chart. The domain level description defines each process at the level of the business operations. The systems level description of a process guides the development of a supporting system for this process. The business architecture view at systems level means the business requirements for the systems to be developed. In one enterprise, the enterprise level architecture descriptions are binding for the whole enterprise. There are parallel domains, and the enterprise level EA work means also the coordination of the possibly concurrent planning and development efforts in different domains. Similarly, the systems level EA work means coordinating the development efforts with any system in the enterprise (Pulkkinen 2006).

A large number of frameworks, models, methods and processes have been suggested for the EA related work (Pulkkinen and Hirvonen 2005a). The earlier frameworks are consistent with the so-called waterfall model of information systems development, where the business needs come first, the logical structures are planned then and only at the final phases, the technological implementation takes shape. However, today the activities of any organization are to such an extent dependent on information systems, that the enabling IC technologies as well as the systems currently in use have to be considered already in the initial planning phases and together with the strategic business development plans. Thus, the *EA planning* is a demanding collaborative effort where the current state of the EA (i.e. the comprehensive ICT systems and infrastructure) and the business development visions are considered together. Among other business environment factors, the new technologies as possible business enablers are taken into account. For the domain and the systems level, the planning means materializing the strategic enterprise level decisions with concrete, practicable business and systems development plans. Often alternative plans are made, and planning projects deliver also evaluations of e.g. applicability and cost of different solutions and technologies, the business implications etc.

EA development may follow the planning effort, however, most commonly as a separate project after one of the alternative plans is chosen. In a development project, detailed architecture is designed for at least one domain, and preparations are made for system implementation or integration, meaning either launching a systems development project or an integration project.

Method Engineering

A *method* means generally a collection of approaches, beliefs, principles, models, techniques and procedures to carry out a development effort. It is some kind of carrier of collective knowledge and experience that are made 'visible' in order to enable its exploitation and advancement in forthcoming situations (Tolvanen 1998, Fitzgerald et al. 2002). Methods carry four kinds of knowledge (Leppänen 2005): knowledge of development process, knowledge of application domain, knowledge of IC technology, and knowledge of human and social issues. We can distinguish between four types of methods depending on how generic or specific is the knowledge on development they contain: generic methods, domain-specific methods, organization-specific methods, and project-specific methods. *Generic methods* provide general support for development endeavors in a wide range of contexts. *Domain-specific methods* offer more special support in particular domains. *Organization-specific* and *project-specific methods* provide customized support for a specific organization or project, correspondingly.

For software engineering (SE) and information systems development (ISD), a multitude of methods have been constructed since the 1960's. Recently, special methods have been engineered for particular domains such as BPR, ERP, CRM and EAP. Yet more methods are needed due to changes in business and its environment, application areas, and approaches and technologies of development environments. Constructing a new method is not a simple task. It comprises the elicitation and analysis of requirements for a new method, the evaluation of current method(s), and the design and implementation of a new method, as well as the assessment of the method under engineering in each of the stages. We use the term '*method engineering*' (ME) to stand for all those actions by which an ISD method is developed and later customized and configured to fit the needs of a domain, an organization or a project (Leppänen 2005).

We distinguish between three main strategies of ME: creation from scratch, integration and adaptation. *Creation from scratch* is applied when no existing methods is suitable. *Integration* implies that a new method is engineered through assembling parts, called method components (e.g. Gupta et al. 2001, Leppänen 2005) or method fragments (e.g. Brinkkemper et al. 1999, Ralyte et al. 2003), from other methods. *Adaptation* is deployed when some method is available for the customization for the needs of an organization or a project, by dropping off or modifying some parts of it, or extending it with new parts (Ralyte et al. 2003, Karlsson et al. 2004). ME approaches can be categorized, for instance, based on the degree to which practical and theoretical knowledge is applied (e.g. consultancy approach, method engineering approach and system development research in (Kaasboll & Smordahl 1996); deductive approach and inductive approach in (Fitzgerald et al. 2003)), or how urgently a new method is wanted into use (e.g. “typical” approach with pilot projects, “fast” approach with greater risks in Kruchten 2000).

The ME literature suggests a large variety of strategies and approaches (Kumar et al. 1992, Ralyte et al. 2003, Harmsen 1997), meta models (Heym et al. 1992, Kelly et al. 1996, Jarke et al. 1995), techniques (e.g. Kinnunen et al. 1996, Saeki 2003, Domingues et al. 2007) and procedures (e.g. Song 1997, Harmsen 1997, Tolvanen 1998, Leppänen 2005) for the engineering of ISD methods. Most of these suggestions are general-purpose and applicable to engineering methodical artifacts in other domains as well. Regardless of what strategy or approach is applied, the selection of ME artifacts and their customization are highly based on the recognition of and reasoning from essential features of the ME situation at hand, as well as the features of those contexts, called the target ISD contexts, for which a method is to be engineered. This is considered in more detail in the next section.

Contingency Approach

Contingency approach is based on the idea according to which there is no universal ‘solution pattern’ that would fit every problem situation. Instead, there is a need to select a pattern which has the best fit with the situation at hand. For this purpose, *contingency factors*, or situational factors, are used to characterize the situation and match them with the properties of the patterns. The contingency approach has been first applied in organizational design (e.g. Galbraith 1977, Pfeffer 1982, Kast et al. 1981, Drazin et al. 1985). In ISD, some of the first suggestions for contingency factors were presented by Naumann & Davis (1978), McFarlan (1981), Davies (1982), Iivari (1983) and Burns et al. (1985). Later, a large array of contingency frameworks (e.g. van Slooten et al. 1996, van Offenbeek et al. 1996, Kettinger et al. 1997, Fitzgerald et al. 1999, Hardgrave et al. 1999, Backlund 2002, Mirbel et al. 2006, Iivari et al. 2007) has been proposed to support the selection and construction of ISD approaches, models, techniques and methods. There are also different ways (e.g. Zhu 2002) and procedures (e.g. van Offenbeek et al. 1996) of applying the contingency approach. Regardless of some critics against the contingency approach (e.g. Lyytinen 1986, Avison et al. 1991, van Slooten et al. 1994, Avison 1996) it is widely applied in a number of fields.

The contingency factors in the ISD field are related to those aspects of an ISD project and its environment, which are seen important to making decisions on which kind of ISD approach is selected, which kinds of models are deployed, how the ISD process is structured, and how end users participate in the process. Typically factors include, for example, management commitment; size, and complexity of a project; availability, clarity and stability of goals; motivation, skills and experience of stakeholders, and business and technology innovativeness. We have constructed an integrative view of ISD contingency factors given in literature and present it in a structured form in Appendix 1. It goes beyond this paper to discuss them here.

The ME literature does not suggest any contingency frameworks for supporting decisions in method engineering. However, there are conceptual foundations with concept categories (e.g. Harmsen 1997, Leppänen 2005) and descriptions of ME projects in practice (e.g. Alderson et al. 1998, Fitzgerald et al. 2003, Karlsson et al. 2004) which can be used to elicit characterizing factors. In addition, based on the perceived analogy between ISD and ME (e.g. Olle et al. 1988, Kumar et al. 1992) we can derive ME contingency factors from those defined for ISD. Resulting from the latter, we can recognize generic contingency factors such as commitment of management, motivation, skills and experience of ME stakeholders, size and complexity of an ME effort etc. Of course, there are also factors that are specific to ME, such as level of generality and desired lifecycle of the method under engineering.

Research Framework and Process

This study aims to derive a preliminary set of contingency factors to be applied in the engineering of EA planning methods. These contingency factors refer to features of EA method engineering that have affects upon decisions on which kinds of ME approaches, ME principles, ME procedures and EA frameworks should be applied in a particular EA ME situation (shortly EA ME). In order to have a full support for these decisions, it is, naturally, necessary that ME artifacts are equally characterized with properties which can be matched with the contingency factors, and that there are empirical evidence on certain fits between the properties of ME artifacts and the contingency factors. The discussion of the properties of the ME artifacts and the 'fits' goes beyond this study.

Contingency factors can be derived in two manners, inductively and deductively. The former means that ME situations in practice are analyzed to find out which kinds of features of the situations are determinant in making decisions on approaches, principles, frameworks, and procedures. The latter implies that contingency factors are derived from existing literature. This literature covers proposed contingency frameworks, conceptual works and descriptive case studies. This study applies both of these manners, as shown in our design-theoretical research framework (cf. Hevner et al. 2004) in Figure 1. The contingency framework produced stands for a design artifact in the terminology of Hevner et al. (2002). It is composed of three layers. The topmost layer, shortly EAP ME, corresponds to a situation where a method for EA planning is engineered. The next lower layer (EAP) stands for a situation where an EA plan is produced. The lowest layer means the realization of an EA plan, or a part thereof. This study covers the topmost layer.

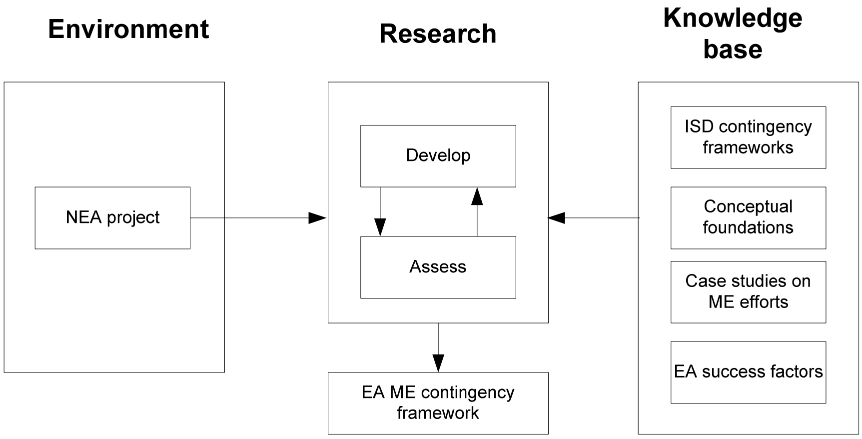


Figure 1. Research Framework

Knowledge base contains literature on ISD contingencies, conceptualizations of the ME and EA domains, descriptions of ME efforts in practice, and EA critical success factors. The research process has progressed as follows. Since the literature does not provide any contingency factors for EAP, not to mention for EAP ME, we paid attention to those ISD contingency factors that are generic enough to apply to EAP. The rationale behind this is that there is, on general level, an analogy between ISD and EAP situations. Second, we fleshed out the set of contingency factors with characteristics underlying suggestions for critical success factors for EA. Third, we derived from and compared to experience obtained in a large EA ME project, i.e., the Finnish *National Enterprise Architecture* (NEA) *project*. The project has involved a large number of representatives of state administration, consultancies and vendors. One of the subprojects of the NEA project has been aiming to engineer a generic EA method for state administration in Finland. Thus far, the first version of the EA method is under testing. One of the authors has participated as an active observer in workshops which evaluated existing methods and engineered a new one by integrating and adapting some of the selected methods. Also another subproject crafting the EA Capability Maturity Model (EACMM 2007) has affected the EACon framework constructed in this study. The more general mission of the NEA project is to advance the development of the public administration and its abilities to provide customer-oriented and flexible services and to increase the productivity of service production.

EACon framework - Contingency factors of EA Method Engineering

In this section, we first present a conceptualization of EA method engineering (EA ME) on a general level and then bring out potential contingency factors organized in accordance with this ontological structure (see Figure 2).

An *EA ME endeavor* means an effort to produce a new or an improved *EA method*. It can be organized as a project, or as a more or less non-structured action. It involves a single *enterprise* or a cluster of enterprises, which can be commercial or public organizations. A *cluster* is established around the endeavor, for instance, to create new networked services among them. The ME endeavor may deploy any variation of ME strategies and processes, e.g., construction of a generic EA method, EA method adaptation for a specific situation, situational EA method selection or modular method construction.

The EA ME endeavor is guided by *EA method goals* that are derived from requirements reflecting the views and needs of the members of the cluster. Reaching the goals requires the availability of *resources* including money, manpower, locations etc. The EA ME endeavor is constrained and directed by *EA principles* pertaining to architectural standards and decisions made in the organizations (Janssen and Hjort-Madsen 2007). Potential differences between local principles among members should be recognized and harmonized to yield a shared collection.

Some enterprises may have EA management of their own as a part of the strategy processes of the organizations. *EA management* targets the future strengthening and success of the enterprises (Pulkkinen and Hirvonen 2005b, Pulkkinen 2006). In the ME endeavor, the EA management of all the involved enterprises should be reconsidered in order to achieve a shared understanding of the future EA management. EA governance, in turn, is the annual process of maintenance (Jansen and Hjort-Madsen 2007) that supports the day-to-day operations and provide information for the strategic planning and management of the EA (Pulkkinen and Hirvonen 2005b).

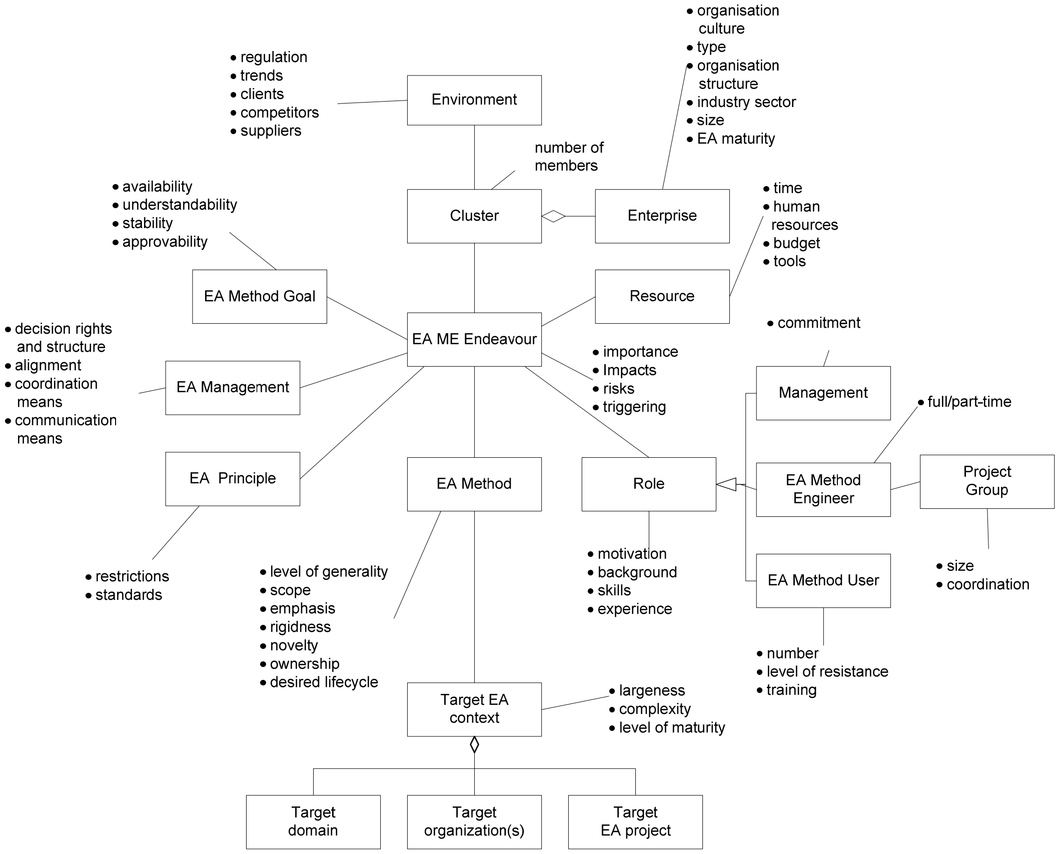


Figure 2. Contingency framework of enterprise architecture method engineering (EA ME).

An EA ME endeavor involves a number of persons in different *roles*. The typical roles in the EA ME are those of management, an EA method engineer and an EA method user. *Management* means here a role person who is in charge of making decisions on the launching the endeavor, allocating resources and approving the outcomes. Persons in this role may be representatives of top management, IT management or operational management. An *EA method engineer* stands for a role person who is engaged in selecting, analyzing, designing and implementing an EA method. The role can be played by an enterprise architect, a vendor or a consultant, each of them equipped with necessary expertise in business, information systems, ICT, ME tools, and/or EA methods. An *EA method user* means a role person who will deploy the EA method under engineering in forthcoming EA planning efforts. If engineering work is carried out in an organized form, a specific *project group* is established to include persons in the aforementioned roles. The EA method under engineering is targeted at some context, called the *target EA context* which is determined by a certain kind of domain, organizations, and EA planning projects, depending on whether the EA method is domain-specific, organization-specific or project-specific.

Next, we introduce contingency factors which can be used to characterize EA ME endeavors.

Characteristics of the enterprises in the cluster are organizational culture, type, organizational structure, industry sector, size, and EA maturity. *Organizational culture* is reflected by management structure and ways of making decisions (cf. Iivari and Huisman, 2007) and is manifested, for instance, as certain attitudes towards changes and ways of communication (Ylimäki 2006). *Type* means business, public administration or the third sector (Janssen & Hjort-Madsen 2007). *Organization structure* may, at its worst, reflect, e.g., silo thinking and strict profit responsibilities of an organization which can cause barriers to EA success (Ylimäki 2006). *EA maturity* means the awareness of, the attitudes toward, and the abilities to the management and development of EA in parallel to business management. EA maturity is seen, for instance, in how organization strategy and IT strategy are aligned (Hirvonen & Pulkkinen 2004, Ylimäki 2006).

EA management, and EA governance as a part of it, can be characterized by e.g. decision rights and structure (Janssen & Hjort-Madsen 2007), alignment (e.g. funding process), and communication and coordination means (Ylimäki 2006) *Decision rights and structure* includes the EA governance process, its roles, tasks, responsibilities and authorizations (Ylimäki 2006). From the EA ME viewpoint, the decision rights and structure tell who budgets the EA ME endeavor and who makes the final decisions about it. *Alignment* means the extent to which EA governance is integrated with Business Management Process, such as investment process or strategy refinement process (Ylimäki 2006). *Communication and coordination* are crucial for EA planning and governance (EACMM 2007), since the goal is to align large strategic entities in the enterprises, which requires intrinsic internal communication. Which kinds of communication channels are available, whether a common language including key concepts is defined (Ylimäki 2006, EACMM 2007), and whether coordination tools such as groupware and shared working places are in use affect on how an EA ME endeavor can be accomplished.

EA work is affected by many regulations and standards, which should be taken into account in engineering an EA method. *Regulation* include e.g. legislation, EU directives (IDABC 2004), national restrictions and agreements (e.g. Juhta 2007) and requirements for compatibility with other international or national models and instructions (EACMM 2007, Ylimäki 2006). *Restrictions* include, e.g., enterprise level architectural decisions, reference and model architectures to cope or follow (EACMM 2007). Reference architectures, for instance, may dictate how to view and act if interoperability between the target domain and that of the reference domains is demanded. *Standards* can be organizational or inter-organizational, and they can pertain to standard business processes, standard technical building blocks, or interfaces and interactions patterns (Janssen and Hjort-Madsen 2007).

The EA ME endeavors differ from each other in terms of their importance, impacts and risks, as well as why they are triggered. *Importance* means how far-reaching is it to have a new EA method. *Impacts* imply which (possibly) positive consequences result from using a new EA method. Every effort, including EA ME efforts, has some *risks* that should be recognized and taken into account (Ylimäki 2006). There are many kinds of reasons behind *triggering* an EA ME endeavor. There may be problems in EA planning and in the use of an EA method, and a new method is desired to solve these problems. Or a novel technology, a new application area, or a new approach to EA planning may require a new kind of methodical support.

Goals of the EA ME endeavors may diverge from one another in how *available*, *understandable*, *stable* and *approvable* they are, as well as what kinds of benefits are pursued. EA ME may appear to be an abstract entity for which it is difficult to set goals and express them in a way, which is understood by the stakeholders (cf. Ylimäki 2006). The larger the number of the involved stakeholders is, the more difficult it is to reach an agreement on the goals. Goals can be expressed in different ways, for instance, in terms of benefits to be reached or problems to be solved.

Success of any effort is dependent on the qualities of stakeholders participating in it. Attached to the notion of role we distinguish contingency factors such as *motivation*, *background*, *skills* and *experience*, related to EA, EA planning and EA method engineering. Particularly important is *commitment* of management on the EA ME endeavor. Also the degree of how intensively (i.e. full-time, part-time) stakeholders in their roles can contribute to the endeavor impact on the ways of working, organizing, controlling and coordinating EA ME. Resources are commonly measured in terms of *time*, *human resources* and money (*budget*). Other resources include *tools* and facilities, e.g. a shared dictionary, visualization techniques and tools (Janssen & Hjort-Madsen 2007),.

The EA method under engineering can be characterized by a level of generality, scope, emphasis, rigidness, novelty, ownership and desired lifecycle. A *level of generality* means the categorization into generic, domain-specific, organization-specific and project-specific EA methods. The *scope* and *emphasis* of the method can be expressed in terms of four architectural viewpoints (i.e. business, information, systems, and technology) and three levels (i.e. enterprise, domain, systems). *Rigidness* means how formally the models, techniques and process is to be specified in the EA method and how strictly they are to be followed in EA planning. *Novelty* refers to innovations expected to be in the EA method compared to existing methods. *Ownership* and *desired lifecycle* are important aspects in making decisions on the usage of ME resources.

Target EA context in Figure 2 is used to build a bridge to the EAP layer in the contingency framework. Factors related to it are general characterizations of aspects that are considered in more detail on the consequent EAP layer. Of the large variety of aspects we only mention three, largeness, complexity, and level of maturity. *Largeness* and *complexity* are qualities for characterizing how large and complex are EA planning efforts that are to be prescribed by the EA method under engineering. *Level of maturity* refers to the degree to which EA planning related work is expected to be familiar in the target contexts.

Summary and Conclusions

This paper has presented a contingency framework, called the EACon framework, which is composed of factors for the characterization of a situation in which an enterprise architecture method is engineered. To construct the framework we first defined the basic concepts for enterprise architecture, method engineering and contingency approach. Contingency factors were derived from literature on ISD contingency factors, conceptualizations of EA and ME, case studies on method engineering, and EA critical success factors. We also deployed the experience obtained from a large project of engineering an EA method for state administration. The contingency framework is composed of three layers, of which we focused on the topmost layer pertaining to an EA ME situation.

The framework can be applied to support the selection and adaptation of EA method engineering strategies, approaches, models, techniques and ways of organizing an EA ME effort in a situational manner, as well as to carry out the retrospective analysis of accomplished EA ME efforts. For instance, the situation in the NEA project can be characterized by the following features: high importance of EA work to the interoperability of information systems of state administration; a large set of stakeholders on various administration levels; a need to have such a general-purpose method for EA domain, which could be later customized into organization-specific and EA project-specific methods; the absence of a suitable EA planning method. These situational features, among other things, affected that the EA ME endeavor was organized in the form of a specific project which involves a large number of representatives of state administration, applies a mixed ME strategy of integration and adaptation, and uses pilot projects to refine the method. Experience got this far from the project suggests that most essential contingency factors to be considered in the customization of the EA method for an organization are industry sector and EA maturity of the enterprise, scope and emphasis of the EA method, and skills and experience of EA method engineer and EA method users.

The results of this study are preliminary and the subject of the future research. First of all, the definitions of the current contingency factors should be elaborated by literature reviews and a more detailed analysis of the experience from practice. Second, the contingency framework should be extended to cover the two other layers concerning an EA planning situation and an EA development situation. Third, in order to fully utilize the EACon framework it is necessary to specify basic properties of EA ME artifacts and 'fits' with the contingency factors.

Fourth, the framework should be equipped with concrete instructions for how to deploy it in EA ME efforts in practice. Fifth, empirical studies in different fields are needed to validate the EACon framework.

Acknowledgements

One of the authors has been funded by Finnish Enterprise Architecture Research (FEAR) project, which is warmly thanked for.

References

Alderson, A.., Hull, M. E., Jackson, K., and Griffiths L. E. (1998). "Method engineering for industrial real-time and embedded systems". *Information and Software Technology*, 40, 443-454.

Applegate, L. M. (1994). "Managing in an Information Age: Transforming the Organizations for the 1990's". In: R. Baskerville, S. Smithson, O. Ngwenyama and J.j. DeGross (eds.), *Transforming Organizations with Information Technology*. Elsevier Science B.V. / IFIP.

Armour, F. J., Kaisler, S. H. and Liu, S. (1999). "Building an Enterprise Architecture Step by Step". *IT Pro Magazine*, July /August, 31-39, IEEE.

Avison, D. (1996). "Information systems development methodologies: a broader perspective", In Brinkkemper J., Lyytinen K., Welke R. (eds.) *Principles of Method Construction and Tool Support.*

Avison D., and Wood-Harper, A. (1991). "Information systems development research: an exploration of ideas in practice", *The Computer Journal*, 34(2), 98-112.

Backlund, P. (2002). "Identifying situational factors for IS development processes: applying the method-in-action framework". *In Proc. of 8th Americas Conference on Information Systems.*

Brinkkemper, S., Saeki, M. and Harmsen, F. (1999). "Meta-modelling based assembly techniques for situational method engineering". *Information Systems*. Vol. 24, No. 3, 209-228.

Burns, R. N. and Dennis, A. R. (1985). "Selecting the appropriate application development methodology". *Data Base*, 17(1), 19-23.

Davis, G. B. (1982). "Strategies for information requirements determination". *IBM Systems Journal*, 21(1), 4-30.

Dominguez, E., Zapata, M. A. (2007). "Noesis: Towards a situational method engineering technique". *Information Systems* 32, 181-222.

Drazin, R. and van de Ven, B. (1985). “Alternative forms of fit in contingency theories”. *Administrative Science Quarterly*, Vol. 30, 514-539.

EACMM 2007. Arkkitehtuurikyvykkyyden kypsyystasomalli. *Enterprise Architecture Capability Maturity Model*. *Instructions.* Version 0.81. 19.3.2007. Working Paper in Finnish, Finnish National EA Project.

Fitzgerald, B. and Fitzgerald, G. (1999). “Categories and contexts of information systems developing: making sense of the mess”. *In Proc. of European Conference on Information Systems.*

Fitzgerald, B., Russo N. L. and O’Kane, T. (2003). “Software development method tailoring at Motorola”. *Comm. of the ACM*, 46(4), 65-70.

Fitzgerald, B., Russo, N. and Stolterman, E. (2002). *Information systems development – methods in action.* London: McGraw Hill.

Fong, E. N. and Goldfine, A. H. (1989). *Information Management Directions: The Integration Challenge.* Special Publication 500-167. National Institute for Standards and Technology (NIST).

Galbraith, J. (1977). *Designing complex organizations*. Addison-Wesley, Reading. MA.

Gupta, D. and Prakash, N. (2001). “Engineering methods from method requirements specifications”. *Requirements Engineering* 6(3), 135-160.

Hardgrave, B. C., Wilson, R. L. and Eastman, K. (1999). “Towards a contingency model for selecting an information system prototyping strategy”. *Journal of Management Information Systems*, 16(2), 113-136.

Harmsen, F. (1997). *Situational method engineering*. Dissertation Thesis, University of Twente, Moret Ernst & Young Management Consultants, The Netherlands.

Hasselbring, W. (2000). "Information System Integration." *Communications of the ACM* 43(6): 32-38.

Hirvonen, A. and Pulkkinen M. (2004). ”A Practical Approach to EA Planning and Development: the EA Management Grid”. *Business Information Systems - BIS*, Poznan, Poland.

Hevner, A., March, S., Park, J. and Ram, S. (2004). “Design science in information systems research”. *MIS Quarterly*, Vol. 28, No. 1, 75-105.

Heym, M. and Österle, H. (1992). ”A reference model for information systems development”. In K. Kendall, K. Lyytinen & J. DeGross (Eds.) *Proc. of Conf. on the Impacts on Computer Supported Technologies on Information Systems Development*. Amsterdam: Elsevier Science, 215-240.

IDABC (2004). *European Interoperability Framework for pan-European eGovernment Services, version 1.0* [online]. <http://europa.eu.int/idabc/en/document/3473/5585>.

Iivari, J. (1983). *Contributions to the theoretical foundations of systemeering research and the PIOCO-model*, Dissertation Thesis, Acta Universitatis Ouluensis, Ser A 150, Institute of Data Processing, University of Oulu, Finland.

Iivari, J. and Huisman, M. (2007). "The relationship between organizational culture and the deployment of systems development methodologies". *MIS Quarterly* 31(1), 35-58.

Janssen, M. and Hjort-Madsen, K. (2007). "Analyzing Enterprise Architecture in National Governments: The cases of Denmark and the Netherlands". *Proceedings of the 40th Hawaii International Conference on System Sciences*, HICSS 40, Jan 3-6, Waikoloa, Big Island, Hawaii.

Jarke, M., Gallensdörfer, R., Jeusfeld, M., Staudt, M. and Eherer, S. (1995). "ConceptBase: a deductive object base for meta data management". *Journal of Intelligent Information Systems*, Vol. 4, No. 2, 167-192.

Juhta (2007). "JHS 145 Palvelutietojen ryhmittely ja osoitteet asiointia varten monta toimialaa kattavissa julkisen sektorin portaaleissa" [online]. *Public Administration ICT Recommendations in Finland (in finnish), Juhta* <http://www.jhs-suositukset.fi/suomi/jhs145>

Kaasboll, J. and Smordal, O. (1996). "Human work as context for development of object-oriented modeling technique". In S. Brinkkemper, K. Lyytinen & R. Welke R. (Eds.) Proc. of the IFIP TC8 WG 8.1/8.2 *Working Conf. on Method Engineering: Principles of Method Construction and Tool Support*. London: Chapman & Hall, 111-125.

Kaisler, S. H., F. J. Armour, et al. (2005). "Enterprise Architecting: Critical Problems". *Proceedings of the 38th Hawaii International Conference on System Sciences*, IEEE Computer Society.

Karlsson, F. and Ågerfalk, P. J. (2004). "Method configuration: adapting to situational characteristics while creating reusable assets". *Information and Software Technology*, 46, 619-633.

Kast, F. and Rosenzweig, A. (1981). *Organization and management: a system and contingency approach (3rd ed)*, McGraw-Hill, New York.

Kelly, S., Lyytinen, K. and Rossi, M. (1996). "MetaEdit+: a fully configurable multi-user and multi-tool CASE and CAME environment". In Y. Vassiliou & J. Mylopoulos (Eds.) *Proc. of the 8th Conf. on Advanced Information Systems Engineering (CAiSE'96).* Berlin: Springer, 1-21.

Kumar, K., and Welke, R. (1992). “Methodology engineering: a proposal for situation specific methodology construction”. In W. Kottermann & J. Senn (Eds.) *Challenges and Strategies for Research in Systems Development.* Chichester: John Wiley & Sons, 257-269.

Kettinger, W., Teng, J. and Guha, S. (1997). „Business process change: a study of methodologies, techniques, and tools”. *MIS Quarterly*, Vol. 21, No. 1, 55-80.

Kinnunen, K. and Leppänen, M. (1996). ”O/A matrix and a technique for methodology engineering”. *Journal of Systems and Software* 33(2), 141-152.

Kruchten, P. (2000). *The Rational Unified Process: An introduction*. Reading: Addison-Wesley.

Leppänen, M. (2005). *An Ontological Framework and a Methodical Skeleton for Method Engineering*, Dissertation Thesis. Jyväskylä Studies in Computing 52, University of Jyväskylä, Finland.

Lyytinen, K. (1986). *Information systems development as social action: framework and critical implications*. Dissertation thesis, Jyväskylä Studies in Computer Science, Economics and Statistics, University of Jyväskylä, Finland.

McFarlan, F. W. (1981). “Portfolio approach to information systems”, *Harwards Business Review*, 59(5), 142-150.

METAGroup (2002). *Enterprise Architecture Desk Reference*, Meta Group Inc. National Association of State Chief Information Officers (Nascio). Enterprise Architecture Maturity Model. [cf. http://www.nascio.org]

Mirbel, I. and Ralyte, J. (2006). “Situational method engineering: combining assembly-bsaed and roadmap-driven approaches”. *Requirements Engineering*, 11, 58-78.

Naumann, J. D. and Davis, G. B. (1978). “A contingency theory to select an information requirements determination methodology”, *In Proc. of the Second Software Life Cycle Management Workshop*, IEEE, New York, 63-65

Offenbeek van, M. A. and Koopman, P. L. (1996). “Scenarios for system development: matching context and strategy”. *Behaviour & Information Technology*, 15(4), 250-265.

Olle, T., Hagelstein, J., MacDonald, I., Rolland, C., Sol, H., van Assche, F. and Verrijn-Stuart, A. (1988). *Information Systems Methodologies – A Framework for Understanding*. 2nd edition. Reading: Addison-Wesley.

The Open Group 2003. *The Open Group Architecture Framework (TOGAF) Version 8.1* "Enterprise Edition". [online] on <http://www.opengroup.org/togaf>.

O'Rourke, C., N., Fishman, et al. (2003). *Enterprise Architecture - Using the Zachman Framework*. Canada, Thomson Course Technology.

Perks, C. and Beveridge T. (2003). *Guide to Enterprise IT Architecture.* New York, Springer.

Pfeffer, M. (1982). *Organizations and organizations theory.* Ballinger Publ., Cambridge, Mass.

Pulkkinen, M. (2006). “Systemic Management of Enterprise Architecture Decisions. Four Dimensions and Three Levels”. *Proceedings of the 39th Hawaii International Conference on System Sciences*, IEEE Computer Society.

Pulkkinen, M. and Hirvonen, A. (2005a). ”EA Planning, Development and Management Process for Agile Enterprise Development”. *Proceedings of the 38th Hawaii International Conference on System Sciences*, IEEE Computer Society.

Pulkkinen, M. and Hirvonen, A. P. (2005b). ”Organisational Organizational Processes in ICT Management and Evaluation. Experiences with Large Organizations” In: Remenyi, D. (ed.): *Proceeding of the 12th European Conference on Information Technology Evaluation* (ECITE). Academic Conferences Limited 2005.

Ralyte, J., Deneckere, R. and Rolland, C. (2003). “Towards a generic model for situational method engineering”. In J. Eder & M. Missikoff (Eds.) *Proc. of the 15th Int. Conf. on Advanced Information Systems Engineering (CAiSE’03).* LNCS 2681, Berlin: Springer-Verlag, 95-110.

Saeki, M. (2003). “Embedding metrics into information systems development methods: an application of method engineering technique”. In Eder J., Missikiff M. (Eds.) *Proc. of the 15th Intern. Conf. on Advanced Information Systems Engineering (CAiSE 2003),* LNCS 2681, Springer-Verlag: Berlin, 374-389.

Schekkerman J. (2003). *How to survive in the jungle of Enterprise Architecture Frameworks.* Creating or choosing an Enterprise Architecture Framework. Canada: Trafford.

Slooten van, K., Brinkkemper, S. and Hoving, P. (1994). *Contingency-based situational systems development in large organizations.* Managing Social and Economic Change with Information Technology. Idea Group Publishing, Harrisburg.

Slooten van, K. and Hodes, B. (1996). “Characterizing IS development projects”. In: Brinkkemper S., Lyytinen K., Welke R. (eds) *Proc. of the IFIP WG 8.1 Conf. on Method Engineering – Principles of Method Construction and Tool Support*, Chapman & Hall, London, 29-44.

Song, X. (1997). “Systematic integration of design methods”. *IEEE Software,* Vol. 14, No. 2, 107-117.

Tolvanen, J.-P. (1998).. *Incremental method engineering with modeling tools – Theoretical principles and empirical evidence*. Dissertation Thesis, Jyväskylä Studies in Computer Science, Economics and Statistics, No. 47, University of Jyväskylä, Finland

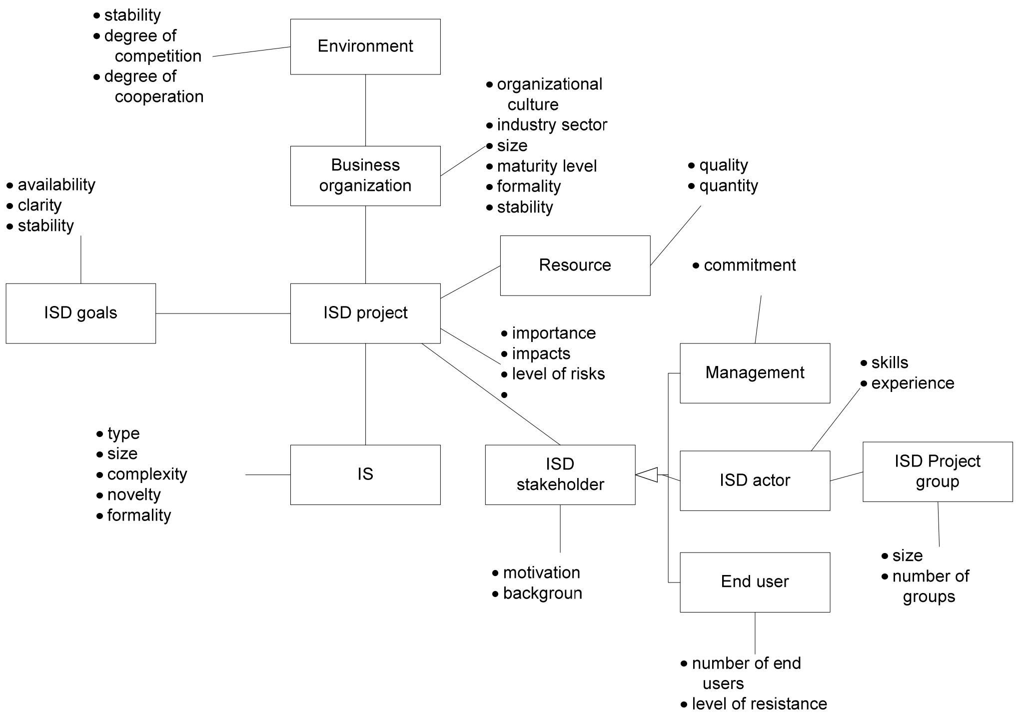
Whitman, L., K. and Ramachandran, et al. (2001). "A Taxonomy of a Living Model of the Enterprise". *Proceedings of the Winter Simulation Conference*, IEEE.

Ylimäki, T. (2006). "Potential critical success factors for enterprise architecture". *Journal of Enterprise Architecture* 2(4), 29-40.

Zachman, J. A. 1987. "A Framework for Information Systems Architecture." *IBM Systems Journal* 26(3).

Zhu, Z. (2002). "Evaluating contingency approaches to information systems design". *International Journal of Information Management*, 2.

Appendix 1



ISD contingency factors