Enterprise Architecture for Decentralized Environments

Thomas Speckert

Department of Computer and Systems Sciences

Degree project 30 HE credits

Degree subject: Computer and Systems Sciences

Degree project at the Master level

Spring 2013

Advisor: Jelena Zdravkovic

External advisor: Irina Rychkova

Reviewer: Janis Stirna

Swedish title: Företagsarkitektur för Decentraliserade Miljöer

Stockholm University

Enterprise Architecture for Decentralized Environments

Thomas Speckert

Abstract

The problem of business-IT alignment is of vital significance for modern enterprises. Solving it allows all components of an enterprise to operate together in a collaborative manner for the purpose of maximizing overall benefit to the enterprise. Enterprise Architecture (EA) is a discipline that aims to solve this problem in a holistic manner from the ground up through proper design.

Decentralization of organizations and subsequent change of their management and operation style requires major changes in organization processes and heavily involves the IT. This thesis project uses a combination of a literature study and a case study to demonstrate, however, that EA is primarily aimed at centralized organizational structures, and as such has some shortcomings when being applied to decentralized organizations. Overcoming these deficiencies requires some new principles to be introduced and incorporated into EA knowledge. Relevant sources for these new principles are decentralized organizations, peer-to-peer technologies, and organizational science. All these areas have tackled the problem of decentralization in certain ways. This thesis project presents prevalent decentralization principles and applies them to EA in order to create a prototype for one part of an EA framework that is fully supportive of decentralization. This prototype is shown to be viable through the use of a case study.

Keywords

enterprise architecture, EA, decentralization, peer-to-peer

Contents

1. Introduction

1.1 Background

Large enterprises have traditionally implemented formal, centralized forms of organizational structure [1], such as hierarchical or matrix structures. In these structures, communication patterns, roles and decision rights are strictly defined. This allows for management to have a high degree of control over the enterprise and therefore enforce compliance with standards, procedures and policies which results in a highly stable enterprise. However, this comes at the expense of agility; it is difficult for these organizations to quickly adapt to a changing environment. While centralized structures were appropriate for the business environments of the past, modern business environments demand a high level of agility.

Common components of modern business environments include cooperation with different organizations, rapidly changing business activities and processes, and a rapidly changing competitive landscape. In order to properly handle these components, a high level of enterprise agility is necessary. In centralized organizations, decisions need to be discussed at all levels of the hierarchy in order to obtain the appropriate justification and approval. This takes time; by the time a decision is made, it is often too late for it to be effective. In contrast, having decision making on the operational level allows for quick decisions that enables an organization to take advantage of opportunities quickly. More decentralized structures, such as networked organizations [1], are examples of this. It is important to note that a lack of rigidity and formal structure does not mean a lack of organization. It is still important for a decentralized enterprise to maintain order in its activities; this organization just needs to be based on an underlying decentralized structure instead of centralized one. Consequently, decentralized organizations need solutions to the same problems faced by centralized organizations – such as business-IT alignment – but the solutions need to be supportive of decentralization over centralization.

Decentralization and agility in organizations needs to be supported by IS. IS architectures exist that can support decentralization, for example Service-Oriented Architecture (SOA), but organizations need to use them to implement IS that is supportive of the organization. This issue of alignment between business and IT is one that is addressed by the practice of Enterprise Architecture (EA).

Enterprise Architecture is a practice for creating an architecture for an entire enterprise or organization. EA takes a holistic view of an enterprise in order to bring its many components – such as goals, strategies, information systems, processes, and governance styles – into alignment with each other. Many different EA frameworks currently exist, for example The Open Group Architecture Framework (TOGAF) [2], the Zachman Framework [3] and Federal Enterprise Architecture (FEA) [4].

All frameworks address one or more of the following three different aspects: artifacts that describe an enterprise's architecture, the process of creating these artifacts, and a way to ensure that the architecture implementation is an ongoing success. In this thesis, these three elements are respectively referred to as the EA description, EA method, and EA engine (Fig.1.1).

By creating an architecture for all components of an enterprise, EA is a solution to the problem of business-IT alignment. Therefore, in this thesis we analyze modern

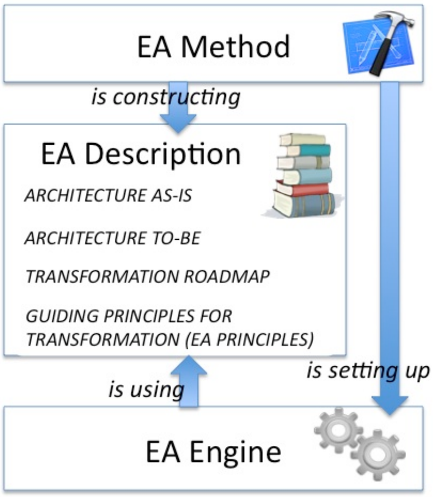


Figure 1.1: Enterprise Architecture

EA frameworks for their support of decentralization in organizations. Support for decentralization in EA frameworks would allow for EA to be a solution for business-IT alignment in decentralized organizations.

1.2 Problem

The problem of business-IT alignment is a relevant problem for all types of organizations, regardless of whether they are centralized or decentralized. Solving it allows all components of an enterprise to operate together in a collaborative manner for the purpose of maximizing overall benefit to the enterprise. Enterprise Architecture frameworks seek to outline a formal way in which to solve this problem. This thesis argues, however, that modern EA frameworks are designed for centralized organizations, and as such, might not be suitable for application in decentralized organizations. As modern organizations are becoming increasingly more decentralized, this suitability of modern EA frameworks for decentralization is becoming increasingly relevant. Consequently, this thesis addresses the problem of creating a suitable EA for decentralized organizations.

An important part of this thesis project is to analyze the suitability of existing EA frameworks for decentralization. Therefore, the problem will be fully explicated in the results chapter, Section 4.1.

1.3 Goals and Purpose

The purpose of this thesis project is to contribute to the field of EA by improving it's support for decentralization. This is supported by two primary goals: The first goal is to demonstrate specific aspects of existing EA frameworks that are supportive and not supportive of decentralized organizations. The second goal is to contribute towards an EA framework supporting decentralization by design an artifact that overcomes a subset of the identified shortcomings.

1.4 Research Questions

In order to adequately solve the problem presented in section 4.1, this thesis will answer two separate research questions. The first question will address the problem of whether or not current Enterprise Architecture methods and frameworks are suitable for use in decentralized environments. The second question addresses the problem of creating an EA framework that is suitable for decentralized environments. Answering these two research questions will enable this thesis to make a contribution the field of EA by providing a basis for how EA can be extended to an area (i.e. decentralized organizations) where, currently, it is not typically applied.

The two research questions for this thesis are:

1. What aspects of existing EA frameworks are supportive of decentralized organizations? What aspects are not supportive?

2. What are the principals of an EA framework that is supportive of decentralized organizations?

1.5 Limitations

Due to time constraints, this thesis project is not able to address the entirety of the problem identified. Instead, only a subset of identified issues with EA will be addressed.

1.6 Chapter Layout

This paper is structured as follows:

Chapter 2 - Extended Background An extended background on decentralization in organizations and on three modern EA frameworks is presented here. Additionally, the related subjects of Enterprise Integration and Enterprise modeling are covered as related works.

Chapter 3 - Method This chapter contains: a justification for following design science; the chosen research methods and strategies along with reasons for these choices and a discussion of alternatives; how these methods and strategies are actually applied for this research; and lastly ethical considerations are covered.

Chapter 4 - Results This chapter contains: relevant aspects of EA frameworks to decentralization and whether they are supportive of it or not; relevant aspects of the case's EA to decentralization and whether they are supportive of it or not; requirements for a decentralized EA framework; requirements for an EA for the case; a prototype of an artifact of a partial EA framework; and an illustrative demonstration of this artifact in the case.

Chapter 5 - Conclusions & Discussions This chapter contains a discussion of the results and recommendations for future works.

2. Extended Background

2.1 Enterprise Architecture

According to Sessions [5], the field of Enterprise Architecture (EA) emerged in order to combat two increasingly prevalent problems facing enterprises: system complexity and business-IT alignment. As enterprises rely more and more on information systems of increasing complexity, these problems become even more important. The field of EA views the solution to these problems to be one of concurrent design. It is not enough simply try and fit IT to the business; business and IT aspects should be designed concurrently.

While there is no singular agreed-upon definition for EA, different definitions [1,2,5–9] do have much in common. EA is a discipline that takes a holistic approach to transforming high-level business vision and goals into the integration of an enterprise's organizational structure, business processes, and information systems. This transformation involves identifying and implementing the necessary change for this to occur. In order to view different EA frameworks from a common perspective, this thesis will break them down into three separate components that contribute to this transformation: the EA method, the EA description, and the EA engine.

The Method aims to lay the groundwork for the EA project. Typically, this involves setting up teams, ownership, responsibilities and gaining commitment. Also it defines the overall process of collecting, validating and approving the EA artifacts (e.g. descriptions As-Is, To-Be, gap analysis, principles) that will form the second component - The EA description. The Engine involves setting up a support structure for ensuring the ongoing adoption of the to-be EA description. This can involve gaining commitment fr om stakeholders, setting up some compliance checking procedures, and deciding upon a prioritization of tasks to be completed. The remainder of this section will look at three different EA frameworks from the perspective of of these three phases: The Open Group Architecture Framework (TOGAF), the Zachman Framework, and the Federal Enterprise Architecture (FEA).

2.1.1 TOGAF

The Open Group Architecture Framework, more commonly know as TOGAF, is a freely available EA framework created by The Open Group[1], a consortium of IT organizations. TOGAF is comprised of a number of different aspects, mainly: the Architecture Development Method (ADM), "a method for developing and managing the lifecycle of an enterprise architecture" [2, Ch. 5.1]; the Architecture Content Framework, a companion to the ADM which describes the content of the products of the ADM; the Architecture Landscape and Enterprise Continuum, which provide a means to organize the produced architectures; a set of reference models; and the Architecture Capability Framework, a set of "reference materials" for successfully operating an "architecture function" within an enterprise [2, Ch. 45].

[1] `www.opengroup.org`

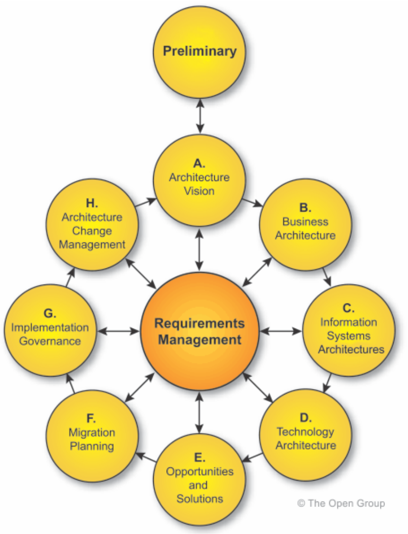

Preliminary
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management
© The Open Group

Figure 2.1: TOGAF: ADM basic structure from [2, Sec. 5.2.2]

EA Method

The TOGAF ADM falls under our EA Method component of EA. The TOGAF ADM is made up of a preliminary phase, six core phases (labeled A-H), and a requirements management component.

In TOGAF, the preliminary phase lays the groundwork for the rest of the EA process. Some important aspects are to set up a governance structure and EA team for the EA process and to establish a repository for storing all architectural information [2, Ch. 6].

Phase A of the TOGAF Process, the architectural vision phase, is aimed at setting a clear vision for the enterprises future architecture. This involves creating the initial as-is architecture as well as setting clear, management approved goals and requirements, and transforming them into a high-level vision of the enterprises to-be architecture [2, Ch. 7].

At this point, TOGAF suggests that the outputs of the preliminary phase and phase A be organized into a “Statement of Architecture Work”. This document is to be approved by project sponsors [2, Sec. 7.4.11] and can be used to form the basis of a contract between the architecture provider and the client [2, Sec. 36.2.20].

The next three phases, B-D are concerned with creating the as-is and to-be business architecture, information systems architecture, and the technology architecture [2, Ch.8-12]. TOGAF suggests two different approaches to creating the architectures: baseline first or target first [2, Ch. 19.4]. Baseline-first involves analyzing the as-is architecture for areas where improvements can be made. Target-first aims at creating a detailed target architecture and then mapping it back to the as-is architecture in order figure out what needs to change. The main aspects of these phases are to develop the as-is and to-be architectures, analyze the gap between them, and create an initial road-map of the steps needed to cross the gap.

Phase E and F, Opportunities and Solutions and Migration Planning, are concerned with organizing the work to be done into projects, and then creating a schedule for executing the projects [2, Ch. 13-14].

Phase G is concerned with the implementation and setting up a framework for its governance and its compliance to the target architecture [2, Ch. 15].

EA Description

TOGAF views architecture from the perspective of four different architecture domains [5]: business, application, data, and technical. Business architecture is concerned with processes and functions used to meet business goals, application architecture is concerned with the design of specific applications and their interactions, data architecture is concerned with managing enterprise data, and the technical architecture is concerned with the infrastructure (hardware and software) used to support the applications. The architectures in these four domains are created through the ADM phases B (Business Architecture Phase), C (Information Systems Architectures Phase) and D (Technology Architecture).

Architecture Landscape

The various architectural artifacts in TOGAF are organized across an Architectural Landscape [2, Ch. 20] of three dimensions: breadth, level, and time. Breadth refers to the area of subject matter for an architecture. Levels refer to the level of detail of an architecture. TOGAF specifies three levels of detail: strategic, for overall direction setting at the executive level; segment, for architectures at the level of a program or portfolio; and capability, for architectures concerned with how the architecture process is itself enabled and governed. The time dimension of the landscape keeps the state of architectures as they evolve over time. Additionally, the Architecture Landscape can be partitioned into independent partitions for supporting different organizational units [2, Ch. 40].

Enterprise Continuum

At each level of the Architecture Landscape, architectures are further organized through the Enterprise Continuum which provides a way to organize the architectures from generic to organization-specific [2, Ch. 39]. The most generic are called Foundation Architectures [2, Ch. 39.4.1], which are applicable to all enterprises. A core aspect of a Foundation Architecture is to provide a high-level taxonomy which can provide a basis for the more specific architectures [2, Ch. 43].

The second set of architectures in the continuum are called the Common Systems Architectures [2, Ch. 39.4.1]. These architectures are specific to a generic problem domain (e.g. security management), and are thus applicable to a wide range (but not all) of enterprises.

The third set of architectures in the continuum are called Industry Architectures. These architectures are applicable to a specific problem within a specific industry. They are thus useful to many members of that industry, but not necessarily outside of it. The most specific level in the continuum are Organization-Specific architectures. As the name implies, they are relevant only to a specific enterprise. These outline the architectural solution for a particular enterprise and provide "a means to communicate and manage business operations across all four architectural domains" [2, Ch. 39.4.1].

Reference Models

TOGAF includes two reference models. The Technical Reference Model (TRM) is a Foundation Architecture which describes a number "of generic services and functions that provides a foundation on which more specific architectures and architectural components can be built" [2, Sec. 43.1.1]. The second provided reference model, the Integrated Information Infrastructure Reference Model (III-RM), is a subset of the TRM and is

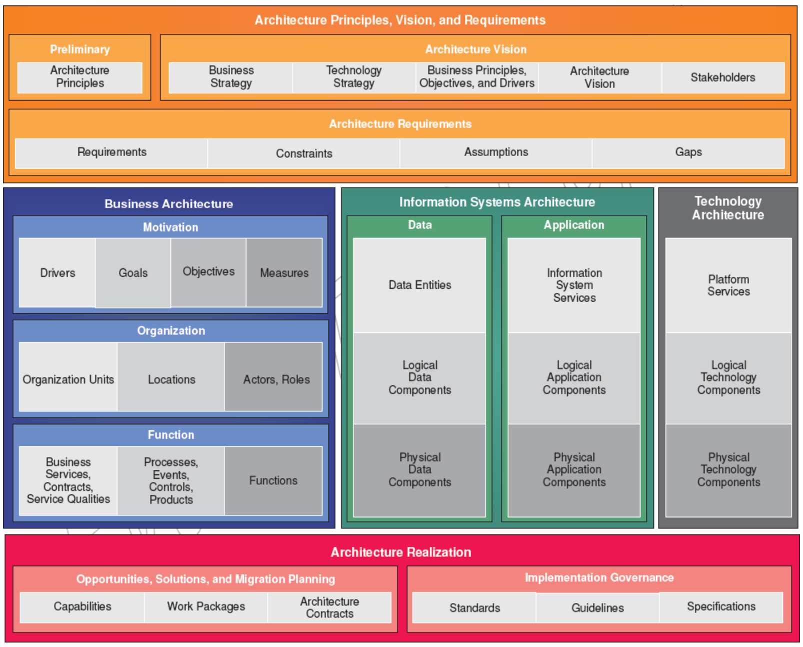


Figure 2.2: TOGAF: Content Metamodel from [2, Sec. 34.2]

classified as a Common System Architecture on the Enterprise Continuum [2, Ch. 44]. The III-RM is a reference model for enabling information integration across an enterprise.

Architecture Content Framework

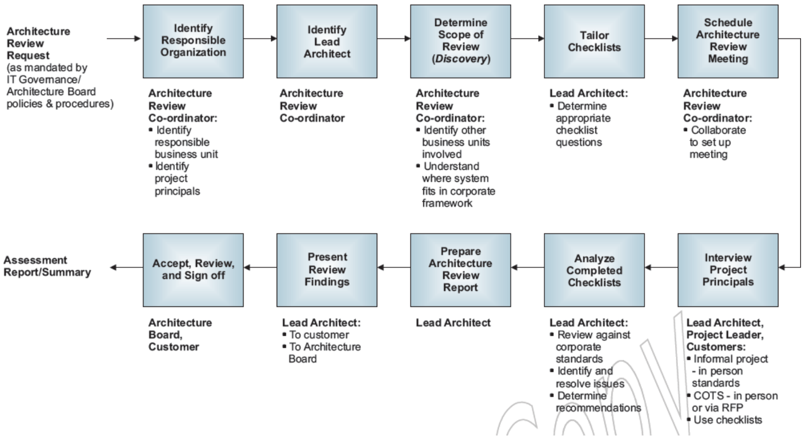
The Architecture Content Framework (ACF) describes the outputs of the architecture efforts from the TOGAF's method, the ADM. The content of the outputs is described by the Content Metamodel [2, Ch. 34], which is summarized in Figure 2.2. The outputs can are seen from three different viewpoints [10, Ch. 4] described by the content framework: deliverables, artifacts, and building blocks [2, Ch. 33]. A *deliverable* is an output that is specified in the architecture contract, and is therefore subject to formal review and sign-off. A deliverable is composed of one or more *artifacts* which describe some aspect of the architecture. Artifacts, in turn, describe *building blocks*, which represent either: a) a specific functionality of the enterprise, or b) an actual component which implements a specified functionality.

EA Engine

The engine component TOGAF is composed of an on-going change management process and a framework for managing architecture capability in an enterprise.

Architecture Change Management

The final ADM phase, phase H, outlines an ongoing change management process for the architecture of an enterprise. It is concerned with managing changes to the architecture throughout its lifecycle [2, Ch. 16]. In this phase, a governance body sets criteria for determining if a change requires an architecture update if a new cycle of the ADM needs to be started. An important aspect of this process is to deploy tools for monitoring for business and technological changes and measuring performance indicators.



Architecture Review Request (as mandated by IT Governance/ Architecture Board policies & procedures)
Identify Responsible Organization
Architecture Review Co-ordinator: ▪ Identify responsible business unit ▪ Identify project principals
Identify Lead Architect
Architecture Review Co-ordinator
Determine Scope of Review (*Discovery*)
Architecture Review Co-ordinator: ▪ Identify other business units involved ▪ Understand where system fits in corporate framework
Tailor Checklists
Lead Architect: ▪ Determine appropriate checklist questions
Schedule Architecture Review Meeting
Architecture Review Co-ordinator: ▪ Collaborate to set up meeting
Interview Project Principals
Lead Architect, Project Leader, Customers: ▪ Informal project - in person standards ▪ COTS - in person or via RFP ▪ Use checklists
Analyze Completed Checklists
Lead Architect: ▪ Review against corporate standards ▪ Identify and resolve issues ▪ Determine recommendations
Prepare Architecture Review Report
Lead Architect
Present Review Findings
Lead Architect: ▪ To customer ▪ To Architecture Board
Accept, Review, and Sign off
Architecture Board, Customer
Assessment Report/Summary

Figure 2.3: TOGAF: Architecture Compliance Review Process from [2, Sec. 48.4.1]

Architecture Capability Framework

TOGAF describes its Architecture Capability Framework as providing a set of reference materials for creating appropriate “organization structures, processes, roles, responsibilities, and skills” in order to “successfully operate an architecture function” [2, Sec. 45.1]. Key components of this framework are the creating of an Architecture Board, a formal architecture compliance process, the use of architecture contracts, and guidelines for architecture governance.

The Architecture Board is responsible for overseeing architecture governance activities. Specific responsibilities include “[p]roviding the basis for all decision-making with regard to the architectures”, ensuring the various partitioned architectures are consistent, and enforcing compliance [2, Ch. 47].

TOGAF describes a formal review process for determining compliance. The main goal of this process is to “[f]irst and foremost, catch errors in the project architecture early, and thereby reduce the cost and risk of changes required later in the lifecycle” [2, Ch. 48.3.1]. This process is outlined in Figure 2.3.

TOGAF suggests the use of contracts between architecture developers and sponsors (i.e. the party who wants the architecture) in order to have a formal agreement on the “deliverables, quality, and fitness-for-purpose” [2, Ch. 49]. The goal of having contracts is to guarantee: continuous monitoring of architecture decisions, changes, and integrity; adherence to architecture “principles, standards and requirements”; that all risks are identified; the existence of processes to ensure “ensure accountability, responsibility, and discipline”; and a formal definition of governance and decision-making authority. Adherence to the contracts is guaranteed by architecture governance.

In TOGAF, “architecture governance is an approach, a series of processes, a cultural orientation, and set of owned responsibilities that ensure the integrity and effectiveness of the organization’s architectures” [2, Ch. 50].The suggested organizational structure for governance involves three separate phases–Develop, Implement, and Deploy–supported by the Enterprise Continuum, and overseen by a CIO/CTO. This is visualized in Figure [**?**]. In order for the architectural effort to be successful, the architecture governance strategy needs to specify an Architecture Board, a set of architecture principles, and ensure compliance.

Figure 2.4: TOGAF: Architecture Governance Framework – Organizational Structure from [2, Sec. 50.2.2.1]

2.1.2 The Zachman Framework

The Zachman Framework was the first EA, first introduced by John Zachman in 1987 [3, 5]. It consists only of a taxonomy, and as such only fits into the EA Description aspect of EA.

EA Method and Engine

Despite not specifying an engine or method for use with ZF, these components are still necessary for a successful EA effort. ZF can be used however an organization wishes, though some formal guidelines do exist for its use. One option is to fit ZF into another EA framework, replacing that framework's EA description with ZF. TOGAF, for example, states that it is possible to replace the TOGAF content framework with ZF [2, Ch. 33]. A second option is pay for formal training from Zachman International [11], where it is possible to get certified in the use of ZF. A third option is through hiring a consultant who is already familiar with ZF. Zachman International, for example, offers consulting services.

EA Description

The Zachman Framework (ZF) breaks down EA into a grid of perspectives. Each perspective is characterized by two things; its target audience and the issue is aimed at. ZF covers six issues: What (data and entities), How (functional), Where (locations and interconnections/networks), Who (people relationships), When (events and performance criteria), Why (motivations and goals) [6]. For each issue, it views it from six different perspectives: executive, business management, architect, engineer, technician, and enterprise users.

The executive perspective is meant for executives or planners and needs to provide an estimate of a system's functionality and cost [6]. The business management perspective is a business view of how an owner thinks the business operates [12]. The architect perspective takes a systems viewpoint and describes the operations and interactions of the variety of systems in an enterprise. The engineer perspective views describes the physical technology and design of the individual systems. The technician perspective takes the perspective of a "sub-contractor" who is implementing a specific system and the high, out-of-context level of detail associated with that. The enterprise users perspective describes the perspective of the people who actually use the system.

2.1.3 Federal Enterprise Architecture

The Federal Enterprise Architecture (FEA) is an effort by the federal government of the United States to create an EA for the entire government. The FEA is a complete EA framework, covering all three components of EA. The Federal Enterprise Architecture Program Management Office describes FEA as "...a common language and framework to describe and analyze IT investments, enhance collaboration and ultimately transform the Federal government into a citizen-centered, results-oriented, and market-based organization as set forth in the President's Management Agenda." [4] FEA takes takes an approach where individual organizational units develop their own architectures that fit into an overall framework of common standards and interoperability.

FEA is composed of six core elements [5]:

- The organization is broken-down into different segments of varying scopes, and architecture is developed for each segment
- A set of five reference models which are used as a basis to describe the important elements of the FEA in a consistent manner
- A process for creating each segment EA
- A transitional process for moving from the current state of the enterprise to the visioned state
- A taxonomy for organizing the various assets of the FEA
- Guidelines for measuring the degree of success of the FEA

Compared to TOGAF and Zachman Framework, FEA defines both the taxonomy for EA artifacts (EA description in Fig. 1.1) and the EA process for creating these artifacts and using them by organization (EA method and EA engine in Fig.1.1).

EA Description

FEA develops architecture for segments and enterprise services. A segment is a "major line-of-business functionality" [5] for an individual organizational unit (such as an agency or department). Two types of segments exist: core mission-area segments and business service segments [4]. Core mission-area segments are at the scope of a single organizational unit (though they may be shared by different units) and are essential to its purpose [4, 5]. Business service segments are also at the scope of an individual organizational unit, however these segments exist in all organizational units and are defined for the entire enterprise. Like business service segments, enterprise services are defined organization-wide. However, they are different in that they also function at the enterprise level, e.g. a single security management service that is shared by the entire enterprise.

The EA artifacts defined by FEA include baseline segment architectures, target segment architectures and transition strategy. The EA transition strategy describes the overall plan and schedule to achieve the target ("to-be") architecture.

In order to have a common language for describing the enterprises assets, FEA describes five reference models for mapping assets to segments and enterprise services [4]. The five reference models are the performance reference model, the business reference model, the service component reference model, the technical reference model, and the data reference model.

The performance reference model provides a framework for developing consistent measurement. The business reference model provides a framework for developing a functional view of the enterprises line of business. The service component reference model provides a framework for describing how the services offered by IT systems support business functionality. The data reference model provides a framework for describing data in a consistent way that enables enterprise-wide sharing.

EA Method

FEA defines a four step iterative process for creating architectures for each segment and service [4]:

1. Architectural analysis
2. Architectural definition
3. Investment and funding strategy
4. Program management plan and execute projects

The first step, architectural analysis, is concerned with defining the scope of the segment, its baseline architecture, current problems in the segment, and a high-level vision of the desired final state for the segment [4].

The second step, architectural definition, is concerned with defining the detailed target architecture of the segment [4]. Aside for the architecture itself, it is also necessary to define a roadmap of projects to get there, the segment transition strategy, and the performance goals of the architecture.

The third step, the investment and funding strategy, is concerned with specifying how the projects identified in the segment transition strategy are to be funded [4].

The fourth step, program management plan and execute strategies, is concerned with making detailed plans for the individual projects, executing the plans, and defining performance measurements for the initiative [4].

EA Engine

FEA describes an "engine" to maintain the architecture in order ensure that it stays relevant over time. FEA calls this engine an activity it calls "segment architecture maintenance" [4]. In this activity, it is important to monitor for, list and prioritize new architectural change drivers as they appear. The impact of these drivers needs to be defined.

FEA defines an EA value measurement process - "a continuous, customer- focused process relying on feedback from EA stakeholders and other value measures to increase the quality and effectiveness of EA products and services to support business decisions." [4, Sec. 5].

2.2 Decentralization in Organizations

This section will first discuss the forms of organizational structure defined in the literature. Second, the (de)centralization of current organization and, as a consequence, their styles of IT governance will be explored. We conclude this section by underpinning the challenges organizations have to face due to their progressive decentralization.

2.2.1 What is a Decentralized Organization?

An organization can be structured in many different ways. Sachdeva [13] defines organizational structure as "... institutional arrangements and mechanisms for mobilizing human, physical, financial and information resources at all levels of the system..." According to Jacobides [14], "Organizational structure provides the frames through which individuals see their world. Thus, the way each organization is structured shapes an ecology of different, distinct frames that exist at the level of the organizational sub-unit."

There has been a lot of research on specific forms of organizational structure. Taxonomies of organization forms are defined in [15], [16]. *Classic* and *modern* types of organizational structure are often recognized. Classic types include simple centralized organizations [17], bureaucratic organizations [18], divisional structure and functional structure. Modern types include matrix structure, flat organizations, adhocracy. New forms of organizational structure emerged recently: collaborative networks, virtual organizations and coopetition.

According to Robbins [19], organizational structure has three components: complexity, formalization and centralization. Complexity refers to the degree to which activities within the organization are differentiated; Formalization refers to the degree to which work is standardized; Centralization refers to the degree to which decision making is concentrated at one point in the organization.

Following Luthens [20], centralization and decentralization can be also defined according to three factors: geographical or territorial concentration or dispersion of operations, functions, extent of concentration or delegation of decision making powers. In [1], the following characteristics of centralization are defined: the allocation of decision rights, the structure of communication lines and the choice of forms of coordination.

In a centralized organization, all decision making authority would reside with a single, top-level authority. In a completely decentralized enterprise all members would have equal decision making rights. Here, hierarchy manages the interdependencies between the different sub-units of organization and often makes direct interactions and communications unnecessary [21]. Decentralized organizations instead have less formalized communication lines [1], and more fluid, project oriented teams. [22]

Centralized organizations lean towards primarily vertical style of coordination [23], which is characterized by formal authority, standardization and rules in operations and in IT, and planning and control systems. Decentralized organizations lean towards lateral coordination characterized by meetings, task forces, coordinating roles, matrix structures, and networks [23].

Below, popular forms of organizations focusing on their degree of centralization will be considered.

2.2.2 Forms of Organizational Structure and Decentralization

Classic Organizational Structures

Pearlson and Saunders offer a thorough description of a pure hierarchical organization structure [1]: Except for the top level position, each position has one superior and zero or more subordinates. Decision rights and communication lines are strictly defined and work their way down from the top (i.e. the centre). The scope of a position is specialized and strictly defined by your superior and one works in assigned teams. The primary benefit of a hierarchy is that the high levels of management have strict governance and control over the company. Hierarchical organization structures are suited for stable, certain environments.

Hierarchical organizations can be subdivided into simple centralized and bureaucratic organizations:

In simple centralized organizations, both strategic planning and operational decision making authority belongs to one person at the top. This structure can be found in small and single-person-owned organizations with only two hierarchical levels.

Bureaucratic organizations [18] are characterized by multi-level hierarchical structure and use of standard methods and procedures for performing work.

Hierarchical organizations generally divide their labor either in terms of function, a grouping of common activities, or in terms of division, a grouping based on output (product). Two organizational structures, divisional and functional, can be identified accordingly.

Modern Organizational Structures

Matrix structure is another popular style of organization structure [1] that can be seen as a mixture of functional and divisional structures. In this form, individuals are assigned two or more supervisors covering different (usually product and functional) dimensions of the enterprise. Pearlson and Saunders state that matrix organization structures are suited for dynamic environments with lots of uncertainty, presumably because their authority structure allows them to cover multiple aspects when making decisions. However, like a hierarchical structure, a matrix structure is a rigid construct with strictly defined roles, communication lines and decision rights. Authority still comes from the top in a centralized manner, even though it becomes more distributed among matrix managers at the lower levels [1].

Flat organization is a novel type of organizations where only one or maximum two hierarchical levels are defined (similarly to simple centralized organizations). For example, Valve Corporation, a software company in the video game industry released their handbook in 2012 [24]. Unlike simple centralized organization described above, individual employees have complete freedom despite there being a president/founder at the top: Nobody reports to anyone, and everyone is free to work on whatever they want to. This is an example of high decentralization.

Adhocracy [1, 22] aims to discard traditional hierarchies in favor of decentralized decision rights and flexible communication lines connecting the entire enterprise. Specifically, instead of hierarchies, an adhocracy has a rapidly changing set of project oriented groups that have decision making authority and other powers [19]. Mintzberg describes an adhocracy as "a loose, flexible, self-renewing organic form tied together mostly through lateral means" [17].

Post-Modern Organizational Structures

New forms of organizational structure enabled uniquely by modern information and communication technologies Internet emerged recently: collaborative networks [25] and coopetitions [26].

Related to the idea of adhocracy, is the concept of collaborative networks (CN). Camarinha-Matos and Afsarmanesh define collaborative networks as being composed of "a variety of entities (e.g., organizations and people) that are largely autonomous, geographically distributed, and heterogeneous in terms of their: operating environment, culture, social capital, and goals." [25] Three common characteristics in various CNs are autonomy in the individual entities, a drive towards meeting common or complementing goals, and the use of an agreed-upon framework for collaboration.

Under the umbrella of CNs, Camarinha-Matos and Afsarmanesh define virtual organizations, virtual communities, and virtual breeding environments [25]. Virtual organizations are a group of independent organizations working together to achieve some goal(s); virtual communities are communities of individuals that interact with each other through the use of computer network-based technologies; and virtual breeding environments are frameworks for inter-operability set up by groups of organizations in order to enable the potential for forming a virtual organization.

Another organizational form emerged recently is coopetition. Bengsston and Kock describe coopetition as a complex relationship between firms where they simultaneously compete and collaborate and benefit from both [26]. Coopetition allows the participating organizations to take advantage of a heterogeneity of resources. Organizations may seek to create competitive advantage through a unique resource they own (e.g. skill). At the same time, it might be beneficial for them to cooperate with another organization that possesses a unique resource that is of value to them.

Virtual organizations and coopetitions differ from the organizational structures defined above since they do not represent a single legal entity but a group of autonomous and independent entities with different (and possibly concurrent) strategic goals. These entities are engaged into collaboration in response to factors such as specific market situations, customer demand, etc. The heterogeneous structure of such organizations remains invisible for a customer, while the service level agreements should be maintained at the same high level any other organization would maintain. Such organizational structures are grounded on a sustainable collaboration between partners without any centralized control.

Decentralization in Organizational IT

According to Rockart et al. [27], changes in business and technology as well as progressive decentralization of organization as a whole drives the changes in roles and structure of IT units. The works presented in [27–30] focus on the relation between the structure of an organization and its IT.

Fulk [28] discusses the interplay between communication technology and various organizational forms. The authors consider communication technologies as one of the key enablers of inter-organizational and intra-organizational changes.

In [29], authors study how different organizational forms affect the knowledge transfer in organization. They claim that "Organizational forms enable different kinds of motivation and have different capacities to generate and transfer tacit knowledge."

Weill [30] defines six forms of organizational structures in IT (called IT Governance archetypes) based on how the five major IT decisions in organizations are made. These archetypes are: business monarchy, IT monarchy, feudal, federal, IT duopoly and anarchy. In a *business monarchy* all IT related decisions are made in a centralized manner by the top-level executives (e.g. the CxOs). In an *IT monarchy*, a group of IT professionals are responsible for making the decisions. This is also highly centralized as the authority resides with this group. An *IT duopoly* is characterized by two groups, one of IT executives and the other of business executives, coming to agreements in order to make decisions. This is more centralized than the federal form, as the decisions are only made by the two groups, rather than each individual business unit having input. The *feudal* is much less centralized. It is where individual organizational units are responsible for their own decisions. *Federal IT* would aim to balance these through a combination of central IT and IT in the business units. *Anarchy* is a highly decentralized style of governance. It is similar to the feudal archetype, however the size of the units is much smaller. Instead of being an entire business unit, small teams or even individuals are responsible for their own decisions.

Figure **??** maps the organizational forms presented above to the centralization axis, ranging them from highly centralized (e.g. bureaucratic) to decentralized (e.g. virtual organizations)

We agree with Rockart, Earl and Ross [27], who describe a continuum of IT governance styles ranging from centralized to decentralized, with federalism in the middle. Indeed, many organizations today tend to combine both centralization and decentralization in order to obtain the advantages of both styles: global integration and efficiency due to centralized management in some key areas and agility and high quality of local customer services due to decentralized decision making in others.

For the purpose of this thesis work, three types of organizational structure in IT are considered: *Centralized IT*, where all IT related decisions are made in a centralized manner by the top-level executives, *Decentralized IT*, where each organizational subunit manages its IT in completely autonomous and independent manner, and *Federal IT* that can be seen as a combination of central IT management and IT management in the subunits. Here a primary task would be to maintain standards for the entire enterprise while supporting flexibility on the subunit level. The business units would still have ownership of many of their own systems, allowing them to implement them as they deem best.

2.2.3 Challenges of Progressive Decentralization in Organizational IT

Modern organizational structures show a strong tendency towards de-centralization [31].

Decentralization of organizations and the subsequent change to their management and operation styles heavily involves IT and requires major changes in organization processes. This transformation is not a mere question of “flattening” the organization by shifting authorities and decision making power from top to bottom hierarchical levels or from one person to a group. In classic organizations, not only does hierarchy ensure control and coordination, it also manages interdependencies between different subunits of an organization which often makes direct interaction and communication unnecessary [21]. As a result, a challenge related to decentralization and a “weakened hierarchy” is a lack of interaction and communication between organizational subunits. Another major risk of IT decentralization, according to [27], is poor synergy and integration due to a lack of standardization.

Caruso, Rogers and Bazerman [32] highlight the importance of information sharing and coordination for these organizations. In order to succeed at these aspects, they outline three barriers that decentralized organizations need to overcome. The first barrier is intergroup bias; the tendency to treat one’s own group better than other groups. The second barrier is group territoriality; the tendency for a group to protect their territory (physical or informational). The third barrier is poor negotiation strategies used by different groups when interacting with one another.

Intergroup bias is direct result of having separate, autonomous groups within an enterprise [32]. The individual groups have a tendency to promote their own group over other groups, especially in situations where they are competing for a resource, such as a portion of the budget. A certain level of competition can be beneficial, however if it leads to hostility or distrust between groups, this can have a detrimental effect on their ability to share information and collaborate. This can prevent the groups from taking advantage of situations where they have to ability to work together for the benefit of everyone.

The second barrier identified by Caruso et al. is group territoriality [32]. Group territoriality is characterized by group members taking action in order to protect their perceived territory. This can include physical territory such as space or tangible resources, as well as intangible territory, such as roles or information. Group territoriality is supported by a group's need to maintain its identity, its reputation of competence and sense of value, and a group's need for a stable home within the organization from which they interact with the rest of it. Group territoriality encourages "a sense of psychological ownership" [32] for a group's members which can enforce the belief that they are the sole responsible party for a role or specific knowledge. This "inward-looking" behavior works against collaboration and information sharing. On the other hand, group territoriality can be beneficial; it can foster a sense of security in its members that "facilitates planning and execution of activities" [32].

The third barrier identified by Caruso et al. in decentralized organizations is related to negotiations between groups, and how these negotiations are often conducted using "poor negotiation strategies" [32]. These poor strategies are the result of three common errors made while negotiating. The first error is a false belief in a "fixed pie" of value that is to be divided when negotiating. This prevents negotiating parties from recognizing situations where they are able to help each other, and therefore increase the size of the figurative pie. The second error is a failure to properly consider the other group's perspective. Understanding the other group's decision process, valuing process, and interests is key to discovering opportunities for helping one another, and the organization as a whole. The third error is when groups fail to even recognize they are in the process of negotiating. Instead, they see it as a competitive or hostile behavior where, again, they only see a fixed pie that is to be split up. This also prevents groups from taking advantage of opportunities to increase the size of the pie.

2.3 Related Works

The practice of EA is just one potential solution to the problems of business-IT alignment. EA is a "heavy-weight" approach in that it aims to be a complete solution. However, other "light-weight" solutions do exist that focus on specific aspects of business-IT alignment and this section will briefly outline a few of them: specifically, enterprise modeling and enterprise integration.

2.3.1 Enterprise Modeling

"Enterprise Modeling is the art of externalizing enterprise knowledge which adds value to the enterprise or needs to be shared. It consists in making models of the structure, behavior and organization of the enterprise" [33]. Through modeling, practitioners aim to better understand the current or future organization and function an enterprise. To this end, common enterprise categories of enterprise modeling include goal, process, and value modeling.

Goal Modeling

Goal modeling aims to describe the goals of an enterprise, their interrelations, means for achieving these goals, and additional factors that impact them. A specific example of a goal modeling technique is the Business Motivation Model (BMM) [34]. In BMM, means, ends and influencers of an organization are modeled and assessed. Here, the focus is on understanding what an organization wants to achieve (i.e. goals). The relationship between an organizations goals and its means is described, though the specifics of the means are not.

Process Modeling

According to Roshen, "[a] business process is a collection of related, structured activities or tasks that produces a specific product" [35]. Business processes are a complicated matter, and as such, process modeling is used to describe them in a detailed and accurate manner, in order to understand how an organization creates some output. Many different modeling languages exist, for example, Business Process Model and Notation (BPMN) [36], Event-driven Process Chain (EPC) [37, Ch. 6], and UML Activity Diagrams [38]. Processes can exist within an organization (intraorganizational) or they can interact with processes from other organizations (interorganizational). According to Weske [39], "the primary focus of intraorganizational business processes is the streamlining of the internal processes by eliminating activities that do not provide value". Interorganizational processes, on the other hand, aim to specify and streamline interactions with other organizations.

Business Value Modeling

Business value modeling depicts the exchange of value between entities. Examples of value modeling languages are e3 Value [40] and REA [41]. Business value modeling is used to understand what an organization does in order to create value. This allows it to be used as a starting point for the exploration of business ideas, design of processes, or development of systems [42].

2.3.2 Enterprise Integration

According to Vernadat, "Enterprise Integration (EI) consists in breaking down organizational barriers to improve synergy within the enterprise so that business goals are achieved in a more productive and efficient way" [33]. To accomplish this, Vernadat states that EI relies "free but controlled flow of information and knowledge, and the coordination of actions". To this end, three general perspectives on EI exist: information-oriented, service-oriented, and process-oriented [43].

Information-oriented integration is aimed at the integration of data. Two key components of information-oriented integration are standardizing how data is represented and enabling efficient access to it throughout an enterprise. Typical approaches to this are: data warehousing [44], where data from across the enterprise is consolidated to a single data warehouse in batches; data federation [45], where a single system is used to query multiple data sources; and data replication [46], where data is copied between data sources at regular intervals.

Service-oriented integration is aimed the integration of functionality. The dominant architecture for service-oriented integration is Service-Oriented Architecture (SOA). Here, functionality is organized into services–worked performed by one application for another application–which have the characteristics of reusability and composability [35]. These characteristics are important as they allow for services to be shared across and between enterprises (reusability), and for multiple services to be used together to create a new service (composability).

Process-oriented integration uses enterprise knowledge in conjunction with systems knowledge in order to integrate on the business process level [33]. Process-oriented integration builds on service-oriented integration in order to automate and order services for the production of some product. This can be done in both intra- and interorganization contexts, and is accomplished with the use of process models and systems dedicated to process management [47]. A key advantage of process integration is that it provides a basis for communication between the IT and business sides of an enterprise [47].

3. Method

3.1 Choice of Research Method

Empirical research "aims at describing, explaining and predicting the world" [48, Ch. 1]. In comparison, design research additionally wants to improve upon the world through the development of artifacts. This thesis project aims to improve EA by proposing artifacts that can extend the support of existing EA frameworks for decentralization. As a result, a design research approach, specifically design science, will be utilized. The remainder of this section seeks to further demonstrate the suitability of design science and outline the specific research strategies and methods chosen.

3.1.1 Design Science and its Relevance to this Thesis Work

Design science is concerned with the development and application of *artifacts* aimed at solving some practical problem [48,49] in a manner that is of general interest [48, Ch. 1].

In order to be relevant, Design Science must exist in some context. Johannesson and Perjons [48, Ch. 1] define a generic context for design science in terms of people, practices and problems. A practice is a set of related activities performed regularly by people. In the performance of a practice, people encounter practical problems. Two general kinds of problems exist; one where the current state of affairs problematic and the desirable state is neutral, and a second where the current state is neutral and the desirable state is an improvement. The artifacts created through design science can be used by people to solve these practical problems. These concepts are easily related to this thesis work: Enterprise Architecture is a practice with a problem of the second type. The current state of EA can be seen as being neutral; the problems outlined in this thesis are not necessarily ones that EA practitioners are concerned with. However, this thesis argues that existing EA frameworks – each of which can be seen as an artifact composed of smaller artifacts – can be improved by increased support for decentralization through the development of artifacts addressing the issue of decentralization.

According to Johannesson and Perjons [48, Ch. 1], artifacts themselves have an "inner construction", exist in an environment, and have a function. The inner construction refers to the inner components of the artifact and the relations between them. The environment refers to the artifacts practice, the people using it, and anything else in its surroundings that have an effect on it. Lastly, the function of an artifact is the result of using it in its practice. This definition of an artifact also relates well to this thesis work: the inner construction of our artifact (an EA framework for decentralized organizations) can, at a high level, be seen as an EA method, EA description, and EA engine. The relations for these components are outlined in Figure **??**. The environment of our artifact includes the practice of EA and all affected components of the decentralized organization using it, such as involved stakeholders and implementers. The function of our artifact is, on a high level, to bring the benefits of traditional EA to decentralized organizations (e.g. business-IT alignment).

There exist four different types of artifacts: constructs, models, methods, and instantiations [48, 49]. Enterprise architecture is concerned with the first three of those types: constructs, models and methods.

Constructs are ways to describe some phenomenon. They give a common language for talking about something, but do not make any assertions about reality. For example, the EA description component of an EA framework provides a common taxonomy for the different parts of an organization covered by EA.

Models represent other objects. EA makes use of models, specifically descriptive and prescriptive models. Descriptive models are used to represent a current situation and its challenges, such the “as-is” architecture from the EA description. Prescriptive models represent potential future solutions, such as the “to-be” architecture, also from the EA description.

Methods define “guidelines and processes for how to solve problems and achieve goals” [48, Ch. 1]. The EA method and EA engine are primarily methods, the former to construct the EA description, the latter to ensure its proper use throughout its lifecycle.

3.1.2 A Design Science Method Framework

Having established the relevance of design science to this thesis project, this thesis will therefore follow the framework for a design science method presented by Johannesson and Perjons in [48, Ch. 4]. This method is composed of five activities with input-output relationships: Explicate Problem, Outline Artifact and Define Requirements, Design and Develop Artifact, and Evaluate Artifact. Each activity has an output which serves as an input to the next activity (e.g. an explicated problem is the input to the Outline Artifact and Define Requirements phase). These activities are carried out in an iterative manner, meaning that the practitioner will move back and forth between them as opposed to working in a sequential manner.

The Explicate Problem activity is concerned with outlining the problem addressed by the research work in detail. To this end, the problem’s significance needs to be clearly stated and its underlying causes can be possibly identified and analyzed. The output of this phase an explicated problem.

The Outline Artifact and Define Requirements activity is where the explicated problem is transformed into the requirements for a solution to said problem. The output of this phase is the set of requirements for the artifact.

The Design and Develop Artifact activity is where the artifact itself is built based on the requirements for the artifact. The output of this phase is the artifact itself.

The Demonstrate Artifact activity takes the developed artifact and implements it in either a real or illustrative case in order to demonstrate its viability. The output of this phase is the demonstrated artifact.

The Evaluate Artifact activity is to demonstrate the artifact’s fulfillment of the requirements and the degree to which it solves the problem. The output here is an evaluated artifact.

3.1.3 The Role of Research Strategies and Methods in this Design Science Framework

Each of these activities can make use of controls and resources. Controls are the knowledge used to govern an activity [48, Ch. 4], and resources are the knowledge used as a basis for the activity. In this method for design science, controls are the research strategies and research methods used. A research strategy is the overall approach used to answer a research question [48, Ch. 3], and research methods are the concrete methods used to generate and analyze data.

3.1.4 Choice of Research Strategy

Alternative strategies exist for undertaking research in the field of design science. A number of common strategies will be briefly outlined in order to discuss their suitability for this thesis.

Surveys aim to take a comprehensive look at something by gathering data from a large number of different sources. This data is then analyzed in some manner. Surveys offer a wide view [48, 50], and as such, are not well suited for a depth view of something. This does not fit in with this project which takes an in-depth look at EA frameworks.

Experiments employ a controlled and artificial environment in order to isolate a small number of specific factors to study them in detail. The effects of manipulating variables in the environment needs to be precisely measured [50]. This poses a problem for EA as organizations are highly complex entities where it would be exceptionally difficult to exert precise control and precisely measure the effects. For this reason, experiments are not a suitable strategy for this project.

In action research, the researcher is an active participant in affecting the environment they are researching. Here, the research is done as part of the practice, as opposed to it being a separate activity [50]. This could be a highly effective strategy for this thesis topic as it would allow the researcher to experience the problems of decentralization first-hand. Furthermore, action research is a cyclical process, meaning the researcher could repeatedly try out different solutions and evaluate their effects in order to come to a good solution. This would allow for a researcher in a decentralized organization the flexibility to find a solution that works. Despite this fit to the thesis topic, the practical issue of finding a decentralized organization that is willing to go through this process is a significant one. As a result, action research is not used in this project.

Ethnography is similar to action research in that the researcher becomes a member of the environment being researched. The difference lies in that they are there to integrate themselves into it, rather than affect change [50]. This could be an applicable research strategy for understanding problems from the perspective of stakeholders, however finding a decentralized organization with some sort of EA (or at least an interest in it) is quite the challenge in itself. Additionally, ethnography requires a large time investment in order to integrate adequately into the environment of study, which is not feasible for a Master's project. For these reasons, ethnography is not used in this project.

Case studies take an in depth view of a single instance of the practice where the problem of interest exists. Case studies are ideal when "a researcher wants to investigate an issue in depth and provide an explanation that can cope with the complexity and subtlety of real life situations" [50]. This project is interested in an in depth view of the problem of suitability of EA for decentralized organizations and decentralized organizations are real life entities that are highly complex. Furthermore, in contrast with ethnography and action research, conducting a single case study fits in well with the scope of a Master's project; it is not necessary for the subject organization to invest large amounts of resources into the project and time investment of a case study fits in with a short-term project. For these reasons, this thesis project will employ a case study research strategy.

3.1.5 Choice of Research Methods

Research strategies do not prescribe any concrete ways to generate and analyze data. Specific research methods for data generation and data analysis are needed.

Data generation methods

This thesis employs the use of interviews and document studies for data generation. Document studies are used as a large amount of data on the structure of the organization being studied is available. Documents are a good source of authoritative, objective, and factual data [50], which therefore gives a solid foundation on understanding the studied organization. Interviews were chosen in order to supplement this data with x'information from stakeholders about the organization. Interviews are suited for gaining insight into complex phenomena, which is supportive of our need for an in-depth view of a complex entity that is an organization. Furthermore, interviews are practical for this project as; a) the organization being studied does not need to invest large amounts of time and b) I have physical access to potential interviewees.

Other common data generation methods are questionnaires and observations. Questionnaires are not particularly suitable for this project as they are most useful when used for specific, straightforward information [50]. This project, on the other hand, is interested in the complexities of an organization. An observation study would require spending time in an organization in order to directly observe its operations. As this thesis is conducted as an individual project, this is not a feasible activity, due to the size and complexity of an organization.

Data analysis methods

After the data has been obtained, it is necessary to analyze it in order to understand the object being studied. Data can be analyzed in either a quantitative or a qualitative manner. Quantitative data analysis is concerned with numeric data, whereas qualitative deals with words and visuals. According to Denscombe [50], some other differences between the approaches are; quantitative research is generally associated with large-scale studies whereas qualitative research is concerned with small-scale studies, and quantitative research is concerned with "specific variables" while qualitative research takes a "holistic perspective". This project follows a qualitative approach because; a) the data being analyzed will composed of words coming from interviews and document studies, b) this is a small-scale study, and c) this project is interested in a holistic perspective on our case study subject.

3.2 Application of Method

This thesis work follows the framework by Johannesson and Perjons [48, Ch. 4] presented above in Section 3.1.2. This thesis deviates slightly from their proposed framework as the formal "evaluate artifact" activity will not be performed. This section will first elaborate on how the different activities will be accomplished and then outline the overall process.

As suggested in the framework, the IDEF0 notation will be used for visualizing the various activities. In this notation, each activity as an input and output, controls in the form of research strategies and methods, and resources which is the knowledge base for the activity.

3.2.1 Case Study: An Institution of Higher Education in Sweden

This thesis work will use an institution of higher education in Sweden as an illustrative case study. This case was chosen as an example of a decentralized organization with an implicit EA, i.e., there is no formal EA framework used, but as they use IT extensively, some form of implicit architecture must exist. An advantage of this case is that, as a

Document	Description
Institution's homepage	Contains descriptions of the different organizational areas of the institution as well its organizational structure
Authority delegation documents	These publicly available documents specify authority and delegations of said authority of the institution's organizational units
Rule book	The official rule book of the institution detailing rules and decisions that must be followed by the institution

Table 3.1: ***Documents forming the document study***

public institution, many official documents are available on its organizational structure, thus making a document study a viable research method. The documents that formed this study are described in Table 3.1.

A disadvantage of this case is that there is no use of modern EA frameworks in the institution, which weakens the link between the institution and the practice of EA.

This thesis is not aiming at effecting change in this institution. The focus is instead on: analyzing the state of its EA in order to assess the decentralization support provided in contrast with what is needed; and proposing part of an EA that can provide the needed support.

Four separate interviews are conducted in order to get a holistic view of the institution. The roles of the interviewees are: deputy department head, head of PhD studies, head of undergrad studies, and head of IT. The interviews are conducted in a semi-structured manner, starting with a set of open-ended questions that promote the interviewees to elaborate on their views.

3.2.2 Research Activities

Iterations Between Activities

This research is conducted through two general iterations between the research activities:

Iteration #1 The first iteration focuses only on using literature for conducting the research activities. In this iteration, the activities explicate problem, outline artifact and define requirements, and the generate sub-activity of design and develop artifact were performed.

Iteration #2 The second iteration then makes use of the case study in order to supplement and confirm findings from the first iteration. In this iteration, the activities explicate problem, outline artifact and define requirements, the search-and-select sub-activity of design and develop artifact, and demonstrate artifact were performed.

Explicate problem

Figure 3.1 outlines the major components of this activity:

Sub-activities Define Precisely, Motivate Problem and Find Root Causes [48, Ch. 5]

Input The initial problem as described in Section 4.1

Resources A literature study on centralization/decentralization in organizational theory, and on the modern EA frameworks TOGAF, Zachman, and FEA. These three frameworks were chosen due to their popularity and extensive available literature.

Controls A case study research strategy that will make use of interviews and a document study. The case is detailed in Section 3.2.1.

Output A fully explicated problem, specifically, the set of specific shortcomings of modern EA frameworks when applied to decentralized organizations determined in the "find root causes" sub-activity.

Iteration #1

The first sub-activity, Define Precisely, is accomplished with the use of the literature review. Here, the differences between centralized and decentralized organizations are precisely specified in order to show that there is a problem.

In the second sub-activity, the problem is motivated through the use a literature review on decentralized organizations. For this, a classification for decentralized organizations will be built from the literature review.

The third sub-activity, find root causes, is accomplished through a literature review. An in-depth analysis of each of the three EA frameworks will done to find specific shortcomings; aspects where the framework provides support for centralized organizations and not decentralized organizations. Aspects that are supportive of decentralization will be presented as well.

Iteration #2

The first sub-activity, the problem in the case is precisely defined

In the second sub-activity, the problem is motivated through the use of the case study. To this end, specific issues in the case arising from their implicit EA and their organizational structure are identified.

The root causes of these issues are then determined in the third sub-activity. This will be done by developing a lightweight "as-is" architecture for the case.

Outline artifact and define requirements

Figure 3.2 outlines the major components of this activity:

Sub-activities Outline Artifact and Define Requirements [48, Ch. 6]

Input Set of specific shortcomings of modern EA frameworks when applied to decentralized organizations

Resources Literature study on centralization/decentralization in organizational theory

Controls None

Output Requirements for a decentralized EA

Iteration #1

In the first sub-activity Outline Artifact, the type of artifacts being developed is specified.

In the second sub-activity, Define Requirements, the requirements for the developed artifact are elicited. This is based on the specific shortcomings of the EA frameworks and the literature study on centralization/decentralization in organizations.

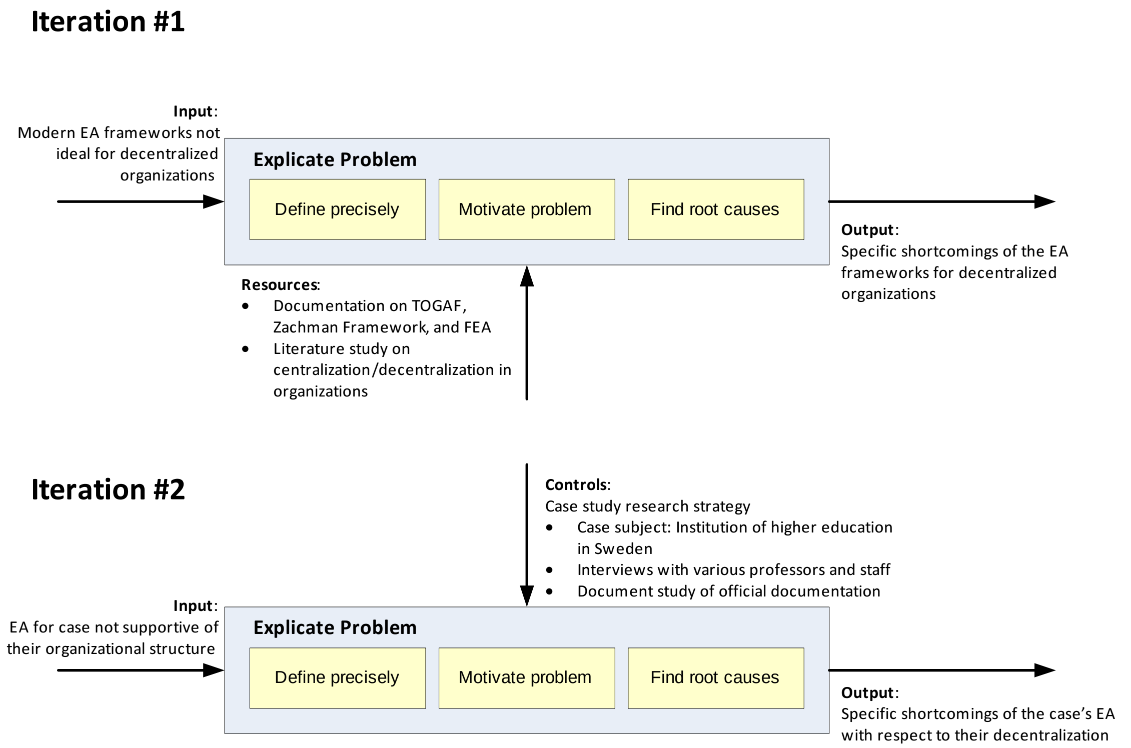

Iteration #1
Input:
Modern EA frameworks not ideal for decentralized organizations
Explicate Problem
Define precisely
Motivate problem
Find root causes
Output:
Specific shortcomings of the EA frameworks for decentralized organizations
Resources:
Documentation on TOGAF, Zachman Framework, and FEA
Literature study on centralization/decentralization in organizations
Iteration #2
Controls:
Case study research strategy
Case subject: Institution of higher education in Sweden
Interviews with various professors and staff
Document study of official documentation
Input:
EA for case not supportive of their organizational structure
Explicate Problem
Define precisely
Motivate problem
Find root causes
Output:
Specific shortcomings of the case's EA with respect to their decentralization

Figure 3.1: Explicate problem activity

Iteration #2

In the first sub-activity Outline Artifact, the type of artifacts being developed is specified.

In the second sub-activity, Define Requirements, the requirements for the developed artifact are elicited. This is based on the specific shortcomings of the case's EA.

Design and develop artifact

Figure 3.3 outlines the major components of this activity:

Sub-activities Generate and Search and Select [48, Ch. 7]

Input Requirements for a decentralized EA

Resources Literature study on peer-to-peer architectures

Controls None

Output Prototype of one artifact for one aspect of an EA framework for decentralized organizations

Iteration #1

In this iteration, only the Generate sub-activity is performed. Here, different potential solutions, each covering a subset of the requirements for a decentralized EA are outlined. This requires multiple artifacts as an EA framework is a large entity composed of a number of different artifacts. In order to develop these artifacts, a literature study on peer-to-peer architectures is conducted. Principles from peer-to-peer are taken and applied as the basis for solutions meeting the specified requirements. Peer-to-peer architectures were chosen because they have offered solutions to decentralization in

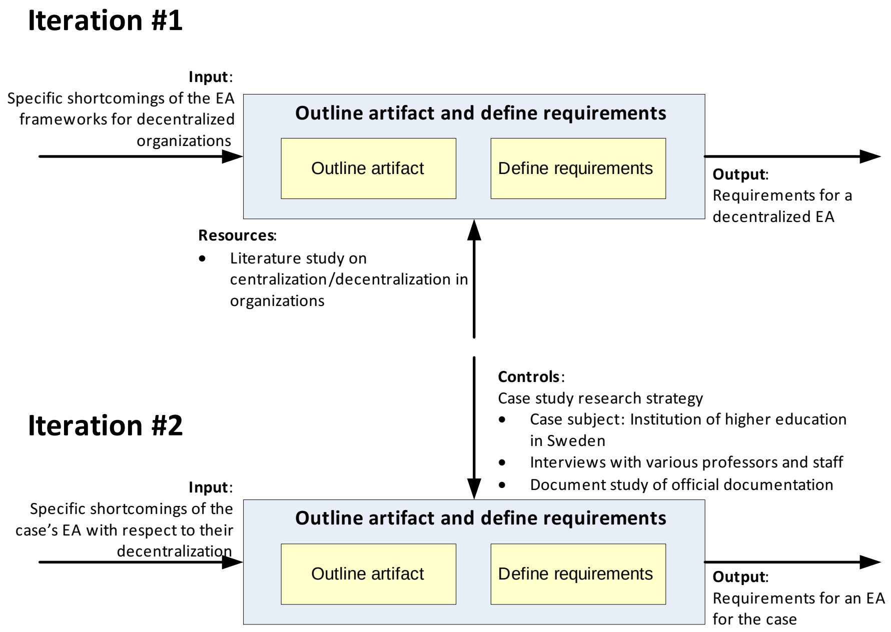

Iteration #1
Input:
Specific shortcomings of the EA frameworks for decentralized organizations
Outline artifact and define requirements
Outline artifact
Define requirements
Output:
Requirements for a decentralized EA
Resources:
• Literature study on centralization/decentralization in organizations
Controls:
Case study research strategy
• Case subject: Institution of higher education in Sweden
• Interviews with various professors and staff
• Document study of official documentation
Iteration #2
Input:
Specific shortcomings of the case's EA with respect to their decentralization
Outline artifact and define requirements
Outline artifact
Define requirements
Output:
Requirements for an EA for the case

Figure 3.2: Outline artifact and define requirements activity

other domains (e.g. technical) and might therefore be able to provide solutions for the practice of EA.

Iteration #2

In this iteration, one of the outlined solutions (from the Generate sub-activity in Iteration #1) is selected and elaborated on to create a prototype of one artifact for an EA framework for decentralized organizations. This selection is based on its applicability to the case.

Demonstrate artifact

Figure 3.4 outlines the major components of this activity:

Sub-activities Choose or Design Case and Apply Artifact [48, Ch. 8]

Input Prototype of one artifact for one aspect of an EA framework for decentralized organizations

Resources Knowledge on the selected case (institution of higher education in Sweden)

Controls A case study research strategy that will make use of interviews and a document study

Output Proof-of-concept artifact for one part of an EA framework prototype

Iteration #1

This activity is not performed in the first iteration.

Iteration #2

For the Choose or Design Case sub-activity, an institution of higher education in Sweden has been selected. Details on the case are described in 3.2.1.

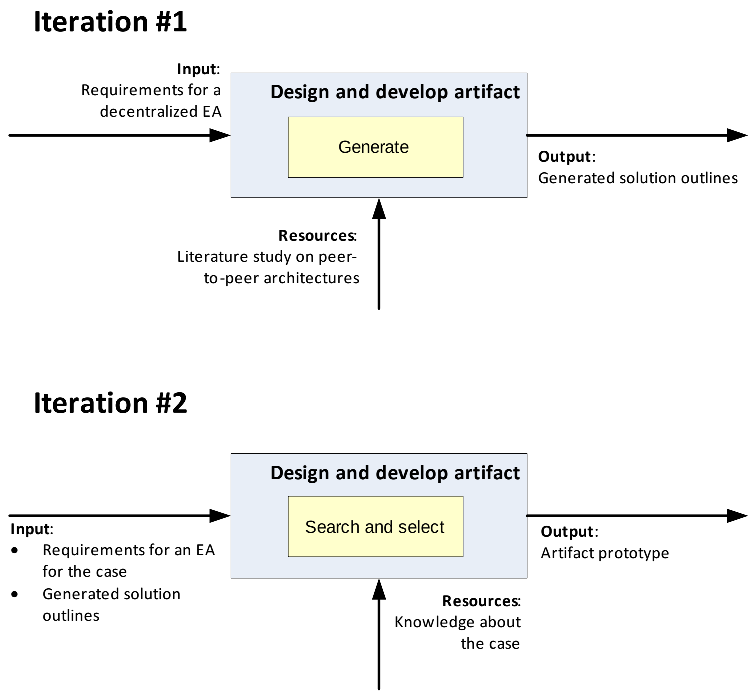

Iteration #1
Input:
Requirements for a decentralized EA
Design and develop artifact
Generate
Output:
Generated solution outlines
Resources:
Literature study on peer-to-peer architectures
Iteration #2
Design and develop artifact
Search and select
Input:
Requirements for an EA for the case
Generated solution outlines
Output:
Artifact prototype
Resources:
Knowledge about the case

Figure 3.3: Design and develop artifact activity

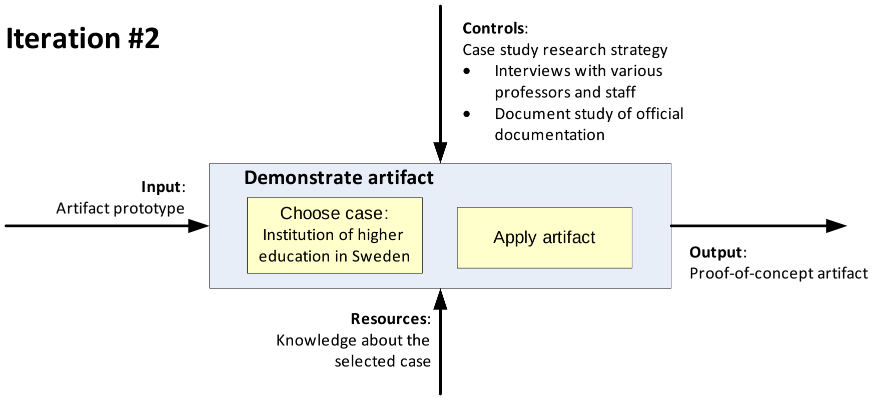

Iteration #2
Controls:
Case study research strategy
Interviews with various professors and staff
Document study of official documentation
Demonstrate artifact
Input:
Artifact prototype
Choose case:
Institution of higher education in Sweden
Apply artifact
Output:
Proof-of-concept artifact
Resources:
Knowledge about the selected case

Figure 3.4: Demonstrate artifact activity

In the Apply Artifact sub-activity, the artifact is applied to the case by creating a to-be architecture for the relevant aspect of the case. This to-be architecture is the proof-of-concept artifact for one part of an EA framework prototype.

Evaluate artifact

A formal evaluation for the developed artifact would be conducted by actually implementing it in the case and analyzing the results of the implementation. However, the focus of this thesis work is on explicating the problem of decentralization for EA and demonstrating a proof-of-concept of a partial solution. Consequently, the evaluate artifact phase is not done.

3.3 Ethical Considerations

As the proof-of-concept does not involve any actual implementation, the ethical considerations for this research project are minimal and only involve withholding the identity of the case institution.

4. Results

4.1 Explicate Problem

4.2 Outline Artifact and Define Requirements

4.3 Design and Develop Artifact

4.4 Demonstrate Artifact

5. Conclusion and Discussion

Bibliography

[1] K. E. Pearlson and C. S. Saunders, *Strategic Management of Information Systems*, 4th ed. John Wiley & Sons, 2009.

[2] The Open Group, *TOGAF Version 9.1.* The Open Group, 2011.

[3] J. A. Zachman, "John Zachman's Concise Definition of The Zachman Framework," 2008. [Online]. Available: http://www.zachman.com/about-the-zachman-framework\

[4] Federal Enterprise Architecture Program Management Office, "FEA Practice Guidance," 2007.

[5] R. Sessions. (2007) A Comparison of the Top Four Enterprise-Architecture Methodologies. [Online]. Available: http://msdn.microsoft.com/en-us/library/bb466232.aspx

[6] J. Schekkerman, *How to survive in the jungle of enterprise architecture frameworks: Creating or choosing an enterprise architecture framework*, 2nd ed. Trafford Publishing, 2004.

[7] Gartner Inc, "Enterprise Architecture (EA)." [Online]. Available: http://www.gartner.com/it-glossary/enterprise-architecture-ea/

[8] J. W. Ross, P. Weill, and D. Robertson, *Enterprise architecture as strategy: Creating a foundation for business execution.* Harvard Business Press, 2006.

[9] M. Lankhorst, *Enterprise architecture at work: Modelling, communication and analysis*, 2nd ed. Heidelberg: Springer, 2009.

[10] S. Bente, U. Bombosch, and S. Langade, *Collaborative Enterprise Architecture: Enriching EA with Lean, Agile, and Enterprise 2.0 Practices.* Waltham, MA: Morgan Kaufmann, 2012.

[11] Z. I. Inc. (2013) Zachman Interational: The Official Home of the Zachman Framework. [Online]. Available: http://www.zachman.com

[12] J. A. Zachman. (2000) Conceptual, Logical, Physical: It Is Simple. [Online]. Available: http://www.zachman.com/ea-articles-reference/58-conceptual-logical-physical-it-is-simple-by-john-a-zachman

[13] P. Sachdeva *et al.*, "Analytical framework for the organization and structure of nars." in *Organization and structure of national agricultural research systems: selected papers....*, 1990, pp. 1–8.

[14] M. G. Jacobides, "The inherent limits of organizational structure and the unfulfilled role of hierarchy: Lessons from a near-war," *Organization Science*, vol. 18, no. 3, pp. 455–477, 2007.

[15] B. McKelvey, *Organizational systematics: Taxonomy, evolution, classification.* Univ of California Press, 1982, vol. 72.

[16] P. Rich, "The organizational taxonomy: definition and design," *Academy of Management Review*, pp. 758–781, 1992.

[17] H. Mintzberg, “The structuring of organizations: A synthesis of the research,” *University of Illinois at Urbana-Champaign’s Academy for Entrepreneurial Leadership Historical Research Reference in Entrepreneurship*, 1979.

[18] ——, “Organizational design: fashion or fit?” *Harvard Business Review*, vol. 59, no. 1, pp. 103–116, 1981.

[19] S. P. Robbins, *Organisational behaviour: Concepts, controversies and applications: Australia and New Zealand.* Royal Victorian Institute for the Blind. Special Request Service, 1997.

[20] F. Luthans, *Organizational behavior.* McGraw-Hill/Irwin, 2006. [Online]. Available: http://books.google.fr/books?id=1axXAAAAYAAJ

[21] J. D. Thompson, *Organizations in action: Social science bases of administrative theory.* Transaction Pub, 1967.

[22] L. M. Applegate, J. I. Cash, and D. Q. Mills, *Information technology and tomorrow’s manager.* Harvard Business Review, Reprint Service, 1988.

[23] L. G. Bolman and T. E. Deal, *Reframing Organizations*, 4th ed. San Francisco, California: John Wiley & Sons, 2008.

[24] Valve Corporation. (2012) Valve Handbook for New Employees. [Online]. Available: www.valvesoftware.com/company/Valve_Handbook_LowRes.pdf

[25] L. M. Camarinha-Matos and H. Afsarmanesh, “Collaborative networks: a new scientific discipline,” *Journal of Intelligent Manufacturing*, vol. 16, no. 4-5, pp. 439–452, 2005.

[26] M. Bengtsson and S. Kock, “Coopetition in business networks–to cooperate and compete simultaneously,” *Industrial marketing management*, vol. 29, no. 5, pp. 411–426, 2000.

[27] J. Rockart, M. Earl, and J. Ross, “Eight imperatives for the new IT organization,” *Sloan management review*, pp. 43–56, 1996.

[28] J. Fulk and G. DeSanctis, “Electronic communication and changing organizational forms,” *Organization science*, vol. 6, no. 4, pp. 337–349, 1995.

[29] M. Osterloh and B. S. Frey, “Motivation, knowledge transfer, and organizational forms,” *Organization science*, vol. 11, no. 5, pp. 538–550, 2000.

[30] P. Weill, “Don‘t just lead, govern: How top-performing firms govern it,” *MIS Quarterly Executive*, vol. 3, no. 1, pp. 1–17, 2004.

[31] R. L. Daft and A. Y. Lewin, “Where are the theories for the new organizational forms? An editorial essay,” *Organization Science*, vol. 4, no. 4, pp. i–vi, 1993.

[32] H. M. Caruso, T. Rogers, and M. H. Bazerman, *Boundaries Need Not be Barriers: Leading Collaboration Among Groups in Decentralized Organization.* Harvard Business School, 2008.

[33] F. Vernadat, “Enterprise modeling and integration (emi): Current status and research perspectives,” *Annual Reviews in Control*, vol. 26, no. 1, pp. 15 – 25, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367578802800062

[34] The Business Rules Group. (2010) The Business Motivation Model: Business Governance in a Volatile World. [Online]. Available: http://www.businessrulesgroup.org/bmm.shtml

[35] W. Roshen, *SOA-Based Enterprise Integration: A Step-by-Step Guide to Services-based Application*. McGraw-Hill, Inc., 2009.

[36] Object Management Group, "Business process model notation (bpmn) version 2.0," *OMG Specification*, 2011.

[37] A.-W. Scheer, O. Thomas, and O. Adam, "Process modeling using event-driven process chains," *Process-Aware Information Systems*, pp. 119–146, 2005.

[38] O. M. Group. (2011) UML 2.4.1. [Online]. Available: http://www.omg.org/spec/UML/2.4.1/

[39] M. Weske, *Business process management*. Springer, 2012.

[40] J. Gordijn and H. Akkermans, "E3-value: Design and evaluation of e-business models," *IEEE Intelligent Systems*, vol. 16, no. 4, pp. 11–17, 2001. [Online]. Available: http://www.computer.org/portal/web/csdl/magazines/intelligent#4

[41] P. Hrubỳ, J. Kiehn, and C. V. Scheller, *Model-driven design using business patterns*. Springer, 2006.

[42] P. Johannesson, "Lecture 2: Business Models Overview," unpublished, Stockholm University, Stockholm, 2011.

[43] J. Zdravkovic, "Lecture 1: Introduction to SI," unpublished, Stockholm University, Stockholm, 2012.

[44] R. Kimball, *The data warehouse toolkit*. John Wiley & Sons, 2006.

[45] L. M. Haas, E. T. Lin, and M. A. Roth, "Data integration through database federation," *IBM Systems Journal*, vol. 41, no. 4, pp. 578–596, 2002.

[46] M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, and G. Alonso, "Database replication techniques: A three parameter classification," in *Reliable Distributed Systems, 2000. SRDS-2000. Proceedings The 19th IEEE Symposium on*. IEEE, 2000, pp. 206–215.

[47] M. Dumas, W. M. Van der Aalst, and A. H. Ter Hofstede, *Process-aware information systems: bridging people and software through process technology*. Wiley-Interscience, 2005.

[48] P. Johannesson and E. Perjons, *A Design Science Primer*, 1st ed. CreateSpace, 2012. [Online]. Available: http://designscienceprimer.wordpress.com

[49] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[50] M. Denscombe, *The good research guide: for small-scale social research projects*. Open University Press, 2010.