Vol. 0, No. 0, month 0000

Hend*erik* A. Proper and Marc M. Lankhorst

Enterprise Architecture

Towards essential sensemaking

In this position paper, we discuss our view on the past and future of the domain of enterprise architecture. We will do so, by characterizing the past, and anticipated future, in terms of a number of trends. Based on these trends, we then discuss our current understanding of the future concept and role of enterprise architecture. We conclude by suggesting vantage points for future research in the field of enterprise architecture.

1 Introduction

Increasingly, organizations recognize enterprise architecture as an important instrument to steer (or influence) the direction of transformations [Buckl et al. 2011; Greefhorst and Proper 2011; Lahrmann et al. 2010; e. a. Lankhorst M. 2012; Op 't Land, Proper, et al. 2008; The Open Group 2009]. Over the past decades, the domain of enterprise architecture has seen a tremendous growth, both in terms of its use and development in practice and as a subject of scientific research. The roots of the domain can actually be traced back as far as the mid 1980s.

In this position paper, which builds on [Proper 2012], we will review the evolution of the field of enterprise architecture. We do so by characterizing both its history (section 2), as well as its anticipated future (section 3), in terms of a number of trends. Based on these trends, we also discuss our current understanding of the concept and role of enterprise architecture (section 4). We conclude with a brief discussion of our view on research in the field of enterprise architecture in terms of key vantage points for further research.

2 A History of Enterprise Architecture

In this section we discuss the history of the field of enterprise architecture in terms of a number of trends as observed by us.

2.1 From Computer Architecture to IS Architecture

The origins of *enterprise architecture* can be traced back to the concept of *information systems architecture* (IS Architecture), which in turn has its roots in the concept of *computer architecture*. One of the first references to the term *architecture*, in the context of IT, can be found in a paper from 1964 on the architecture of the IBM System/360 [Amdahl et al. 1964]. There it was used to introduce the notion of *computer architecture*.

Later, in the 1980s, the term architecture started to become used in the domain of information systems development as well. This occurred both in Europe and North America. The North American use of the concept of architecture in an information systems context can (at least) be traced back to a report on a large multi client study, the PRISM[1] project [Hammer & Company 1986] conducted in 1986, as well as the later paper by John Zachman [Zachman 1987]. The European origins can be traced back to the early work of August-Wilhelm Scheer on the ARIS framework, also dating back to 1986 [Scheer 1986, 1988, 2000].

In Europe, the ARIS framework as developed by August-Wilhelm Scheer eventually formed the base of the well known IDS-Scheer toolset. In

[1] Not to be confused with the present day concept of PRISM `http://en.wikipedia.org/wiki/PRISM_(surveillance_program)`

North America, the PRISM project was a multi-year research project, led by Michael Hammer, Thomas Davenport, and James Champy. PRISM, short for Partnership for Research in Information Systems Management, was sponsored by approximately sixty of the largest global companies (DEC, IBM, Xerox, Texaco, Swissair, Johnson and Johnson, Pacific Bell, AT&T, etc.). This research effort produced an architecture framework known as the PRISM Architecture Model, which was published in 1986. The PRISM framework has strongly influenced other enterprise architecture standards, methods and frameworks [Beijer and De Klerk 2010; Davenport et al. 1989; Richardson et al. 1990; Rivera 2007].

Many years later, the PRISM report also influenced the IEEE definition of architecture, as many of the IEEE 1471 committee members were employed by the original sponsors of their earlier work on PRISM. Key people involved in PRISM later also spearheaded the wave on Business Process Reengineering [Davenport et al. 1989; Hammer 1990], which is essentially an early business architecting effort.

The Zachman [Zachman 1987] paper is often referred to as one of the founding papers of the field of enterprise architecture. It should be noted, however, that both the PRISM and ARIS frameworks pre-date the Zachman framework, although these frameworks have indeed been published in less accessible sources.

The important message of the ARIS, PRISM and Zachman frameworks is the need to consider *information systems* from multiple perspectives based on stakes, concerns, as well as different aspects of the information systems and its business or technology context, while at the same time focusing on the key properties of the information system. The latter focus is also captured by the phrase *fundamental organization* in the IEEE 1471 [IEEE 2000] architecture definition: "*the fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution.*", where *fundamental* is dependent on the key concerns/stakes of the stakeholders involved in an architecting effort.

The basic idea to consider information systems in a holistic way, i.e., from multiple related perspectives, was actually already identified before being linked to the term *information systems architecture*. For example, Multiview [Wood–Harper et al. 1985] already identified five essential viewpoints for the development of information systems: Human Activity System, Information Modelling, Socio-Technical System, Human-Computer Interface and the Technical System. Even though the authors of Multiview did not use the term *architecture*, one can argue that Multiview is effectively one of the earliest explicit information systems architecture frameworks. During the same period in which Multiview was developed, the so-called CRIS Task Group of the IFIP working group 8.1 developed similar notions in 1982 [Olle, Sol, and Tully 1983; Olle, Sol, and Verrijn–Stuart 1982], where stakeholder *views* were captured from different perspectives. Special attention was paid to disagreement about which aspect (or *perspective*) was to dominate the system design (viz. "process", "data" or "behaviour"). In the early 1980s, the CRIS Task Group already identified several *human roles* (stakeholders!) involved in information system development, such as *responsible executive*, *development coordinator*, *business analyst*, *business designer*, quite similar to the stakeholder dimension of e.g. the Zachman framework.

In the 1990s, challenges such as interoperability and distributed computing resulted in the creation of reference architectures, including the CIMOSA (*Open System Architecture for CIM*) framework for computer integrated manufacturing systems [ESPRIT Consortium AMICE 1993] and the RM-ODP (*Reference Model for Open Distributed Processing*) framework for information systems [ISO 1996a,b, 1998a,b]

2.2 From IS Architecture to Enterprise Architecture

The awareness that the design of information systems needed to be seen in a broader business and enterprise context, triggered several authors to shift towards the use of the term *enterprise architecture* rather than *information systems architecture*. One of the first authors to use the term *enterprise architecture* was Spewak [Spewak 1993].

The initial architecture approaches focused on the development of information systems, while taking the models/architectures of other relevant aspects of the enterprise as a given. However, due to the strong connection between business processes and the underlying information systems, it was only natural to not just treat such perspectives as a *given*, but rather to *co-design* these in tandem with the information systems and their underlying IT support.

Earlier versions of TOGAF [The Open Group 2005], including TAFIM [TAFIM 1996], treated *business architecture* as a *given* thing. By defining *Enterprise Architecture Planning (EAP)* as "*the process of defining architectures for the use of information in support of the business and the plan for implementing those architectures*", Spewak [Spewak 1993] also seems to suggest to take *business architecture* as a *given*. Boar [Boar 1999] in "*Constructing Blueprints for Enterprise IT architectures*" does the same.

The shift from taking a *business architecture* as a given input, to the realisation that business and IT should be co-designed as a whole, could be seen as the birth of modern day *enterprise architecture*. The *strategic alignment model* by Henderson & Venkatraman [Henderson and Venkatraman 1993] has played an important role in taking this step to the co-design of *business architecture* and *information systems architecture*. Henderson & Venkatraman [Henderson and Venkatraman 1993] indeed suggests that aligning business and IT should not necessarily require that the business strategy should be treated as a given. There are several ways to align business and IT. Also the work by e.g. Tapscott and Caston [Tapscott and Caston 1993] contributed to this realisation, as well as the work by Ross et al. [Ross et al. 2006]. The earlier mentioned work on Business Process Reengineering [Davenport et al. 1989; Hammer 1990], essentially an early business architecting effort, also contributed to this shift.

Without an attempt to be complete, some enterprise architecture approaches that indeed take a more co-design oriented perspective include: the Integrated Architecture Framework (IAF) [Goedvolk et al. 1999; Wout et al. 2010], the ArchiMate [Jonkers et al. 2003; e. a. Lankhorst M. 2012] language, as well as the DYA [Wagter et al. 2001, 2005] and DEMO [Dietz 2006] methods. Also the most recent version of TOGAF [The Open Group 2009] does indeed suggest to co-design the business architecture and the information systems architecture.

2.3 From Business-to-IT-stack to Enterprise Coherence

The realisation that information systems architecture and business architecture need to be co-designed in tandem, led most enterprise architecture approaches to capture a business architecture in terms of building blocks such as business services, business processes, business actors, etc. These business building blocks were then linked to information systems, and ultimately IT infrastructures, resulting in a '*Business-to-IT-stack*'. Among an increasing group of researchers and practitioners, the 'reduction' of 'the architecture of the enterprise' to a '*Business-to-IT-stack*' caused unease. In particular Graves [Graves 2008], Fehskens [Fehskens 2008] as well as Wagter et al [Wagter 2009] have argued that such a Business-to-IT-stack centricity is a major weakness of contemporary enterprise architecture approaches, and that enterprise architecture should involve many more aspects of an organization, such as a clear connection to its strategy, its financial structures, the abilities of its work force, etc. More specifically, Wagter et al [Wagter 2009] argue that enterprise architecture should not just be concerned with Business-IT alignment, but rather with the alignment of all relevant aspects of an enterprise. Therefore, rather than using the term *alignment*, Wagter et al [Wagter 2009] suggest to use the term *enterprise coherence* to stress the multi-faceted nature.

A first enterprise architecture method to indeed explicitly move beyond a Business-to-IT-stack centricity is the GEA method [Wagter 2009]. GEA argues that the coherence between several aspects of an enterprise needs to be governed explicitly by means of an enterprise architecture. To indeed co-design the different aspects of an enterprise architecture, and to use it (both the co-design process, and the resulting architecture) in governing enterprise coherence, it is necessary to take the concerns and associated strategic dialogues of senior management as a starting point. In other words, the way in which architecture is integrated into the strategic dialogue should take the concerns, language, and style of communication of senior management as a starting point, and not the typical domains, layers, or columns, as identified in the traditional architecture frameworks.

The shift from Business-to-IT-stack centricity to the broader notion of enterprise coherence also required a change in perspective on change processes in organizations [Wagter, Proper, et al. 2011]. De Caluwé and Vermaak [De Caluwé and Vermaak 2003] have identified a number of core perspectives on change processes in organizations:
1: *Yellow-print thinking*: Bring the interests of the most important players together by means of a process of negotiation enabling consensus or a win-win solution.
2: *Blue-print thinking*: Formulate clear goals and results, then design rationally a systematic approach and then implement the approach according to plan.
3: *Red-print thinking*: Motivate and stimulate people to perform best they can, contracting and rewarding desired behaviour with the help of HRM-systems.
4: *Green-print thinking*: Create settings for learning by using interventions, allowing people to become more aware and more competent on their job.
5: *White-print thinking*: Understand what underlying patterns drive and block an organization’s evolution, focusing interventions to create space for people’s energy.

As argued in [Wagter, Proper, et al. 2011], most traditional approaches and frameworks, including the Zachman [Sowa and Zachman 1992] and IAF [Wout et al. 2010] frameworks, the ArchiMate [Iacob et al. 2012; e. a. Lankhorst M. 2012] language, as well as the DYA [Wagter et al. 2005] and TOGAF [The Open Group 2009] architecture methods, essentially take a Blue-print perspective on change. The need to really involve senior management, however, suggests the use of another style of thinking, involving internal or external stakeholder interests, strategy formulation processes, formal and informal power structures, and the associated processes of creating win-win situations and forming coalitions. In terms of De Caluwé and Vermaak [De Caluwé and Vermaak 2003] this would suggest to complement the Blue-print perspective with the Yellow-print perspective, and arguably also a mix of the other perspectives.

In the development of the GEA method [Wagter 2009], this line of thinking was taken as a starting point. As a result, the actual political power structures, and associated strategic dialogues, within a specific enterprise were taken as a starting point, rather than the frameworks suggested by existing architecture approaches. This leads to enterprise specific frameworks of *coherence governance perspectives*, to manage enterprise coherence. For example, in terms of ‘mergers & acquisitions’, ‘human resourcing’, ‘clients’, ‘regulators’, ‘culture’, ‘intellectual property’, ‘suppliers’, etc. The existing Blue-print oriented frameworks can still be used to further structure the dialogue between the coherence governance perspectives, especially where it concerns issues pertaining to the Business-to-IT-stack.

It is to be expected that organizations aiming to use enterprise architecture to steer major transformations, will increasingly move from a Business-to-IT-stack centricity perspective to an enterprise coherence perspective on their enterprise architectures.

2.4 From Big-Design-Up-Front to Fit-for-Purpose

Early frameworks and languages for enterprise architecture [e. a. Lankhorst M. 2012; The Open Group 2005; Wout et al. 2010; Zachman 1987] were primarily concerned with the identification of the aspects, concepts and domains that should be included in an architecture; hence the resulting *content frameworks*. This orientation brings along the risk that architects focus more on completeness of architecture descriptions, rather than on ensuring that the descriptions meet the purposes for which they are actually needed. Accepted standards for defining architecture, such as the earlier quoted IEEE 1471 [IEEE 2000]: "*the fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution.*" do not provide a clear 'stop criterion' for architects that allows them to provide just enough architecture. This definition points primarily at what the things are that an architecture is concerned *about*: "*its components, their relationships to each other and to the environment, and the principles guiding its design and evolution*". The risk is that inexperienced (and method obeying) architects loose themselves in meticulous designs of the future enterprise. The reference to "*the fundamental organization*" only implicitly refers to the purpose for having an architecture, i.e. understanding or expressing the fundamental organization of a system. But why? And what part of *organization* is to be regarded as *fundamental*? This is of course dependent on the *purpose* for which the architecture (description) is created. The more recent ISO [ISO 2011] version of this definition: "*fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution*", does not remedy this.

In our observation, the focus on completeness indeed quite often results in overly-detailed architecture descriptions, involving long lists of architecture principles, meticulously worked out models for each of the cells from the architecture framework used, etc. This situation triggered the agile software development community to talk about [Ambler and Jeffries 2002; Beck et al. 2001; Cockburn 2002; M. Lankhorst 2012] "*Big-Design-Up-Front*" (BDUF). Of course, experienced architects knew when to *stop* architecting. However, early architectural approaches did not provide clear guidelines to ensure that architectures stayed *Fit-for-Purpose*, and rather invited architects to be over-complete.

The need to tune an enterprise architecture to the purpose at hand and avoid overly detailed architectures, triggered the authors of [Wagter et al. 2001, 2005] to create the DYnamic Enterprise Architecture approach, which incorporates notions such as "*just enough architecture*", resembling the ideas that were also put forward (in parallel) by the agile system development community. The most recent version of TOGAF [The Open Group 2009] also provides indications for different (purpose specific) ways to use its ADM to ensure the resulting architecture descriptions are indeed fit-for-purpose.

In [Greefhorst and Proper 2011] the authors suggest to make a clear distinction between:
1: The *purpose* that an enterprise architecture serves. For example, to understand (make sense of) the current/past situation of an enterprise in terms of its fundamental properties and concepts, to articulate and motivate (make sense of) a desired future situation in terms of fundamental properties and concepts.
2: The *meaning* of an enterprise architecture as an artefact. For example, to express (for some purpose) the fundamental properties and/or concepts that underly the present structure of an enterprise, or to express the fundamental properties and/or

concepts that should inspire, guide, or steer, the evolution towards the future.
3: The *elements* of an enterprise architecture in terms of the typical concepts used to capture this meaning, such as its elements, relationships, and the principles of its design and evolution as mentioned by the IEEE and ISO definitions, which may be captured by means of models and views.

This distinction enables a clear top-down reasoning on the level of detail and completeness needed from an architecture description. Given the purpose of a specific architecture (description), one can identify the desired meaning of the architecture, and following that, the kinds of elements needed to capture/express this meaning. For example, in [Greefhorst and Proper 2011] the authors focus on using enterprise architecture for the *purpose* to align the enterprise to its essential requirements and ultimately its strategy: "... *the main purpose of an enterprise architecture is to align an enterprise to its essential requirements. As such, it should provide an elaboration of an enterprise's strategy to those properties that are necessary and sufficient to meet these requirements*". Even though it is only normative in nature, the "*necessary and sufficient*" and the reference to the enterprise's strategy provide a (possible) stopping criterion to keep an architecture Fit-for-Purpose (i.c. steering transformations that aim to establish an enterprise's strategy changes).

2.5 From a Constructing to a Constraining Perspective

The shift from computer architecture to information systems architecture, and then to enterprise architecture at large, also resulted in an increase of scope of architecture efforts. Where at the start the focus was typically on a limited number of applications in support of an information system, the organizational scope gradually broadened to business-unit wide, then to enterprise wide, or sometimes even to a sector/branch wide scope. At the same time, the potential time-horizon for architectures increased, from focusing on the situation after the next development stage, to mid-term and longer-term planning activities covering several intermediary stages. The shift from Business-to-IT-stack centricity to more overall enterprise coherence also resulted in a wider range of aspects to be covered in an architecture.

The resulting increase in scope and complexity, combined with the Big-Design-Up-Front to Fit-for-Purpose trend as discussed in the previous section, resulted in the awareness that another means was needed next to the traditional architecture descriptions involving the enterprise's construction in terms of actual building blocks (value exchanges, transactions, business processes, actors, objects, roles, collaborations, etc). This resulted in a strengthening of the role of *architecture principles* as a way to translate an enterprise's strategic intentions to more specific directing/guiding statements, without immediately 'jumping' to the use of actual building blocks of an actual (high level) design. Several architecture approaches indeed position architecture principles as an important element of enterprise architecture [Beijer and De Klerk 2010; Davenport et al. 1989; Op 't Land, Proper, et al. 2008; Richardson et al. 1990; Tapscott and Caston 1993; The Open Group 2009; Wagter et al. 2005; Wout et al. 2010], while some authors even go as far as to position principles as being the essence of architecture [Dietz 2008; Fehskens 2010; Hammer & Company 1986; Hoogervorst 2009]. In our view, the challenges of dealing with increased scope and complexity really emancipated the role of principles as ways to constrain design space.

Fundamentally, we can see a shift from considering an architecture as being primarily concerned with *constructing* the (high level) design of an enterprise in terms of building blocks to being concerned with *constraining* the space of allowable/desirable constructions. A prime example of an architecture from a *constraining* point of view is the NORA [Nederlandse Overheid Referentie Architectuur (NORA) 2012] reference architecture for the Dutch government. It focuses primarily on architecture principles that should be applied in the elaboration of more specific architectures and designs.

It is important to note that the distinction between *constructing* an assembly of building blocks and *constraining* the set of possible assemblies to an allowable/desirable subset, is orthogonal to the *deontic modality*[2] of an architectural description. This refers to the question if the architectural description is, for example, intended to be a suggestion (*could*), guidance (*advisable*), indicative (*should*), or a pure directive (*must*).

2.6 From Building to Integrating

Another trend also resulted in a similar shift towards to the constraining of design space. Instead of developing their own software, most organizations today use packaged solutions, cloud services and other pre-defined solutions to support large parts of their business activities. These solutions may be configured with the organization's business rules, business processes, information models, etc., but they inherently limit the design freedom of the architect. The upside, of course, is in the common gains of re-use: lowering costs and risks, and speeding up development.

This trend, combined with the growing scope and complexity outlined in the previous section, also leads to a growing emphasis on the integration of various business processes and IT components, within and across organizations. Anyone who has spent some time in a large organization will recognize that the most common and at the same time most pernicious problems in architecture are at these integration points. The service-oriented architecture (SOA) paradigm [Erl 2005] was an important attempt to alleviate this problem, but has not been the panacea that it was once thought to be.

This shift towards integration also influences the design and development process. Whereas in the past, a large system was often designed in one go and as a single, coherent whole, an integrative approach will need to be more piecemeal and iterative: adding and integrating various components one-by-one.

2.7 From One-shot to Iterative Approaches

The agile movement in software development [Ambler and Jeffries 2002; Cockburn 2002] has received much attention over the last two decades. Light-weight, iterative methods have gradually taken over much of the software development community. Since the 1990s, evidence has been mounting that agile ways of working, using short iterations and close customer contact, have a higher success rate than traditional, waterfall-like methods for software development, at least for many types of software projects. Recent studies provide theoretical and empirical evidence for the effectiveness of agile methods; see for example the extensive overview by Lee and Xia [Lee and Xia 2010].

The Agile Manifesto values "*responding to change over following a plan*" [Beck et al. 2001]. Many proponents of agile methods are opposed to the use of architecture, categorically classifying it as Big-Design-Up-Front. They argue that stakeholders cannot know what they really need and the problem will change anyway before the project is completed, so one cannot provide any useful designs up-front. Moreover, the changing business environment makes stable requirements an illusion to begin with. Hence, complex socio-technical systems cannot be designed solely behind the drawing board.

On the other hand, many architects and managers resist the agile movement, arguing that one should think before planning actions and building systems. They fear a loss of control and claim that all these agile projects will build their own silos, resulting in the same fragmentation of IT landscapes that the architecture discipline promised to fix.

[2] See for example `http://en.wikipedia.org/wiki/Deontic_modality`

Both positions are misguided about the role of architecture. A well-defined architecture helps in positioning new developments within the context of the existing processes, IT systems, and other assets of an organization, and in identifying necessary changes. A good architecture and infrastructure is an up-front investment that makes later changes easier, faster and cheaper, and a good architectural practice helps an organization innovate and change by providing both stability and flexibility [M. Lankhorst 2012]. But this does not mean that everything should be architected up-front. As addressed in sections 2.4 and 2.5, a good enterprise architecture is not overly detailed and focuses on the essential inspiration and guidance needed to foster enterprise-wide coherence.

Architecture processes in many organizations still give the impression that architects should do all the thinking beforehand and software developers and others can only start their work after the architects are done. Methods like TOGAF's ADM [The Open Group 2009] are also easily interpreted in this way. The measurable success of agile methods and related developments such as continuous delivery [Humble 2010] creates an increasing need for the architecture discipline to follow suit and embrace a more iterative way of working, closely tied to the entire development process and not merely as a starting phase.

The trend towards less detailed and more normative enterprise architecture, as outlined in sections 2.4 and 2.5, matches well with this need for an iterative approach. Agile enterprise architects provide assistance to projects to help them fit within the big picture, while refraining from too much and too detailed guidance. Moreover, as Ciborra [Ciborra 1992] argued, bricolage, emergence and local improvisation, instead of central control and top-down design, may lead to strategic advantages: the bottom-up evolution of socio-technical systems will lead to something that is deeply rooted in an enterprise's organizational culture, and hence much more difficult to imitate by others. Agile enterprise architects leave room for such local, bottom-up improvements and fit these within the larger scheme of things.

3 Future trends

In this section we discuss the anticipated future of enterprise architecture in terms of a number of anticipated trends.

3.1 From IT to IT

In most enterprises the role of IT started with the 'automation of administrative work'. In modern day organizations, there continues to be a clear role for IT to automate administrative information processing. However, the use of IT has moved far beyond this. In some situations, IT has given rise to new social structures, and business models. Consider, for example, the development of social media, the (acclaimed) role of twitter in time of social unrest, the emergence of on-line music stores, app-stores, music streaming services, etc. The advent of 'big data' [Hurwitz et al. 2013] is expected to drive such developments even further by allowing IT based systems to use statistical data to tune their behaviour to observed and learned trends.

At the same time, IT is becoming firmly embedded in existing technological artefacts. The cars in which we drive now contain more lines of code than typical banking applications do. The next generation of cars will even be able to (partially) do the driving for us. The so-called smart (power) grid, is likely to lead to the 'smartening' of household appliances. Our houses are already being vacuumed by dedicated robots, while in some cases robots even play a role in the care of elderly people [Tamura et al. 2004]. The military use of all sorts of drones also spearheads more peaceful applications of such self-reliant devices that can e.g. perform tasks on behalf of us in hostile or unpleasant environments.

In sum, we argue that we are moving towards smart and more 'sociable' technology that is enabled by computer technology. One might indeed say, from *information* technology to *intelligent* technology, i.e. from IT to IT. When architecting modern day enterprises, one should treat these as (evolving) collectives of human actors and computerised actors, where the latter might operate in a pure software world, or might be embedded/embodied in other forms of (connected) technology. Needless to say, however, that human actors will always need to remain (socially and legally) responsible for the actions of the computerised actors that operate on their behalf.

3.2 From Syntax to Semantics

The trend towards an increased scope of integration, described in subsection 2.6, brings its own set of design issues. Although paradigms such as service orientation promised to facilitate this integration, they function mainly on a syntactic level, providing a stack of interconnection standards for software systems.

When the integration scope grows, the associated semantic problems grow as well. The information shared across organizational borders may be interpreted in ways that were not intended and do not match with the context in which this information originated. The same holds for the behavioural semantics of cross-border business processes. The Semantic Web [W3C Semantic Web Activity 2013] provides some partial solutions, but the premise of its methods is the unification of semantics in a single overarching ontology, basically trying to standardize the meanings of information. It is simply not feasible to build such ontologies for the size and variety of real-world integration problems. Local variety in semantics cannot be avoided or ‘standardized away’, because of the inevitable loss of meaning this causes.

This problem is exacerbated by the rapidly growing volume, variety and velocity of ‘big data’ [Hurwitz et al. 2013], as already mentioned in subsection 3.1. Applying statistical methods will not suffice to create meaningful interpretations. This implies that novel methods are needed for architecting the semantics of information and behaviour, taking into account the variety and context of meaning and the social processes needed to create understanding and agreement at different scales. It is not feasible to provide complete top-down designs for large-scale socio-technical systems, as we have already argued in subsection 2.7. The shift from building towards integration (subsection 2.6), also puts more emphasis on the need for semantic interoperability. Different semantic backgrounds in a multi-organizational setting make this even more complicated. We need gradual, iterative approaches for coherent and collaborative design, development and deployment of these socio-technical systems.

3.3 From State-thinking to Intervention-thinking

We argue that contemporary approach to architecture ‘think’ in terms of *as-is* and *to-be* states of the enterprise. Some approaches may indeed go as far as identifying several intermediary stages between *as-is* and *to-be*, e.g. leading to the concept of *transition architecture* in TOGAF [The Open Group 2009] and plateaus in ArchiMate [Iacob et al. 2012; e. a. Lankhorst M. 2012]. What remains common, however, is the focus on several *states* of the (construction of the) enterprise. This state-oriented thinking might have worked well in the past when the focus was on architecting an enterprise’s IT support. However, as soon as other other aspects are taken into consideration, the story becomes more complicated.

As soon as non-technological aspects are taken into consideration, this brings about a shift of focus from *technical* systems to *socio-technical* systems involving a mix of human and technological actors. The enterprise and its environment, being socio-technical systems, will evolve out of themselves. People working in an enterprise will make changes to the ‘design’ of the enterprise, if only to make the ‘design’ (continue to) work in day-to-day practise. The people making up the organization, collectively ‘author’ their enterprise [Taylor and Van Every 2010].

Even without the use of architecture as a planning instrument, there are likely to be a plethora of projects and related efforts that will continuously change the enterprise in response to external and/or internal stimuli. Some of these changes might not even be 'visible' as projects, as they are based on local initiatives taken within the operational processes (i.e. actors switching between a role in the operational capability to the transformation capability).

We argue that a shift is needed from thinking of enterprise transformations as being a change of an enterprise from one state (the *as-is*) to a future state (the *to-be*), but rather as primarily being an intervention in the natural evolution of the enterprise, resulting in a changed course of its evolution towards a presumably more desirable direction. So, from an as-is *trajectory* to a to-be trajectory.

For the focus of an enterprise architecture this would lead to an even stronger emphasis of the *constructing to constraining* trend as discussed in subsection 2.5, as constraints are more suitable to articulate desired trajectories than specific building blocks. Using e.g. *architecture principles* enterprises can distinguish between desirable and less desirable directions of its evolution, and from that infer interventions that can be undertaken to drive, or lure, the natural evolution of the enterprise in the desired direction. These interventions might indeed involve (re-)constructions of building blocks of the enterprise.

3.4 From Operational Capability to Transformation Capability

In line with the previous trend, an enterprise is likely to evolve continuously. The capabilities needed to change an enterprise are quite different from the capability needed to run its day-to-day business. The latter capabilities of an enterprise can be referred to collectively as its *operational capability*, while the capabilities needed to transform itself are the *transformation capability*. Teece et al [Teece et al. 1997] stress the need for modern day organizations to have a transformation capability that meet its rapidly changing environment, leading to a highly dynamic transformation capability: "*the firm's ability to integrate, build, and reconfigure internal and external competences to address rapidly changing environments*". Teece et al [Teece et al. 1997] refer to this dynamic transformation capability as "*dynamic capability*".

It is important to realize that the humans involved in an enterprise can play a role towards both the operational capability and the transformation capability simultaneously. For human beings this is actually quite natural. While executing our daily activities, we typically also learn how to do these activities better and/or adapt them to changing needs/circumstances. In these cases, we decide to 'on the fly' innovate our operational capability. In doing so, we (briefly) use our transformation capability. As a consequence, it is advised to regard the operational capability and transformation capability of an enterprise as *aspect systems* and not as *sub systems*.

When considering an enterprise from an architectural perspective, one can of course opt to focus the architecture efforts on one of these capabilities or both. In most cases that we know of, as well as the illustrating case studies discussed in the existing architecture approaches, the focus is on architecting the *operational capability* only. An exception would be enterprises who have created a so-called *development architecture* focusing on the way the enterprise will go about developing new information systems. An example is the development architecture from the Dutch Tax Administration [Achterberg et al. 2000].

Whether an enterprise's architecture effort should focus on the operational capability and/or the transformation capability depends on the enterprise's strategy. For example, in terms of the Discipline of Market Leaders from Treacy and Wiersema [Treacy and Wiersema 1997], it would be logical for enterprises focusing on:
1: *operational excellence*, that the operational capability requires architecting priority,

2: *product leadership*, that the parts of the transformation capability dealing with product/service innovation require architecting priority,
3: *customer intimacy*, that the parts of the operational capability and the transformation capability that deal with client interaction require architecting priority.

When indeed also architecting the transformation capability, it is again recommendable to realize that the operational and transformation capabilities are *aspect systems*, and that the different actors (be they human or be they technology) can play roles towards both capabilities simultaneously.

In recent work [M. Lankhorst 2012] on agile service development, it was also argued that an agile services context requires enterprises to move from having only an efficient operational capability to an effective combination of operational and transformation capabilities. One should focus on designing the operational capability in such a way that it lends itself to quick changes within given boundaries and ambitions, while the transformation capability should be designed in such a way that it can use this built in agility of the operational capability to meet anticipated changes in the environment, as well as the ability to take appropriate actions to transform the operational capability when having to meet unanticipated changes (in terms of Teece, it would have to be *dynamic*).

In [M. Lankhorst 2012] some guidelines are provided on how to balance an architecting effort between the transformation and operational capabilities. However, more research is needed. At the same time, the need for enterprises to be agile, does stress the need to be able to make explicit tradeoffs on how to deal with this agility across the two capabilities.

3.5 From Intuition-based to Evidence-based Management

Modern day enterprises need to change in order to survive. At the same time they need to do so in the face of an increasing number of regulations on compliance and transparency. Furthermore, a considerable part of an enterprise's shareholder value is '*tied*' up in the needed transformations. As a consequence, the processes needed to transform the enterprise become a *core* business process themselves, requiring ample management attention.

In addition, due to the increasing amount of shareholder value (and/or taxpayer's money) that is tied up in such transformations, one can expect that the requirements on the transparency with which such decision are made, will increase. Would it not be logical for companies that are listed on the stock market, to also report annually on their ability to transform in an effective way? In other words, not just how well their operational capability is able to earn a revenue for its shareholders, but also how well their transformation capability is able to ensure the continuation of this revenue in a cost-effective way?

In this sense, one can expect that senior management will increasingly be held responsible (by shareholders, tax payers, and ultimately auditors) for their ability to steer and control transformations. Even more, senior management should not only worry about the cost effectiveness of change, but also about governance, risk management, compliance, etc., associated to these transformations. Given the earlier discussion on the purpose of enterprise architecture, and its role for informed governance, it shall not be surprising that we take the point of view that enterprise architecture would indeed provide a means to senior management to take more control over the transformations and the associated decision making on the future of the enterprises for which they are *responsible*. Using enterprise architecture, one can more crisply analyse problems in an existing situation, articulate desired directions (using architecture in a prescriptive way), analyse the costs and benefits of different options (using architecture in a more descriptive way), and guard that transformation projects are indeed moving in the desired direction.

In parallel to this, one can also observe an interesting trend in the field of management. As argued in [Pfeffer and Sutton 2006, 2011], there is an increasing call for evidence-based management instead of (yet not fully replacing) intuition-based management. The authors draw an interesting analogy to the trend in medicine towards evidence-based medicine [Evidence-Based Medicine 2012], which is defined in [Sacket et al. 1996] as: "*the conscientious, explicit and judicious use of current best evidence in making decisions about the care of individual patients.*". If you think that doctors would always base their diagnose on sound evidence and reasoning, then [Pfeffer and Sutton 2011] invites us to rethink this.

When considering the promise of evidence-based management, there is indeed a strong analogy to the potential contribution of enterprise architecture. Some early examples of how enterprise architecture can be used for evidence-based management of enterprise transformation can be found in e.g. [Op 't Land 2006, 2007; Op 't Land and Dietz 2008]. We indeed argue that enterprise architecture can become a leading mechanism in enabling evidence-based management of transformations. Or rather, the field of enterprise architecture should take upon it as its *mission* to enable evidence-based management of transformations. We explicitly use the word *enable* to stress the fact that it is senior management who have to take the *responsibility* for making decisions based on evidence. It remains their choice not to take that responsibility, and explain to the shareholders, tax payers and auditors, why they did not.

4 Redefining Enterprise Architecture

Based on the future trends as identified in the previous section, we will now revisit our understanding of enterprise architecture. In line with the definition provided in [Greefhorst and Proper 2011] we regard architecture as essentially being about: "*Those properties of an artefact that are necessary and sufficient to meet its essential requirements*". This view is shared by Fehskens [Fehskens 2008], who defines architecture as "*those properties of a thing and its environment that are necessary and sufficient for it to be fit for purpose for its mission*". The focus on the properties *that matter*, is also what distinguishes architecture from design. It also resonates well with the reference to *fundamental organization* in the original IEEE definition [IEEE 2000] and the reference to *fundamental concepts* in the ISO definition [ISO 2011].

The reference to properties that are necessary and sufficient to meet its *essential requirements* does indeed introduce a strong form of relativity to architecture: *Who/what determines what the essential requirements are?* We argue that these essential requirements follow from the key stakeholders and their core concerns. *What concerns them most about the artefact?* In the case of an enterprise, the essential requirements can be linked directly to the enterprise's (past/current) strategy, next to other core concerns of the key stakeholders [Greefhorst and Proper 2011]. As such, we argue that enterprise architecture should first and foremost be about *essential sensemaking* in that it should primarily:
1: make sense of the past and future of the enterprise with regards to the way it has/will meet its *essential requirements* as put forward by its core stakeholders and captured in its strategy,
2: provide clear motivations/rationalisation, in terms of the above essential requirements, as well as e.g. constraints, of the trade-offs that underly the presence of the elements (e.g. building blocks or architecture principles) included in the architecture.

In line with this, we argue that the *purpose*, *meaning* and *elements* of an enterprise architecture should evolve:
1: Its *purpose* is (i) to understand the current evolution of the enterprise, including its past and its likely future evolution and (ii) formulate, as well as motivation/rationalise, the desired future evolution and the interventions needed to achieve this.
2: Its *meaning* is that it expresses, in relation to the (current) essential requirements: (i) the understanding how the enterprise has evolved so-far, (ii) what the expected natural evolution

of the enterprise is and (iii) the desired future evolution of the enterprise and actions needed to change/strengthen its current evolution.
3: Its *elements* will focus on the fundamental properties that have played a role in its past evolution, as well as its expected/desired future evolution. These properties can be expressed from a *constraining* perspective in terms of architecture principles and/or from a *constructing* perspective in terms of the building blocks of the enterprise.

It is important to note that during the evolution of an enterprise, it is likely that the understanding of what the *essential requirements* are will change. This means that the boundary between what was included in the architecture and what is considered design may also shift over time. For the modelling languages used (be it from a *constructing* or a *constraining* perspective), this means that they should better take a broad perspective focus on enterprise modelling in general, where what is considered to be "architecturally relevant" may shift over time; modelling approaches with a narrow view of what is "proper" architecture may find themselves obsolete before they know it.

5 Conclusion

In this position paper we discussed our view on the history, and the potential future evolution, of the field of enterprise architecture. It is our firm belief that enterprise architecture can, and *should*, play a crucial role in enabling senior management of enterprises to take their responsibility in steering, controlling and/or guiding enterprise transformations, based on sensemaking and evidence-based insights. It is certainly one of the driving hypotheses in our work.

We suggest that future research into the enterprise architecture domain should do so from at least three important vantage points, that are also likely to need different types of research methodologies:
1: An *engineering* perspective that focuses on strategies, methods and techniques to provide evidence-based underpinning of the design decisions underlying enterprise architectures (both in the constructing and the constraining sense).
2: A *modelling* perspective focussing the role of the different models, frameworks, modelling languages, model transformations, and associated modelling processes for enterprise architecture.
3: A *sociological* perspective concerned with the role of culture, skills, attitudes, communication, etc, needed/involved during the formulation of an enterprise architecture, as well as in the intervention needed to establish the changes proposed by a future architectural direction.

6 Acknowledgements

This work has been partially supported by the *Fonds National de la Recherche Luxembourg*, via the ASINE *(Architecture-based Service Innovation in Network Enterprises*), ACET (*Architecture-based Coordination of Enterprise Transformations*) and RationalArchitecture projects.

References

Achterberg R. v., Frankema B., Jong-Ellenbroek M. d., Molen P. v. d., Proper H., Schut W. (Mar. 2000) Handleiding SysteemConcept en ApplicatieArchitectuur – Startarchitectuur. Technical Report Version 2.0. Dutch Taxation Office. Last Access: In Dutch

Ambler S., Jeffries R. (2002) Agile Modeling: Effective Practices for Extreme Programming and the Unified Process. John Wiley & Sons, New York, New York

Amdahl G., Blaauw G., Brooks F. (Apr. 1964) Architecture of the IBM System/360. In: IBM Journal of Research and Development

Beck K., Beedle M., Bennekum A. v., Cockburn A., Cunningham W., Fowler M., Grenning J. Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R., Mellor S., Schwaber K., Sutherland J., Thomas D. (2001) Manifesto for Agile Software Development. http://www.agilemanifesto.org Last Access: Accessed 14 June 2013.

Beijer P., De Klerk T. (2010) IT Architecture: Essential Practice for IT Business Solutions. Lulu

Boar B. (1999) Constructing Blueprints for Enterprise IT architectures. Wiley, New York, New York

Buckl S., Matthes F., Schweda C. (2011) A Method Base for Enterprise Architecture Management. In: Ralyté J., Mirbel I., Deneckere R. (eds.) Engineering Methods in the Service-Oriented Context: 4th IFIP WG 8.1 Working Conference on Method Conference on Method Engineering, ME 2011, Paris, France, April 20-22, 2011. Springer, Berlin, Germany, pp. 34–48

Ciborra C. (1992) From Thinking to Tinkering: The Grassroots of Strategic Information Systems. In: The Information Society 8, pp. 297–309

ESPRIT Consortium AMICE (1993) CIMOSA: Open System Architecture for CIM, 2nd. Springer, Heidelberg, Germany

Cockburn A. (2002) Agile Software Development: The Cooperative Game, 2nd. Addison Wesley

Davenport T., Hammer M., Metsisto T. (Mar. 1989) How executives can shape their company's information systems. In: Harvard Business Review 67(2), pp. 130–134

De Caluwé L., Vermaak H. (2003) Learning to Change: A Guide for Organization Change Agents. Sage publications, London, United Kingdom

Dietz J. (2006) Enterprise Ontology – Theory and Methodology. Springer, Berlin, Germany

Dietz J. (2008) Architecture – Building strategy into design. Netherlands Architecture Forum, Academic Service – SDU, The Hague, The Netherlands http://www.naf.nl

Evidence-Based Medicine. http://en.wikipedia.org/wiki/Evidence-based_medicine Last Access: Last visited 02-10-2012

Erl T. (2005) Service-Oriented Architecture (SOA): Concepts, Technology, and Design.. Prentice Hall

Fehskens L. (Oct. 2008) Re-Thinking architecture. In: 20th Enterprise Architecture Practitioners Conference. The Open Group

Fehskens L. (July 2010) What the "Architecture" in "Enterprise Architecture" Ought to Mean. In: Open Group Conference Boston. The Open Group

Goedvolk J., Bruin H. d., Rijsenbrij D. (1999) Integrated Architectural Design of Business and Information Systems. In: Bosch J. (ed.) Proceedings of the Second Nordic Workshop on Software Architecture (NOSA'99). vol. 1999. Research Report 13 University of Karlskrona/Ronneby, Ronneby, Sweden

Graves T. (2008) Real Enterprise Architecture: beyond IT to the whole enterprise. Tetradian Books, Colchester, England, United Kingdom http://tetradianbooks.com

Greefhorst D., Proper H. (2011) Architecture Principles – The Cornerstones of Enterprise Architecture Enterprise Engineering Series. Springer, Berlin, Germany

Hammer & Company (1986) PRISM: Dispersion and Interconnection: Approaches to Distributed Systems Architecture, Final Report.. CSC Index, Inc.. Cambridge MA

Hammer M. (Apr. 1990) Re–engineering work: don't automate, obliterate. In: Harvard Business Review 68(4), pp. 104–112

Henderson J., Venkatraman N. (1993) Strategic alignment: Leveraging information technology for transforming organizations. In: IBM Systems Journal 32(1), pp. 4–16

Hoogervorst J. (2009) Enterprise Governance and Enterprise Engineering. Springer, Berlin, Germany, Berlin, Germany

Humble J. (2010) Continuous delivery: reliable software releases through build, test, and deployment automation. Addison Wesley

Hurwitz J., Nugent A., Halper F., Kaufman M. (2013) Big Data For Dummies For Dummies. John Wiley & Sons Inc.

Iacob M.-E., Jonkers H., Lankhorst M., Proper H., Quartel D. (2012) ArchiMate 2.0 Specification. The Open Group

IEEE (Sept. 2000) Recommended Practice for Architectural Description of Software Intensive Systems. IEEE P1471:2000, ISO/IEC 42010:2007. The Architecture Working Group of the Software Engineering Committee, Standards Department, IEEE. Piscataway, New Jersey

ISO (1996a) Information technology – Open Distributed Processing – Reference model: Architecture ISO/IEC 10746–3:1996(E)

ISO (1996b) Information technology – Open Distributed Processing – Reference model: Foundations ISO/IEC 10746–2:1996(E)

ISO (1998a) Information technology – Open Distributed Processing – Reference model: Architectural semantics ISO/IEC 10746–4:1998(E)

ISO (1998b) Information technology – Open Distributed Processing – Reference model: Overview ISO/IEC 10746–1:1998(E)

ISO (July 2011) Systems and software engineering - Architecture description is an international standard for architecture descriptions of systems and software.. ISO/IEC 42010. ISO

Jonkers H., Veldhuijzen van Zanten G., Buuren R. v., Arbab F., De Boer F., Bonsangue M., Bosma H., Ter Doest H., Groenewegen L., Guillen Scholten J., Hoppenbrouwers S., Iacob M.-E., Janssen W., Lankhorst M., Van Leeuwen D., Proper H., Stam A., Torre L. v. d. (Sept. 2003) Towards a Language for Coherent Enterprise Architecture Descriptions. In: Steen M., Bryant B. (eds.) 7th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2003), Brisbane, Queensland, Australia. IEEE, Los Alamitos, California, pp. 28–39

Lahrmann G., Winter R., Fischer M. (Nov. 2010) Design and Engineering for Situational Transformation. In: Harmsen A., Proper H., Schalkwijk F., Barjis J., Overbeek S. (eds.) Proceedings of the 2nd Working Conference on Practice-driven Research on Enterprise Transformation, PRET 2010, Delft, The Netherlands. vol. 69. Lecture Notes in Business Information Processing Springer, Berlin, Germany, pp. 1–16

Lankhorst M.M. e. a. (2012) Enterprise Architecture at Work: Modelling, Communication and Analysis, 3rd Enterprise Engineering Series. Springer, Berlin, Germany

Lankhorst M. (ed.) Agile Service Development: Combining Adaptive Methods and Flexible Solutions Enterprise Engineering Series. Springer, Berlin, Germany

Lee G., Xia W. (Mar. 2010) Toward Agile: An integrated analysis of quantative and qualitative field data on software development agility. In: MIS Quarterly 34(1), pp. 87–114

Nederlandse Overheid Referentie Architectuur (NORA). http://www.e-overheid.nl/onderwerpen/e-overheid/architectuur/nora-familie/nora Last Access: Last checked: 02-10-2012

Olle T., Sol H., Tully C. (eds.) Information Systems Design Methodologies: A feature analysis North–Holland/IFIP WG8.1, Amsterdam, The Netherlands

Olle T., Sol H., Verrijn–Stuart A. (eds.) Information Systems Design Methodologies: A Comparative Review. North–Holland/IFIP WG8.1, Amsterdam, The Netherlands

Op 't Land M. (Oct. 2006) Applying Architecture and Ontology to the Splitting and Allying of Enterprises: Problem Definition and Research Approach. In: Meersman R., Tari Z., Herrero P. (eds.) On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops - OTM Confederated International Workshops and Posters, AWESOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Acadamy Doctoral Consortium, RDDS, SWWS, and SebGIS, Proceedings, Part II, Montpellier, France. vol. 4278. Lecture Notes in Computer Science Springer, Berlin, Germany, pp. 1419–1428

Op 't Land M. (Sept. 2007) Towards Evidence Based Splitting of Organizations. In: Ralyté J., Brinkkemper S., Henderson-Sellers B. (eds.) Proceedings of the IFIP TC8 / WG8.1 Working Conference on Situational Method Engineering: Fundamentals and Experiences (ME07), Geneva, Switzerland. vol. 244. IFIP Series Springer, Berlin, Germany, pp. 328–342

Op 't Land M., Dietz J. (Mar. 2008) Enterprise ontology based splitting and contracting of organizations. In: Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC'08), Fortaleza, Ceará, Brazil.

Op 't Land M., Proper H., Waage M., Cloo J., Steghuis C. (2008) Enterprise Architecture – Creating Value by Informed Governance Enterprise Engineering Series. Springer, Berlin, Germany

Pfeffer J., Sutton R. (Jan. 2006) Evidence-Based Management. In: Harvard Business Review Last visited: 02-10-2012 http://hbr.org/2006/01/evidence-based-management/ar/1

Pfeffer J., Sutton R. (2011) Trust the Evidence, Not Your Instincts. In: New York Times New York edition, September 4th, BU8 http://www.nytimes.com/2011/09/04/jobs/04pre.html?_r=2&ref=business&

Proper H. (2012) Enterprise Architecture – Growing up to evidence-based management? In: Baldinger F., Rijsenbrij D. (eds.) Netherlands Architecture Forum, The Netherlands chap. 8.1, pp. 317–328

Richardson G., Jackson B., Dickson G. (1990) A Principles-Based Enterprise Architecture: Lessons from Texaco and Star Enterprise. In: MIS Quarterly 14(4), pp. 385–403 http://www.jstor.org/stable/249787

Rivera R. (2007) Am I Doing Architecture or Design Work? In: It Professional 9(6), pp. 46–48

Ross J., Weill P., Robertson D. (2006) Enterprise architecture as strategy: creating a foundation for business execution. Harvard Business School Press, Boston, Massachusetts

Sacket D., Rosenberg W., Gray J., Haynes R., Richardson W. (Jan. 1996) Evidence based medicine: what it is and what it isn't. In: British Medical Journal 312, pp. 71–72 http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2349778/pdf/bmj00524-0009.pdf

Scheer A.-W. (1986) Neue Architektur für EDV-Systeme zur Produktionsplanung und -steuerung In German. Institut für Wirtschaftsinformatik im Institut für Empirische Wirtschaftsforschung an der Universität des Saarlandes, Saarbrücken, Germany

Scheer A.-W. (1988) Computer integrated manufacturing : CIM. Springer, Berlin, Germany

Scheer A.-W. (2000) ARIS – Business Process Modeling. Springer, Berlin, Germany

Sowa J., Zachman J. (1992) Extending and formalizing the framework for information systems architecture. In: IBM Systems Journal 31(3), pp. 590–616

Spewak S. (1993) Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology. Wiley, New York, New York

TAFIM (Apr. 1996) Department of Defence Technical Architecture Framework for Information Management – Overview.. Defence Information Systems Agency Center for Standards, United States of America

Tamura T., Yonemitsu S., Itoh A., Oikawa D., Kawakami A., Higashi Y., Fujimooto T., Nakajima K. (Jan. 2004) Is an entertainment robot useful in the care of elderly people with severe dementia? In: J Gerontol A Biol Sci Med Sci 59(1), pp. 83–85

Tapscott D., Caston A. (1993) Paradigm Shift – The New Promise of Information Technology. McGraw–Hill, New York, New York

Taylor J., Van Every E. (2010) The Situated Organization: Case studies in the pragmatics of communication research. Routledge

Teece D., Pisano G., Shuen A. (Aug. 1997) Dynamic Capabilities and Strategic Management. In: Strategic Management Journal 18(7), pp. 509–533

The Open Group (2005) TOGAF – The Open Group Architectural Framework – Version 8.1 Enterprise Edition. The Open Group, Van Haren Publishing, Zaltbommel, The Netherlands

The Open Group (2009) TOGAF Version 9. Van Haren Publishing, Zaltbommel, The Netherlands

Treacy M., Wiersema F. (1997) The Discipline of Market Leaders – Choose your customers, narrow your focus, dominate your market. Addison Wesley, Reading, Massachusetts

W3C Semantic Web Activity (2013).. http://www.w3.org/2001/sw/. Last Access: Last visited 14-06-2013

Wagter R. (2009) Sturen op samenhang op basis van GEA – Permanent en event driven In Dutch. Van Haren Publishing, Zaltbommel, The Netherlands

Wagter R., Proper H., Witte D. (Sept. 2011) Enterprise Coherence Assessment. In: Harmsen A., Grahlmann K., Proper H. (eds.) Proceedings of the 2nd Working Conference on Practice-driven Research on Enterprise Transformation, PRET 2010, Delft, The Netherlands. vol. 89. Lecture Notes in Business Information Processing Springer, Berlin, Germany, pp. 28–52

Wagter R., Berg M. v. d., Luijpers J., Steenbergen M. v. (2001) DYA: snelheid en samenhang in business en ICT architectuur. Tutein Nolthenius

Wagter R., Berg M. v. d., Luijpers J., Steenbergen M. v. (2005) Dynamic Enterprise Architecture: How to Make It Work. Wiley, New York, New York

Wood–Harper A., Antill L., Avison D. (1985) Information Systems Definition: The Multiview Approach. Blackwell, Oxford, United Kingdom

Wout J. v., Waage M., Hartman H., Stahlecker M., Hofman A. (2010) The Integrated Architecture Framework Explained. Springer, Berlin, Germany

Zachman J. (1987) A framework for information systems architecture. In: IBM Systems Journal 26(3)

Hender*ik* A. Proper

Public Research Centre Henri Tudor
Luxembourg, Luxembourg
and
Radboud University Nijmegen
Nijmegen, the Netherlands
e.proper@acm.org

Marc M. Lankhorst

BiZZdesign
Enschede, the Netherlands
m.lankhorst@bizzdesign.nl