The Data Administration Newsletter
Since 1997

On the Notion and Use of Architecture Principles in Transformation

by Danny Greefhorst, Erik Proper
Published: November 1, 2011

(Article URL: http://www.tdan.com/view-articles/15655)

Danny Greefhorst and Erik Proper discuss the concepts and principles of enterprise transformations, and suggest a practical approach.

Enterprises evolve continuously and need some means of deliberate control in order to attain their goals. Enterprise architecture as a whole enables such controls, but may lead to many models that do not focus on what is really important. Architecture principles focus on the essence of architecture and prevent analysis paralysis. They guide organizations in their transformations by describing a vision on the future and translating this vision into more concrete artifacts and actions. The concept of principles has not been studied much yet, and is elaborated by the authors in a book that describes the concepts and provides a practical approach.[1] This paper provides a summary of the concepts and shows how principles can help to direct enterprise transformations in practice, even beyond what is described in the book.

Introduction

Enterprise architecture and the associated formulation, implementation and governance processes are increasingly recognized by organizations as important capabilities. We believe that enterprise architecture plays a pivotal role in governing the continuous improvement process of an enterprise. We see them as the key means to govern the direction of the transformation of an enterprise. In our view, governing enterprise transformations first and foremost entails the perspective on an enterprise as a purposely designed and implemented artifact. This enables management to govern the enterprise transformation in terms of a clear goal, its current state, and the desired future states of the enterprise. Doing so implies a perspective on properly governed enterprise transformation as being a form of engineering. Consequently, they can be re-designed and re-implemented if there is a need for change. The field also recognizes that enterprises are primarily social systems, supported by technical systems. This means that the dominant system elements are social individuals, and that the essence of an enterprise's operation lies in the entering into and complying with commitments between these social individuals.

The governance of an enterprise transformation process involves multiple disciplines such as enterprise strategy, programme management, project management and enterprise architecture. Enterprise strategy typically remains at a high level, requiring other disciplines to operationalize it. Project management is very much focused on delivering solutions, increasing the risk of conducting "local optimisations." Projects have the tendency to provide the best cost/benefits trade-off within the scope of a specific project but that are also likely to damage the overall quality of the result of the transformation. Programme management improves the project management perspective by looking at a portfolio of projects and their dependencies. However, these dependencies are not made very explicit and agreed upon at an enterprise level. Enterprise architecture operationalizes the strategy in terms of the essential properties of the enterprise being engineered. It provides a normative restriction of design freedom towards transformation projects and programmes.[2] This makes it possible to assess whether projects contribute to the realization

of the strategy.

Enterprise architecture is described by various methods, but these methods do not seem to agree on the viewpoints and artifacts that are needed. In general, they propose to make all sorts of models ranging from business to information and technology. However, the exact contribution of these models to the specific goals and problems is often not very clear, leaving important questions unanswered. This is why we see architecture principles as an important part of enterprise architecture. Architecture principles provide a clear view on what is essential in order to reach the goals and address these problems. They fill the gap between high-level strategic intentions and concrete design decisions. Also, they tend to provide a stable point of reference. The concept of architecture principle is not new; they are described in several approaches such as PRISM[3], TOGAF[4] and the HP IT Strategy and Architecture approach.[5,6] However, these approaches do not provide a solid conceptual foundation and show a wide variation in the actual use of principles. Also, the various contributions they have in enterprise transformations are also not made explicit. The goal of this paper is, therefore, to clarify principles and relate them to other concepts. In addition, we show how they actually contribute to organization transformations.

Defining Principles

The term *principle* is said to originate from the Latin word of *principium*, which means "origin", "beginning" or "first cause." Vitruvius, an architect in ancient Rome, used principles to explain what is true and indisputable, and should apply to everyone. Vitruvius considered principles as the elements, the laws of nature that produce specific results. Laws of nature simply are, and influence, the things we do. Examples of such principles are the law of gravity and the Archimedes principle. The latter is defined by Archimedes in the third century BC and states that "any object, wholly or partially immersed in a fluid, is buoyed up by a force equivalent to the weight of the fluid displaced by the object." These principles are used in a wide range of engineering disciplines such as industrial engineering, chemical engineering, civil engineering, electrical engineering and systems engineering. They can be seen as a form of design knowledge that should be shared to increase the quality of designs. We refer to these principles as *scientific principles*, based on the view of the American Engineers' Council for Professional Development. They define engineering as "the creative application of *scientific principles* to design or develop structures, machines, apparatus, or manufacturing processes, or works utilizing them." Scientific principles are likely to be cross-disciplinary in the sense that they will be applicable in various design disciplines.

In the field of enterprise architecture a different interpretation of the term principle seems to have evolved. Principles in this category are based on beliefs and used as rules of conduct, intended to influence behavior. An example of such a principle is "no wrong doors", which suggests that clients should be served at whichever channel they approach the enterprise. Examples outside the field of enterprise architecture can be found in the Bible in the form of the Ten Commandments stating for example that "You shall not murder" and "You shall not commit adultery." Normative principles are not simply true; they are a conscious decision for a certain direction. They consist of a statement, as well as a rationale and implications (**Figure 1**). The rationale shows why the principle is important and should refer to business drivers. The implications show what the specific impact of the principle is on the organization, so that people understand what it means to their work. We name principles in this category *normative principles*, since they intend to provide a norm that people should adhere to. We define a normative principle as "a declarative statement that normatively prescribes a property of something."

Architecture principles are a specific form of normative principle; they guide/direct the enterprise by normatively restricting design freedom. This is in line with the common interpretation of the term. TOGAF states that "principles are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an enterprise sets about fulfilling its mission." The

use of principles in the context of enterprise architecture can be traced back to a research project, called the Partnership for Research in Information Systems Management (or PRISM). PRISM defined principles as "simple, direct statements of an organization's basic beliefs about how the company wants to use IT in the long term."

Normative principles exist at various levels of precision. The level of precision influences the ability to assess the compliance of a design or architecture to the principle. When considering the role of principles bridging between strategy – via architecture – to design, this is quite natural. At first, a principle will be formulated rather informally and refined later on in order to use it as a means to restrict design freedom. At the start of their life-cycle, normative principles are just statements that express the enterprise's fundamental belief of how things ought to be. At this stage, their exact formulation is less relevant. Architecture at this stage is very future-oriented and mostly a creative process. Architecture principles are not yet specific enough to actually assess compliance to them. They are primarily used as a source of inspiration. When enterprises want to use normative principles as a way to actually limit design freedom, principles need to be more specific. This is when the exact formulation of the principle becomes important. They need to be formulated in such a way that compliance to them can be assessed.

Regarding an architecture as a normative restriction of design freedom, raises the question: What the difference is between architecture and design? As suggested by the IEEE[7] and TOGAF, architecture should focus on fundamental aspects. An architecture is typically positioned at a level concerned with a class of systems. A design focuses on the remaining requirements and design decisions pertaining to a specific system being developed, having a limited impact on the key concerns of the stakeholders. Inspired by Len Fehskens[8,9] we believe that architecture should explicitly address alignment, relating the role of architecture to the mission. He redefines 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." In his view, architecture should focus on what is essential, on "the stuff that matters." This equates to those properties that are necessary and essential. This is also what distinguishes architecture from design. A different architecture implies a different mission, whilst different designs may address the same mission. We define a design principle as "a normative principle on the design of an artifact." As such, it is a declarative statement that normatively restricts design freedom. In contrast, we define an architecture principle as "a design principle included in an architecture". As such, it is a declarative statement that normatively prescribes a property of the design of an artifact, which is necessary to ensure that the artifact meets its essential requirements."

Relationship with Other Artifacts

Normative principles do not exist in isolation. They are based on all sorts of other elements, such as strategy, issues, the existing environment and external developments. On the other hand, they also influence all sorts of artifacts, such as guidelines, requirements, designs and implementations. Normative principles limit design freedom. They are, however, not the only statements which limit design freedom. Requirements also limit design freedom. However, requirements state what properties one or more systems should have. Design principles provide policies on how the design of systems will ensure that the system will meet these requirements. They provide the direction for a solution, which can be translated to lower level requirements. These lower level requirements are the basis for solutions, expressing their required characteristics. We define a requirement as "a required property of an artifact."

The impact of architecture principles on the design should be made more concrete, down to the level of work instructions for designers. We refer to these statements as *design instructions*, since they tell designers specifically what to do and what not to do. Design instructions will refer to the concepts used in the actual construction of the enterprise, such as value exchanges, transactions, services, contracts, processes, components, objects, building blocks, etc. They can be expressed in textual form, but also in

the form of models in languages such as UML and BPMN. We define a design instruction as "an instructive statement that describes the design of an artifact."

Not all normative principles have impact on the design of organizations or IT systems. There are also principles that provide direction to the behavior of people in general. The most general form of normative principle is what we refer to as *business principles*, using the term "business" in the most general sense: the organization. A subset of these principles provide policies to govern IT, which can be referred to as *IT principles*. Design principles and thereby architecture principles can be seen as a subclass of business principles (or IT principles) when they influence the design. So not all business principles or IT principles are architecture principles.

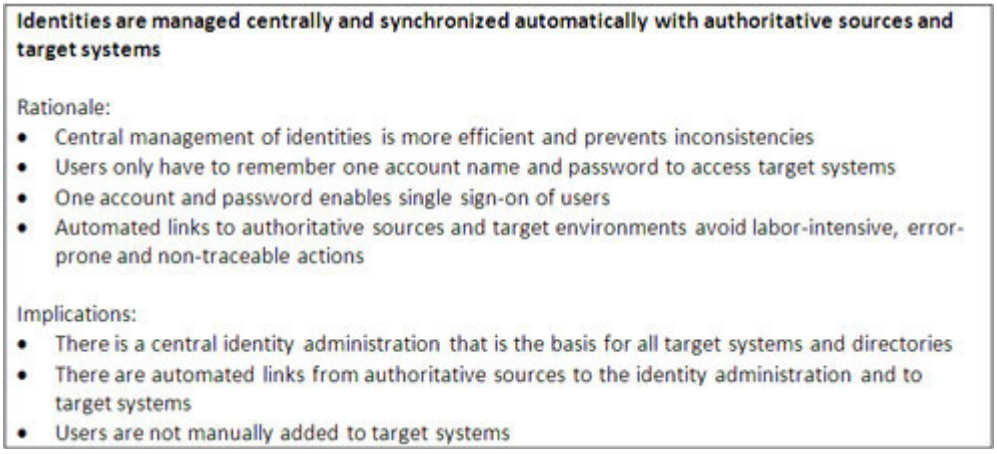
Using Architecture Principles in Transformation

Architecture principles can be used in various ways to actually influence transformations. This starts with the development of the architecture principles themselves, using them as a means to express a vision. Further downstream they can be used as a means to encode knowledge, align projects with the strategy, make decisions explicit and traceable, and as input for the planning process. We will look at all these application areas in turn.

As mentioned earlier, normative principles are based on the beliefs of people. These beliefs are an important force in the formulation of a vision. In general, having a vision is an important prerequisite in transformations. It provides an attractive view of the future that is valid for a number of years and can be used as an instrument to infer innovation programmes and projects. Beliefs may be personal, but should become shared in order to function as a vision for the organization. At the highest level, the business vision provides a view on the organization as a whole from a business perspective. Normative principles can already be seen at this stage. They can be used as an instrument to formulate a vision. The advantage of using principles at this level is that they provide a standard structure, with an explicit rationale and implications that make them concrete. Also, there is a standardized approach for describing them.[1,11] An interesting source of inspiration for principles at this level is Steve Jobs, one of the most innovative leaders of our time. Through first-person interviews with Apple employees, experts, and analysts, as well as Steve Jobs' own words over the past thirty years, Carmine Gallo discovered that there are 7 principles largely responsible for Jobs' breakthrough success.[12] For example, the first principle states that you should "do what you love." Jobs has followed his heart his entire life and that passion, he says, has made all the difference. It's very difficult to come up with new, creative, and novel ideas unless you are passionate about moving society forward. Note that this principle is really a business principle and not an architecture principle, since it has no direct impact on the design of the organization.

At an architectural level principles can be used to define an architecture vision, which is the translation of the business drivers from an architectural perspective. This is part of the "architecture vision" phase in TOGAF, and can be seen as an alternative to the business scenario technique proposed in this phase in TOGAF. The architecture vision provides the foundation for the rest of the architecture, and may touch upon all architecture domains. Architecture principles at this level are very specific to the organization, and are also called the "key" or "guiding" architecture principles. An example of an architecture vision that strongly relies on architecture principles is the European Interoperability Framework.[13] One of the main goals of this framework is to promote and support the delivery of European Public Services by fostering cross-border and cross-sectoral interoperability. To this end it provides principles that should underlie European Public Services. An example architecture principle is that of "inclusion and accessibility" which states that "the use of ICT should create equal opportunities for all citizens and businesses due to open, inclusive services that are publicly accessible without discrimination

Principles can also be used as a means to encode knowledge and best-practices. These best-practices will increase the quality of solutions, since they have been tested in multiple situations. They prevent typical problems, and thereby reduce risks. Consider for example the architecture principle in **Figure 1** that describes a general best-practice related to identity management. This principle is based on experience in a large number of organizations and has proven valuable numerous times. Principles such as these are mostly at a lower level than guiding architecture principles, and are typically defined in reference architectures. Reference architectures are generic architectures, based on best-practices.[14] They can be reused in all various contexts and organizations. In a certain sense these architecture principles are comparable to design patterns[15]; they also describe common solutions for recurring problems. Many patterns can even be reformulated in a principle form. The advantage of using principles at this level is that they provide a compact and structured form. The rationale helps in getting commitment from others on using the best-practice and why. The implications make explicit what the impact is on their own work.


Identities are managed centrally and synchronized automatically with authoritative sources and target systems

Rationale:

- Central management of identities is more efficient and prevents inconsistencies
- Users only have to remember one account name and password to access target systems
- One account and password enables single sign-on of users
- Automated links to authoritative sources and target environments avoid labor-intensive, error-prone and non-traceable actions

Implications:

- There is a central identity administration that is the basis for all target systems and directories
- There are automated links from authoritative sources to the identity administration and to target systems
- Users are not manually added to target systems

Figure 1: Example of an Architecture Principle

Another important contribution of architecture principles is that they help in aligning projects with the strategy and other important drivers. Before starting a project, the architecture principles can be translated to requirements for the project. You basically answer the question: What does this architecture principle imply for this project? This translation can be performed using the "guiding architecture principles" as well as using more generic principles as defined in reference architectures. The result is specified in a document that can be seen as a contract with the project, which accompanies the project initiation document. In TOGAF this is actually called an "architecture contract." It is an agreement between architecture design and development partners to design and developing the enterprise architecture, or significant parts of it. Wagter et al.[16] introduce the concept of a "project start architecture" for this purpose, which translates the enterprise architecture to the project. Such a contract is only meaningful if architecture governance is also in place, including an architecture compliance review process. Such a review process should include reviews at specific phases of the project lifecycle, and is typically part of an overall multi-disciplinary project review process. The architecture function is responsible for checking compliance with the architecture(s). They can look at project documents (typically designs), and determine whether they comply with the requirements that were formulated in the architecture contract. The result is typically discussed in the "architecture board", which may decide to provide a dispensation for deviations if there are very good reasons for it. Architecture compliance assessments provide management with insight on the actual implementation of the architecture. It provides them with an instrument to highlight potential problems, and the opportunity to act upon it before it is too late.

Another contribution of architecture principles is that they make decisions explicit and traceable. This is very important since management wants to be in control, and needs to know which decisions have been made and why. Architecture principles are a form of design decision (at an architectural level). By explicitly determining and discussing these decisions with relevant stakeholders, commitment is created

for them. You could say that they stimulate the right discussions. Also, by explicitly linking the architecture principles to goals and drivers at a higher level, it is ensured and shown how architecture principles contribute. Architecture principles can consist at multiple levels, and these levels can be linked to each other. The rationale for one principle can be found in another principle. This creates a form of traceability from strategy to design, and allows impact analysis of changes at any of these levels. At a low level of architecture maturity architects do not make their decisions explicit. They are basically local heroes that the organization is dependent upon. However, the motivation behind their opinions should be made explicit since in the end they should contribute to business outcomes. Also, without explicitly documented decisions the organization has a problem when such local heroes leave the organization. The knowledge and motivations behind all the decisions that have been made should therefore be made explicit.

The last area that we want to touch upon is that architecture principles also help in the planning process, both at a business level (business planning) as well as an information level (information planning) and technology level (technology planning). Principles describe a future state that often is not the current state. They show how the practices of the organization can be improved, but improvement actions should be made explicit and embedded in the planning process. By explicitly determining the gaps between the baseline and target state, actions can be derived. For example the architecture principle shown in **Figure 1** describes a state in which there is a central administration of identities. When such a central administration does not exist a project needs to be initiated to implement such a system.

Conclusions

In this paper we have explored the concept of architecture principle, and the elements that it relates to. We have shown that principles are a powerful instrument, since they provide the rationale for their existence as well as insights into their impact on people that need to use them. In contrast, models simply are and do not show the rationale behind the decisions that they encode. Also, principles are generic instruments that can be applied in all sorts of situations.

We have also described how architecture principles contribute to transformation programmes and projects. They can be used to express a vision, encode knowledge, align projects with the strategy, make decisions explicit and traceable, and as input for the planning process. These are just the major contributions of architecture principles; we have also seen other areas of application such as using them as a means to measure maturity. This is certainly an interesting area for future research.

The concepts described are the result of a first iteration in a design science -driven[10] research effort in which we aim to more clearly define the concept of architecture principles. While the concepts are a synthesis of existing theoretical perspectives as well as empirical insights, more validation and refinement is needed. Especially more research is needed to come up with better criteria to determine what the nature is of a specific statement; how can we detect whether a given statement is an architecture principle or not?

References

1. Greefhorst, D., Proper, E.: Architecture Principles – The Cornerstones of Enterprise Architecture, 1st Edition, ISBN 978-3-642-20278-0, Springer, 2011.
2. Dietz, J.: Architecture – Building strategy into design. Netherlands Architecture Forum, Academic Service – SDU, The Hague, The Netherlands (2008). ISBN-13: 9789012580861.
3. CSC Index, Inc. and Hammer & Company, Inc.: PRISM: Dispersion and Interconnection: Approaches to Distributed Systems Architecture, Final Report. Technical report, CSC Index, Inc. and Hammer & Company, Inc., Cambridge MA. (1986).
4. The Open Group – TOGAF Version 9. Van Haren Publishing, Zaltbommel, The Netherlands

(2009). ISBN-13: 9789087532307
5. Beijer, P., De Klerk, T.: IT Architecture: Essential Practice for IT Business Solutions. Lulu (2010).
6. Rivera, R.: Am I Doing Architecture or Design Work? It Professional 9 (2007) 46–48.
7. The Architecture Working Group of the Software Engineering Committee, Standards Department, IEEE: Recommended Practice for Architectural Description of Software Intensive Systems. Technical Report IEEE 1471:2000, Piscataway, New Jersey (2000). ISBN-10: 0738125180.
8. Fehskens, L.: Re-Thinking architecture. In: 20th Enterprise Architecture Practitioners Conference, The Open Group (2008).
9. Fehskens, L.: What the “Architecture” in “Enterprise Architecture” Ought to Mean. In: Open Group Conference Boston, The Open Group (2010).
10. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research, MIS Quarterly 28 (2004) 75–106.
11. Danny Greefhorst, Erik Proper: A Practical Approach to the Formulation and Use of Architecture Principles, Proceedings of TEAR workshop 2011, Helsinki, 29 augustus 2011.
12. Gallo, C.: The 7 Success Principles of Steve Jobs, blogs.forbes.com, 2011.
13. European Commission: Towards interoperability for European public services - Annex II - European Interoperability Framework, Version 2.0, 2010.
14. Greefhorst, D., Grefen, P., Saaman, E., Bergman, P., Van Beek, W.: Herbruikbare architectuur - een definitie van referentie-architectuur, Informatie, 2009.
15. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object–Oriented Software. Addison Wesley, Reading, Massachusetts (1995).
16. Wagter, R., Van den Berg, M., Luijpers, J., Van Steenbergen, M.: Dynamic Enterprise Architecture: How to Make It Work. Wiley, New York, New York (2005). ISBN-10: 0471682721.

Go to Current Issue | Go to Issue Archive

Danny Greefhorst - Danny Greefhorst, MSc., is a Principal Consultant and Director of ArchiXL in Amersfoort, The Netherlands, and acts as an architect and consultant for clients in the financial and public sector. He has extensive experience with the definition and implementation of enterprise architectures, application architectures and technical architectures. In addition, he coaches organizations in setting up and executing their architecture function. Danny is responsible for the EA portal Via Nova Architectura and is a member of the governing board of the architecture department of the Dutch Computing Association. Danny is active in the architecture community, regularly publishes on IT- and architecture related topics and is co-author of the book Architecture Principles: The Cornerstones of Enterprise Architecture. He can be reached at dgreefhorst@archixl.nl.

Erik Proper - Prof. Dr. H.A. (Erik) Proper is a Senior Research Manager at the Public Research Centre Henri Tudor in Luxembourg, where he leads the services-oriented enterprise engineering research program. He is also a Professor at the Radboud University Nijmegen, The Netherlands, where he heads up the Theories for Enterprise Engineering group. His general research drive is the modeling of systems. He applies this drive mainly in the fields of service science, enterprise modeling, enterprise engineering and enterprise architecting. He was co-initiator of the ArchiMate project, and currently also serves on the board of the ArchiMate forum of The Open Group. He has also co-authored and edited several books on architecture and information systems, including Architecture Principles: The Cornerstones of Enterprise Architecture. He can be reached at e.proper@acm.org.