

# ArchiMate<sup>®</sup> for Integrated Modelling Throughout the Architecture Development and Implementation Cycle

Henk Jonkers, Dick A. C. Quartel, Henry M. Franken  
 BiZZdesign, Enschede, the Netherlands  
 {h.jonkers, d.quartel, h.franken}@bizzdesign.nl

## Abstract

The ArchiMate standard offers an integrated language for enterprise architecture modelling. It allows for the description and visualization of different architecture domains, as well as their underlying relationships and dependencies. Since its adoption as a standard of The Open Group, the international interest in ArchiMate has been growing rapidly. ArchiMate complements TOGAF, the standard of The Open Group for developing enterprise architectures. To provide modelling support throughout the architecture development and implementation cycle as defined by TOGAF, Version 2.0 of ArchiMate includes two extensions to the original ArchiMate core language: a *Motivation* extension and an *Implementation and Migration* extension. The aim of this paper is to provide the reader with an introduction to the ArchiMate language, highlighting the new features of Version 2.0, and to illustrate how the language can be used in the different phases of the TOGAF architecture development cycle.

**Keywords:** Enterprise Architecture, modelling, ArchiMate, TOGAF.

## Izveleček

### ArchiMate – integrirano modeliranje podjetniških arhitektur skozi razvojni in implementacijski cikel

ArchiMate ponuja integrirani jezik za pristop k modeliranju podjetniških arhitektur. Omogoča opisovanje in vizualiziranje različnih arhitekturnih domen kot tudi odnosov in odvisnosti v njihovem ozadju. Mednarodno zanimanje za ArchiMate močno narašča, odkar ga je The Open Group privzel kot standard. ArchiMate namreč dopolnjuje TOGAF, ki je standard skupine The Open Group za razvijanje podjetniških arhitektur. Da bi zagotovili podporo modeliranju skozi celoten razvoj arhitekture in ciklus implementacije, kot ga določa TOGAF, vsebuje ArchiMate verzija 2.0 dve razširitvi izvirnega osnovnega jezika ArchiMate: motivacijsko razširitev ter implementacijsko in migracijsko razširitev. Namen tega prispevka je uvesti bralca v jezik ArchiMate, tako da osvetlimo nove lastnosti verzije 2.0 in prikažemo, kako lahko jezik uporabimo v različnih fazah razvojnega cikla arhitekture TOGAF.

**Ključne besede:** arhitektura podjetja, modeliranje, ArchiMate, TOGAF.

## 1 INTRODUCTION

Within larger organizations, one can typically find various architecture domains, such as: organizational structures, products, business processes, information systems, applications and technical infrastructure. Traditionally, each architecture domain employs specific models and visualizations, which simplifies communication, discussion and analysis *within* the domain. However, the relations between these different domains are in many cases unclear. Moreover, these domains tend to (at least partially) overlap.

The ArchiMate language for enterprise architecture modelling language has been developed with the aim to provide a

uniform representation for enterprise architecture (EA) descriptions (Lankhorst et al., 2009; The Open Group, 2012). It offers an integrated architectural approach by which organizations can describe and visualize different architecture domains, as well as their underlying relationships and dependencies. ArchiMate provides a unified way to model enterprise architectures, while integrating the various domains and describing them in an easily readable way, as illustrated in Figure 1. In addition, a distinction is made between a business layer, an application layer, and a layer with the underlying (IT) technical infrastructure.

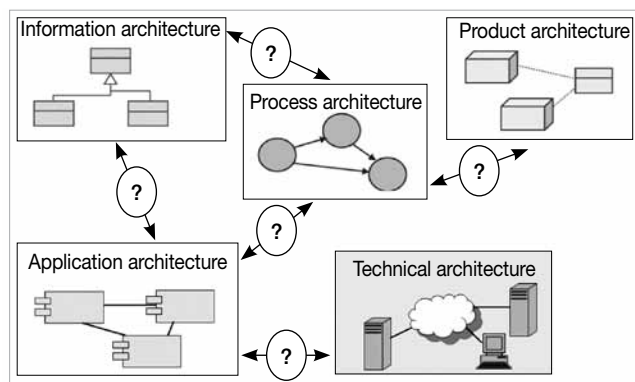


Figure 1. Integration of architectural domains

Since ArchiMate is positioned at the level of enterprise architecture, this also implies that the ArchiMate language does not provide the level of detail one would typically find in languages used at the »design level« (Lankhorst, Proper, & Jonkers, 2010). For example, while ArchiMate features concepts such as business event and junction, it does not provide the rich detailed set of gateways, et cetera as offered by a language such as BPMN (Object Management Group, 2008). Similarly, in contrast to languages such as UML (Object Management Group, 2005), it does not provide concepts to model the details of software applications. At the same time, refinement/abstraction mechanisms can be used to maintain the connection between, e.g., a BPMN or UML model and an ArchiMate model (Lankhorst et al., 2009).

The ArchiMate language has been designed in a structured way, by defining a generic structure that is made specific for the different architectural layers (as will be explained in Section III). Also, ArchiMate has a limited set of relation types that are used throughout the language. Finally, ArchiMate provides a standard graphical notation for the modelling concepts and relations.

The purpose of this paper is to provide the reader with an introduction to the ArchiMate language, highlighting the new features of Version 2.0, and to illustrate how the language can be used in the different phases of the TOGAF architecture development cycle. The remainder of this paper is structured as follows. In Section II, we introduce the ingredients of an integrated enterprise architecture approach, and we sketch the TOGAF architecture development cycle. Section III describes the general structure of the ArchiMate language. In Section IV, we show how the different types of architectures as defined by TOGAF

can be modelled with the ArchiMate Core language, while Section V introduces the two extensions that have been added in Version 2 of the ArchiMate language: the *motivation* extension and the *implementation and migration* extension. Finally, in Section VI we present some conclusions and directions for future work.

## 2 AN INTEGRATED APPROACH TO ENTERPRISE ARCHITECTURE

Frameworks for enterprise architecture vary in the types of support that they offer. They may have, among others, any combination of the following ingredients:

- A process (»way of working«) for creating architectures; this may be accompanied by guidelines, techniques and best practices.
- A set or classification of viewpoints.
- A language for describing architectures (defining concepts and relationships, but also a notation).
- The concept of a (virtual) architecture repository, possibly containing predefined architectural artefacts and (reference) models.

The core of TOGAF (The Open Group, 2011) is formed by the Architecture Development Method (ADM), a step-wise, iterative process for the development and implementation of an enterprise architecture (see Figure 2).

The ten phases of the ADM can be grouped into four main parts, also shown in Figure 2:

1. »Getting the organization committed and involved«: the Preliminary Phase prepares the organization as a whole for »working under architecture«, and involves activities such as establishing an architecture capability, tailoring the architecture methods and techniques for the specific characteristics of the organization, and defining an initial set of architecture principles; Phase A, Architecture Vision, prepares for a single architecture development cycle, and includes the formulation of an architecture vision with a high-level overview of the change that is envisaged.
2. »Getting the architecture right« concerns the description of the actual baseline and target architectures, and an analysis of the gaps between baseline and target. The three phases in this group are concerned with three main types of architectures: business architecture (Phase B), information systems architectures (Phase C), and technology architecture (Phase D).

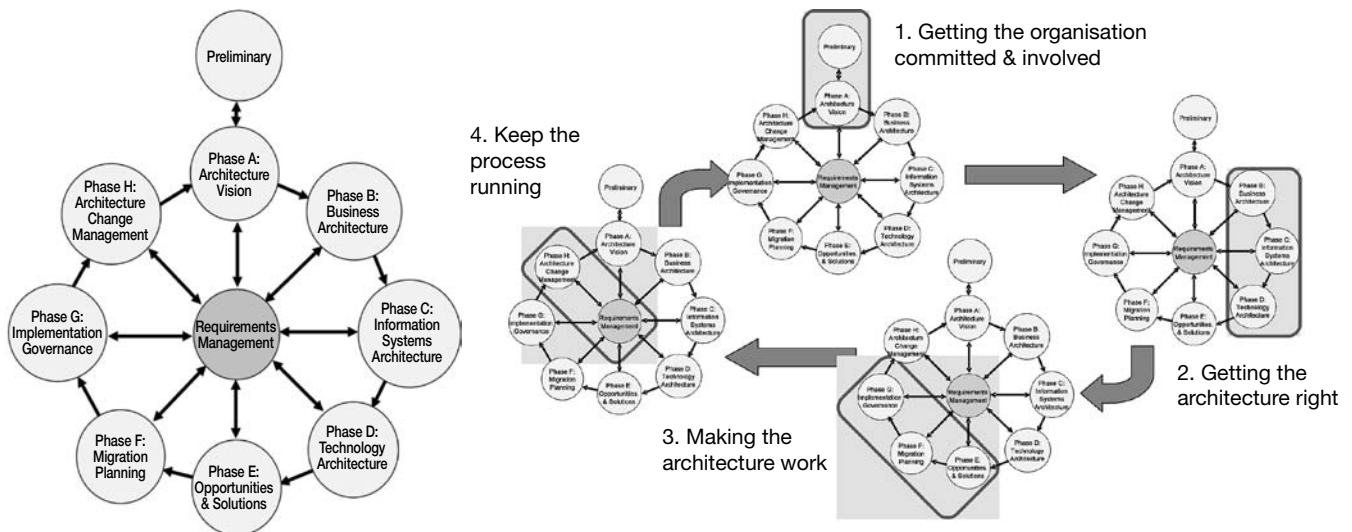


Figure 2. TOGAF Architecture Development Method (The Open Group, 2011)

3. »Making the architecture work« is concerned with the implementation of the developed architecture, and planning the migration to the new situation. It includes Phase E, Opportunities and Solutions, in which the gap analysis results are consolidated and potential implementation work packages are identified; Phase F, Migration Planning, in which work packages are prioritized and a migration plan is established; and Phase G, Implementation Governance, safeguarding the compliance of implementation projects with the architecture.
4. »Keep the process running« is concerned with the management, prioritization and version control of requirements on the architecture. The central Requirements Management process manages the requirements during an architecture development cycle. In Phase H, Change Management, new requirements are identified that may lead to the start of the new architecture development cycle.

TOGAF also includes the identification of viewpoints, techniques and reference models. However, it does not define an actual modelling language. The TOGAF Architecture Content Framework does indeed identify relevant architecture building blocks, but it does not constitute a precisely defined language, nor does it provide a notation for these building blocks. ArchiMate complements this by defining a fully worked out (graphical) modelling language, including the definition of relevant viewpoints. This language also provides a concrete visualization of the views identified in TOGAF.

TOGAF and ArchiMate share their view on the use of viewpoints, and the concept of an underlying common repository of architectural artefacts and models; i.e., they have a firm common foundation. However, TOGAF and ArchiMate complement each other with respect to the definition of an architecture development process and the definition of an enterprise architecture modelling language. Together, they make up a complete, integrated approach for delivering enterprise architecture.

### 3 STRUCTURE OF THE ARCHIMATE LANGUAGE

In this section we briefly discuss the core structures of the ArchiMate language. In (Lankhorst et al., 2010) a more detailed account is provided of the requirements on the language, and the design decisions that underpin its design.

#### A. Core Concepts

To arrive at a language that is easy to learn and understand, a conscious decision was made to limit the set of core modelling concepts. Therefore, a small number of generic modelling concepts have been created that essentially re-appear (in different variations) on the various layers of the language. First, we distinguish between the *structural* or *static* aspect and the *behavioural* or *dynamic* aspect. Structural concepts are assigned to behavioural concepts, to show who or what performs the behaviour. In addition to *active* structural elements (the business actors, application components and devices that performs actual behav-

our, i.e., the ‘subjects’ of activity), we also recognize *passive* structural elements, i.e., the *objects* on which behaviour is performed.

Second, we make a distinction between an *external view* and an *internal view* on systems. The concept of *service* plays a central role in ArchiMate. A service represents a unit of essential functionality that a system exposes to its environment, and can be used to »bind« together the layers (applications provide services to business processes, while applications on their turn use infrastructure services). Furthermore, within a layer, services can be used as well to encapsulate behaviour. This enables the use of a services-oriented architecture (SOA) style from business processes, via applications to the underlying infrastructure. For the external users, only this external functionality, together with non-functional aspects such as the quality of service, costs etc., are relevant. Services are accessible through *interfaces*, which constitute the external view on the structural aspect.

Figure 3 summarizes the resulting generic core concepts of the language, as well as their main relationships

to support and automate its business processes). The application layer uses the services supplied by the technology layer (e.g., to make use of the physical resources – servers, networks etc. – in order to run its applications).

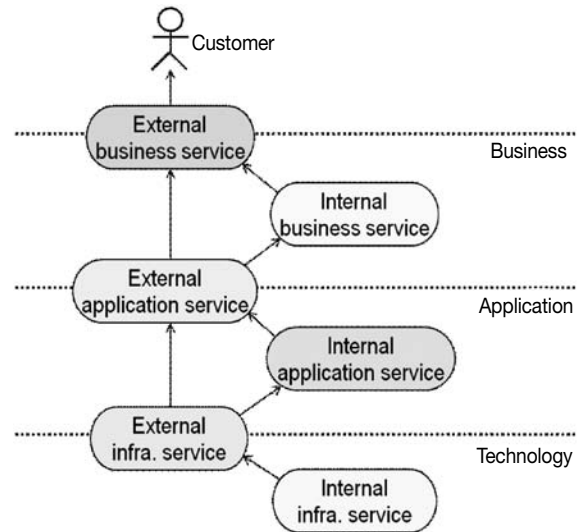


Figure 4. Services

In addition to the types of services mentioned above, the ArchiMate language distinguishes within each layer between internal and external services. Internal services are the services (added values) supplied to entities within the same layer. External services are the services made available to entities outside that layer.

## 4 CREATING ARCHITECTURE MODELS WITH ARCHIMATE

The primary use of ArchiMate in the context of TOGAF is in the representation of architecture models. TOGAF distinguishes four architectures: the Business Architecture (created in Phase B of the ADM), the Application Architecture and Data Architecture (both part of the Information Systems Architectures, Phase C) and the Technology Architecture (Phase D). In all of these phases, baseline (»as is«) and target (»to be«) architectures are created. In Phase A (Architecture Vision) of the ADM, first global versions of these architectures are already sketched; for this, a simplified subset of ArchiMate can be used.

We illustrate the different architectures with a small example based on a fictitious insurance company. ArchiSurance is a merger of three previously independent companies: Home & Away for home-

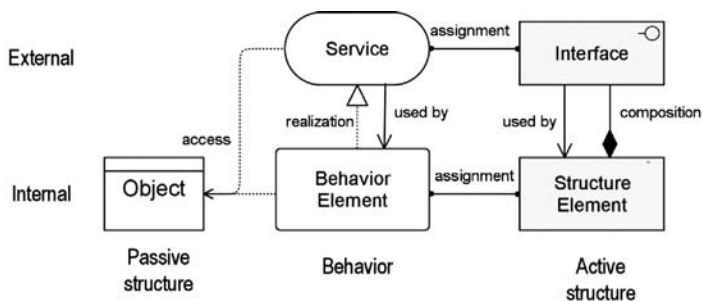


Figure 3. Core Concepts of the ArchiMate Language

### B. Services as a Linking Pin Between Layers

In ArchiMate, the concept of service is defined as *the externally observable behaviour of a system<sup>1</sup> that may have some added value for that system’s environment*. It is therefore natural to expect that, in the case of architecture layers, higher layers use the services supplied by the lower layers, since higher layers can be seen as the »environment« of the lower layers (see Figure 4). The enterprise’s environment is the »end-user« of the services offered by the business layer of the enterprise. The business layer makes use of the services exposed by application layer (e.g., in order

<sup>1</sup> System in the general sense, and not just as a synonym to application.

owner's and travel insurance, PRO-FIT for car insurance, and Legally Yours for legal aid insurance. The new company has a single Front Office and three separate Back Offices. ArchiSurance has plans to rationalize their application portfolio, by integrating legacy applications with similar functionality from the old companies that are still in use. Note that these examples give an impression of the ArchiMate language, but do not show all the concepts. For a com-

plete overview of the language, please refer to (The Open Group, 2012).

### A. Business Architecture

The Business Architecture provides the context for system development trajectories, showing, among others, the main business processes, the actors (or roles) performing these processes, and the information (objects) exchanged between the processes.

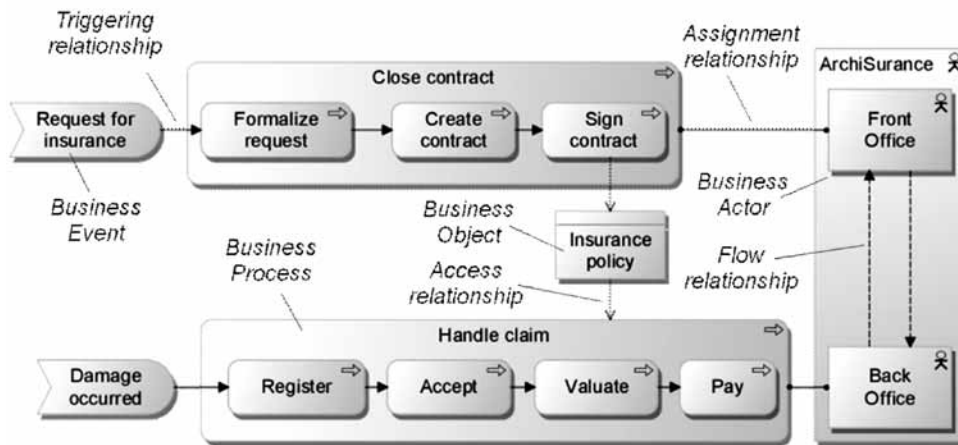


Figure 5. Baseline and Target Business Architecture

Figure 5 shows an example of a Business Architecture expressed in ArchiMate. We assume that the business architecture of ArchiSurance does not change in the application rationalization process.

### B. Application Architecture

The Application Architecture shows the applications or application components, their relationships and

their functionality. Figure 6 shows the baseline Application Architecture of ArchiSurance. The functionality that the applications offer to their environment is modelled with services. The service concept plays a central role in ArchiMate, also in the Business Architecture and the Technology Architecture (although this is not shown in our example), and in particular as a linking pin between the different architectures.

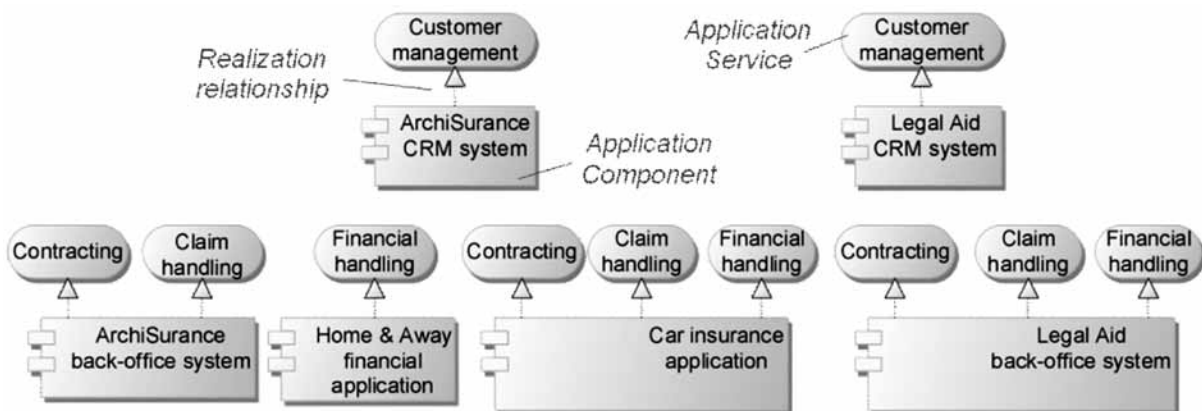


Figure 6. Baseline Application Architecture

Figure 7 shows the target Application Architecture of ArchiSurance, in which the legacy applications have been replaced by a single back-office system and a single CRM system for the whole company.

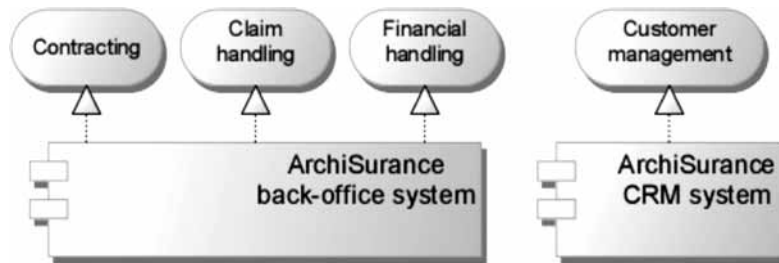


Figure 7. Target Application Architecture

In ArchiMate, separate views can be used to show the relationships between the different architectures. As an example of this, Figure 8 shows how the services from the Application Architecture are used in the processes of the Business Architecture.

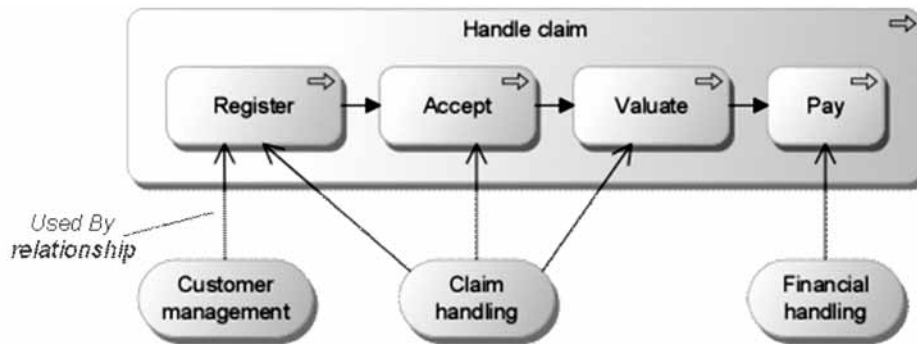


Figure 8. Business-Application Alignment (Target)

### C. Data Architecture

The Data Architecture shows the main data objects used within the applications, as well as their relationships.

Figure 9 shows the Data Architecture of ArchiSurance, which we assume will not change in the application rationalization process.

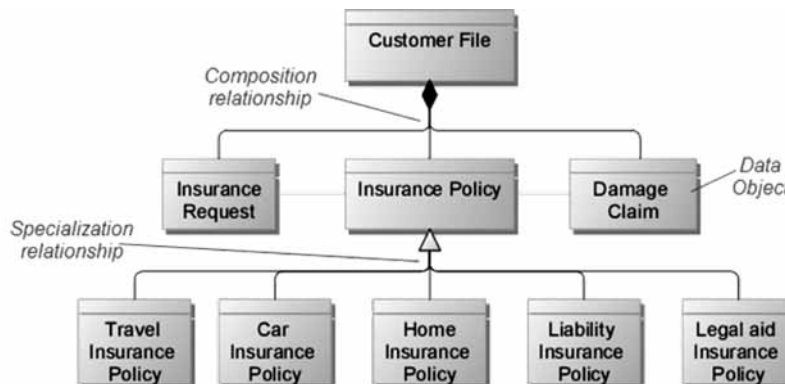


Figure 9. Baseline and Target Data Architecture

### D. Technology Architecture

The Technology Architecture shows, among others, the devices and system software on which applications run, the networks connecting devices, and artefacts that form the physical implementation of appli-

cation components or data objects. Figure 10 shows the baseline Technology Architecture of ArchiSurance. There are separate application servers for the different back-office applications.

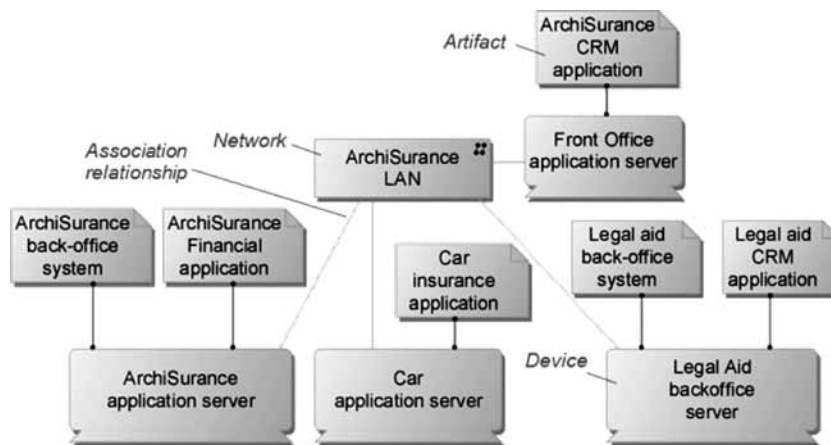


Figure 10. Baseline Technology Architecture

In the target Technology Architecture, as shown in Figure 11, some of these application servers become

redundant. However, to increase reliability and availability, an additional backup server is introduced.

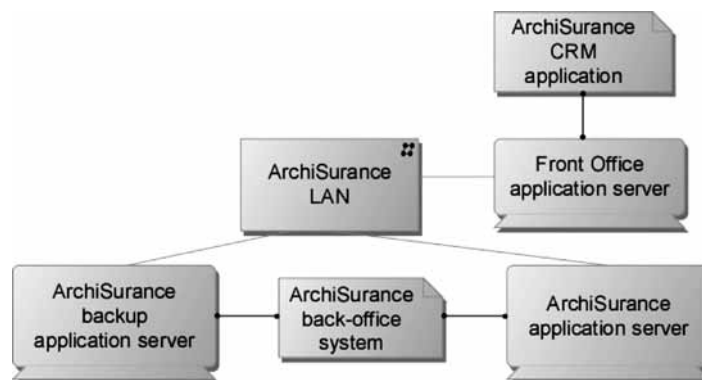


Figure 11. Target Technology Architecture

### E. Gap Analysis

An important step in Phases B, C and D of the TOGAF ADM is a gap analysis, which reviews the differences between the baseline and target architecture. It shows which building blocks are carried over from baseline to target, which building blocks are new in the target architecture (which can be used as a basis to decide whether to buy or build these building blocks), and which elements have been eliminated from the baseline architecture (on purpose or accidentally; i.e., a gap analysis can also be used as a

mechanism for validation of the target architecture). Subsequently, Phases E, F and G of the TOGAF ADM deal with the implementation of the proposed target architecture.

TOGAF suggests the use of a gap matrix as a technique for gap analysis. However, ArchiMate models also form a useful starting point for gap analysis, and the results can also be presented as an ArchiMate view. Figure 12 shows an example of this for the Technology Architecture.

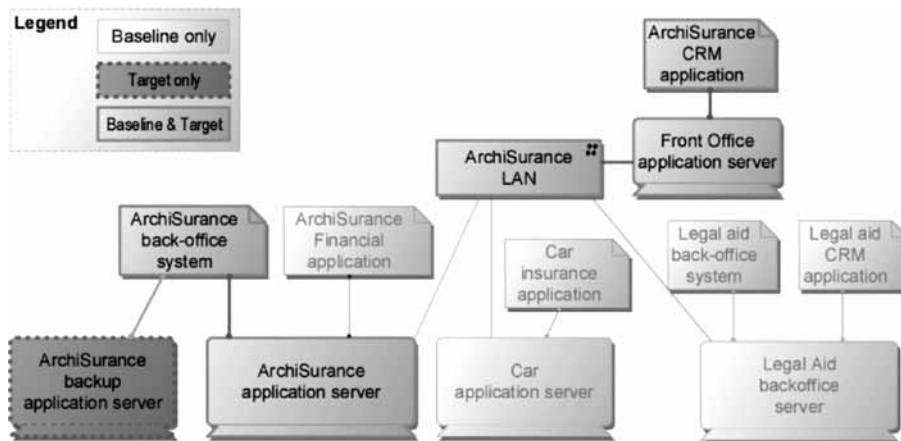


Figure 12. Technology Architecture Gap Analysis

## 5 EXTENDING THE ARCHIMATE COVERAGE OF TOGAF

As described in the previous sections, ArchiMate Version 1.0 chiefly supported modelling of the architectures in Phases B, C and D in the TOGAF ADM (»Getting the architecture right« in Figure 2), as is illustrated in Figure 13. The resulting models are used as input for the subsequent ADM phases. However, modelling concepts specifically aimed at the other phases – e.g., concepts for modelling principles, goals and requirements, or concepts to support migra-

tion planning – were still missing in the language. ArchiMate Version 2.0, which was launched early in 2012 by The Open Group, provides two extensions for this: one for describing motivation (e.g. stakeholders, goals and requirements), supporting the phases for »Getting the organization committed and involved« and »Keep the process running« from Figure 2, and one for implementation and migration planning, supporting the phases for »Making the architecture work« from Figure 2. The next subsections outline these two extensions.

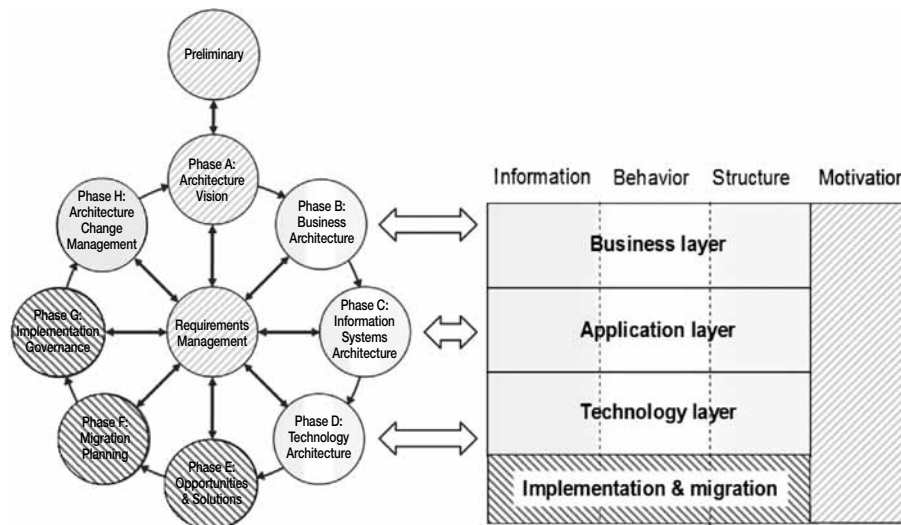


Figure 13. TOGAF ADM and ArchiMate

### A. Motivation Extension

The motivation extension of ArchiMate (Engelsman, Jonkers, & Quartel, 2011) adds concepts related to goal modelling and business requirements management. They can be used for the identification,

description, analysis and validation of goals and requirements at business level and their realization in enterprise architecture models as described with the ArchiMate core concepts.

The motivational concepts, based on sources



such as OMG’s Business Motivation Model (Object Management Group, 2006), architecture principles (Proper & Greefhorst, 2010; Greefhorst & Proper, 2011) and goal-driven requirements engineering (Yu & Mylopoulos, 1994; Van Lamsweerde, 2001; Regev & Wegmann, 2005) are used to model the motivations, or intentions, that underlie the design of an enterprise architecture. These intentions influence, guide and constrain the design. Intentions are pursued by stakeholders, which can be individuals or groups such as a project team, enterprise or society. In addition, intentions may be organized into certain areas of interest, called drivers, such as customer satisfaction, compliance to legislation or profitability. Optionally, assessments of these drivers may be used to decide whether existing intentions need to be adjusted or not.

The actual intentions are represented by goals, principles and requirements. Goals represent some desired result – or end – that a stakeholder wants to achieve; e.g., increasing customer satisfaction with 10 percent. Principles and requirements represent desired properties of solutions – or means – to realize the goals. Principles represent desired properties that are required from all possible solutions in a given context; requirements represent desired properties of specific, individual solutions. For example, the requirement »Use a single CRM system« is a specialization of the principle »Data should be stored only once« by applying it to the current organization’s architecture in the context of the management of customer data. Figure 14 shows an example of the use of the motivation concepts and their relationships, and also how requirements can be realized by core concepts.

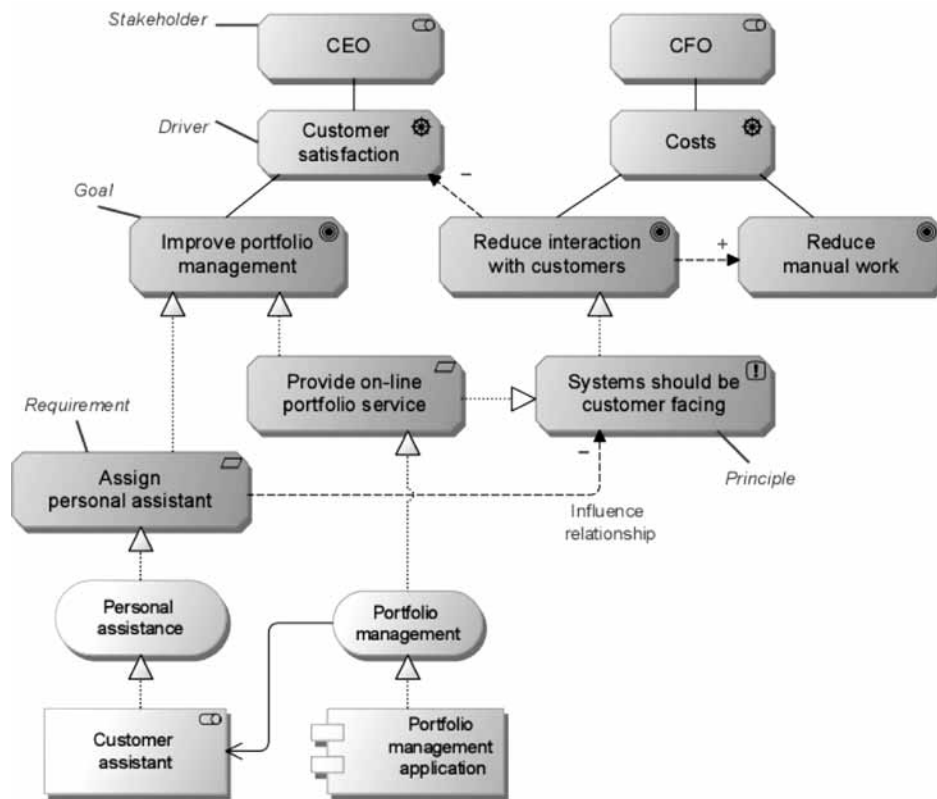


Figure 14. Motivation Extension Example

## B. Implementation & Migration Extension

The Implementation and Migration Extension defines a number of additional concepts that enable the modelling of the architecture change process and increase the insight into these changes as well

as their manageability in terms of portfolio and project management and decision-making. By defining concepts such as work package (to model implementation work at different levels of granularity, e.g., programs, projects, or project tasks), delivera-

ble and plateau it is possible to connect ArchiMate with program and project management standards and best practices, such as MSP (Chittenden & Van

Bon, 2006)[1], PRINCE2 (The Stationary Office, 2009), and PMBoK (Project Management Institute, 2001).

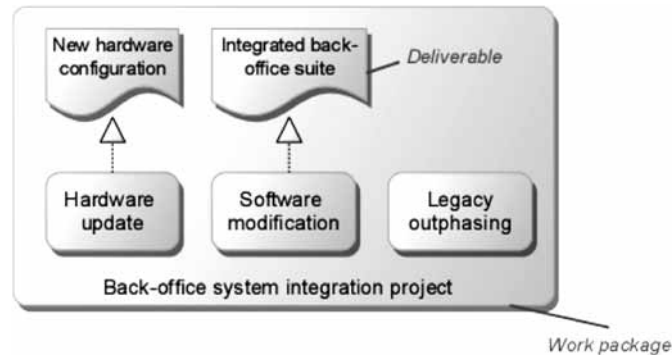


Figure 15. Programs, Projects, Project Roles and Project Results

Figure 15 shows an example of the use of work packages and deliverables. A project may be subdivided into a hierarchy of project tasks. Multiple projects which are managed together coherently, and which all contribute to a common outcome, can be grouped into a program. Work packages produce deliverables. These may be results of any kind, e.g., reports, papers, services, software, physical products, etc. A deliverable may also realize (a part of) architecture, or a solution that implements (a part of) architecture. To each work package, one or more business roles can be assigned.

An important premise in TOGAF is that the various architectures are described for different stages in time. In each of the Phases B, C, and D of the ADM, a Baseline Architecture and Target Architecture are created, describing the current situation and the desired future situation. In Phase E, »Opportunities and Solutions«, Transition Architectures are defined, showing the enterprise at incremental states reflecting peri-

ods of transition between the Baseline and Target Architectures. Transition Architectures are used to allow for individual work packages and projects to be grouped into managed portfolios and programs, illustrating the business value at each stage. In order to support this, the plateau concept was introduced.

Relationships can be established between the enterprise architecture models created at different moments in time and the migration models. Subsequently, analysis tools can be used to emphasize the differences between the different versions of models through the linked plateaus. These differences are captured by the concept of gap. A gap is an important outcome of a gap analysis in Phases B, C, and D of the TOGAF ADM, and forms an important input for the subsequent implementation and migration planning. The gap concept is linked to two plateaus (e.g., baseline and target architecture, or two successive transition architectures), and represents the differences between these plateaus (Figure 16),

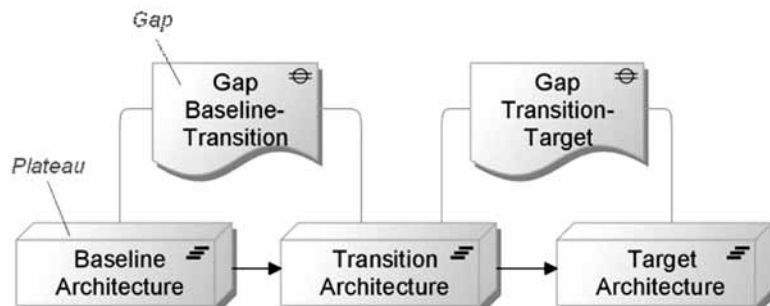


Figure 16. Migration Planning Concepts

## 6. CONCLUSIONS AND FUTURE DIRECTIONS

TOGAF is a leading enterprise architecture method of The Open Group. ArchiMate has recently been adopted as an Open Group standard for modelling enterprise architectures. TOGAF and ArchiMate share their view on the use of viewpoints, and the concept of an underlying common repository of architectural artefacts and models; i.e., they have a firm common foundation. However, they complement each other with respect to the definition of an architecture development process and the definition of an enterprise architecture modelling language. ArchiMate provides a concrete visualization for the architectures and views proposed in TOGAF.

From the previous sections, it is clear that TOGAF and ArchiMate can be used in conjunction and cover much of the same ground. TOGAF itself provides no guidance on creating a consistent overall model of the architecture, but refers to tools that should provide this support (The Open Group, 2011, Chapter 31):

»In order to achieve the goals of completeness and integrity in an architecture, architecture views are usually developed, visualized, communicated, and managed using a tool.

In the current state of the market, different tools normally have to be used to develop and analyze different views of the architecture. It is highly desirable that an architecture description be encoded in a **standard language**, to enable a standard approach to the description of architecture semantics and their re-use among different tools.« (Emphasis ours).

This is where ArchiMate nicely complements TOGAF: it provides a vendor-independent set of concepts that would help to create a consistent, integrated model »below the waterline«, which can be depicted in the form of TOGAF's views.

Presently, The Open Group is actively pursuing a closer integration between ArchiMate and TOGAF. An outline of this convergence is given by Jonkers, Proper, and Turner (2009). Some parts of TOGAF were not yet covered by ArchiMate Version 1.0 concepts. Therefore, two extensions to the language have been defined and included in Version 2.0 of the standard to fill these gaps. In particular, these concern on the one hand concepts for modelling the goals, motivations, principles and requirements used as inputs in defining an architecture, and on the other hand concepts for TOGAF's implementation and migration phases. With these two extensions, the new

version of ArchiMate has full coverage of TOGAF. Thus, these two complementary open standards will reinforce each other and help to advance the enterprise architecture discipline in general.

Future research is concerned with potential additional extensions of the language in other directions. In the practical use of ArchiMate, a number of fields have been identified in which such future extension of the language may be advisable: e.g., concepts for modelling business policies, decisions and rules, concepts for better support of the design process, or concepts that provide the link to (business) models at a more strategic level.

## 7. REFERENCES

- [1] Chittenden, J. & Van Bon, J. (2006). *Programme Management Based on MSP: A Management Guide*. Van Haren Publishing.
- [2] Engelsman, W., Jonkers, H., & Quartel, D.A.C. (2011). *ArchiMate® Extension for Modeling and Managing Motivation, Principles, and Requirements in TOGAF™*, White Paper, The Open Group.
- [3] Greefhorst, D. & Proper, H. A. (2011). *Architecture Principles – The Cornerstones of Enterprise Architecture*. Springer, Berlin.
- [4] Jonkers, H., Proper, H. A., & Turner, M. (2009). *TOGAF and ArchiMate: A Future Together. A Vision for Convergence & Co-Existence*. Whitepaper, The Open Group.
- [5] Jonkers, H., Van den Berg, H., Iacob, M.-E., & Quartel, D. A. C. (2010). *ArchiMate® Extension for Modeling the TOGAF™ Implementation and Migration Phases*, White Paper, The Open Group.
- [6] Lankhorst, M. M., et al. (2009). *Enterprise Architecture at Work – Modelling, Communication and Analysis*, Second Edition, Springer.
- [7] Lankhorst, M. M., Proper, H. A., & Jonkers, H. (2010). The anatomy of the archimate language. *International Journal of Information System Modeling and Design (IJISMD)*, 1(1), 1–32.
- [8] Object Management Group (2005). *Unified Modeling Language: Superstructure v2.0*. Technical Report formal/05-07-04.
- [9] Object Management Group (2006). *Business Motivation Model (BMM) Specification*. Technical Report dtc/06-08-03.
- [10] Object Management Group (2008). *Business Process Modeling Notation, v1.1*. Technical Report formal/2008-01-17.
- [11] Project Management Institute (2001). *Project Management Body of Knowledge*. Technical report.
- [12] Proper, H. A. & Greefhorst, D. (2010). The role of principles in enterprise architecture. *Proceedings of the 5th Workshop on Trends in Enterprise Architecture Research*, Delft, The Netherlands.
- [13] Quartel, D. A. C., Engelsman, W., Jonkers, H., & Van Sindereen, M. J. (2009). A Goal-Oriented Requirements Modeling Language for Enterprise Architecture. *Proceedings of the 13th IEEE International EDOC Enterprise Computing Conference*, Auckland, New-Zealand, 3–13.
- [14] Regev, G. & Wegmann, A. (2005). Where do goals come from: the underlying principles of goal-oriented requirements engineering. *Proceedings of the 13th IEEE International Conference on Requirements Engineering*, Paris, France.
- [15] The Open Group (2011). *TOGAF® Version 9.1*, Van Haren Publishing.
- [16] The Open Group (2012). *ArchiMate® 2.0 Specification*, Van Haren Publishing.

- [17] The Stationary Office (2009). *Managing Successful Projects with PRINCE2*.
- [18] Van Lamsweerde, A. (2001) Goal-oriented requirements engineering: A guided tour. *Proceedings of the 5th International Symposium on Requirements Engineering*.
- [19] Yu, E. S. K. & Mylopoulos, J. (1994). Understanding 'why' in software process modelling, analysis, and design. *Proceedings of the 16th international conference on Software engineering*, Sorrento, Italy, 159–168.

■

Henk Jonkers is a Senior Research Consultant at BiZZdesign. In this capacity, he is involved in the company's new developments in the area of enterprise architecture and enterprise engineering. He also participates in multi-party research projects, contributes to training courses, and performs consultancy assignments. Previously, he worked as a Member of Scientific Staff at Telematica Instituut (currently Novay), where he was involved in various applied research projects in the areas of business process modeling and analysis, enterprise architecture, service-oriented architecture, and model-driven development. Henk was one of the main developers of ArchiMate and an author of the ArchiMate 1.0 and 2.0 Specifications, and is actively involved in the activities of the ArchiMate Forum of The Open Group.

■

Dick Quartel is a Senior Research Consultant at BiZZdesign. In this role he contributes to the development and improvement of BiZZdesign's products and services, is involved in research projects, supervises MSc students and interns, and performs consultancy assignments. In addition, he is an author of many scientific and professional publications, and an author of the ArchiMate 2.0 Specification. Previously, he worked as a Senior Researcher at Novay (formerly Telematica Instituut), where he acted as researcher and project manager and contributed to the definition and acquisition of research projects, and as an Assistant Professor at the University of Twente in the areas of distributed systems design, protocol design and implementation, and middleware systems.

■

Henry Franken is Managing Director of BiZZdesign, and is responsible for the research and innovation portfolio of the company. As Chair of The Open Group ArchiMate Forum, Henry has led the development of the ArchiMate Version 2.0 standard. He has been a speaker at many conferences and has co-authored several international publications and Open Group White Papers. Henry is co-founder of the BPM Forum in the Netherlands.

■

BiZZdesign is an innovative company that offers complete and integrated solutions to design and improve businesses. These solutions consist of proven and easy-to-use software tools, best practice models and methods, training courses and business consultancy. The BiZZdesign service lines include Enterprise Architecture Management, Business Requirements Management, Business Strategy and Business Model Management, Business Process Design and Improvement, Lean Management, Business Process Management and Structured Implementation and Governance. BiZZdesign embraces open standards and actively participates in The Open Group (TOGAF, ArchiMate) and the BPM Forum in the Netherlands. The BiZZdesign training courses for TOGAF and ArchiMate have been accredited by The Open Group, and the tool BiZZdesign Architect is certified for TOGAF and ArchiMate.