Adopting ArchiMate ?

Expressing Architecture

Master's Thesis
MSc e-Technology, cohort 2002-01
Roland Ettema

May 23, 2005

Supervisor
Hans Bot

Master of Science Programme e-Technology

CIBIT
CONSULTANTS | EDUCATORS

Proffessor Bronkhorstlaan 10-XII
Postbus 2
3723 MB Bilthoven
Tel: +31 30 230 89 00
Fax: +31 30 230 89 99
www.cibit.com

Management Summary

The goal of this research project is investigating if ArchiMate should be adopted as leading architectural description language for LogicaCMG analyzing the quality aspects and the business potential.

This document presents a methodology to address the quality of an architectural language. This methodology is based upon the semiotic theory that addresses the quality aspects of presentations. The choice for an architectural language has effect for the quality of a representation. This research compares the semiotic qualities of UML and ArchiMate with each other to make a qualitative statement whether LogicaCMG should adopt the language for the architectural domain.

In this research project we where able to measure these qualities for UML and ArchiMate. With our 'conditioned' framework we measured an increase of the four semiotic qualities when ADAM's ADL was substituted by ArchiMate. In other words, LogicaCMG increases the quality of their architectural services when ArchiMate would be applied in architectural description and representations.

Beside a qualitative statement it is important to address the business potential of ArchiMate. The business potential was studied with a single case study at the University of Maastricht. This is a case study based upon an architectural definition study conducted as a project applying the ArchiMate language.

Based upon this case study we recommend to prescribe ArchiMate as architectural language for LogicaCMG to apply in project proposals and plans. It's worth to invest in this standard because the ArchiMate standard is an enabler for many architectural concepts. The UM experienced that ArchiMate is the 'glue' that could bind all architectural stakeholders and experts in the organization.

We can generalize this single case study by signaling that the ArchiMate initiative triggers organizations to rethink of their current architectural approach. From an ICT Service suppliers perspective the real business potential is the implementation of a new designed architectural approach within these organization following LogicaCMG's implementation framework BASIC.

CIBIT

CIBIT supports the innovation of business processes and ICT by consultancy and education. From a strategic perspective, their services offered aim at combining knowledge of organizational processes and innovative ICT without losing sight of people. CIBIT guarantees knowledge that can be widely used.

CIBIT consultants are also the teachers of the study programme. Therefore the expertise gained from the consultancy activities is also used to keep the courses up-to-date and practically oriented. CIBIT helps the students to update the knowledge and skills by providing Master of Science programmes, Master classes, short courses, workshops and tutor-ships. The majority of the CIBIT courses are also offered in-company.

CIBIT offers a range of MSc programmes: Information & Knowledge Technology, Knowledge Managament & Technology, Co-operative Computing, e-Technology and ICT Management. CIBIT has been accredited by Middlesex University, London. This means that CIBIT has earned the right to develop, teach, validate, monitor, evaluate and review study programmes (in the broad fields of information technology and knowledge management, and other fields at Masters level with the prior agreement of the University) leading to Master Awards of Middlesex University. The quality of these courses is finally assured by Middlesex University.

LogicaCMG

LogicaCMG is a major international force in IT services and wireless telecoms. It provides management and IT consultancy, systems integration and outsourcing services to clients across diverse markets including telecoms, financial services, energy and utilities, industry, distribution and transport and the public sector. The group holds a 60 per cent controlling interest in Edinfor, S.A. (Edinfor), one of the largest IT service providers in Portugal, with additional operations in Spain and Brazil. LogicaCMG employs around 21,000 staff in offices across 34 countries and has more than 40 years of experience in IT services. Headquartered in Europe, LogicaCMG is listed on both the London and Amsterdam stock exchanges (LSE: LOG; Euronext: LOG). More information is available from www.logicacmg.com

Parties involved

Author:	Roland Ettema Davidbron 8 NL-6367 TJ VOERENDAAL
E-mail:	Roland.ettema@logicacmg.com
Institute:	CIBIT Proffesor Bronkhorstlaan 10-XII Postbus 2 3723 MB Bilthoven
Supervisor:	Hans Bot
E-mail:	hbot@cibit.nl
Supervisor:	Michel Saleh
E-mail:	Michel.saleh@logicacmg.com
Supervisor:	Michiel Perdeck
E-mail:	Michiel.perdeck@logicacmg.com

Table 1-1 Author and supervisor information

Acknowledgements

I am indebted to many people. I would like to thank Hans Bot for his support during the initiation of this research. Hans inspired me and taught me the principles of doing scientific research for which I am very grateful. I cannot leave this part with mentioning the inspiring discussions with Hans about our common interest in ICT Architecture.

Furthermore I would like to thank my colleagues, in particular to Michiel Perdeck, Martin Pekarek, Ilske Verburg and Michel Saleh for their continuing interest in my work.

I owe gratitude to my family and friends. In particular I thank my wife for her unconditional help and support during difficult times and for putting up with my absence during busy times.

The last words are for the most important persons in my life which are my children. Seeing them grow inspires me to create them a wonderful legacy.

Voerendaal, 23 May 2005 Roland Ettema

Research Outline

This document reports all relevant research results in the same sequence as the research conduction. The document is contains five sections that addresses the major milestones in the project.

PART I Introduction

The first section addresses the structure of the project. The chapter 1introduces the project by addressing the context of the project. Beside the context the chapter addresses the business and research problem. Chapter 2addresses the design of the project wherein the research model and the research questions play a central role. Chapter 3 addresses the topics for the technical research design. These topics are the studied research material, the strategy, reliability and validity of the research.

PART II ADL Quality Evaluation

The second section of this document contains the results of the desk research concerning the quality evaluation of architectural description languages (ADL). Before the actual evaluation, it has to be proofed that the studied ADL's are comparable. Chapter 4 addresses the common grounds of both ADL's. Chapter 5 describes the design of the evaluation criteria where against both ADL's will be evaluated on quality aspects. The last chapter in this section addresses the actual evaluation, which leads to an ADL recommendation, which is based upon theoretical quality aspects.

PART III ADL Business Potential

The third section of this document describes the conducted architecture case study in a large organization where the studied ADL was applied. The case study will not be focused on the project goals but studies the business potential and attractiveness of the applied ArchiMate standard

PART IV Conclusions & Recommendation

The fourth section concentrates on the overall results and conclusions of this research. The conclusions and results are aligned with the project goal and the research question.

PART V Reflection

The last and fifth section addresses the recommendation and reflection. This section describes based upon the results the recommendation if LogicaCMG should adopt ArchiMate. An important aspect is beside this recommendation the generalization of the results. The reflection describes based upon the research process as conducted what went well and what not?

Legenda:

💣 Indicates that the research results must be interpreted under certain conditions or circumstances

c:<nr> Indicates a conclusion that is captured for the research questions and the recommendation

Contents

Part III ADL Business Potential

Part IV Conclusions & Recommendations

Part V Reflection

Figures:

Tables:

Part I

Introduction

1. Project Introduction

This chapter will describe the research context regarding the organization, involved stakeholders, the business problem and the research problem. The description will provide the information of the relevance of this research.

1.1 The organization

LogicaCMG as a global organization employs 21,000 staff across 34 countries and provides management and IT consultancy, systems development and integration as well as outsourced management of targeted business processes. LogicaCMG creates and implements solutions for global clients that embrace the best technological solutions for tangible business results.

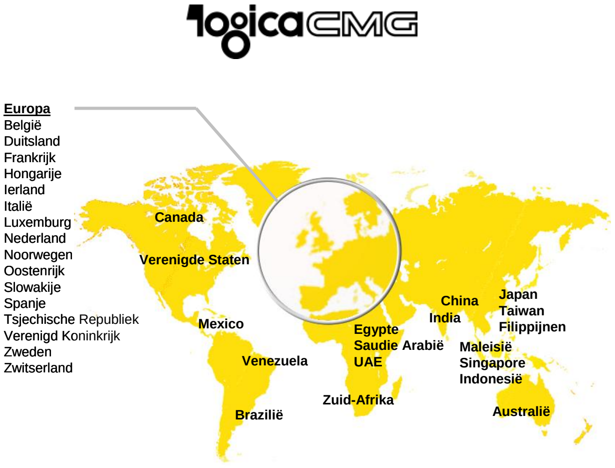


Figure 1-1 LogicaCMG, a major international force

Our focus in this research is the Dutch LogicaCMG organization. All further descriptions refer to the Dutch organization of LogicaCMG. The Dutch LogicaCMG organization consists of divisions, which are: “Public Sector”, “Financial Services”, “Energy, Utility & Telecom” and “Industry, Distribution & Transport”.

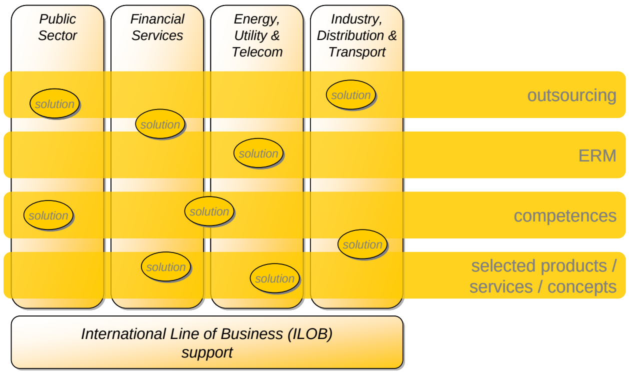

Public Sector
Financial Services
Energy, Utility & Telecom
Industry, Distribution & Transport
outsourcing
ERM
competences
selected products / services / concepts
solution
International Line of Business (ILOB) support

Figure 1-2 LogicaCMG Dutch organization

LogicaCMG also has horizontal propositions which are ‘common’ services or products that are division independent. Some examples of our Dutch propositions are: consulting, enterprise application integration, healthcare, ICT management, mobile business, offshore, outsourcing and security

The author and researcher of this project, Roland Ettema, is working as a business consultant/architect for LogicaCMG IDT (division: Industry, Distribution & Transport). His office location is LogicaCMG Maastricht which is the regional office for the province Limburg in the south of the Netherlands.

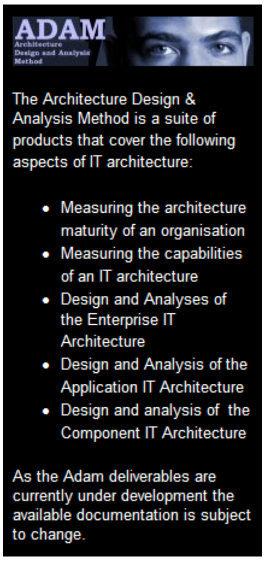
1.2 The business problem

A strategic vision is more than a simple company slogan. It is a vision of where an organization needs to head. A strategic vision starts with a mission statement which gives an organization its own special identity, business emphasis and path for development. The roots of our business problem can be found in LogicaCMG’s mission. The next quote is the relevant part of the mission statement of LogicaCMG for this research:

> *Our solutions must create maximum business impact for the customers. Its our vision that an optimal alignment of the project aspects with the mission and business goals of our customers delivers the maximum contribution to our customers business.*

One of the means for achieving this mission is enterprise architecture starting with the Business – IT alignment. LogicaCMG strives to offer high-quality architectural services to their customers. To achieve this high level of quality, LogicaCMG has two main architectural initiatives: BASIC and ADAM.

The BASIC Framework is aimed at helping our clients deal with actual IT and Business questions. It helps clarify the question and define a solution aligned with the organizational goals and its ability to change. To ensure these goals, BASIC covers the change aspects of the Business (e.g. Market, Products & Services, Business models, Organization additional concepts, methodologies and models need to be applied to close the practical gaps.



The Architecture Design & Analysis Method is a suite of products that cover the following aspects of IT architecture:

- Measuring the architecture maturity of an organisation
- Measuring the capabilities of an IT architecture
- Design and Analyses of the Enterprise IT Architecture
- Design and Analysis of the Application IT Architecture
- Design and analysis of the Component IT Architecture

As the Adam deliverables are currently under development the available documentation is subject to change.

One important aspect in every architectural suite is the central role of the architectural definition language (ADL). This is the language which architects use to analyze, express and model architecture (for this purpose ADAM uses the Unified Modeling Language of OMG.) The architectural fit of a language is a critical success factor for an applicable architectural suite because architectural governance is about communication with the stakeholders so it requires a strong communicative language.

A generic business problem for LogicaCMG arises when BASIC as Framework and ADAM as architectural suite will be surpassed by open initiatives of the community, because customers prefer open standards versus propriatry solutions.

It is important that LogicaCMG determines the correct moment to leave its proprietary techniques to switch over to more generic and open architectural techniques. LogicaCMG must strive towards maximum compatibility with the architectural service market.

The specific context of the business problem for this research is the identification of one real life example that proofs that the described business problem can happen. The lack of an ADL standard brought a group of large Dutch companies together in the ArchiMate project of the telematica institute. They formed a broad consortium of companies and knowledge organizations. The goal of the ArchiMate project was to develop an architectural language and visualization techniques that show the connection and relationships between the various architectural domains.

The Telematica Instituut, Ordina, the Radboud University of Nijmegen, the Leiden Institute for Advanced Computer Science (LIACS) and the Centrum voor Wiskunde en Informatica (CWI) (National Research Institute for Mathematics and Computer Science in the Netherlands) conducted the research. While ABN AMRO, the Dutch Tax and Customs Administration and the ABP Pension Fund contributed practical experience and applied the project results in practice.

This project realised at the end of 2004 a mature ADL specification and a starter kit for Microsoft Visio®. Several companies that where involved are positive about their experiences that they want to adopt ArchiMate as a corporate ADL standard. However they hesitate because they wait until enough knowledge is available by their own employees and professional ICT consultants.

The fact that large customers of LogicaCMG are involved in this project could lead towards a real life example of the generic business problem. As result of this LogicaCMG could be confronted with a significant amount of customers that are ready to define ArchiMate as corporate standard and our propriatry ADAM ADL will loose market value which leads to loss of projects.

1.3 The research problem

This research concentrates on two fundamental aspects of and ADL in the perspective of a large IT service supplier like LogicaCMG namely: the quality and the business potential. The biggest challenge for this research is the way how we express ADL Quality in an objective manor. There are less academic theories and research results available to construct objective evaluation criteria.

The business potential is hard to express in an convincing way. It is important to use the right arguments to win thrust by our audience that a trustworthy approach is applied. The reader decides at the end if he embraces the results of the business potential based on his/her confidence in our approach.

1.4 Parties involved

LogicaCMG is the constituent of the research. The research results, a recommendation, will be addressed to LogicaCMG's architecture competence centre and LogicaCMG's national management. In the research we will use the term "problem owner" which refers to the architecture competence centre of LogicaCMG. Involved employees for this research are:

Michiel Perdeck Title: LCMG Architect Role: Architecture Expert, Mentor Tel : 020-5033202	**Michel Saleh** Title: LCMG Consultant Role: Mentor of this project. Tel: 043-3524200	**Wim Groenendaal** Title: LCMG Principal Consultant Role: ADAM Expert Tel: 020-5033011
Ben Paters Title: Human resource manager Role: Personal education plan. Tel: 043-3524200	**Wouter Paul Trienekens** Title: LCMG Principal Consultant Role: Problem Owner. Tel: 020-5033011	**Nico van Bugenum** Title: Team Manager Role: BC Competence manager Tel: 043-3524200

email = <firstname>.<lastname>@Logicacmg.com

The chairman of the ArchiMate project is Marc Lankhorst (Telematica Instituut.) Telematica Instituut is an unique consortia between business, science and government that on the basis of research develop solutions for the application of information and communication technology in businesses and society. The emphasis lies on a fast translation of fundamental knowledge to pragmatic, market-oriented applications for example electronic business and knowledge management. Members of the Telematica Institute consortium are:

ABN AMRO - ABP/USZO - Basell - Belastingdienst - Corus - CWI - DSM - FEI - Het Roessingh - IBM - ING - KPN - Leiden University/LIACS - Lucent Technologies - Océ - Ordina - Philips Research - ProRail - SURFnet - TNO - TU Delft - Universiteit van Amsterdam - Universiteit van Nijmegen - Universiteit van Tilburg - Universiteit Twente.

With financial contributions of the Dutch government and the consortia members, the telematica institute started the ArchiMate research. The goup of members can be categorised in:

Universities - IT Organizations – Government – Industry – Research Organizations – Tool Vendors

This research will emphasis these categories and is aware of the special interests of these groups. Involved people from this group are:

Marc Lankhorst
Title: Chairman ArchiMate research
Role: ArchiMate Expert
Email: Marc.Lankhorst@telin.nl
Tel: 053-4850456

1.5 The relevance of the project

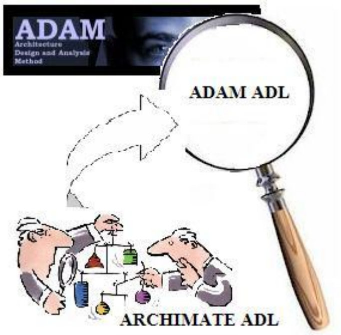
The telematica institute will end the ArchiMate project at the end of 2004. The results of the ArchiMate project will be assessed to close officially all project activities. The result of this research reflects in a transparent way the business value of ArchiMate according a large IT organization. These conclusions are valuable material which can be used by the Telematica Instituut to address the opinion of a large IT service supplier.

If the assessment of an ADL is based on modern convincing architectural theories, the assesment can be reused and maintained to asses other ADL's. The effort is to bring up relevant evaluation criteria that can be used by Architects. Also the results of the evaluation can be used as feedback for the ArchiMate specification.

This research delivers a business case for further investigations. If the result indicates a high business potential and the results are accepted by our targeted audience, we have a business case for further business development.

2. Conceptual Research Design

ADL Substitution - As LogicaCMG we have to review our repeatable solutions to verify if they are still relevant or competitive enough. This research brings up the ADL of ADAM for discussion. The ADL of ADAM has to be reviewed against criteria that are objective and based on modern architectural standards and visions. The result of this evaluation is a qualitative statement about the theoretical and practical value of the ADAM ADL.



Beside a statement about the architectural value of the ADAM ADL it is important to know if one of the available open ADL standards could be a better alternative. In this research ArchiMate was chosen as a potential ADL substitute candidate for LogicaCMG.

ArchiMate motivation - The ArchiMate project has full attention of the architecture community and the academic world. This project under the guidance of telematica institute is able to deliver an architecture language and visualization techniques that picture architectural domains and their relations. The ArchiMate ADL has the potential to become a powerful instrument for the architect that improves the quality of work by the architects. One aspect that deserves attention is that ArchiMate integrates existing and emerging standards wherever possible. Also the efforts that the telematica institute takes to participate in national and international fore and standardization organizations indicate that the organization behind the standard has the drive to find acceptance of their dissemination of the project results.

2.1 Theoretical approach and focus

The research will focus on potential of ADL’s. The research has two main research pathway’s: the quality and business potential of the investigated ADL’s. These two viewpoints are relevant to emphasize in an advice towards the problem owner.

Architecture quality - A very important research part is the qualification of both ADL’s. Defining the criteria for qualifying the ADL’s is a critical success factor within this research. This criteria has to be an objective quality criteria which is a critical factor in success concerning the acceptance of this project result by our stakeholders. The problem owner, The value of the project results concerning the quality of ADL will increase if the applied quality criteria are objective, valid and relevant for the architectural domain.

Business Potential - From a business perspective it is possible that an ADL with less quality can have a higher business potential. If the business potential is high enough it could grow out into a successful architectural solution for LogicaCMG. The pitfall that quality guarantees business potential is a common mistake and will be researched separately from the quality perspective.

2.2 The goal of the research project

The goal of this research project is as follows:

Recommend, based on thorough analysis on the quality aspects and business potential of both ADL’s, whether ArchiMate should, be adopted as the leading ADL for LogicaCMG.

2.3 The research model

Convincing an organization to use an alternative for their standards, methodologies and processes is not easy. In the researchers opinion they should be based on two types of arguments. The first arguments should make a statement of quality improvement that the alternative offers. The second argument should make an indication about the business potential of the alternative. Using these types of arguments in a recommendation is the minimum to expect that the recommendation will be heard. The research model in its project plan definition is presented in the next figure expresses this vision by two research pathway’s [quality research pathway & business potential pathway] converging in a research result, a recommendation.

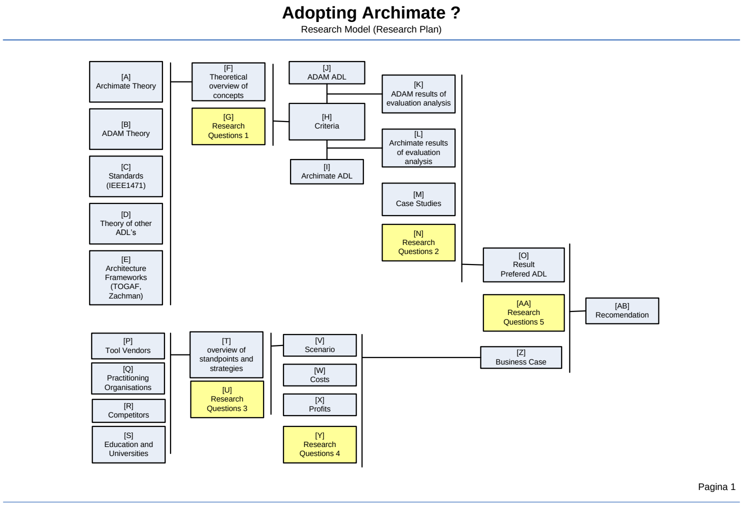


Figure 2-1 Research model from the research plan "Adopting ArchiMate ?"

2.3.1 Changes in the research plan

During the desk research of the quality research pathway problems raised as result of the terminology of all studied ADL’s. How do I identify all concepts and terms that are common? How do I prevent that I evaluate different things. A change in the research plan is necessary to identify the common grounds of both ADL’s.

Another change in the research plan was made as result of an business opportunity in my work. We could apply the ArchiMate language as solution for an architectural problem for a large organization. This opportunity fits in the research pathway “ADL Business Potential” and delivers much qualitative research material. The impact on the research plan was that part [V] – [Z] was changed in case study approach based on this opportunity. The new research plan is represented in the next picture.

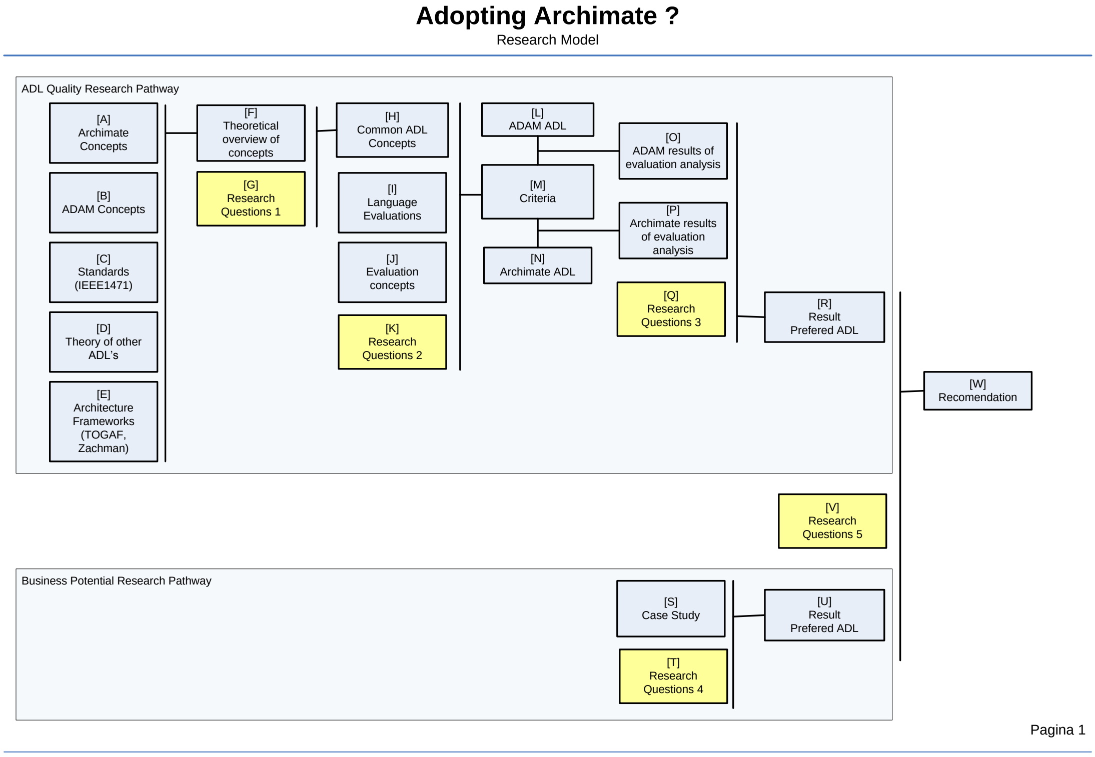


Figure 2-2 Research Model "Adopting ArchiMate ?"

Research is initiated by theory development, when a literature survey is carried out to gather relevant concepts from the concepts and theories of ArchiMate[A], ADAM [B], Architecture standards [C], other relevant ADL's [D] and architectural frameworks [E] (A-E, explanatory, normative) into a theoretical overview.

Theoretical overview of concepts [F] (explanatory, normative) will be confronted with research questions 1 [G] of a theoretical nature. The result of this confrontation will lead to common concepts [H]. These common concepts with objective evaluation criteria formulated after studying other language evaluations [I] and evaluation concepts [J] will lead to objective evaluation criteria [M] after an confrontation with research question [2]

The evaluation results will be confronted with research questions 3 [Q] of an quantitative nature. The confrontation with the research questions 3 delivers the qualitative arguments if ArchiMate is an alternative ADL.

The results of the case study [S] can be confronted with the research questions 4 [T]. This leads to generic approach wherefrom a business potential can be expected Confronting the quality statement [R] and the business solution [U] with the research questions 5 [V] will lead to a recommendation [W] which is the objective of this research.

2.4 Research questions

The central research question has to be efficient and leading through the complete research. Efficiency is in this context the amount of knowledge which is the source where the answer on the research questions will be based on without loosing relevance. The leading factor in the research question gives the direction on which knowledge is necessary for answering the research question. From this we can formulate which research material is necessary. The central research question is:

To what extent is the recently developed ADL of Archimate applicable for LogicaCMG's business and how does it improve the architects quality of work?

The main research questions are followed by a set of sub questions that elaborate the problem:

1 **Which relevant and common concepts share both ADL's?**
   a. Which objective concepts can be derived from the ArchiMate standard?
   b. Which objective concepts can be derived from the ADAM standard
   c. Which objective concepts can be derived from Architecture standards?
   d. Which objective concepts can be derived from other ADL concepts?
   e. Which objective concepts can be derived from Architecture frameworks?
   f. Which common objective concepts do both ADL's contain?

2 **Which objective evaluation criteria can be applied on both ADL's?**
   a. Which important subjects can be derived from 1a – 1e to serve as the grounds of ADL evaluation?
   b. What can we learn from other language evaluations?
   c. Which evaluation frameworks can be identified?
   d. Which subject's 2a – 2c are applicable for an objective quality evaluation for languages in the architectural domain?

3 **How are the results of the evaluation studied?**
   a. What essential aspects are derived from the ArchiMate evaluation ?
   b. What essential aspects are derived from the ADM ADL evaluation?
   c. Which lessons can be learned from the studied evaluation material in 2a-2c?

4 **How is the case study studied?**
   a. Which important concepts 1a – 1e are used in the case study?
   b. Identify the research objects
   c. Which business attractiveness can be identified?
   d. Identify the critical success factors in the case study?
   e. Which conclusion can be drawn based on the case?

5 **What is the interpretation and added value of the results?**
   a. To what extent do the theories apply (1a – 1e) to the practice of ADL practitioners (2b, 2c)?
   b. What is learned about the theory and practice of using the ArchiMate ADL (2.d, 2e)?
   c. What is learned about the case study 4d?
   d. Translate the results 5a-bc towards the central research question in combination with the business problem.

2.5 Description and concepts used

Architecture - 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.

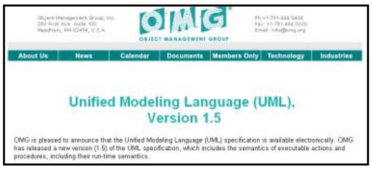
Architecture description - An architectural description (AD) is a collection of products to document an architecture

Architecture description Language - Architecture description language (ADL) is a language which that is designed to serve for architecture descriptions.

ADAM - “Architectural design and analyze method” the architectural suite of LogicaCMG contains methods, architecture products and LogicaCMG's joined experience in the field of architecture. The theoretical fundaments where based on IBM’s enterprise modeling technique LOVEM®.

LOVEM® - Line of Visibility Enterprise Modeling (LOVEM)® is a proven IBM offering for many process-related projects from simple process capture to serious Business Process Reengineering (BPR). It is meant for business and systems professionals. It uses an integrated set of graphical modeling techniques that helps you to analyze and redesign interactions between your customers and internal processes. It also helps you to develop requirements for customer- and employee-oriented automated system. LOVEM is a common specification language that lets business and systems professionals manage all aspects of business processes. But ultimately, LOVEM is a structured methodology for Business Process Reengineering (BPR), Business Process Management (BPM), business process mapping and analysis, as well as business process enabling and implementation. (http://sunset.usc.edu/publications/TECHRPTS/1999/usccse99-514/usccse99-514.pdf)

ADAM ADL - The architectural description language that is preferred in ADAM is UML. The unified modeling language is a graphical language that can be used for modeling perspectives on systems and their behavior.



Object Management Group, Inc.
250 First Ave. Suite 100
Needham, MA 02494, U.S.A.

Ph:+1-781-444 0404
Fax: +1-781-444 0320
Email: info@omg.org

Unified Modeling Language (UML),
Version 1.5

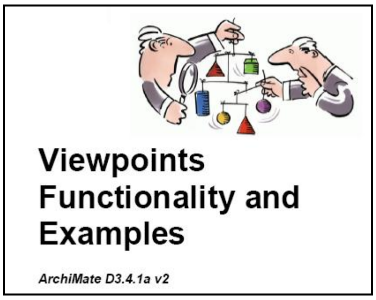
OMG is pleased to announce that the Unified Modeling Language (UML) specification is available electronically. OMG has released a new version (1.5) of the UML specification, which includes the semantics of executable actions and procedures, including their run-time semantics.

Figure 2-3 OMG UML Specification V1.5

This research is based on the “new” UML V 1.5 specifications of the OMG group [31]. This specification can be found on the site of OMG on the following URL: http://www.omg.org/technology/documents/formal/uml.htm

ArchiMate - ArchiMate was introduced in earlier sections of this document. The ArchiMate project has delivered many documents without a version baseline. Each document has its own version number which makes it difficult to address the used specification with one version number. This research used all ArchiMate deliverables as they where public on the ArchiMate project site (http://ArchiMate.telin.nl) on 1 November 2004.

The evaluation between UML and ArchiMate found common ground in the viewpoint concept which will be addressed in the result section. Its important to highlight one frequently used ArchiMate deliverable concerning ArchiMate's viewpoint implementation as specified in ArchiMate's deliverable D3.4.1a v2 [30]



Viewpoints Functionality and Examples

ArchiMate D3.4.1a v2

Figure 2-4 ArchiMate Deliverable 3.4.1a V2

3. Technical Research Design

This chapter will describe the research methods available regarding purpose of research, research approach, research strategy, data collection method, sample selection, analysis of data, and quality standards, as well as providing motivations of the specific methods selected.

3.1 Research Purpose

According to Zikmund [17], business research can be classified based on the purpose of the research. Depending on the nature of the problem, the research will be exploratory, descriptive, or causal (ibid).

ADL Quality Research Pathway - The research purpose of this pathway is descriptive **research.** Since measuring quality of an architecture modeling language is a somewhat new area, not many studies have been performed within this area. It is therefore difficult to identify appropriate theories relevant to this research problem, and not one single study dealing with the exact same research problem has been found. The variables found in the theories reviewed are describing the essence of an ADL from the authors point of view, not one was objective enough to formulate a recommendation for a large ICT supplier whether they should adopt the ADL based on quality criteria. However, considering the shortage of studies performed within this specific area, new variables are not unlikely to be found. The research performed in this study aims at exploring whether or not the variables proposed in the semiotic theories reviewed are relevant for quality evaluation for an ADL as well.

Business Potential Research Pathway – An **exploratory research** will be performed in order to:

- diagnose to which sort of problems the new ADL's could be an solution
- screening for alternative approaches for the problems.
- discover new idea's (potential) with ArchiMate

3.2 Research Approach

3.2.1 Induction or deduction

The emphasis of my research approach is split in two parts. The quality evaluation research path will be based upon deduction. With other words we try to formulate an evaluation framework that is based upon the architectural theories. With this theory we analyze / observe the re search objects (UML & ArchiMate as ADL).

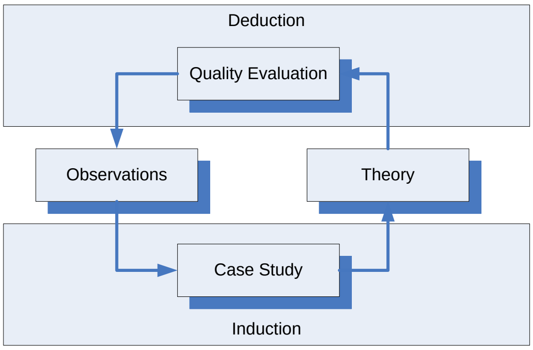


Figure 3-1 Induction & Deduction Research

The case study approach will be **induction**. A deduction emphasis would have too many constraints. Induction emphasis offers in this project more suitable mechanisms like:

- Gain and understand meanings of humans, think of using the opinions of relevant stakeholders and architectural domain experts.

- Close understanding of research context

- The collection of qualitative data, think of collecting the best architecture practices with ADAM and ArchiMate.

- A more flexible structure to permit changes of research emphasis as the research progresses. If the collecting data did not offer the necessary data, but offers a relevant interesting new research optic, the research could undergo minor new changes. This is relevant for the research of the business potential.

- A realization that the researcher is part of the research process. The researcher is involved in the case study and the recommendation phase of the project.

- Less concern with the need to generalize. The research results will be suitable for a small audience.

3.2.2 Qualitative or quantitative research

The nature of the business problem makes it a qualitative research. For both research pathways a qualitative research approach delivers the best facts to formulate the recommendation whether LogicaCMG should adopt ArchiMate

ADL Quality Research Pathway - The research strategy for the part that is concerned with the qualitative analyses of the ADAM/ ArchiMate ADL [A – R] is a performed as **fundamental theory approach**. (Dutch: "gefundamenteerde theoriebenadering"). Research part [A - R] hast the characteristics that fits to the remarks of this research strategy.

- Exploring (tentative, hermeneutic, understanding) attitude of the research worker.
- Continuously mutual and with each other compare of empirical data and theoretical terms.
- A careful and consequent application of research procedures and techniques.

Business Potential Research Pathway - Categories of Exploratory Research are: Experience surveys, Secondary data analysis, Case studies & Pilot studies. The research strategy for the research part [S-U] is as result of the opportunity that a case study can be conducted in a large organization a qualitative research. The researcher becomes the instrument of data collection, and results will be depended on the researcher who conducts the research. This is conform the **qualitative research** paradigm corresponding Zikmund [17]

3.3 Research Strategy

ADL Quality Research Pathway - For research part [A-O] the research is a **desk research** I will use desk research to find answers on my research questions in the theory. This research takes place by means of the method of Vorst. Vorst defines a literature research as: `a range coordinated activities which it makes possible to reliably and purely note what is stated in the professional literature concerning a certain subject.

Business Potential Research Pathway – During the research it happened, that a customer asked for an advice for an architectural problem. We offered our customer a solution that was based upon ArchiMate. The customer was charmed by our approach and accepted our services for several months'. From that time we could use this context for a **case study.** More information about the case study design can be found in 8.2.

3.4 Research material

During the conduction of the research to find answers for research question 1 we saw that the material that was studied did not deliver enough objectiveness that could be translated into objective quality criteria. Most of the studied theories are based on the vision of the author what an ADL should contain. Which is then less useful for our research while its not objective enough for a proper evaluation.

Appendix 1 gives an impression which material was studied from the original project plan perspective. The desk research led us towards new insights and new perspectives based upon the semiotic theory, method point analyses and other evaluation methodologies. These perspectives are objective because they where used to address the quality aspects of languages in general. The problem was that they where never applied on modeling language of the architectural domain.

The research material in a subtract of the new perspectives, the full studied scope can be found in Appendix 1 Useful research sources and their open up strategies for the quality research pathway are:

Persons (Experts):

- Marc Lankhorst (Telematica Instituut), ArchiMate project leader (Informant, Expert)
- John Krogstie (SINTEF), Quality Expert (Informant, Expert)
- Dirk Roeleveld (University Cape Town), Evaluation expert (Informant, Expert)
- Xavier Castelani (CEDRIC: Research Laboratory in Computer Science of the CNAM), complexity metrics (Expert)

Open up strategy regarding persons:

- face-to–face communication is preferred, interview techniques helps to extract relevant knowledge. This is a time consuming approach but delivers qualitative useful knowledge and insights.

Documents regarding:

- Semiotic Theory
- Method point analysis
- Complexity metrics of modeling languages

Open up strategy regarding documents:

- Desk research
- A content analysis by positioning this on a model to express the relevance and the knowledge domain where it reflects on.

3.5 Reliability

Zikmund [17] defines reliability as “the degree to which measures are free form errors and therefore yield consistent results”. Meaning that if a researcher is to repeat the exact procedure and research as described from an earlier researcher, he or she should arrive at the same results and should also able to draw the same conclusions Yin [14]. A high reliability is attempted to be reached by carefully explaining every step taken in every chapter. Furthermore, a structured approach has been adopted, in which every sequential chapter is based on the previous one and aligned with the research model, making it easy for readers and other researchers to follow the logical structure and flow for reading and to use in future research.

3.6 Validity

Yin [14] alleges that construct validity concerns using the correct operational measures for the concepts being studied. It regards whether the researcher has succeeded to develop an operational set of measures and if subjective judgments have been used in the data collection procedure, and if so, how it was done. Yin (1994) discusses tactics available in order to increase construct validity. One is to use multiple sources of evidence during the data collection procedure. Another is to establish a chain of evidence to allow the reader of the case study to follow the derivation of any findings from the question stated to the respondent to the conclusion drawn from it. The case studies performed in this study were able to use both tactics mentioned above for increased construct validity. Both documents and interviews have been used as sources of evidence and therefore a multiple source strategy has been used. The respondents interviewed held similar responsibilities, although different titles, within the case companies, strengthening the cross-case analysis. Also, efforts were made in order to find respondents with the right knowledge and experience to answer the questions needed for the study. Strong chains of evidence have been created since continuous citations have been made throughout the research from where evidence was collected. Reporting each workshop and interview, the possibility to double-check answers and reduce the risk of misinterpretation, was created. There is, however, a risk of translating errors since the interviews / workshops were held in Dutch and that they therefore had to be translated into English. This risk is reduced due to the fact that the respondent was willing to answer any questions or obscurities that may occur further down the line after the actual interviews.

3.7 Visualized Summary

Figure 3-2 provides a visualization of the discussion above, showing the methodological path for each research pathway selected for this study.

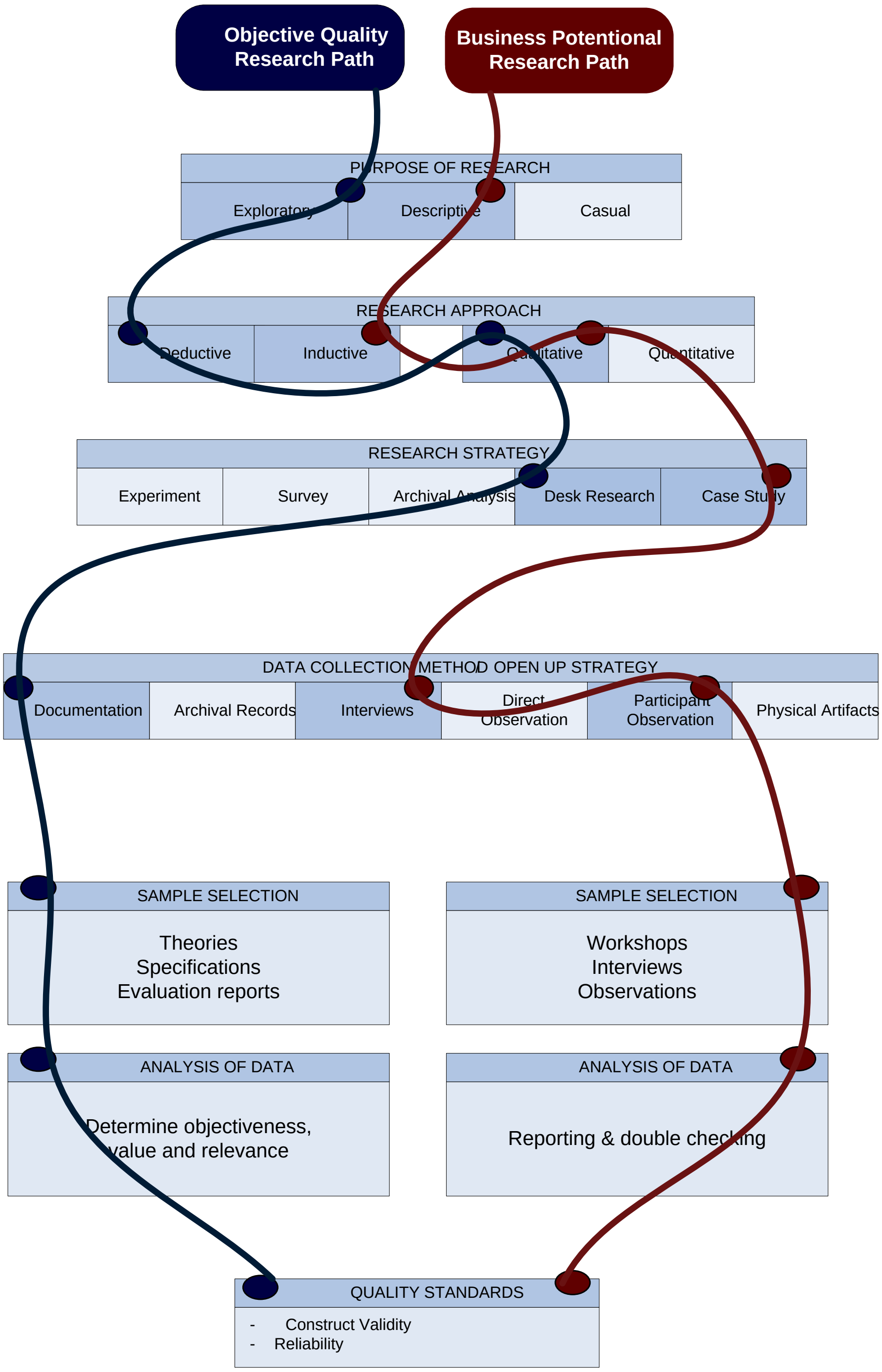


Figure 3-2 Visual Representation of the technical research designed

Part II

ADL Quality Evaluation

4. Common ground of both ADL's

This chapter will describe the common ground for evaluation of both ADL's. Both ADL's use own terms, concepts and definitions in heir specifications. It has to be guaranteed that both ADL's have a common baseline that allows us to base an objective evaluation on.

The work in this stage of the research was conducted and reported in work package 4. The results of this investigation will be reported in a compact form. For more and information and backgrounds the work package 4 report is included in appendix.

4.1 Common understanding

For academically research its important to rely on definitions that has a large group of persons and organization that adopt the definition. If this group is large enough we speak of a standard, in this perspective the IEEE 1471 finds a large group of adopters. In the desk research we identified several references from the UML and ArchiMate specifications towards this standard. To identify the similarities between UML and ArchiMate our research studied both specifications from the IEEE 1471 perspective. This perspective differs from other resresearch those studiese similar concepts like the study of M.Lankhorst [1]. The results where astonishing what terms, concepts and definitions which looked at a first glance very different are in fact similar from this point of view. To share this experience it is essential to understand the basic concepts from IEEE 1471.

4.1.1 The IEEE-1471 concepts

The following concepts are essential for the architectural domain which addresses the topic of viewpoints and views. The concepts have been adapted from the more formal definitions contained in ANSI/IEEE Std 1471-2000; Recommended Practice for Architectural Description of Software-Intensive Systems. [1]

System - A system is a collection of components organized to accomplish a specific function or set of functions.

Architecture - The architecture of a system is the system's fundamental organization, embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution.

Architecture Description - An architecture description is a collection of artifacts that document architecture.

Stakeholders - Stakeholders are people who have key roles in, or concerns about, the system: for example, as users, developers, or managers. Different stakeholders with different roles in the system will have different concerns. Stakeholders can be individuals, teams, or organizations (or classes thereof).

Concerns - Concerns are the key interests that are crucially important to the stakeholders in the system, and determine the acceptability of the system. Concerns may pertain to any aspect of the system's functioning, development, or operation, including considerations such as performance, reliability, security, distribution, and resolvability.

View - A view is a representation of a whole system from the perspective of a related set of concerns.

In capturing or representing the design of system architecture, the architect will typically create one or more architecture models, possibly using different tools. A view will comprise selected parts of one or more models, chosen so as to demonstrate to a particular stakeholder or group of stakeholders that their concerns are being adequately addressed in the design of the system architecture.

Viewpoint - A viewpoint defines the perspective from which a view is taken. More specifically, a viewpoint defines: how to construct and use a view (by means of an appropriate schema or template); the information that should appear in the view; the modeling techniques for expressing and analyzing the information; and a rationale for these choices (e.g., by describing the purpose and intended audience of the view).

A view is what you see. A viewpoint is where you are looking from - the vantage point or perspective that determines what you see.

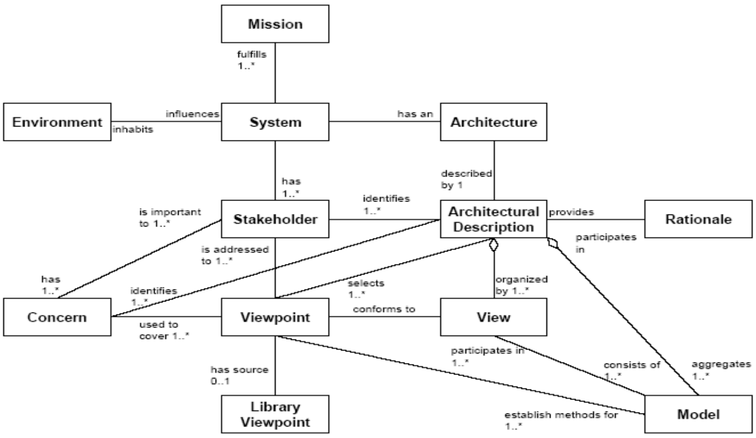


Figure 4-1 IEEE-1471

4.1.2 The IEEE-1471 logical tool implementation

At this point it is for the research process important to address the logical implementation of IEEE 1471 as seen in Architectural tool implementations. It is in the context of the research important to address the choice of tool support in the case study for Bizzdesign Architect®. Again the logical structures of this type of modeling environments are equal with other implementations (Mavim, Popkin Enterprise Architect).

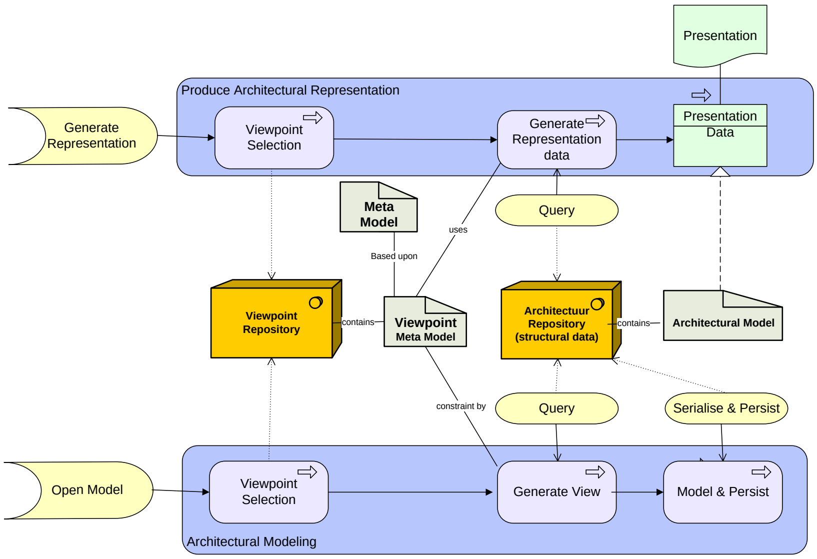


Figure 4-2 Logical representation of ADL Tooling support

Figure 4-2 represents the structure of ADL tooling with the focusing on the IEEE 1471 aspects in the processes of modeling and generating a representation of the architectural description in the repository. We see in this figure the role of the meta-model of the ADL. Viewpoints are based upon these meta models where we define which architectural concepts are involved in the viewpoint. (= concern of stakeholder) These viewpoints act as a sort of filter for the content wherein the modeling takes place or as filter for the content used within the representation that is generated.

The meta model is essential in this logical model. It bridges the graphical modeling concept with the syntactical domain. A viewpoint is a subset of concepts from the language meta model and bridges the semantic contract with the stakeholder with the syntactic data in the repository.

4.1.3 Generic system elements

The major challenge is to detect the common ground between UML and ArchiMate as a fundamental statement that they are comparable. UML and ArchiMate speak both of models. This is a common terms in both techniques. Veryard stated: “Every model can be expressed in four metaphorical directions” [23]

1. ‘inwards’, toward the internal composition of the element;
2. ‘upwards’, towards the elements that are supported by it;
3. ‘downwards’, toward its realization by other elements;
4. ‘Sideways’, towards peer elements with which it cooperates.

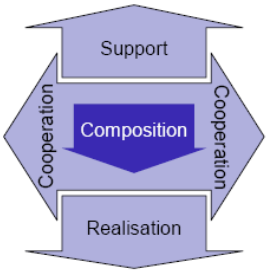


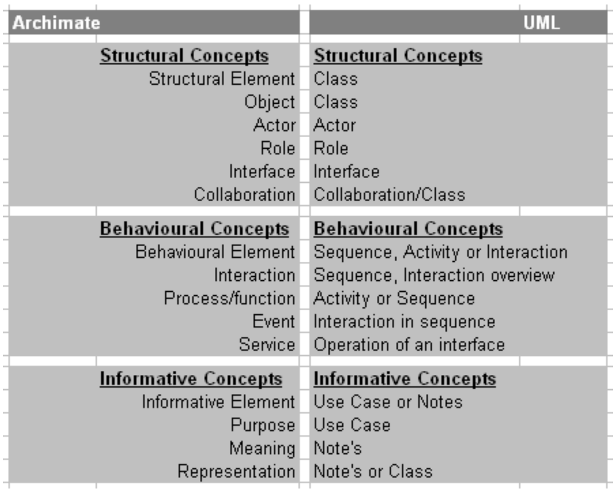
Figure 4-3 Metaphorical directions of modeling

The authors of ArchiMate’s deliverable D3.4.1a v2 used veryards view of models to identify the basic types of elements in architectural descriptions. They are addressed as:

1. active elements, e.g., the composition of a business actor from sub actors, i.e., an organization structure;
2. behavior elements, e.g., the structure of a business process in terms of sub processes;
3. Passive elements, e.g., the information structure in terms of data objects.

Veryard makes a difference between elements that are active or passive in communication. Active elements manipulate data and passive elements are a source or a target of data. The behavior elements are objects that response with a behavior when the are stimulated. This theory of generic system elements delivered us the first similarity experience between UML and ArchiMate.

The essence is of this experience is that the concepts as specified in the specifications of UML and ArchiMate can be grouped from this point of view. In the representation the term informative concepts are used which addresses the generic passive concepts.



Archimate	UML
Structural Concepts	**Structural Concepts**
Structural Element	Class
Object	Class
Actor	Actor
Role	Role
Interface	Interface
Collaboration	Collaboration/Class
Behavioural Concepts	**Behavioural Concepts**
Behavioural Element	Sequence, Activity or Interaction
Interaction	Sequence, Interaction overview
Process/function	Activity or Sequence
Event	Interaction in sequence
Service	Operation of an interface
Informative Concepts	**Informative Concepts**
Informative Element	Use Case or Notes
Purpose	Use Case
Meaning	Note's
Representation	Note's or Class

Figure 4-4 UML-ArchiMate Concept mapping

4.1.4 Generic viewpoints on systems

Combining the generic elements of systems with the IEEE 1471 concepts deliver us generic viewpoints on systems. In these context systems indicates to the whole enterprise architecture as system. We addressed the following generic viewpoints on systems:

1. Behavior viewpoint: Interest in the behavior of a systems when stimulus occurs
2. Interaction viewpoint: Interest in interaction between elements in a system
3. Implementation viewpoint: Interest in structural decomposition of a system
4. Purpose viewpoint: Interest how the system relates with its environment.

If we translate this list and allow viewing the enterprise architecture as a system we specify the generic viewpoints for the architectural domain:

1. Composition viewpoint: Interest in the composition of elements in the architecture.
2. Cooperation viewpoint: Interest in interaction between the active elements in the architecture.
3. Realization viewpoint: Interest in structural decomposition of the architecture which element realizes a service/product.
4. Support viewpoint: Interest in the legitimacy of elements in the architecture.

Knowing these generic and architectural viewpoints both specifications specify views. Diagrams in the UML specification are views. The UML specification did not specify the viewpoint from where the diagram conforms to. (see figure IEEE 1471). An mapping of the views specified in the ArchiMate specification proves that diagrams can be substitute architectural views which are predefined for the architectural domain according the ArchiMate's deliverable D3.4.1a v2 [30]

Composition		Cooperation	
ArchiMate	**UML V1.5**	**ArchiMate**	**UML V1.5**
Organization (active)	Class Diagram	Actor Cooperation	Collaboration Diagram
Business Function (behavior)	Use Case with sequence diagram	Business Process Cooperation	Activity Diagram
Business Process (behavior)	Sequence Diagram / Activity	Application Cooperation	Activity Diagram
Information Structure (passive)	Class Diagram		
Application Structure (active)	component diagram		
Application (behavior)	State machine, Activity Diagram		
Infrastructure (active)	Deployment Diagram		
Support		**Realization**	
ArchiMate	**UML V1.5**	**ArchiMate**	**UML V1.5**
Product	Class Diagram	Service Realization	Class-, Component diagram
Application Usage	Activity, Sequence, Component Diagrams	Implementation and Deployment	Component, Deployment Diagram

Table 4-1 Research baseline "Viewpoints and their diagram equivalent"

4.2 Conclusions

In this chapter we designed a fundament for our research. By applying standards and theories during this process we identified remarkable conclusions that form the must be involved in our final conclusions.

c:1 **UML has restricted support for IEEE 1471 .**

UML Concepts are strictly related to diagrams. This is a big disadvantage for the architectural domain in relation to IEEE 1471 standard. This standard states that viewpoint definitions are useful to serve the corresponding stakeholder with accurate views. The predefined viewpoints as defined in ArchiMate's deliverable D3.4.1a v2 [30] showed that every viewpoint contained a corresponding meta-model. This viewpoint meta-model constrains the architect in usage of concepts in the views for the specific stakeholder. In other words in the architectural domain we must be capable to define own diagram types by defining a viewpoint that can be constrained with a corresponding Meta model.

c:2 **UML cannot constrain semantics for a viewpoint.**

This is only possible if the specification of a modeling language allows us to use all concepts in to design a specific Meta Model for a viewpoint with respect of the Meta model of the modeling language.

c:3 **ArchiMate concepts can be mapped to UML Concepts.**

The conclusions after studying the deliverable 2.2.3b is that almost all discrete ArchiMate concepts can be mapped to all discrete UML concepts. What have to be addressed are the conditions that made this mapping possible:

- Massive use of the stereotype extension point in UML
- Language Meta model is not regarded during the mapping.

The mapping deliberdly denies the extra capabilities of the UML concept. (I.e. ArchiMate: Role -> UML: Class, a class does contain methods a role does not need any. It is mappable under the condition that we may not use the whole UML concept)

c:4 **UML and ArchiMate are comparable.**

This conclusion is based upon the following arguments: systems:

- UML and ArchiMate support the four metaphorical directions of modeling [18]
- Both ADL's can group their specified concepts in according the three basic architectural model elements. (active-, behavior- & passive elements).
- The terms "diagram" and "view" are in the research context the same.
- The concepts of ArchiMate can be mapped on the concepts of UML
- Through concept mapping, the basic elements and applying veryard we can identify for every ArchiMate viewpoint a UML equivalent diagram.

c:5 UML cannot constrain semantics for a viewpoint.

Comparing the concepts of UML and ArchiMate we identified that UML is capable to express almost all concepts of ArchiMate by using the stereotype extension mechanism. In the UML domain we call the modification of UML for a specific domain a UML profile. If we design a UML based Meta model for a viewpoint we are only allowed to use UML Concepts eventually tagged with a stereotype. The problem is that on the level of the modeling language all relationships between the UML concepts do not respect the value of the stereotypes. This causes for the architectural domain semantic wrong models, for example:

A Meta model containing two classes with a unidirectional relationship is syntactically correct. But when I provide one class with a stereotype “business function” and the other with “node” the view will be syntactically correct but semantically incorrect.

5. Designing objective ADL Evaluation criteria

This chapter reports the insights and design of objective evaluation criteria where against both ADL's will be evaluated. The chapter reports the process, the evaluation model and conclusions based on the gathered insights.

5.1 Specifying evaluation criteria.

The design of the research model gave insight which topics could be studied to formulate objective evaluation criteria. Appendix 1 gives an overview which papers where studied to find answers of the sub questions of research question 1., addresses in short the answers and findings on the sub-questions of research question 1.

The answers that where formulated based upon the desk research was dissatisfying because each paper has a certain perspective from the author. If evaluation criteria where based upon this information they wouldn't be objective and useless for this research. The scope of the used resources as stated in the research model [a] - [e] must be increased to lead to objective quality criteria.

5.2 Finding objective evaluation criteria

As shown in Appendix 1 the studied resources where increased in the research model. The scope of the desk research increased by including research topics that addresses : "The Architectural Domain", "Evaluations in general" & "Evaluation Frameworks". The increase was fertile and delivered a broader quality perspective on languages in general that was unknown at the time the research plan was formulated. This broader scope delivered two objective theories regarding the quality of languages which can be applied on ADL's

The first theory is the Method Points Analysis [21] which is a metric for indicating a method complexity. It helps to choose between competing methods like the modeling methodology between UML and ArchiMate in the Architectural domain. The method point's analysis is based upon a generic method representation model that is developed by the Graham McLeod [21]. The theory in this research context can be studied in section 5.3

The second theory is based on the semiotic theory and delivers objective insights, aspects and subjects on languages and signs to base evaluation criteria on. This theory and the translation towards this research can be studied in section 5.4

5.3 Complexity evaluation framework

Graham McLeod was inspired of the Function point analysis [22]. McLeod compared method fragments behavior with the behavior of a software system and came till the following conclusions:

- It interacts with its environment.
- It has users (practitioners) who provide it with information and receive outputs (deliverables) from the processes it performs.
- It manages information to support the various processes and outputs.
- It interfaces with other methods (or method fragments) which may precede or follow it, or operate in parallel to it.

The advantage of this view is that the methodology can be expressed in an abstract model. The process that McLeod presented is similar with the process of expressing a software system in an entity/object model. The methodology model of McLeod has the advantage that it is possible to retrieve complexity metrics. McLeod proposes the following counting procedure:

1. **Express the method** to be counted in the method model in terms of Tasks, Resources and Deliverables.
2. **Determine the counts for each type of deliverable** specified in the method. We distinguish three types of deliverables:
   - *Graphical Deliverables* such as diagrams and models. Examples would be entity relationship diagrams, data flow diagrams, class hierarchy diagrams and project management network diagrams
   - *Tabular Deliverables* which can be expressed as columns and rows or as records in a relational table. Examples would include the definition of the attributes of a data group, which may have columns for name, type, length and valid ranges; and a system consistency matrix
   - *Textual Deliverables* which include long descriptions and narratives as well as more structured reports and hypertext documents We will discuss the counting of each type of deliverable in the following sections.
3. **Determine and add the count for task complexity**. This will normally not come into play, unless the tasks are more complex than would be apparent from the deliverables produced.

However not every deliverable (instance of a description) has to be counted. An equivalent deliverable expressed in a different notation should be assigned to an earlier count.

This research uses only the counting methodology for the graphical deliverables. The reasons for this choice are:

- ArchiMate has no specifications regarding textual and tabular deliverables.
- ArchiMate model persistency is based on the eclipse modeling framework. The EMF XML format is map able to the XMI/UML standards. We can conclude there is no difference in structure. There is only difference in field usage.

5.3.1 Counting graphical Deliverables

We identify several components of a graphical model which are relevant to our purpose:

- *Symbol types (count 1).* These are unique shapes that represent something in the domain being modeled. Examples include entity boxes in an entity relationship model, process boxes in a data flow diagram, and class symbols in an object model.
- *Link types (count .5).* These are unique types of connection between symbols. Examples would include a data flow arrow on a data flow diagram, a relationship line on an entity relationship diagram, and inheritance relationships in an object model.
- *Embellishments (count .5).* These are any unique type of modifier which can be added to a symbol, a link or the model canvas. Examples include: text label of a data flow, cardinality indicator on an entity to entity relationship, and indication of a key field identity on a data group symbol. Further examples include: a duplicate marker added to a data store on a data flow diagram, a boundary around symbols indicating geographical location or mutual exclusion, and a business rule related to a symbol on an event model.
- *Decomposition (count .5).* It is common that a symbol on one model can be expanded into another model at a greater level of detail. An example would be a process box (representing a complete system) within a context diagram which may be decomposed to a data flow diagram expressed as a separate model. For each type of symbol which can be expanded into another model (which is cross referenced from this one) count .5. If this diagram (model) can contain a reference to a parent model, count .5.

5.4 Semiotic evaluation framework (theory)

Semiotic is the study of signs and can be used to describe the form-, meaning, and use-related aspects of information. Semiotics can serve as a theoretical framework to integrate the different approaches required to define quality criteria for information aspects.

This semiotic idea's where already found in the old Greek era. Modern philosophers like Charles Pierce (1931-1935) and Charles Morris (1938) describes semiotics in terms of logical components.

The semiotic theory until 1990 contained three levels in sign interpretation:

- Syntactic level, expresses the form of signs
- Semantic level, expresses the meaning
- Pragmatic level, expresses the application

These logical components where used in a research of Stamper (1992) where Stamper investigated the meaning of information and communication in large organizations. Stamper introduced the semiotic ladder with three additional semiotic levels.

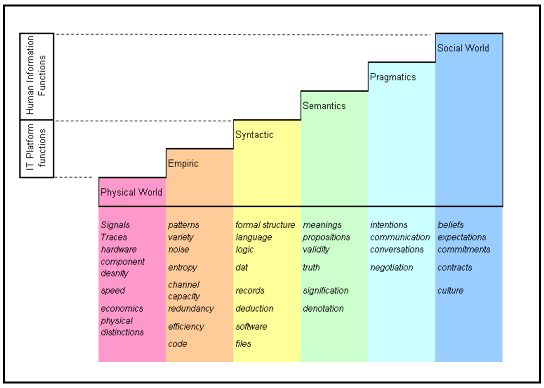


Figure 5-1 Semiotic Ladder R, Stamper

Although these levels have no scientific foundation other writers and publishers used these definitions while they make sense realizing that there is no scientific proof.

- Social level, shared social context
- Empiric level, statistical properties of sign representation
- Physical level, physical/material properties of sign representation.

These six identified levels leads to six views on signs that together can be depicted as a semiotic ladder. This semiotic ladder consists of the views on signs from the perspective of physics, empirics, syntactic, semantic, pragmatics, and the social world. The addition of a view on information from the social world stresses that information use is always a part of human behavior in a social setting, where norms or social conventions govern people's behavior [27]. The semiotic ladder shows that there are six views on information that together form a complex conceptual structure. This means that seeing 'information' as a primitive or atomic concept is wrong [28].

5.4.1 Semiology in relation to ADL's

Manny architecture modeling languages and frameworks treat the information output as a primitive or atomic concept. The first attempt to position architectural information in a more complex conceptual structure is IEEE 1471. IEEE 1471 is a specific conceptual model for the architectural domain.

Ronald Stamper's semiotic ladder is a more generic conceptual structure. This structure can be applied on a broad range on information like magazine covers, natural languages or even traffic signs. This research tries to apply the semiotic views and principles on architectural information. We adapt the vision of IEEE 1471 that architectural information has relationships with stakeholders and their concerns. But in my personal opinion we want to achieve more.

Instead of producing architectural information when want to achieve norms, and specific information field paradigm. At the core of this paradigm are fields of norms, binding together groups of people (stakeholders). The norms allow meaning and responsibilities to be clearly specified, thus fostering the active construction of social reality, shared understanding and mutual commitments. The semiotic ladder helps to indicate how the researched phenomena (an architectural modeling language) contribute to this architectural information paradigm.

Read the last section again and
think of all the communications between parties
when shared understanding and mutual
commitments can be achieved.

i.e.
principles and business objectives
become social norms of the community.

-- Vision of the author "Roland Ettema" --

IEEE 1471 addresses some particular aspects of this vision but it does cover all aspects. The semiotic theory (theory of signs) goes beyond the aspects of IEEE 1471 and addresses the pragmatic- and social aspects of architectural signs. The discussion in this research uses the model of Ronald Stamper's that is known as the semiotic ladder. This paper discusses the value of architectural signs referring to this model.

5.4.2 A semiotic based quality framework

Krogstie, Sindre and Lindland [4] [8] have developed a framework for discussing the quality of models such as those found in the architecture designs. This framework can be applied to every model that can be found in the architectural domain, and fits into our research scope which are modeling architecture with UML and ArchiMate. For the ease of use, we identify the quality framework with KSL-QF where KSL stands for the authors of the quality framework .

The KSL-QF has three unique properties:

- It distinguishes between quality goals and means.
- Since modeling is essentially making statements in some language, it is closely linked to linguistic and semiotic theory. This addresses the objectiveness of the approach
- It is based on a constructivist world-view, recognizing that models are usually created as part of a dialogue between the participants involved in modeling, whose knowledge of the modeling domain changes as modeling takes place.

Further details on the framework can be found in [5][6][9] where several modeling approaches including OMT and approaches for flexible workflow modeling have been evaluated. What one is able to evaluate using the framework is the potential of a modeling approach to support the creation of models of high quality. Used in this way we only utilize parts of the framework as will be illustrated below. How the framework can be specialized for requirements specification models is discussed in [11].

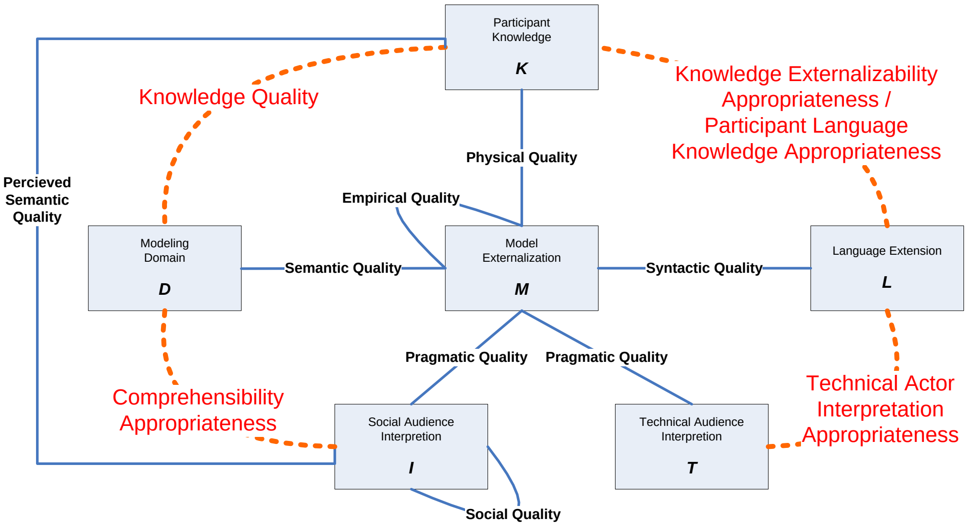

Participant Knowledge
K
Knowledge Quality
Knowledge Externalizability Appropriateness / Participant Language Knowledge Appropriateness
Physical Quality
Percieved Semantic Quality
Empirical Quality
Modeling Domain
D
Semantic Quality
Model Externalization
M
Syntactic Quality
Language Extension
L
Pragmatic Quality
Pragmatic Quality
Comprehensibility Appropriateness
Technical Actor Interpretation Appropriateness
Social Audience Interpretion
I
Technical Audience Interpretion
T
Social Quality

Figure 5-2 Quality Framework (Krogstie, Sindre & Lindland)

The main concepts of the framework and their relationships are shown in Figure 5-2 and are explained below. Quality has been defined referring to the correspondence between statements belonging to the following model interpretation aspects:

- L, the language extension, i.e. the set of all statements that are possible to make according to the graphemes, vocabulary and syntax of the modeling language.
- D, the domain, i.e. the set of all statements which can be stated about the situation at hand.
- M, the externalized model, i.e. the set of all statements in someone's model of part of the perceived reality written in a language.
- K, the relevant explicit knowledge of the audience.
- I, the social audience interpretation, i.e. the set of statements which the audiences (i.e. those that need to understand the model) think an externalized model contain.
- T, the technical audience interpretation, i.e. the statements in the conceptual model as they are interpreted by the different modeling tools used.

The main quality types are indicated by solid lines between the sets, and are described briefly below.

(The KSL Quality framework uses the term set to indicate the group of (L,D,M,K,I,T). It was not possible to find a more specific term. For this research we use the term KSL-QSet)

- **Physical quality**: There are two basic quality means on the physical level:
    - Externalization, that the explicit knowledge of some person has been externalized in the model by the use of a modeling language
    - Internalizeability, that the externalized model is persistent and available, enabling the other persons involved to make sense of it.

- **Empirical quality** deals with error frequencies when a model is read or written by different users, coding, and ergonomics of computer-human interaction for modeling tools.
- **Syntactic quality** is the correspondence between the model and the language extension of the language in which the model is written.
- **Semantic quality** is the correspondence between the model and the domain. The framework contains two semantic goals:
    - Validity, which means that all statements made in the model are correct and relevant to the problem
    - Completeness, which means that the model contains all the statements that are correct and relevant about the domain.

  These goals are made more applicable by introducing the notion of feasibility.
- **Perceived semantic quality** is the similar correspondence between the participants' interpretation of a model and his or her current explicit knowledge. Whereas the primary goal for semantic quality is a correspondence between the externalized model and the domain, this correspondence can neither be established nor checked directly. To build a model, one has to go through the participants' knowledge regarding the problem at hand, and to check the model one has to compare with the participants' interpretation of the externalized model.
- **Pragmatic quality** is the correspondence between the model and the audience's interpretation of it.
- **Social quality**: The goal defined for social quality is agreement among participants' interpretations.

It is important to note that the framework deals with quality on the individual model level rather than the quality of modeling languages and techniques. Illustrative examples are syntactic quality, the degree to which an individual model is in accordance with the language and semantic quality, the degree to which an individual model is in accordance with the domain. However the framework is applicable if the framework will be conditioned for the architectural domain. (see 5.5) It's worth to mention that it will be difficult to evaluate ADL's on social, and perceived semantic quality as result that we do not evaluate an architectural description with an audience but we evaluate the language.

5.4.3 Quality as maturity indicator

The authors of the paper “Achieving Quality in Natural Language Requirements” [20] achieved to express a models maturity in the terms of the identified qualities of the KSL-QF . This representation is drawn in [20]. The results of the KSL-QF can be drawn in this model to indicate the maturity of UML/ArchiMate.

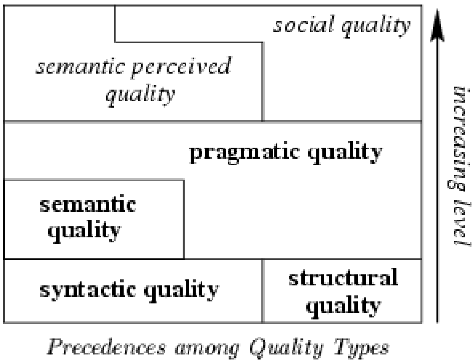


Figure 5-3 Semiotic Maturity Model

5.5 Semiotic evaluation framework for ADL’s

The KSL-QF addresses two perspectives of a modeling language:

1. The constructs of the language (the actual language syntax and notation)
2. How the constructs are visually represented (through a computerized, supporting tool).

The practical application of the framework requires the analysis of each of these two perspectives, starting by identifying and applying the following five main quality groups. The definitions as outlined by the framework are given in the next sections.

5.5.1 Domain Appropriateness

Domain appropriateness can be evaluated from two points of views. The first view is that a language is capable to face the challenges of architecture modeling [26]. The second view is to check if architectural statements can be expressed. The practical evaluation guidelines regarding these two points of vies will be explained in the next sections:

Language capability to face the architectural challenges

The architectural challenges are retrieved from a case study with the title “A UML-driven Enterprise Architecture Case Study” conducted by Dr. Frank Armour [26]. These challenges are based upon the views of the Enterprise IT Architecture Framework [26].

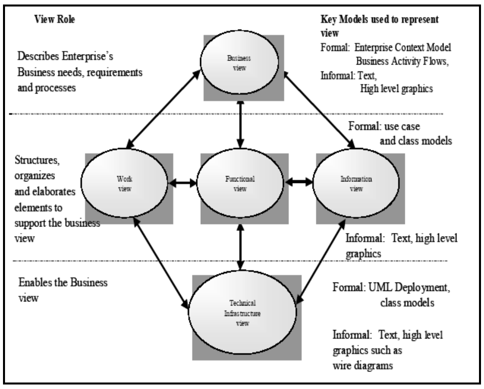

View Role
Key Models used to represent view
Describes Enterprise's Business needs, requirements and processes
Business view
Formal: Enterprise Context Model Business Activity Flows,
Informal: Text, High level graphics
Formal: use case and class models
Structures, organizes and elaborates elements to support the business view
Work view
Functional view
Information view
Informal: Text, high level graphics
Enables the Business view
Technical Infrastructure view
Formal: UML Deployment, class models
Informal: Text, high level graphics such as wire diagrams

Figure 5-4 Enterprise IT Architecture Framework

The strategy to make a statement for domain appropriateness is that we ask the assessor to answer the question: "Is the language is capable to face the architectural challenge" This strategy is based upon an open questionnaire and is not suitable to quantify. The architectural challenges are defined as shown in Table 5-1

View	Architectural Challenges
Business	▪ Capture "horizontal" processes or workflows that can cross multiple use cases and Business Application Packages ▪ Present the information in as "user" friendly form as possible, but still define the problem in enough rigors to highlight routes, roles and the information that is acted on. ▪ Clearly identifying what is in, and what is outside, the enterprise
Functional	▪ Capture the interfaces across multiple applications ▪ Identify and define various interfaces to both external and internal actors
Information	▪ Given the large amount of information that can be modeled related to an enterprise is to determine what data is relevant to the overall enterprise, model it at the right level of abstraction.
Work	▪ Capturing the key aspects about departments, locations and roles
Technical Infrastructure	▪ Robustly documenting the key technical components such as server platforms, client platforms, legacy systems, networks, middleware, etc. ▪ Determining the key "paths" and their characteristics, based on usage, through the technical components.

Table 5-1 Architectural challenges following Dr. Frank Armour

Capabilities to express architectural statements

All statements in the architectural domain should be expressible, and the modeling language should not allow for statements outside of the architectural domain to be expressed. This can be easily checked by evaluating the architectural concepts in the Meta models of UML and ArchiMate. The amount of specific architectural concepts in the meta-models are indicators that ca be quantified.

Questions (formulated as requirement for the architectural domain):

- The language should be organizational independent
- The language should not allow expressiveness outside the architectural domain.
- The language should be able to express all architectural concepts in the following architectural dimensions
  - Generic : IEEE 1471, Architecture principles etc…
  - Strategy : Principles, Goals, Drivers
  - Business : Structure, Actor, Roles, Collaboration, etc…
  - Process : Process, Workflow, Products, Services etc…
  - Application : Interface, Service, Collaboration etc…
  - Technical : Service, Device, Network, Platform etc…
- The language should be able to express the following seven general perspectives
  - Structural : static structure (entities and relationships)
  - Functional : the processes, activities and transformations
  - Behavioral : states and transactions
  - Rule : rules for certain processes, activities..
  - Object : objects methods, attributes and classes
  - Actor and role : role, society and organization
  - Communication : language actions, meaning

5.5.2 Participant language Knowledge Appropriateness

All statements made in the modeling language are explicit knowledge of the participants; therefore the conceptual basis must correspond with the way in which the participant perceives the problem. Participant language knowledge appropriateness is primarily a means to achieve ***physical and pragmatic quality***.

Questions:

- The terms or concepts must be same as those for the organization. At least all specific organizational terms must be "mappable"
- It must be easy to learn the language.
- The external representation must be intuitive, meaning that the symbol represent the concept better then another symbol would.

Indicators:

- Intuitive expressiveness
- Terms of concepts are equal of the architectural domain at least map able.
- Availability extension points

5.5.3 Knowledge Externalisability Appropriateness

There should be no statements in the participants' knowledge that cannot be expressed in the language. Knowledge externalisability appropriateness is primarily a means to achieve ***physical quality***.

Questions:

- Does the Meta model provide the right concepts for modeling
- Is it possible to use the language to model ones knowledge of an architectural domain like business or information in an efficient way?

Indicators:

- Structural data deliverables, Graphical deliverables, Textual deliverables

5.5.4 Comprehensibility Appropriateness

Participants in the modeling effort must understand all possible statements of the language. This means that:

- the language phenomena should be easily distinguishable,
- the number of the phenomena should be reasonable,
- the use of phenomena should be uniform,
- the language must be flexible in the level of detail,
- the language must allow for separation into areas of concern, and
- the language must have expressive economy (the most frequent and important statements are brief).

Comprehensibility appropriateness is primarily a means to achieve ***empirical and pragmatic quality***.

Questions:

- Is it easy comprehending the model?
- Are the relationships between the concepts clear and understandable

Indicators:

- Easily distinguishable concepts
- Reasonable number of concepts
- Uniform usage of concepts
- Flexible level of detail
- Support for separation into areas of concern
- Expressive economy (the most frequent and important statements are brief).

5.5.5 Technical Actor Interpretation Appropriateness

This relates the language to the technical audience interpretation, which should lend itself to automatic reasoning through formality and executability. The power of formal semantics lies in three aspects:

- The process of making a more formal specification may reveal errors and ambiguities at an early stage in the development process
- Formal and even automated proofs may be available
- The remaining (or improvable) rules may be translated into executable constraints in some imperative language.

The different aspects of technical actor interpretation appropriateness are a means for achieving ***syntactic, semantic and pragmatic quality***.

Questions:

- Has the methodology a well-defined syntax.

Indicators:

- formal syntax, metrics, automated proofing, simulation

The areas of the Krogstie's Model that were used were those of Domain Appropriateness, Comprehensibility Appropriateness and Technical Actor Interpretation Appropriateness, as these are context independent.

The remaining two areas of evaluation deal closely with "participants" and thus require the context of a project or development initiative to be analyzed. The analysis of the specific quality types as specified by the framework were also omitted, as these are dependent on the tool used to render the language. The evaluation of such tools falls outside the scope of my research paper.

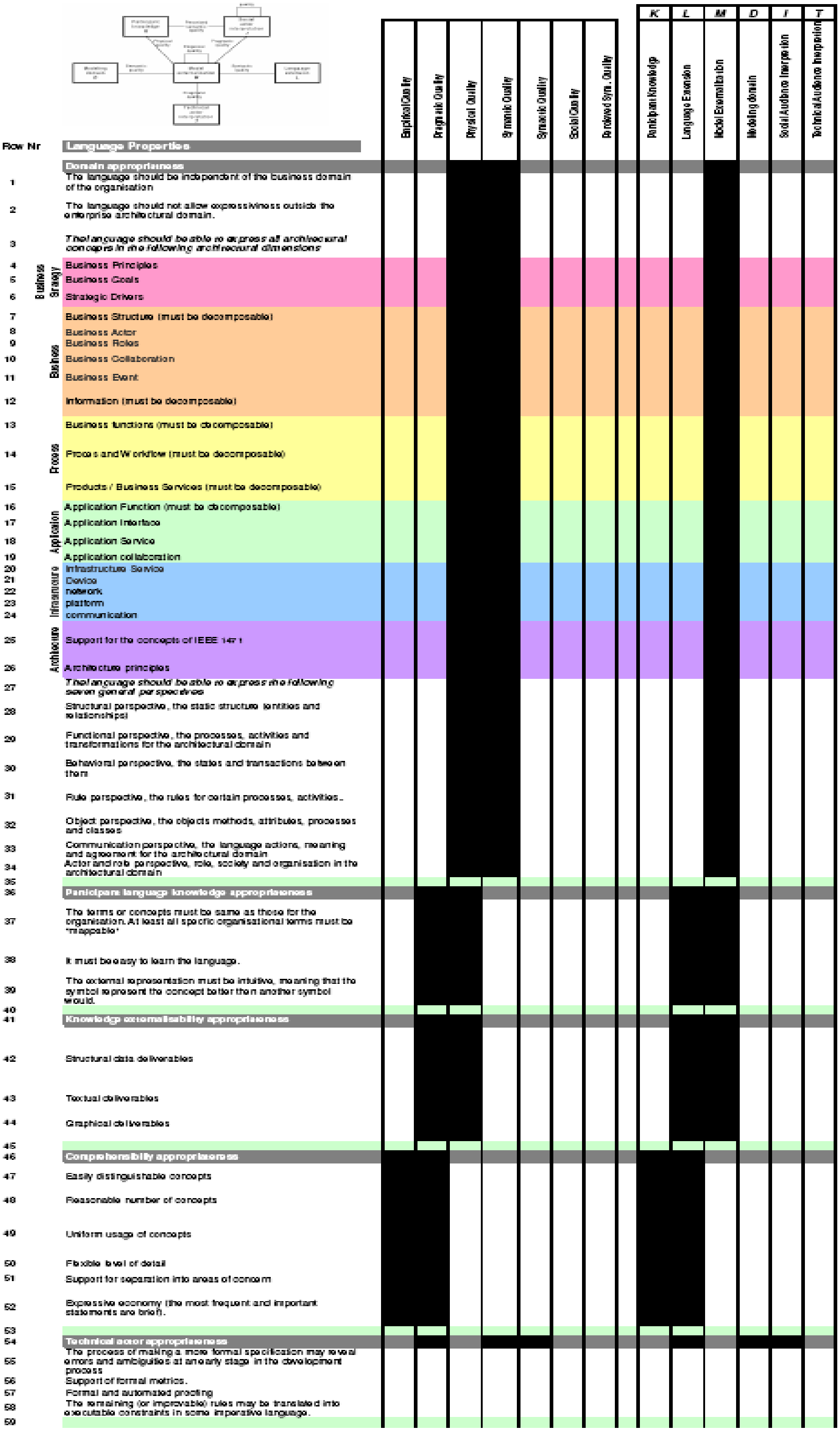
Krogstie (2000) also notes that in the evaluation of any modeling language, the following should be remembered:

- It is possible to make good models with a poor modeling language,
- It is possible to make poor models with a comparatively good modeling language,

You will always find some deficiencies in any language (often due to the trade-offs involved in language creation), but it may be useful to know the weak spots in order to avoid possible problems.

5.6 The results evaluation criteria

The evaluation criteria is a 'short' aspect list based upon the KSL-QF framework. We decided to quantify on a scale from 0-10 the positive contribute of the language towards the aspect. The next model represents this model that was created within Microsoft Excel. The colors represent the specific aspects of the architectural domain. Many other aspects have a more generic character and can be applied on languages in generic.



Row Nr		Language Properties	Empirical Quality	Pragmatic Quality	Physical Quality	Semantic Quality	Syntactic Quality	Social Quality	Perceived Sem. Quality	K Participant Knowledge	L Language Extension	M Model Externalisation	D Modeling domain	I Social Audience Interpretation	T Technical Audience Interpretation
		Domain appropriateness			■	■						■			
1		The language should be independent of the business domain of the organisation			■	■						■			
2		The language should not allow expressiveness outside the enterprise architectural domain.			■	■						■			
3		*The language should be able to express all architectural concepts in the following architectural dimensions*			■	■						■			
4	Business Strategy	Business Principles			■	■						■			
5		Business Goals			■	■						■			
6		Strategic Drivers			■	■						■			
7	Business	Business Structure (must be decomposable)			■	■						■			
8		Business Actor			■	■						■			
9		Business Roles			■	■						■			
10		Business Collaboration			■	■						■			
11		Business Event			■	■						■			
12		Information (must be decomposable)			■	■						■			
13	Process	Business functions (must be decomposable)			■	■						■			
14		Proces and Workflow (must be decomposable)			■	■						■			
15		Products / Business Services (must be decomposable)			■	■						■			
16	Application	Application Function (must be decomposable)			■	■						■			
17		Application Interface			■	■						■			
18		Application Service			■	■						■			
19		Application collaboration			■	■						■			
20	Infrastructure	Infrastructure Service			■	■						■			
21		Device			■	■						■			
22		network			■	■						■			
23		platform			■	■						■			
24		communication			■	■						■			
25	Architecture	Support for the concepts of IEEE 1471			■	■						■			
26		Architecture principles			■	■						■			
27		*The language should be able to express the following seven general perspectives*			■	■						■			
28		Structural perspective, the static structure (entities and relationships)			■	■						■			
29		Functional perspective, the processes, activities and transformations for the architectural domain			■	■						■			
30		Behavioral perspective, the states and transactions between them			■	■						■			
31		Rule perspective, the rules for certain processes, activities..			■	■						■			
32		Object perspective, the objects methods, attributes, processes and classes			■	■						■			
33		Communication perspective, the language actions, meaning and agreement for the architectural domain			■	■						■			
34		Actor and role perspective, role, society and organisation in the architectural domain			■	■						■			
35															
36		**Participant language knowledge appropriateness**		■	■						■	■			
37		The terms or concepts must be same as those for the organisation. At least all specific organisational terms must be "mappable"		■	■						■	■			
38		It must be easy to learn the language.		■	■						■	■			
39		The external representation must be intuitive, meaning that the symbol represent the concept better then another symbol would.		■	■						■	■			
40															
41		**Knowledge externalisability appropriateness**		■	■						■	■			
42		Structural data deliverables		■	■						■	■			
43		Textual deliverables		■	■						■	■			
44		Graphical deliverables		■	■						■	■			
45															
46		**Comprehensibility appropriateness**	■	■						■	■				
47		Easily distinguishable concepts	■	■						■	■				
48		Reasonable number of concepts	■	■						■	■				
49		Uniform usage of concepts	■	■						■	■				
50		Flexible level of detail	■	■						■	■				
51		Support for separation into areas of concern	■	■						■	■				
52		Expressive economy (the most frequent and important statements are brief).	■	■						■	■				
53															
54		**Technical actor appropriateness**		■		■	■				■		■	■	
55		The process of making a more formal specification may reveal errors and ambiguities at an early stage in the development process													
56		Support of formal metrics.													
57		Formal and automated proofing													
58		The remaining (or improvable) rules may be translated into executable constraints in some imperative language.													
59															

T

The model shows the relationship between all architectural aspects and the semiotic quality. By quantifying all the aspects we can express the quality with a number. These numbers do not have an absolute value but they can be used as relative values with each other.

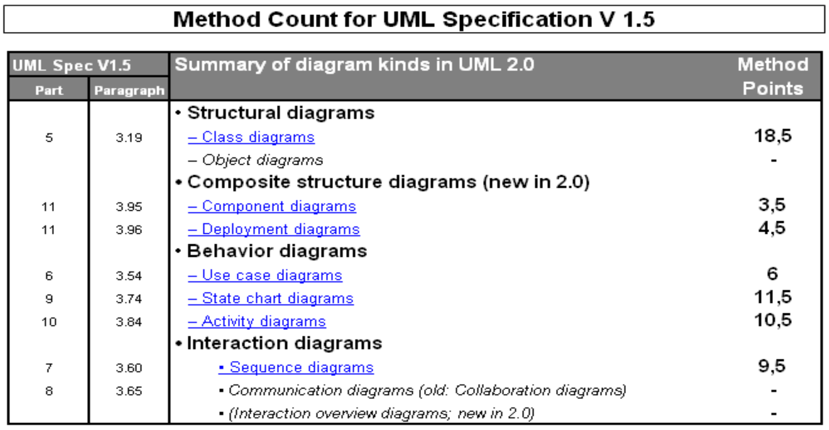
6. ADL Evaluation Results

This chapter reports the insights and design of objective evaluation criteria where against both ADL's will be evaluated. The chapter reports the evaluation based upon the method count methodology as indication for the ADL's complexity and the evaluation results based upon the semiotic framework as indication for the semiotic value of an ADL.

6.1 Complexity evaluation

The method count is conducted in Microsoft excel. The count is based upon the ADL specifications as stated earlier in this document. The complete result is included in Appendix 2

6.1.1 Results of method point analyze for ArchiMate

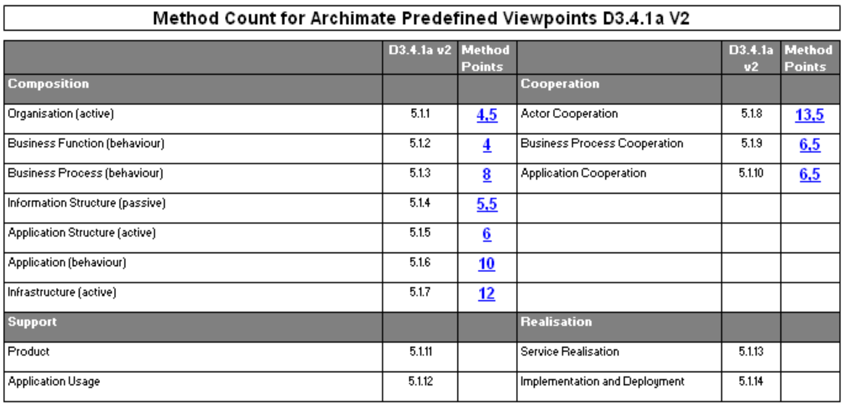


Method Count for UML Specification V 1.5

UML Spec V1.5		Summary of diagram kinds in UML 2.0	Method Points
Part	Paragraph		
		• Structural diagrams	
5	3.19	– Class diagrams	18,5
		– *Object diagrams*	-
		• Composite structure diagrams (new in 2.0)	
11	3.95	– Component diagrams	3,5
11	3.96	– Deployment diagrams	4,5
		• Behavior diagrams	
6	3.54	– Use case diagrams	6
9	3.74	– State chart diagrams	11,5
10	3.84	– Activity diagrams	10,5
		• Interaction diagrams	
7	3.60	▪ Sequence diagrams	9,5
8	3.65	▪ *Communication diagrams (old: Collaboration diagrams)*	-
		▪ *(Interaction overview diagrams; new in 2.0)*	-

Figure 6-1 Results UML Method Count

6.1.2 Results of Method point analysis for UML



Method Count for Archimate Predefined Viewpoints D3.4.1a V2

	D3.4.1a v2	Method Points		D3.4.1a v2	Method Points
Composition			**Cooperation**		
Organisation (active)	5.1.1	4,5	Actor Cooperation	5.1.8	13,5
Business Function (behaviour)	5.1.2	4	Business Process Cooperation	5.1.9	6,5
Business Process (behaviour)	5.1.3	8	Application Cooperation	5.1.10	6,5
Information Structure (passive)	5.1.4	5,5			
Application Structure (active)	5.1.5	6			
Application (behaviour)	5.1.6	10			
Infrastructure (active)	5.1.7	12			
Support			**Realisation**		
Product	5.1.11		Service Realisation	5.1.13	
Application Usage	5.1.12		Implementation and Deployment	5.1.14	

Figure 6-2 Results ArchiMate Method Count

6.1.3 ArchiMate and UML results compared

Composition				Cooperation			
ArchiMate		UML V1.5		ArchiMate		UML V1.5	
Organization Structure (active)	4,5	18,5	Class Diagram	Actor Cooperation	13,5		Collaboration Diagram
Business Function (behavior)	6	15,5	Use Case with sequence diagram	Business Process Cooperation	6,5	10,5	Activity Diagram
Business Process (behavior)	8	9,5 / 10,5	Sequence Diagram / Activity Diagram	Application Cooperation	6,5	10,5	Activity Diagram
Information Structure (passive)	5,5	18,5	Class Diagram				
Application Structure (active)	6	3,5	component diagram				
Application (behavior)	10	11,5/ 10,5	State machine / Activity Diagram				
Infrastructure (active)	12	4,5	Deployment Diagram				
Support				**Realization**			
ArchiMate		UML V1.5		ArchiMate		UML V1.5	
Product			Class Diagram	Service Realization			Class-, Component diagram
Application Usage			Activity, Sequence, Component Diagrams	Implementation and Deployment			Component, Deployment Diagram

Table 6-1 Results of Method Count Analysis

6.1.4 Conclusions of the evaluation of complexity

c:6 “Tabular” deliverable structure are equal

The ArchiMate project did not deliver an official specification how the ArchiMate model is persisted. Unofficially the project team experimented with the eclipse modeling framework (=EMF). It is however to expensive to explain the EMF but what is important is to know that the format is based on xml technology and that the EMF community delivers transformation tools like EMF2XMI which realizes automated transformation to the UML persistency standards.

c:7 **UML and ArchiMate deliver no textual deliverables**

Both modeling languages do not provide any form of textual deliverables like stated in the method point analyses. However in the UML Tooling domain some propriatry solution can be indicated. What is important to mention is that ArchiMate has the potential to deliver textual deliverables because it can be based on the rich semantic architectural concepts.

c:8 **Behavior compositions have the same complexity**

Modeling behavior is a complex exercise where many concepts and relation types are involved. The method point analysis indicates on both sides an equal amount of unique symbols, relationships and embellishments.

c:9 **ArchiMate's structural compositions less complex**

The reason for this is that in several ArchiMate deliverables statements point on the architectural need for a certain detail level. No deliverable made however a statement on which arguments this detail level was based.

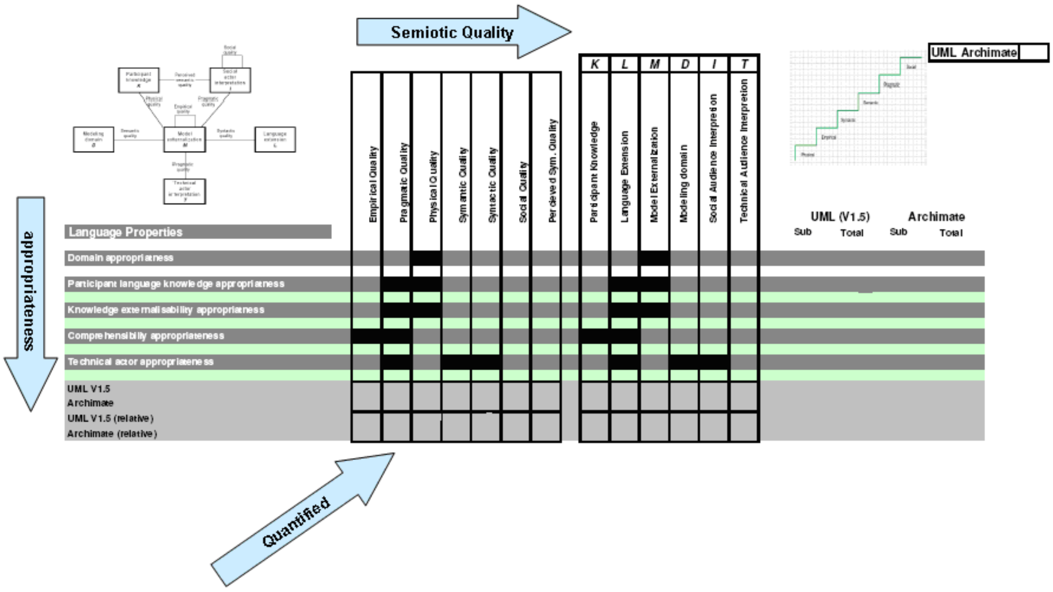
c:10 **ArchiMate's infrastructure viewpoint is more complex**

The deployment diagram has in my point of view enough information for the stakeholders to target. It is questionable if the infrastructure view must have specialized concepts where the deployment diagram works with universal embellishments. (Consequences for the ArchiMate Meta model have to be regarded)

6.2 Semiotic evaluation

The analysis done using Krogstie's quality assessment framework revealed the following under the quality groups of:

- Domain appropriateness
- Participant language knowledge appropriateness
- Knowledge externalisability appropriateness
- Comprehensibility appropriateness
- Technical actor appropriateness



The analyze results based upon the designed evaluation framework is available in Appendix 3

In the next section we explain the conclusions based upon the results after applying the evaluation framework. These conclusions are grouped corresponding the three assessment directions and where conducted in the following order:

1. Results in the language appropriateness direction
2. Results in the semiotic quality direction
3. Global quantified results

Ad1. Results in the appropriateness directions where retrieved by applying the practical assessment questions as stated in 5.5.1 - 5.5.5.

Ad2. Conclusions in the semiotic quality direction are the results of the relationships between the quality types and the KSL-Set's which are related to the semiotic theory.

Ad3. We quantified the language support for every question. This quantification was applied in which degree the language supports the architectural requirement. The support rating legend can be found on the right. Our research strategy is to group all results in the semiotic direction. When all support ratings are grouped for every semiotic level we can position these ratings on the semiotic ladder. The results have no absolute value but are used for comparative semiotic evaluation

Support Rating:

0-3 There is no, or very limeted support
4-6 The aspect is partly supported
7-9 There is satisfactory support
10 The aspect is very wel supported

Although the KSL Framework was never used to quantify the support of languages in numbers we think these numbers have certain value for evaluation. But we must regard some points of attention when we use this method:

We saw a major influence on the rating when the number of aspects increases. We identified this problem as the aspects increase in the domain appropriateness section. Not only the absolute value increases but also the relative positions of a quality group against the other increases (i.e. Domain appropriateness vs. Technical Actor appropriateness).

6.2.1 Language appropriateness

Before we draw conclusions we present a quantified impression of the evaluation in this direction:

	UML	ArchiMate
Domain appropriateness	160	221
Participant language knowledge appropriateness	16	25
Knowledge externalisability appropriateness	18	16
Comprehensibility appropriateness	31	49
Technical actor appropriateness	23	7

Table 6-2 Quantification in the language appropriateness direction

c:11 **Conclusions regarding the domain appropriateness**

- The quantification of domain appropriateness shows a significant better fit of ArchiMate with the architectural domain. ArchiMate achieved this score based on the better fit of the discrete concepts to the architectural domain.
- The core UML specification has only weak support for the architectural domain. Enterprise architecture modeling is possible if the “stereotype” extension point is used. But this point UML violates serious KSL-QF which states that it is not allowed to express statements that do not belong to the specific domain. In other words with UML you can express statements in the stereotype field that is outside the architectural domain.
- The structural perspective is well supported. Traditional abstraction mechanisms (aggregation, classification and generalization) are provided. This could be valuable in the architectural domain regarding relationships between the architectural domains. (think of business-, information-, process- and technical architecture) A positive side effect is the high semantic value of all e these mechanisms.
- The UML supports the behavioral perspective. Although it is supported UML it is not ideal for behavioral modeling in the domain of enterprise architecture.
- The functional process perspective is supported in UML through the combined use of Use Case Modeling and Activity Diagrams. By default UML does not support this mixture of diagrams. Beside that aspect Hommes and Reijswoud [24] argue also that modeling concepts in the business process domain are not easily mapped to the UML.

- The UML does not provide an intuitive means of representing organizational and group structures. All entities must use a class. (see Table 4-1)
- Temporal constraints, business goals and other non-functional requirements are not easily expressed in both languages.

c:12 **Conclusions regarding the Comprehensive Appropriateness**

Many of the following issues are relatively unproblematic, they make the language more difficult to learn and comprehend.

- ArchiMate has for the architectural domain easy distinguishable concepts. To distinguish the concepts in UML is mainly based on the usage of stereotypes. (Manny ArchiMate concepts can only be mapped on UML Classes)
- UML knows diagram types for areas of concerns. ArchiMate supports IEEE 1471
- The use of the UML to define its own meta-model has resulted in circularities and inconsistencies in definitions, partly as a result of inheritance of sometimes meaningless or ill-defined properties.
- UML is unnecessarily complex for the architectural domain, with a total of 233 discrete concepts that castellani addresses in his research, causing some redundancy and overlap.
- ArchiMate contain less symbol differentiation problems all discrete concepts know their own symbol. UML has more problems like :
  - Rectangles for classes and objects
  - Ellipse shapes for Use Cases, State charts and Activities in Activity Diagrams.
- In UML Classes have different shapes and sizes depending on the relative number of attributes and operations that are defined, making these potentially visually complex.

c:13 **Technical Actor Appropriateness**

UML knows many concepts for consistency proofing, identification of metrics and indicators. ArchiMate has two disadvantages concerning these aspects:

- ArchiMate does not known a formal persistency syntax like XMI
- ArchiMate is too young to fulfill, so less research is available.

However architectural metrics and automated proofing is a niche but important scientific domain. For more information we recommend the work of Torre [32].

c:14 **The business strategy domain is not available**

It is remarkable that business strategy with known phenomena as “Business Goal”, “Business Principle” and “Driver” are out of modeling scope of an architectural definition language. The existence of a strategic domain is well known in many research papers and relationships with the other architectural domains are known. [Domain Appropriateness, row 4, 5, 6]

c:15 **Architectural principles must be defined as a concept**

Both languages did come not further then use a “note” or other textual concepts for stating architectural principles. It is however important to have a more embedded concept for this important architectural phenomena. [Domain Appropriateness, row 26]

c:16 ArchiMate has better architectural concepts

In UML we can address many architectural statements but we must use the extension points of UML with that causes a real architectural problem. (see: 1.5) [Domain Appropriateness, row 7-24]

c:17 The detail level of ArchiMate's concepts are unknown

This conclusion was made out of uncertainty because it generates many questions like: [Domain Appropriateness, row 7-24]

- What can we address in a diagram and what not?
- Have the tool vendors the freedom to define their own properties for a concept.
- This makes a standard for model persistency impossible with the cause that the development of architectural metrics and automated analysis will be blocked.

c:18 ArchiMate has better intuitive concepts

This is the result of using unique symbols for every discrete concept. UML is to abstract and knows less variety in expressiveness. This makes a high contribution to the semantic value of ArchiMate. (see quantified results in the semiotic ladder) [Participants knowledge appropriateness, row 39]

c:19 ArchiMate has no specification for model persistency

This is a major problem. UML has the XMI specification, an xml based model persistency. This made it possible to develop metrics, automated proofing, model exchange between tool vendors, transformation capabilities etc… The lack of such persistency format blocks the entire evolution of ArchiMate's language. Using XMI is in my opinion no solution because unique architectural constructs will get lost. [Knowledge externalisability appropriateness, row 42] [Technical actor appropriateness 61-66]

c:20 Architectural comprehensibility has own characteristics

We see this when we compare UML and ArchiMate. The appropriateness of UML in the object oriented domain is very good. Think of reasonable amounts of concepts, flexible in level of detail and separation of concerns is supported with fix diagrams because the stakeholders are known. This excellent comprehensibility in the object oriented domain scores badly in the architectural domain.

The small amount of concepts is in the architectural domain a problem while must use easily distinguishable concepts and uniquely expressiveness (unique icon's). We don't know the stakeholders so we don't need fix diagram types. (IEEE 1471)

[Comprehensibility appropriateness, row 46-52]

6.2.2 Semiotic quality

The semiotic quality is the result of the relationship with the assessment in the appropriateness direction. The relationship between the appropriateness direction and the semiotic direction is stated by the KSL-QF and can be found in section assessment 4.3.1 – 4.3.5. For readability we show the relationships again in Table 6-3 All conclusions for the semiotic quality are based on the values of the appropriateness direction and their relationship with the semiotic quality.

Language Appropriateness	Semiotic Quality
Domain appropriateness	Physical and Semantic Quality
Participant language knowledge appropriateness	Pragmatic and Physical Quality
Knowledge externalisability appropriateness	Pragmatic and Physical Quality
Comprehensibility appropriateness	Empirical and Pragmatic Quality
Technical actor appropriateness	Pragmatic, Semantic and Syntactic Quality

Table 6-3 Relationships between appropriateness and semiotic quality

	Empirical Quality	Pragmatic Quality	Physical Quality	Semantic Quality	Syntactic Quality	Social Quality	Perceived Sym. Quality
UML	31	65	194	183	23	X	X
ArchiMate	49	97	262	228	7	X	X
Uml (relative)	0,388	0,401	0,425	0,445	0,767	X	X
ArchiMate (relative)	0,613	0,599	0,575	0,555	0,233	X	X

Table 6-4 Quantification in the semiotic quality direction

c:21 ArchiMate show up a high semiotic quality

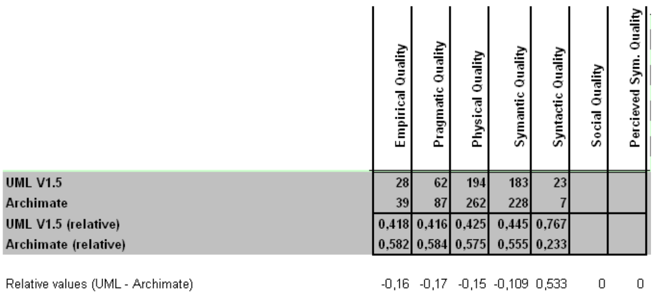
Empirical quality stands for readability and aesthetics. ArchiMate's models are easier to read and understand. The numbers UML 31 and ArchiMate 49 state the same. The pragmatic results cannot be stated without an empirical research. This quality measure is dependent on the domain experts involved in the design of the architectural model so we cannot verify this with this analysis. (Green area)

c:22 UML scores significant higher on syntactic quality

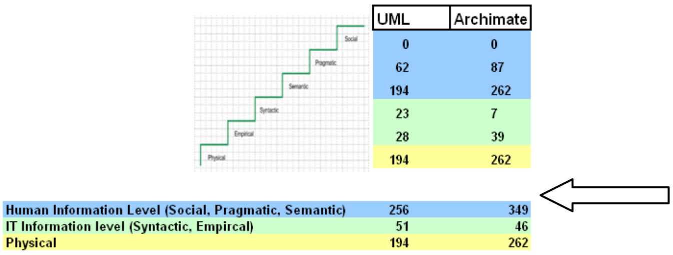
The result of ArchiMate's lack of an official XML Based persistency format.

6.2.3 Semiotic ladder

The conclusions where made after positioning the global results in the semiotic ladder. The semiotic ladder has two information interpretation levels "Human" and "IT". What we identified is that ArchiMate has less information value for IT interpretation then UML. But ArchiMate scored has significant more information value for human interpretation.



	Empirical Quality	Pragmatic Quality	Physical Quality	Symantic Quality	Syntactic Quality	Social Quality	Percieved Sym. Quality
UML V1.5	28	62	194	183	23		
Archimate	39	87	262	228	7		
UML V1.5 (relative)	0,418	0,416	0,425	0,445	0,767		
Archimate (relative)	0,582	0,584	0,575	0,555	0,233		
Relative values (UML - Archimate)	-0,16	-0,17	-0,15	-0,109	0,533	0	0



UML	Archimate
0	0
62	87
194	262
23	7
28	39
194	262

Human Information Level (Social, Pragmatic, Semantic)	256	349
IT Information level (Syntactic, Empircal)	51	46
Physical	194	262

c:23 ArchiMate has a higher human interpretable value

ArchiMate has significant higher syntactic- and pragmatic quality. As result of these qualities ArchiMate is more human interpretable then UML. We may not conclude that this makes ArchiMate a better architectural definition language but human interpretation of architectural views is an important success factor in the architectural domain. This success factor is defined in the IEEE 1471 standard where communication with stakeholder (Human) is the essential part of this standard.

c:24 UML has a higher IT interpretable value

As result of higher syntactic- and empirical quality, UML has a higher interpretable value for the IT. This is an important issue if we want to apply algorithms or when metrics play an essential part in the domain. If ArchiMate does want to evolutes in a standard they should increase these quality aspects. In this way scientific research can be applied which results in a better interpretation value for humans when metrics and algorithms generate calculated information for specific stakeholders.

7. Quality Evaluation Conclusions

This chapter describes the results and conclusions that this deduction approach has delivered. The theory from which the deduction approach is conducted is the method point framework and the 'conditioned' KS-quality framework.

7.1 Conclusions

Revisiting chapter 2.4, the first question in the quality research pathway to be answered concerned the common baseline on which both ADL's could be compared. This question was answered conclusively in chapter 4, systematically introducing:

- the terms IEEE-1471 and its concepts (4.1.1)
- Generic system elements and generic viewpoints on systems (4.1.3 - 4.1.4).

These views / perspectives created a higher generic architectural modeling ontology whereon both modeling techniques could be compared to investigate where both standards meet each other.

From both perspectives (IEEE-1471 & generic system viewpoints) we could identify that both languages are based upon generic system concepts and are both based upon the generic viewpoints on systems based on veryards theory. The result of the similarities in viewpoints where addressed in Table 4-1 Research baseline "Viewpoints and their diagram equivalent" The conclusions [c:3, c:4] address the similarities of both techniques for the architectural domain. However we saw that UML has a lack in support from the architectural IEEE-1471 perspective. This lack of architectural support is addressed in the conclusions [c:1, c:2]

Chapter 5 dealt with the second research question 2 that was focused on the design of evaluation criteria where against both ADL's can be compared. It was difficult to address and to define measurable quality aspects of an architectural language. The first parts of chapter 5, section 5.1, 5.2 address the relevant theories for evaluation criteria design.

The first measurement methodology of the quality aspect "Complexity" can be measured with the method point analysis as described in section 5.3. During the conduction and the desk research we identified the KSL-Quality Framework as a semiotic based quality instrument. The semiotic theories and the framework is presented in section 5.4. However this semiotic based quality framework is meant to evaluate a model and not the modeling technique/language it was suitable for our research but has to be conditioned. This conditioning is described in section 5.5. The evaluation framework is presented in section 5.6

The evaluation based upon the method point evaluation and the conditioned KSL-Quality Framework is described in chapter 6. This chapter is a collection of conclusions drawn upon the evaluation results aligned with research question 3.

7.2 Recommendations

The recommendations are based upon the measurable quality aspects:

- Complexity
- Semiotic Quality (Physical-, Pragmatic-, Empirical-, Semantic-, Syntactic, Social- & Perceived Semantic Quality) referring to (Domain-, Participant language knowledge- , knowledge externalization- , comprehensibility- and technical actor appropriateness) see Table 6-3

During the method point analysis we saw that both languages deliver the same structural data results [c:6]. Both languages use XMI as syntactical spec or is at least mapable to it. To describe behavior aspects from an architecture both languages showed up the same complexity [c:8]. However to express the structural view, ArchiMate is less complex then UML. We see the effects that ArchiMate embraces the idea that architectural modeling has a less abstraction level then is necessary from a generic system perspective that UML embraces. (Architectural domain v.s. System description). The complexity of ArchiMate for the infrastructure architecture is in my opinion high. ArchiMate has a lot of unique concepts for this domain and it's questionable if this specializations increase the insights.

The semiotic evaluation delivers insights regarding domain appropriateness [c:11], Comprehensibility appropriateness [c:12] & technical actor appropriateness [c:13]. The results where also interpreted from other perspective [c:14-c:22].

We see in all these conclusions one general aspect back. UML was designed for a certain domain. The designers' choose that UML should be used as system specification language. ArchiMate's designers choose that ArchiMate should be a architectural modeling language. We see in our conclusions these choices back.

- UML as specification language needs the abstraction to specify which can be seen in complexity and intuitively.
- Its correct that architecture can be seen as a system from a generic system perspective but this not state that an system specification language is a highly effective than a modeling language specialized for the architectural domain.

Based upon these conclusions we may identify that ArchiMate is more effective for the architectural domain. However if a customer of LogicaCMG initiated an architectural project it is important to analyze before the project start which perspective the customer has regarding the Architecture. Does he prefer a more technical system approach or a more coherent approach and what is he planning to do with the description of the architecture.

His answer should be verified with all numbered conclusions of this quality evaluation with the question "Do I need this for this customer ?". The conclusions are formulated from the IEEE 1471 perspective so if a projects needs the support that is described in the conclusions the choice for ArchiMate is defendable towards the customer.

The last important difference between both languages is the effect that UML has to be explained how it should be applied in the architectural domain. The ArchiMate standard can only be applied within the architectural domain.

Based upon the quality evaluation we recommend to prescribe ArchiMate as architectural language standard for LogicaCMG.

7.3 Reliability & Validity

All conclusions are based upon important standards, frameworks and theories from researchers over the world. I verified the application of these frameworks as described in Chapter 5 with the researcher self. John Krogstie (KSL-QF) was involved in the research as Dirk Roeleveld (Method Point Analysis).

The conditioned frameworks can be applied by other researchers. The main advantage of both frameworks is that they have a strict format wherein a research could be repeated. However the questionnaires in the framework asks to express a the correctness of a statement in a quantified number. Depended from which researcher the framework is used other quantifications could be the result. It would be interesting to use the framework as a survey under architects to identify the ranges / bandwidth in answers on each statement. It would increase the reliability of the evaluation.

Part III

ADL Business Potential

8. Case Study (Embargo)

This chapter reports under strict embargo the findings of the case study as conducted by the author within a LogicaCMG project at the University of Maastricht (UM) concerning the description and analysis of the functional- and technical architecture. The chapter starts with an introduction of the case study by describing the UM company profile, the business situation problem and solution. The second part zooms in the case study approach with the corresponding research questions.

8.1 Introduction

Company Profile - The University Maastricht is the youngest university in the Netherlands and growing rapidly. At the time of writing there are 11,500 students and 3, 000 staff. The UM has seven faculties: the Faculty of General Sciences, the Faculty of Arts and Culture, the Faculty of Economic Sciences, the Faculty of Medicine, the Faculty of Health Science, the Faculty of Psychology and the Faculty of Law. Next to that there is the University College Maastricht.

The University Maastricht (UM) is known at home and abroad for its unique education system: Problem-based learning. This type of education is a high-scorer with educational inspectorates and comparative research. A further aspect of the university's profile is its strong international orientation. This profile appeals to students: students from all over the Netherlands and an increasing number of foreign students choose the Maastricht system.

Since the establishment of the university scientific research has been characterized by the matrix organization. This was arranged largely around a limited number of socially relevant themes and further concentrated in research institutes and schools.

Business Situation - In the last years the ICT budgets where a significant part of the whole UM budget. An intervention was justified to reduce the ICT costs and are the project goals of the PRISMA project. This project started after a study of the research results of: "Boer en Kroon" (in the middle of 2002) to the costs of the services at the ICT service centers & LogicaCMG (in the middle of April 2004) to the total cost or ownership (TCO) at FdG and FdEWB.

Both studied reports will be used in the PRISMA project with the reports – "Van missie naar koers", Strategic program of the UM 2002-2005 (Bureau van de Universiteit) & "Aanscherpen van planning en control (Boer en Kroon) – to be translated to ICT aspects. These aspects are alignment to the mission and objectives of the UM and are important to justify the ICT initiatives to management.

In the last years it became clearer by management of the UM that the ICT organization had to be led by architectural thinking. This common understanding embraces the idea that the relationships between the identified architectures (business-, process-, application-, infrastructure architectures) and the business domains cannot be neglected. The island approach of today causes to many fragmented ICT initiatives and leads to an unmentionable architecture that leads to uncontrolled and unpredicted costs. An architecture that can grow in a business aligned architectural governance leads to a better ICT support towards the business and costs that can be controlled.

8.1.1 The problem & Cause

The UM functional- as technical architecture was hardly described from one architectural perspective. The available description of the relationship between applications and the business processes in MAVIM are detailed but there is a lack of other information from an architectural perspective. There has been also determined that MAVIM is used at the UM as an information canal for UM employees to inform this audience concerning processes, procedures and associated instructions.

An important cause of these problems has been the lack of a coherent overview. The availability of sub-overviews cannot be denied but they where isolated developed by island organizations/departments that did not address the relationships with other architectural domains. The UM is however not unique in situation because until recently organization had no effective language to express the architecture coherently. This changed however by the arrival of the ArchiMate standard as a coordinating architectural language.

8.1.2 Solution

An architectural description & overview could provide many stakeholders with enough information to monitor the progress of their policies. (e.g. migrating from a large amount of databases towards one grid, security audits, identification of cost places, -carriers and –causers). One representation in the architectural domain is known as Architectural Landscape Map and is well explained in the paper “Landscape Maps for Enterprise Architectures” [35]. A landscape card provides exactly the missing overview and is based upon the structural and internal related architectural information.

It is maintainable, living instrument that can be tailored to your current problem. The current problem requires specific information that can be projected on the landscape map. We call the third ‘projected’ dimension the landscape maps theme. (e.g. security aspects, owners, performance etc..) So each landscape map is unique and is indicated by their axes and its theme.

A landscape map provides the UM an instrument to analyze rapidly the impact of changes because relations and dependences are made explicit in the landscape representation. LogicaCMG offered an architectural scan based upon the ArchiMate methodology. All relevant architectural information will be stored in a structural repository. The desired landscape maps can be generated by a query on this repository where all relational information is stored. For this project LogicaCMG decided to use the ArchiMate implementation of Bizzdesign

Architect® a product of Bizzdezign. (http://www.bizzdesign.nl) All deliverables must be aligned with the PRISMA project goals. This means that landscape maps must address the problems that are reported in the PRISMA project.

8.2 Case Study Design

In this section we address the specific case study aspects of this project (referring to the UM project). The research quest 4 "How do we study the case ? " plays an central role in this section. We do not focus on the UM problems, the content or other project aspects. We do focus on the practical quality aspects of our research object ArchiMate. Our goal is to identify the business attractiveness, the practical value and the added value for our consultants.

8.2.1 Case Study Type

-- Case studies can deal with either single or multiple cases. There are two types of single case study: the intrinsic and the instrumental. The intrinsic case study is done to learn about a unique phenomenon which the study focuses on. The researcher needs to be able to define the uniqueness of this phenomenon which distinguishes it from all others; possibly based on a collection of features or the sequence of events. The instrumental case study is done to provide a general understanding of a phenomenon using a case -- Quote Yin [14]

The case study type for this thesis is an instrumental case study. The 'phenomenon' in our case is the usage of the new ArchiMate ADL in a case. It is our intention to gather general understanding about the business benefits for LogicaCMG and our customers when ArchiMate is applied in ICT services offered by LogicaCMG. With other words: "Studying the UM case to gather insights into business potential for LogicaCMG when ArchiMate is applied"

8.2.2 The studied research object

The research object of this case study is the LogicaCMG organization / consultant and the UM organization / employee. Both research objects will be confronted with the possibilities of the new ArchiMate ADL for the architectural domain. Aligned with the research question 4 it is important to observe these effects which can be identified as an signal for business potential. These observations from the "Business Potential Research Pathway" perspective deliver qualitative data regarding the goals of this research pathway. (see 2.3)

8.2.3 UM Project Plan

The structure of the conduction of the is aligned with the LogicaCMG time plan from the project plan to describe the technical and functional architecture for the UM. This time plan defines three milestones which can be identified in Figure 8-1.

- Deliverable “Ist”-results – 22 april 2005
- Deliverable “Soll” results – 22 mei 2005
- Deliverable functional decomposition – 22 mei 2005

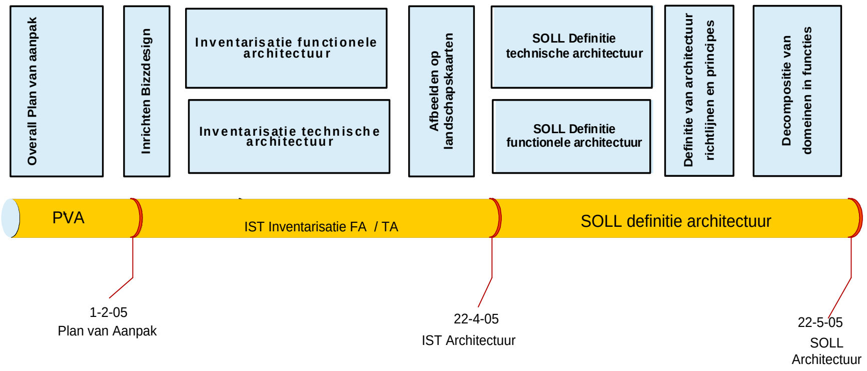

Overall Plan van aanpak
Inrichten Bizzdesign
Inventarisatie functionele architectuur
Inventarisatie technische architectuur
Afbeelden op landschapskaarten
SOLL Definitie technische architectuur
SOLL Definitie functionele architectuur
Definitie van architectuur richtlijnen en principes
Decompositie van domeinen in functies
PVA
IST Inventarisatie FA / TA
SOLL definitie architectuur
1-2-05
Plan van Aanpak
22-4-05
IST Architectuur
22-5-05
SOLL
Architectuur

Figure 8-1 UM Project Time plan (Dutch)

Between the milestones we identify three important project phases

1. Writing and defining the project plan
2. Inventory of the IST functional- and technical architecture
3. Definition of the SOLL functional- and technical architecture

Within these three project phases we observe the effects of the ArchiMate ADL by the research objects (UM organization & LogicaCMG organization).

8.3 Case Study results

8.3.1 ArchiMate in the Project plan phase

During the project plan phase ArchiMate enabled us to offer the UM a concept called Landscape Maps. These Landscape maps where based upon structural information that can be queried to generate a three dimensional charts as a representation of an architectural cross section of the architecture. The attractiveness of such charts is the interpretability of a large quantity of relational information. The UM recognized the attractiveness of the Landscape Maps as a policy tool to monitor the direction of architectural evolution as result of the UM ICT policy.

c:25 Architecture modeling tools with an syntactical ArchiMate implementation are enablers for the Architectural Landscape Map concept.

ArchiMate itself does not provide any a syntactical language specification. The tooling provides a syntactical implementation behind the graphical (semantic) modeling environment. This constructs creates the opportunity to store the model information in a structural format in a central repository. A query mechanism on this repository generates the landscape map as cross section of the database. Several vendors offer these solutions (Bizzdesign, Popkin, Mavim) **Validity -** Results with Bizzdesign Architect are evaluated by several consultants and UM employees

c:26 An architectural policy tool has business value.

ArchiMate usage as policy tool is one purpose of this ADL that has high business attractiveness. This concept was presented towards the UM as described in Dutch in Appendix 2. The concept was presented at the UM but also at several other LogicaCMG customers. They showed up high interest in the idea. With the ArchiMate implementation of Bizzdesign we can demonstrate the practicability of the concept. Customers reflected many idea's after a presentation. This indicates that they see potential in this concept.

The contract / project plan should constrain the scope of the architectural inventory. ArchiMate's Meta model (see Figure 8-2 ArchiMate Meta Model) showed up as an ideal instrument to identify exactly what should be inventoried in the architectural domain of the UM.

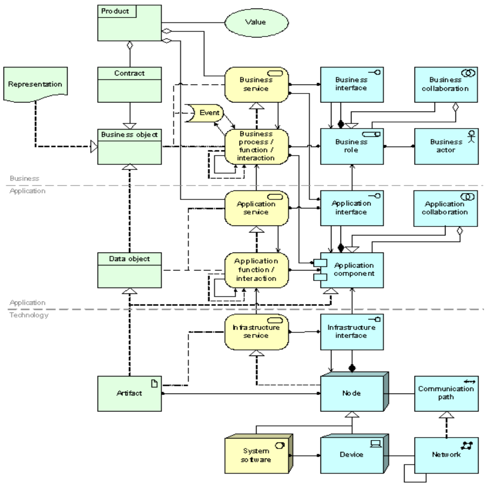


Figure 8-2 ArchiMate Meta Model

To prevent misunderstanding about the concepts used within the Meta Model the project plan could refer to the concepts of the ArchiMate specification. LogicaCMG decided to use a simplified model (see Figure 8-3) based upon this meta model as scope contract.

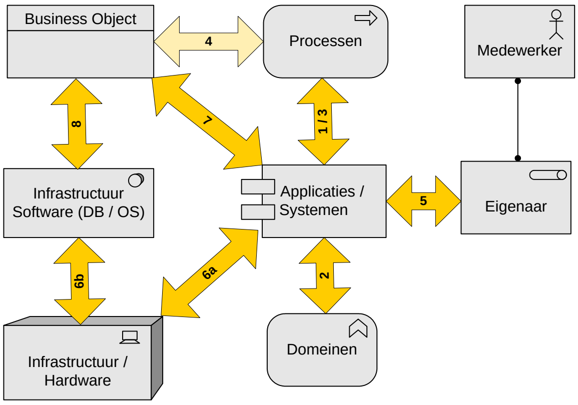
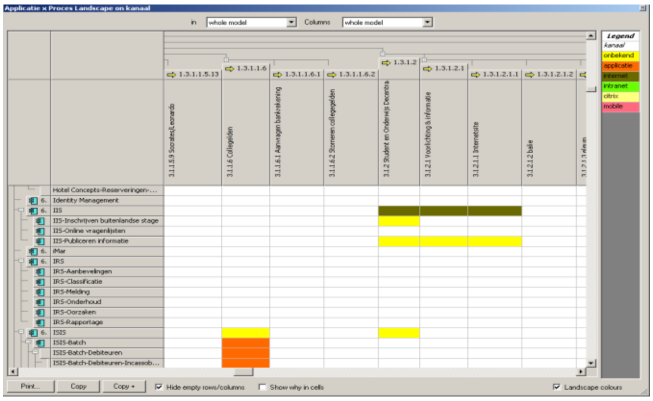


Figure 8-3 Simplified Meta Model

The numbers used in the relationships between the concepts points to the architectural landscape maps where the architectural concepts at both ends are the axes of the landscape map. (see Figure 8-4)



	3.1.1.5.9 Socrates/Leonardo	3.1.1.6 Collegegelden	3.1.1.6.1 Aanvragen bankrekening	3.1.1.6.2 Storneren collegegelden	3.1.2 Student en Onderwijs Decentra	3.1.2.1 Voorlichting & informatie	3.1.2.1.1 Internetsite	3.1.2.1.2 balie	3.1.2.1.3
Hotel Concepts-Reserveringen-...									
6. Identity Management									
6. IIS					🟫	🟫	🟫		
IIS-Inschrijven buitenlandse stage					🟨				
IIS-Online vragenlijsten									
IIS-Publiceren informatie					🟨	🟨	🟨		
6. iMar									
6. IRS									
IRS-Aanbevelingen									
IRS-Classificatie									
IRS-Melding									
IRS-Onderhoud									
IRS-Oorzaken									
IRS-Rapportage									
6. ISIS		🟨			🟨				
ISIS-Batch		🟧							
ISIS-Batch-Debiteuren		🟧							
ISIS-Batch-Debiteuren-Incassob...		🟧							

Figure 8-4 Landscape Map 1 Processes - Applications

c:27 **The ArchiMate meta model is ideal to scope the project.**

Scoping an architectural project upon the concepts in the ArchiMate Meta model has a high contribution in the project plan. Contract partners know exactly which concepts will be identified and which landscape maps can be generated based upon the relationships between the agreed architectural concepts. **Reliability** – The concept was 'invented' by the UM organization. **Validity** – Many stakeholders referred to this model during the project which is an important indicator for usability.

8.3.2 ArchiMate in the inventory phase

This section addresses the business contribution of ArchiMate in combination with the logical IEEE 1471 implementation (see 4.1.2.) as used within Bizzdesign Architect®. In the project plan the activity (Dutch Inrichten Bizzdesign) refers to the activity to configure and prepare the repository for the architectural data.

This repository and Bizzdesign implementation of IEEE-1471 is conform the logical structure as defined in 4.1.2. The structures and mechanisms that are specific for this tool is that:

- The repository is based upon XML
- The structure of the repository can be configured with an profile in a propriatry scripting format. (Like a DDL file for databases.)
- The modeling environment is not completely coupled with the syntactic model information in the repository. This means that modeling a relationship in the graphical modeling environment is not translated in the syntactic format. We have to mention that the graphical information is stored but not involved in the IEEE-1471 implementation.

The inventory is based upon the simplified meta model is presented in Figure 8-3. This model was the contract with the customer that indicates the relationship between the architectural concepts. Beside this 'contract' the table in Appendix 4 represents the content of each concept. The relationships between the concepts are expressed as *cursive* and indicate that the relationship is configured as a foreign key between concepts.

The experiences in the inventory was that that the contract between LogicaCMG and the UM was exactly defined by the Meta Model and the concept definitions as presented in Appendix 4. This was commonly accepted and never a discussion point.

c:28 **A architecture meta model in combination with property definition could serve as an architecture project contract.**

This conclusion is based upon the experiences in the project. An example is the combination between the meta model of Figure 8-3 and the property definition in appendix 4.

During the project phase where the landscape maps where generated upon the syntactic collected data the landscape maps showed up an enormous representation like (process >200 X applications >80). The expectation was that the landscape maps showed up overlapping area's that are easy to indicate. The next figure expresses expectation versus reality.

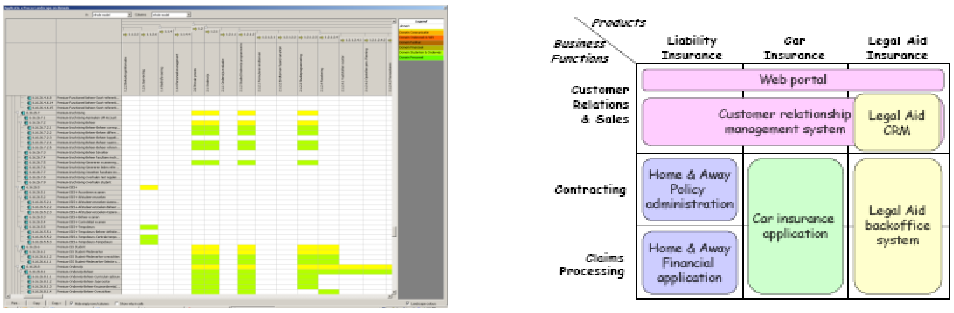

Products
Business Functions
Liability Insurance
Car Insurance
Legal Aid Insurance
Customer Relations & Sales
Web portal
Customer relationship management system
Legal Aid CRM
Contracting
Home & Away Policy administration
Car insurance application
Legal Aid backoffice system
Claims Processing
Home & Away Financial application

Figure 8-5 Landscape Maps reality v.s. Expectation

The following difficulties are identified during the generation and analysis of these landscape maps.

c:29 The Landscape maps derived from a repository delivers highly fragmented area's and is difficult to interprete

The landscape maps theme (color indicators on the third axe) is highly fragmented. Areas aren't easy to indicate. It would be helpful if an auto layout function could group colors together. The axe sequence was defined by our self but it would be interesting to study the axes after an auto layout. (which grouping on the axes can be identified ?)

c:30 Composite constructions of architectural concepts are important for generating architectural landscape maps

The propagation aspect plays an enormous role where components on the axes are composites. With other words if an application module is linked to an information object we must have the opportunity as architect to see that relationship on the also on the application level. This is also necessary for clusters on axes, with other words if applications belong to an cluster "i.e. Financial Applications" the relationships that applications have with the cluster has to lead to propagations towards the cluster. We are recommend that the ArchiMate standard should define propagation rules in the relation type specification that are in line with the relation. (association, realization etc…)

8.3.3 ArchiMate in the SOLL architecture phase

The SOLL architecture phase is still in progress. It is not manageable to describe the results before this thesis deadline.

8.3.4 The workshops

The UM case results are based upon the information from the stakeholders and involved employees of the UM. The most important aspects of constructing reliability and validity for this project was the group process within the workshops.

These workshops had the goal to create an experience for the workshop invitees (25 persons) which contribution architectural thinking delivers towards the quality for ICT and the UM decision process. As workshop facilitator we designed an reflection process based upon three university business situations as described in Figure 8-6 a full except of the workshop is available in Appendix 6

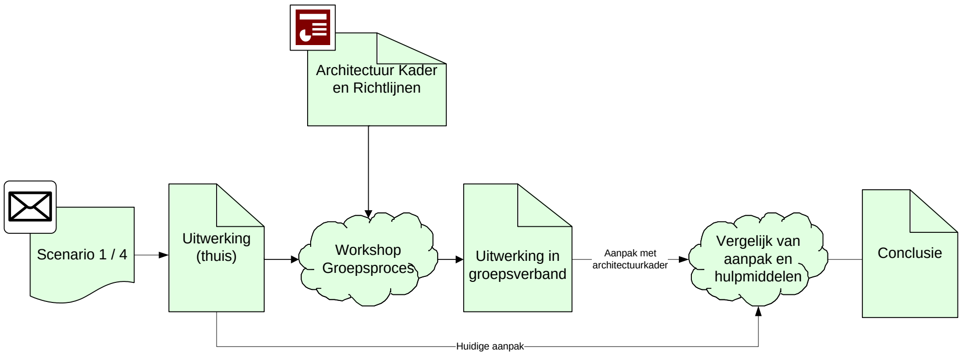

Architectuur Kader en Richtlijnen
Scenario 1 / 4
Uitwerking (thuis)
Workshop Groepsproces
Uitwerking in groepsverband
Aanpak met architectuurkader
Vergelijk van aanpak en hulpmiddelen
Conclusie
Huidige aanpak

Figure 8-6 Workshop Process (Dutch)

The three business situations where emailed as document to 25 UM employees with the homework question. "Try to express the architectural consequences as result of the changed business situation?" The direct result of this question was not important the participants were asked to monitor which approach and resources they approach to solve the homework question.

The workshop goal is to address the current architecture organization wherein no uniform architectural approach is available with the lack of one consistent architectural description. In this workshop we presented the ArchiMate standard with the simplified meta model and the concept of Landscape Maps Appendix 2. The participants where asked to look again to the business situation and to try the 'ArchiMate' approach with instructions as presented in 0

At the whiteboard we collected information from the homework and the output of the workshop activities. The results where even better than expected we could address the differences between the current situation and a situation with the "ArchiMate" approach. The next conclusions where formulated as group by the participants:

c:31 A standard provides a uniform work format

The uniform approach versus a highly fragmented and person depended format in the current situation is preferred by the group.

c:32 More consistency and coordination between several architectures (process, information, application etc..) with ArchiMate

The group saw the relationships between the process-, application-, infrastructure architecture. They prefer the use of one language that covers all architectural domains.

c:33 Structural architectural data approach delivers better analysis

The group saw the positive effects of capturing architectural data in a structural format in the repository. Better analysis are possible by querying this set of architectural data.

c:34 Group understanding of the ArchiMate Concepts

The graphical presentation of the symbols of ArchiMate (semantic value) create a group understanding (This addresses the social quality of the KSL-Framework) and preference for an overall architectural approach versus the architectural segmentation over several departments in the current situation.

c:35 Administrative overhead

The group expressed their thoughts regarding the price for these advantages. They all address the architectural administration overhead and are concerned that it will not be maintained. The overhead will be higher that the positive contribution.

9. Case study results

This chapter describes the results and conclusions that this induction approach by a instrumental case study has delivered. It is our goal to formulate the business potential as result of using ArchiMate.

9.1 Conclusions

In section 8.1 the case study was presented by explaining the organization, the problem and possible cause. As LogicaCMG we proposed a solution based upon ArchiMate and the Architecture Landscape Map. (see 8.1.2)

In the context of this research the case study is approached as an instrumental single case study with the goal to identify the business potential of ArchiMate. The approach was to observe the LogicaCMG consultant when he applies the standard (section 8.3.1 - 8.3.3) and the UM employees when the are confronted with the standard (section 8.3.4).

In the appliance of ArchiMate by our consultants we saw a major advantage of one formalized language for the architectural domain. We used the standard to scope the project but also to define exactly which parts of the architecture is involved in the project. [c:27c:28] These effects deliver an ICT service supplier as LogicaCMG an explended contract situation. The semantic value of the ArchiMate symbols helped to understand what has to be done but had to be translated in a more simplified model (Figure 8-2 v.s. Figure 8-3). The ArchiMate website (http:// ArchiMate.telin.nl) delivered us an excellent reference point. Not only to confirm the project plan on but consultants and UM employees who are less familiar with the standard could easily understand the principles of this standard. In terms of KSL-QF it helps to build up the social (Social Quality) understanding. (section 5.4.2)

However the consultants experienced the attractiveness of the concept of "The landscape map as a policy tool" by themselves and by our customers [c:26], the practical experiences where less positive. The landscape maps generated from the repository delivered us a large map with highly fragmented areas which are difficult to analyze. [c:29] One of the reasons is that the composite structure of the architectural elements was not used for propagation [c:30].

In the confrontation of UM employees with their current approach versus an approach with the ArchiMate standard in a workshop (section 8.3.4.) the employees gave an interesting response. Many conclusions refer to the Social Quality of the KSL-QF [c:31-c:34] as they say that it will bring them together regardless in which architectural domain employees work. This is in line with ArchiMate's theory to achieve one architectural language which to coordinates all sub architectures.

The group had beside these positive conclusions on major concern regarding the administrative overhead [c:35]. The positive effects of describing architecture in a structural format with an persistency in a repository was not experienced by the group. One reason for this is the structure of the current ICT organization and it's architecture governance which is highly fragmented over many faculties with no architectural governance.

9.2 Recommendations

Based upon the conclusions [c:25 - c:28] we may conclude that ArchiMate delivers us a better alternative to specify our project proposals & project plans for the architectural domain or implementation projects to address the relationships with the existent environment.

A little bit outside of this research scope is the architecture landscape map concept. ArchiMate is however an enabler for this concept, but the concept is not our central topic in this research. The concept creates however business attractiveness which can be related to the ArchiMate standard. Landscape Maps based upon UML would not be manageable. We allow the concept in our recommendation phase because it is strictly related with the ArchiMate standard and creates business potential.

The Landscape maps as concept have a large attractiveness by our customers however the implementation in tooling at this stage does not deliver enough potentional to deliver the quality that the landscape map concept promises.

We recommend the ArchiMate implementers to:

- Add propagations of relations in the structural model data following the composite relation rule that can be expressed in the landscape map.
- Add auto layout facility in the landscape maps which are able to group the colors. For this grouping it is necessary that the elements on the axes are allowed to rearrange which delivers the analyst qualitative insights which elements have formed a group.

Based upon the experiences as facilitator in the workshops the group reflected their conclusions in a reliable manor. The conclusions [c:31-c:34] indicates a better approach for this organization to tackle the architectural problems. The price is an higher administrative load as formulated in the last conclusion [c:35]

Based upon the instrumental case study we recommend to prescribe ArchiMate as architectural language for LogicaCMG to apply in project proposals and plans. It's worth to invest in this standard because the ArchiMate standard is an enabler for many architectural concepts where the Landscape Map is only one concept. This organization experienced the advantage of one architectural standard. These advantages are based upon the experience that the ArchiMate language is the 'glue' that could bind the architectural stakeholders and experts in the organization.

9.3 Validity and Reliability

We must address that the case study was a single case study which cannot be compared with other case study results. It was the intention to reach towards a large audience build up from a diverse range of stakeholders with a large spectrum in organizational ranks. This intention was realized in a workshop where 25 UM employees participated in the workshop process. The design of the workshop increased the validity and reliability of the results. This design formulated an approach to compare the current approach with the ArchiMate approach by using architectural cases. (Appendix 6) The participants where asked to solve the architectural case and to monitor how they approach the case without knowledge of the ArchiMate standard. In the workshop meeting the group was confronted with the ArchiMate approach. After this intervention the same workshop case was approached by the participants with the ArchiMate standard. All participants where asked to formulate conclusions based on the two architectural approaches.

Part IV
Conclusion

10. Research Conclusions

Both research pathway's are in it self a complete research with their own conclusions and Recommendations. Chapter 7 contains the conclusions of the quality evaluation research pathway while Chapter 9 formulates the conclusions of the business potential research. This chapter structures these conclusions aligned with the research model and plan with the goal to define an overall conclusion for the whole research project.

10.1 Structure conclusion (recap)

The first research question "**Which relevant and common concepts share both ADL's?**" was answered studying the research model as defined in the research model and open up strategy. The positioning of all studied research papers delivers us an overview where each paper contributes in our research project. (See: Appendix 1)

The answer to this question was formulated from the generic system theory perspective. UML and ArchiMate could be mapped to this generic perspective which makes it possible to compare both languages. The answer was specified in the conclusion as specified in paragraph 4.2 & Figure 4-4

The second research question: "**Which objective evaluation criteria can be applied on both ADL's ?**" Chapter 5 reports all conclusions based upon two quality aspects: Complexity- and Semiotic Quality of ADL. These ADL qualities can be measured with the evaluation instruments:

- Method point analysis, to measure the complexity of an ADL
- 'conditioned' KSL-QF, to measure the semiotic quality

This research question guide us towards two evaluation instruments. Chapter 6 presents the results of the evaluation process. The conclusions based upon these results where discussed in Chapter 7. With chapter 7 we finished the research path of evaluating both ADL's based upon the designed quality framework.

The business potential was studied in a project situation at the university of Maastricht. The research question "**How is the case study studied?"** was explained in the section 8.1 & 8.2. The research objects "UM employees & LogicaCMG consultants" where observed how and why they applied ArchiMate as ADL in the project. The business attractiveness reflected by the involved stakeholders are reflected in the conclusions [c:25 - c:35] The conclusions and recommendation based upon the case where formulated in Chapter 9

10.2 Conclusion based on results

This leaves us the central research question:

To what extent is the recently developed ADL of Archimate applicable for LogicaCMG's business and how does it improve the architects quality of work?

The quality of ArchiMate was evaluated from the complexity perspective and the semiotic quality. Complexity is a metric for the stakeholder's usage which can be related to the semiotic "pragmatic quality". To formulate a conclusion it is the most powerful to introduce a human interpretable model as presented in Figure 10-1.

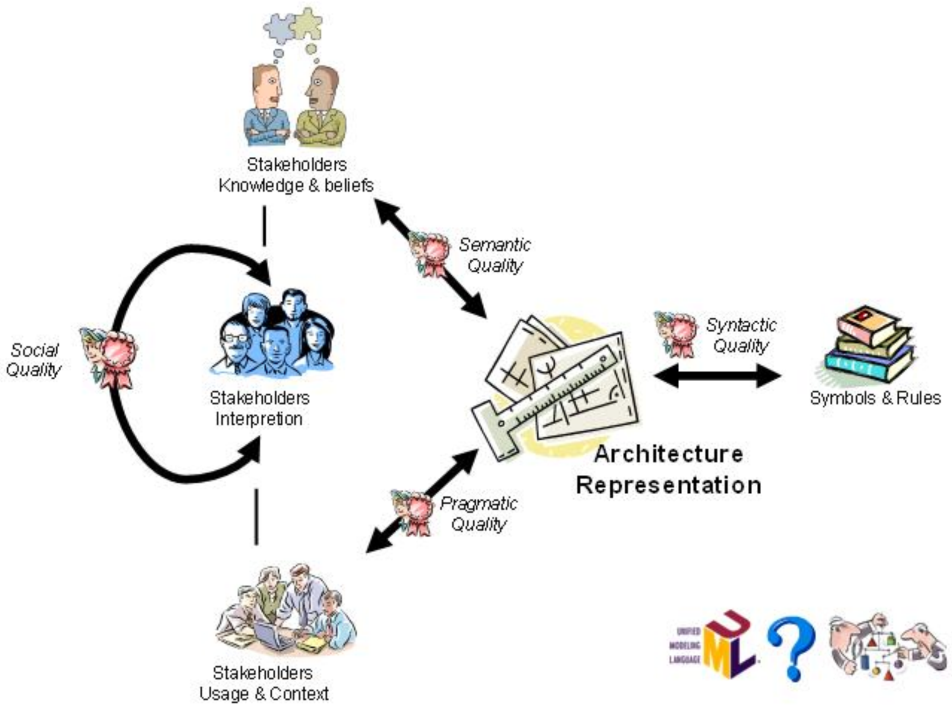


Figure 10-1 Understanding quality in an architectural descriptions

We identify in this figure the ADL as a set of books specifying the symbols and their usage when they are applied to create an architectural description. In this research we have investigated two modeling techniques ADAM's ADL UML and the ArchiMate ADL.

We see that an architectural description when its presented the stakeholders try to interpret the presentation based on his/her knowledge and beliefs. An ADL with an high Semantic Quality achieves to be accurate and complete with powerful symbols that are easy to interpret.

Pragmatic Quality indicates how easy it is for the stakeholders to maintain and to access the representation. Another pragmatic quality is the usefulness to specify the architecture. What are the benefits of an described architecture.

The Social Quality is the most powerful quality but also the less measurable quality of them all. This quality addresses the social understanding of the stakeholders group. It depends on the beliefs, knowledge, context and usage of architecture by the stakeholders

In this research project we where able to measure these qualities for two architectural definition languages UML and ArchiMate. Referring to Figure 10-1 we replaced the set of books by the UML and ArchiMate specifications. With our 'conditioned' framework we where able to measure the increase of the four semiotic qualities when ADAM's ADL is substituted by ArchiMate.

LogicaCMG increases the quality of their architectural services when ArchiMate would be applied in architectural description and representations. This conclusion is based upon the semiotic quality measurement of UML and ArchiMate.

The single case study is a fragile instrument to build conclusions on, however some reflections cannot be ignored. During presentations of ArchiMate we introduced some academically concepts (Landscape Maps, Quantitative Analysis, Viewpoints & Views) that become practical for designing-, informing- and deciding support. The attendees responses where positive when they saw a demonstration of an implementation. (Audience: UM Decision makers, ict managers, architects and project leaders)

We conducted with the same audience a workshop wherein they are confronted with their current architectural approach and the ArchiMate/Landscape Map approach. That audience formulated their own conclusions: "Based upon this experience the UM recognizes the power of an architectural approach based upon the ArchiMate language/ methodology. We experienced also an administrational overhead that costs effort in time and material. We cannot identify exactly if the benefits are greater than the costs of this overhead " We identified the high business attractiveness for this architectural approach, but the challenge is to identify the efforts that an organization has to invest.

As ICT service supplier we have a challenge to address the costs v.s. the benefits of this approach. This single case study identified the business attractiveness and identified that an organization is willing to adopt the methodology if they have identify the price. Verifying this single case study with other organizations in the ArchiMate Forum we identify the same signals. All organizations experience the attractiveness but they have all problems to identify the costs and benefits for their own organizational context.

Based upon the single case study in combination with observations in the ArchiMate forum we conclude:

ArchiMate is highly attractive for organizations and becomes relative easy the attention of an organization. An interested organizations experiences the benefits of the architectural concept and see an advantages of the formalized architectural approach.

Many of these organizations hesitate to adopt this standard because decision makers have no insights in the impacts of an implementation. The UM behavior is in line with these observations within the ArchiMate forum. Before they adopt this standard it has to be clear:

- Which benefits does it bring ?
- Which costs are involved ?
- How do I implement it ?

Based upon the conclusions and the observations define in the following section a conclusion about the business attractiveness:

At this time the ArchiMate initiative triggers organizations to rethink of their current architectural approach. From an ICT Service suppliers perspective the real business potential is the implementation of a new designed architectural approach within these organization that are based upon ArchiMate.

10.3 Value stakeholders

As addressed in section 1.4, Wouter Paul Trienekens (WPT) is identified as the business problem owner. The value of this research for WPT is that the research is very explicit in addressing the quality of our architecture language. The research formulates explicit the quality improvements when we apply ArchiMate in our products and frameworks (referring to BASIC). The business problem is becomes even more urgent that we identify the combinations of ArchiMate with architectural Frameworks from our Competitors like Sogeti of other at this moment even more important when we identify that our competitors enhance their architectural frameworks with ArchiMate. (See Figure 11-1 ‘Sogeti’)

11. Recommendations

In this thesis we have addressed a number of research questions concerning the evaluation of architectural description language. We identified the increase of quality when ArchiMate is applied in architectural descriptions. The business potential for ICT suppliers is the implementation of architecture within organizations, not the language standard itself. Knowing these results what can LogicaCMG do to gain full efforts of these conclusions ?

11.1 Business activities

As result of the first positive signals that ArchiMate is an enabler for the concept landscape maps as ICT policy monitor tool, LogicaCMG started to write a fact sheet for our customers. (see Appendix 7) The factsheet contains a profile in which the customer could recognize himself. The next activity is to write a complementary whitepaper that answers the "How" question of the proposed solution in the factsheet. We must remark that this is only one concept that is based upon ArchiMate other examples of concepts are quantitative analysis, ontology and patterns all based on ArchiMate

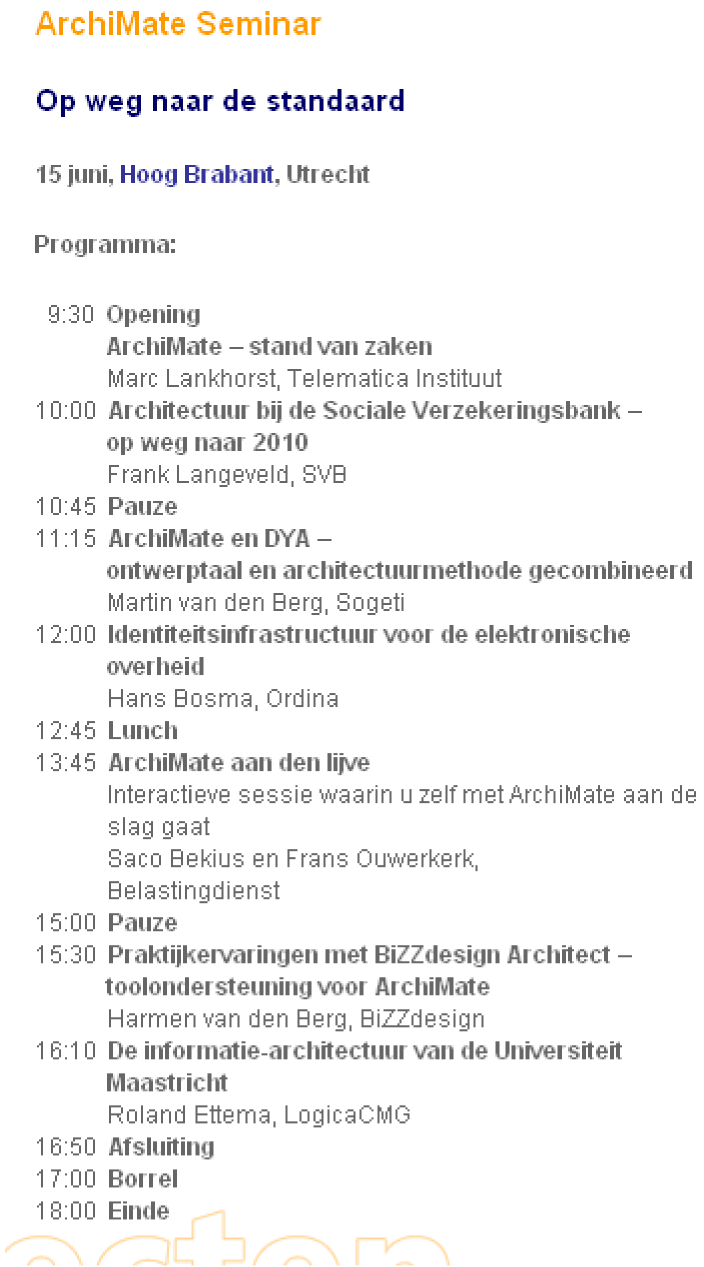
LogicaCMG as no internal architecture proposition, there are activities but they are (just as our customers) highly fragmented and not centralized. ArchiMate could stimulate this centralization because it formalizes and standardizes the architectural descriptions in our work. This first step of standardization of the language is the first step to understand each other. In a large organization as LogicaCMG it is important to start to inform architects and to address the positive effects of this standard. When the UM project is closed we have could share the expertise within this architectural community of LogicaCMG.

In the external oriented direction LogicaCMG should participate in the ArchiMate Forum. This would be a logical step for several reasons:

- Learning from architecture/ArchiMate business cases of forum members.
- External visibility towards our customers
- Joining the architectural network of architects which are aligned with this standard.

Beside the ArchiMate forum LogicaCMG should adopt architecture as theme in de tour d'ÍT. This is an seminar that is organized for our customers to inform them of IT topics as a trigger to think about.

The next step in our activities is to speak at the ArchiMate Seminar and to show up our knowledge of this standard. See the next figure for the program of the seminar 2005.



ArchiMate Seminar

Op weg naar de standaard

15 juni, Hoog Brabant, Utrecht

Programma:

9:30 **Opening**
ArchiMate – stand van zaken
Marc Lankhorst, Telematica Instituut
10:00 **Architectuur bij de Sociale Verzekeringsbank – op weg naar 2010**
Frank Langeveld, SVB
10:45 **Pauze**
11:15 **ArchiMate en DYA – ontwerptaal en architectuurmethode gecombineerd**
Martin van den Berg, Sogeti
12:00 **Identiteitsinfrastructuur voor de elektronische overheid**
Hans Bosma, Ordina
12:45 **Lunch**
13:45 **ArchiMate aan den lijve**
Interactieve sessie waarin u zelf met ArchiMate aan de slag gaat
Saco Bekius en Frans Ouwerkerk, Belastingdienst
15:00 **Pauze**
15:30 **Praktijkervaringen met BiZZdesign Architect – toolondersteuning voor ArchiMate**
Harmen van den Berg, BiZZdesign
16:10 **De informatie-architectuur van de Universiteit Maastricht**
Roland Ettema, LogicaCMG
16:50 **Afsluiting**
17:00 **Borrel**
18:00 **Einde**

Figure 11-1 LogicaCMG Speaker at ArchiMate Seminar 2005

11.2 Further research

In some respects the research that has been carried out is incomplete and shortcomings can be thought of. In order to extend the research that has been conducted and to overcome the shortcomings, I would like to conclude this thesis with the proposal of the following research agenda concerning the evaluation of architectural definition languages:

- It can be concluded that the KSL quality framework gives a reasonable account of the quality of the language but its real power is evaluation of individual models. It would be of important value for the ArchiMate Forum if a research was conducted of architectural descriptions with the KSL quality framework.

- The KSL-framework is not limited to the evaluation of ADL's. Krogstie applied the framework on the UML language for the software specification domain. Other domains with modeling languages could also be studied like the business process modeling & data modeling languages. We would gather more generic understanding of modeling languages which could result in a positive quality effect on how we model in the future.

- The conducted case study is an unique instrumental case study. More case studies should be carried out in order to gain more experience and insights of the business value of the ArchiMate standards. But also other architectural languages should be studied and be monitored if it should be adopted.

- A research of sustainability regarding the topic "Landscape Maps as a ICT policy monitoring tool" could be studied one year after its implementation at the UM. It would be a great contribution to the whole ICT community if this ICT policy tool could increase in status like the balanced scorecard for the business community

Part V

Reflection

12. Reflection

12.1 Process

The research contains two researches corresponding the two research pathways. Both processes had a specific character and difficulties.

The first research pathway was a desk study with the goal to identify the quality aspects of both ADL's. Studying the material delivered new insights and increased my understanding of the topic. It became clear that before we could compare both ADL's we had to address the similarities between both languages. I recognized by studying many rapports, specifications and thesis's that we should address the similarities on a higher meta level. Both languages could be reduced to standard system concepts and system views because both languages describe in essence a system. Reducing objects to a common meta level by questioning: "What have the studied phenomena in essence in common ?" This question is in many situations hard to answer but the answer delivers the key to redefine the phenomena in terms of commonalities. What looks at a first glance at two incomparable languages (UML – ArchiMate) looks if we redefine it in the concepts of the common meta position comparable. For research this is a powerful construct, it is an extreme universal concept and a real eye-opener to me.

In the desk research it was difficult to find objective quality aspects. The pitfall was to think that papers that described the language or other related paper delivered objective criteria. This was not the case in almost hundred percent it the paper was written from a clear choice for one perspective which decreases the objectiveness. I had to postpone my research and took a lot of time in researching the essence of languages or modeling languages. At this time I can formulate this question very explicit but at that time I experienced it as a serious problem for my research it was serendipity that lead me towards the semiotic theory wherein I found my objectiveness and essentials. This could not been foreseen.

As stated before the case study had own characteristics and difficulties. In the research plan we defined another approach to research the business potential. This changed while we had the opportunity to apply the ArchiMate language in an architectural study at the University of Maastricht. At that time it was clear that this opportunity would deliver qualitative data for our research however it is only a single instrumental case study wherein it is very difficult to generalize.

During the case study conduction I realized that the data only could be used if we apply validation techniques that would increase the reliability of our observations. In this context the workshop design went very well. The design increased the reliability and validity of our conclusions but must be interpreted as the result of a group employees working within a specific organization and its culture. For an instrumental single case study is this acceptable.

An identified risk in the project plan was the availability of people, this delayed the research tremendous.

The worst part of the research is the writing of the thesis. Starting early with the thesis report helps to spread the deadline pressure to an acceptable level. However the unbalance between work, study and family life was in this period in big unbalance.

12.2 Generalization

The research delivered generalized concepts ideas and reusable components for future research, business and appliance in projects. The next section provides a list of generalized deliverables that the research delivered.

Evaluation based upon the generic system theory – Many languages describing or specifying a system can be reduced to a model based on Veryard. This model is very generic and can be seen as the essence of describing systems and can be applied in evaluation of models in projects or evaluation of languages related to system descriptions**.** (Referring to: Figure 4-3 Metaphorical directions of)

The semiotic theory in relation with modeling languages – The power of this concept is the KSL-QF. This framework delivers the essential and objective quality aspects regarding a model. It delivers LogicaCMG a framework to evaluate models (graphical representation of something) when they are a deliverable for a client. This instrument is independent from any language and any customer. (Referring to: Figure 5-1 Semiotic Ladder R, Stamper & Figure 5-2 Quality Framework (Krogstie, Sindre & Lindland)

The workshop design – The workshop process was highly effective. This construct can be applied in almost every organization. The confrontation between 'how they work now' with 'How could they work with ArchiMate' is an universal mechanism to identify the 'positive' effects.

Bibliography

[1] Investigating the mapping of an Enterprise Description Language into UML 2.0 M.J. Wiering a, M.M. Bonsangue a, R. van Buuren b, L.P.J. Groenewegen a, H. Jonkers b and M.M. Lankhorst b

[2] Krogstie, J. ' Goal-oriented Modeling of Information Systems' in Proceedings of the Seventh International Conference on Computing and Information (ICCI'95) Peterborough, Canada July 5-8 (1995) 983-1007

[3] Krogstie, J., Conceptual Modeling for Computerized Information System Support in Organization, PhD Thesis, NTH, Trondheim, Norway (1995)

[4] Krogstie, J., Lindland, O.I. and Sindre, G., 'Defining Quality Aspects for Conceptual Models', in Proceedings of the FIP8.1 working conference on Information Systems Concepts (ISCO3); Towards a consolidation of views, Editors Falkenberg, E.D., Hesse, W. and Olive, A. Marburg, Germany, March 28-30 (1995) 216-231

[5] Carlsen, S., Krogstie, J., Sølvberg, A., & Lindland, O.I.(1997). Evaluating Flexible Workflow Systems. In J. F. Nunamaker, & R. H. Sprague (Eds.), *Proceedings of the Thirtieth Annual Hawaii International Conference on System Sciences (HICCS'97). Volume II Information Systems- Collaboration Systems and Technology, January*, (pp. 230-239).

[6] Krogstie, J. (1999). Using Quality Function Deployment in Software Requirements Specification. In A. L. Opdahl, K. Pohl, & E. Dubois. Proceedings of the Fifth International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ'99), June 14-15, (pp. 171-185), Heidelberg, Germany.

[7] Krogstie, J. and Sindre, G. 'Utilizing Deontic Operators in Information Systems Specification'. Requirements Engineering Journal, 1(4) (1996) 210-237

[8] Krogstie, J. 'Integrating the Understanding of Quality in Requirements Specification and Conceptual Modeling'. Software Engineering Notes 23(1) (1998) 86-91

[9] Krogstie, J. & Sølvberg, A. (2000) Information Systems Engineering : Conceptual Modeling in a Quality Perspective, Draft of Book, Information Systems Groups, NTNU, Trondheim, Norway.

[10] Krogstie, J., Berg, E. and Sandvold, Ø. 'Groupware Support for using Quality Function Deployment in Software Requirements Specifications in [4] 107-118

[11] Krogstie, J. (1999). Pulling together the understanding of quality in requirements specifications and modeling. In Proceedings of Norsk Informatikkonferanse (NIK'99) November 15-17 (pp. 315-326), Trondheim, Norway.

[12] Krogstie, J. and Sølvberg, A., Information Systems Engineering - Conceptual Modeling in a Quality Perspective, Draft of Book, Information Systems Groups, NTNU, Trondheim, Norway (1999)

[13] McDonald, M. P., 'Quality Function Deployment - Introducing Product Development into the Systems Development Process', in Seventh Symposium on Quality Function Deployment, Novi , Michigan, June (1995)

[14] Yin, R.K. (1994) Case Study Research: Design and Methods. Thousand Oaks: SAGE Publications, Inc.

[15]Castellani, X. 1998a. "An Overview of the Version 1.1 of the UML Defined with Charts of Concepts", <<the UML98>> Beyond the Notation – International Worksho, Mulhouse, France.

[16] Castellani, X., 1998b. "Evaluation of Models Defined with Chart of Concepts: Application to the UML Model", Proceedings of the Third Caise/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design, Pisa, Italy.

[17] Zikmund, W.G. (2000). Business Research Methods. Orlando: Harcourt Inc.

[18] Mylopoulos, J. , Chung, L. and Tu, E. 'From Object-oriented to Goal-oriented Requirements Analysis' Communications of the ACM. Vol. 42, No.1, January (1999) 31-37

[19] Sindre, G. and Krogstie, J., ‘Process Heuristics to Achieve Requirements Specification of Feasible Quality’, in Second International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ’95), Editors Pohl, K. and Peters. P. , Jyvälskylä, Finland, (1995) 92-103

[20] Achieving Quality in Natural Language Requirements, F.Fabbrini, M.Fusani,V.Gervais, S.Gnesi, S.Ruggieri

[21] Towards a metric for method complexity, Graham McLeod, University of Cape Town, Private Bag, Rondebosch, 7700 South Africa mcleod@iafrica.com

[22] Albrecht, A.J., 1979. Measuring Application Development Productivity, Proceedings IBM Share/Guide symposium, GUIDE International Corp., Chicago.

[23] Veryard, R. (2004), Business-Driven SOA 2 – How business governs the SOA process, CBDI Journal, June 2004.

[24] Thesis Bart-Jan Hommes and Victor van Reijswoud "Analysing the Quality of a Business Modeling Technique" http://is.twi.tudelft.nl/~hommes/pubs.html

[25] Investigating the mapping of an enterprise description language into UML 2.0, Wiering, Bonsangue, Buuren, Groenewegen, Jonkers en Lankhorst. Telematica Instituut, Leiden university

[26] A UML-driven Enterprise Architecture Case Study Dr. Frank Armour, ArmourIT, LLC. Dr. Stephen Kaisler,U.S. Senate Jim Getter, Doug Pippin,U.S. Capitol Police

[27] Liu, K.. Semiotics in information systems engineering. Cambridge, England: Cambridge University Press (2000)

[28] Stamper, R. K. Organizational semiotics: Informatics without the computer? In K. Liu, R. J. Clarke, P. Bøgh Andersen, & R. K. Stamper (Eds.), Information, organization and technology: Studies in organizational semiotics (pp. 115-171). Boston, MA: Kluwer Academic Publishers (2001)

[29] Mapping between ArchiMate and Standards ArchiMate Deliverable 2.2.3b, René van Buuren (TI), Luuk Groenewegen (LIACS), Henk Jonkers (TI), Peter Klaus (Ordina), Marc Lankhorst (TI), Martijn Wiering (LIACS).

[30] Viewpoints Functionality and Examples ArchiMate Deliverable 3.4.1a v2, Hugo ter Doest, Maria-Eugenia Iacob, Marc Lankhorst, Diederik van Leeuwen

[31] OMG Unified Modeling Language Specification March 2003 Version 1.5 formal/03-03-01

[32] Enterprise Architecture Analysis with XML, Frank de Boer, Marcello Bonsangue, Joost Jacob, Andries Stam, and Leendert van der Torre http://homepages.cwi.nl/~torre/papers.html

[33] Arguments and points of attentions of meta models http://www.metamodel.com/staticpages/index.php?page=20021010225607569

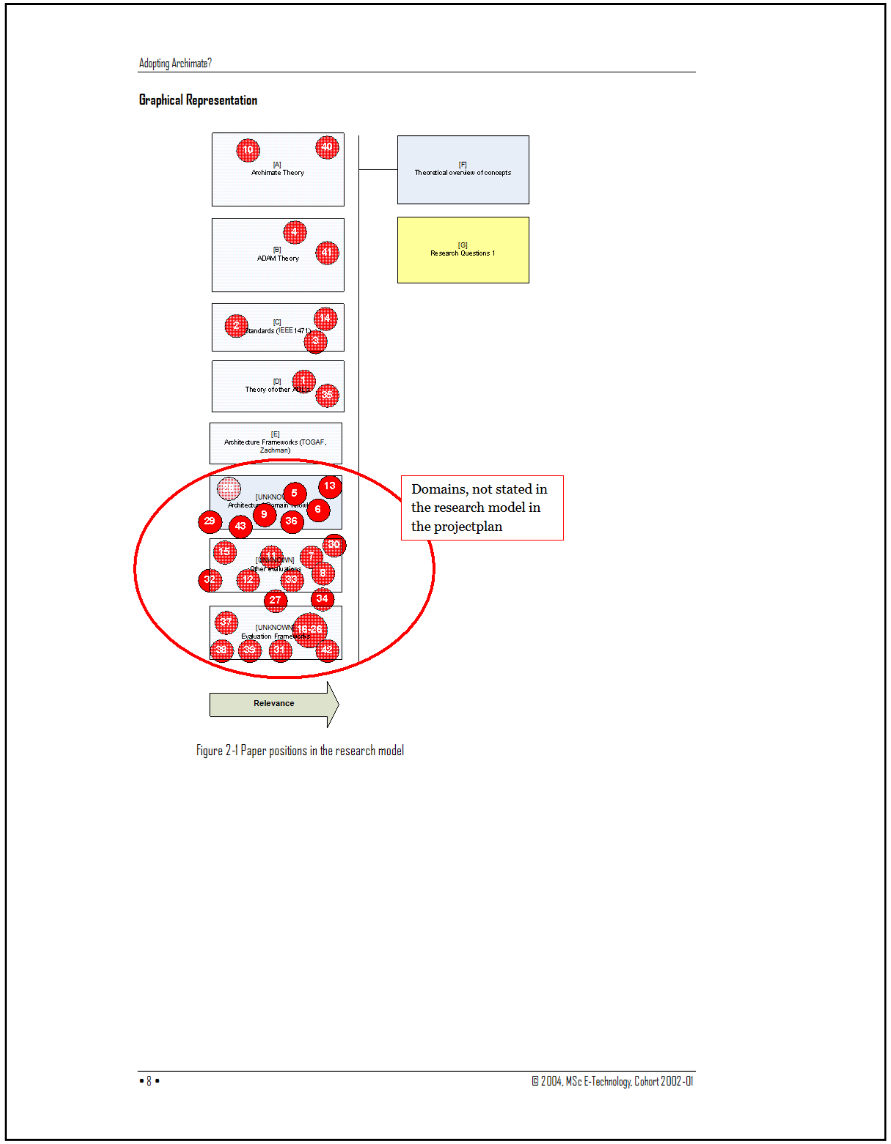
[34] Navigeren aan de hand van een landschapskaart, Dr. ir. W. Bakkeren, Drs. A. van der Krabben, Dr. R. van der Plank

[35] Landscape Maps for Enterprise Architectures, L. van der Torre1, M.M. Lankhorst2, H. ter Doest2, J. Campschroer3, F. Arbab1, 1 CWI, Amsterdam, the Netherlands, 2 Telematica Instituut, Enschede, the Netherlands, 3 Ordina, the Netherlands

[

Appendices

Appendix 1. Desk research for research question 1

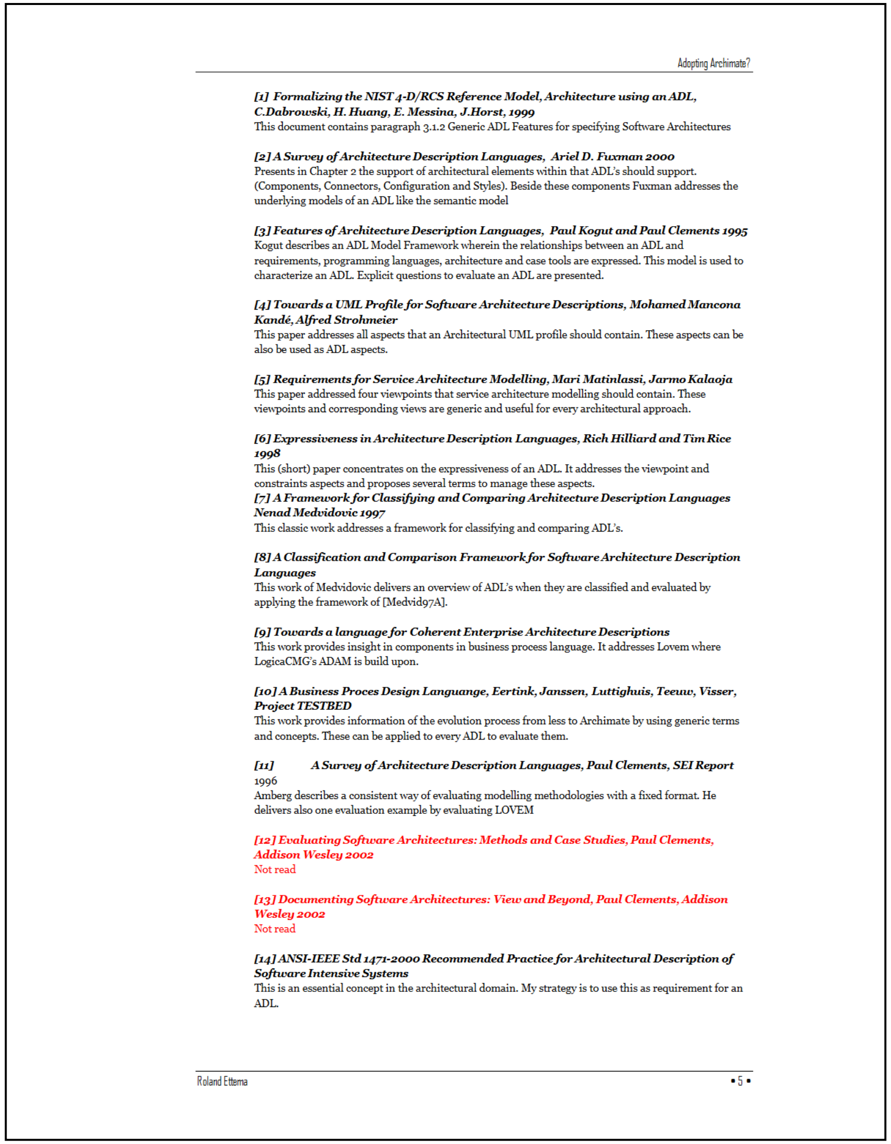


Adopting Archimate?

Graphical Representation

Figure 2-1 Paper positions in the research model

▪ 8 ▪ © 2004, MSc E-Technology, Cohort 2002-01



Adopting Archimate?

[1] Formalizing the NIST 4-D/RCS Reference Model, Architecture using an ADL, C.Dabrowski, H. Huang, E. Messina, J.Horst, 1999
This document contains paragraph 3.1.2 Generic ADL Features for specifying Software Architectures

[2] A Survey of Architecture Description Languages, Ariel D. Fuxman 2000
Presents in Chapter 2 the support of architectural elements within that ADL's should support. (Components, Connectors, Configuration and Styles). Beside these components Fuxman addresses the underlying models of an ADL like the semantic model

[3] Features of Architecture Description Languages, Paul Kogut and Paul Clements 1995
Kogut describes an ADL Model Framework wherein the relationships between an ADL and requirements, programming languages, architecture and case tools are expressed. This model is used to characterize an ADL. Explicit questions to evaluate an ADL are presented.

[4] Towards a UML Profile for Software Architecture Descriptions, Mohamed Mancona Kandé, Alfred Strohmeier
This paper addresses all aspects that an Architectural UML profile should contain. These aspects can be also be used as ADL aspects.

[5] Requirements for Service Architecture Modelling, Mari Matinlassi, Jarmo Kalaoja
This paper addressed four viewpoints that service architecture modelling should contain. These viewpoints and corresponding views are generic and useful for every architectural approach.

[6] Expressiveness in Architecture Description Languages, Rich Hilliard and Tim Rice 1998
This (short) paper concentrates on the expressiveness of an ADL. It addresses the viewpoint and constraints aspects and proposes several terms to manage these aspects.
[7] A Framework for Classifying and Comparing Architecture Description Languages Nenad Medvidovic 1997
This classic work addresses a framework for classifying and comparing ADL's.

[8] A Classification and Comparison Framework for Software Architecture Description Languages
This work of Medvidovic delivers an overview of ADL's when they are classified and evaluated by applying the framework of [Medvid97A].

[9] Towards a language for Coherent Enterprise Architecture Descriptions
This work provides insight in components in business process language. It addresses Lovem where LogicaCMG's ADAM is build upon.

[10] A Business Proces Design Languange, Eertink, Janssen, Luttighuis, Teeuw, Visser, Project TESTBED
This work provides information of the evolution process from less to Archimate by using generic terms and concepts. These can be applied to every ADL to evaluate them.

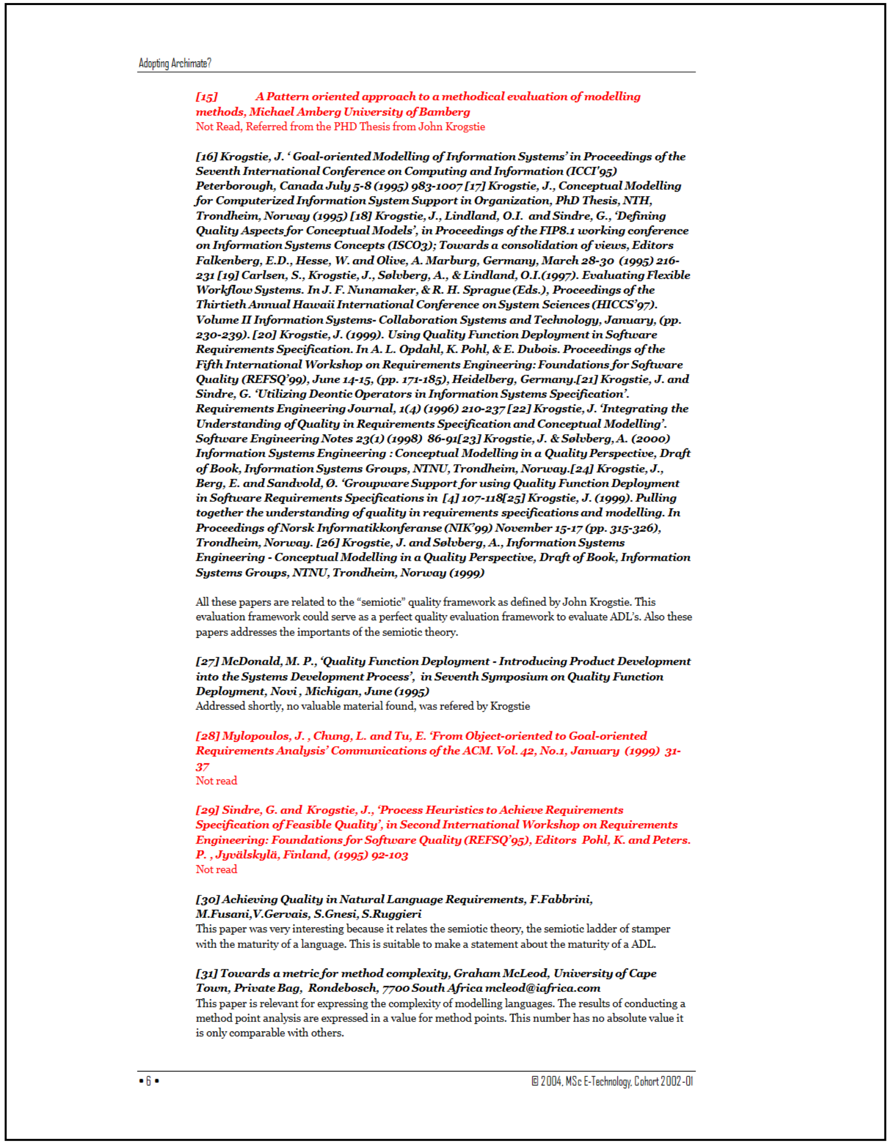
[11] A Survey of Architecture Description Languages, Paul Clements, SEI Report 1996
Amberg describes a consistent way of evaluating modelling methodologies with a fixed format. He delivers also one evaluation example by evaluating LOVEM

[12] Evaluating Software Architectures: Methods and Case Studies, Paul Clements, Addison Wesley 2002
Not read

[13] Documenting Software Architectures: View and Beyond, Paul Clements, Addison Wesley 2002
Not read

[14] ANSI-IEEE Std 1471-2000 Recommended Practice for Architectural Description of Software Intensive Systems
This is an essential concept in the architectural domain. My strategy is to use this as requirement for an ADL.

Roland Ettema ▪ 5 ▪



Adopting Archimate?

[15] A Pattern oriented approach to a methodical evaluation of modelling methods, Michael Amberg University of Bamberg
Not Read, Referred from the PHD Thesis from John Krogstie

[16] Krogstie, J. ' Goal-oriented Modelling of Information Systems' in Proceedings of the Seventh International Conference on Computing and Information (ICCI'95) Peterborough, Canada July 5-8 (1995) 983-1007 [17] Krogstie, J., Conceptual Modelling for Computerized Information System Support in Organization, PhD Thesis, NTH, Trondheim, Norway (1995) [18] Krogstie, J., Lindland, O.I. and Sindre, G., 'Defining Quality Aspects for Conceptual Models', in Proceedings of the FIP8.1 working conference on Information Systems Concepts (ISCO3); Towards a consolidation of views, Editors Falkenberg, E.D., Hesse, W. and Olive, A. Marburg, Germany, March 28-30 (1995) 216-231 [19] Carlsen, S., Krogstie, J., Sølvberg, A., & Lindland, O.I.(1997). Evaluating Flexible Workflow Systems. In J. F. Nunamaker, & R. H. Sprague (Eds.), Proceedings of the Thirtieth Annual Hawaii International Conference on System Sciences (HICCS'97). Volume II Information Systems- Collaboration Systems and Technology, January, (pp. 230-239). [20] Krogstie, J. (1999). Using Quality Function Deployment in Software Requirements Specification. In A. L. Opdahl, K. Pohl, & E. Dubois. Proceedings of the Fifth International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ'99), June 14-15, (pp. 171-185), Heidelberg, Germany.[21] Krogstie, J. and Sindre, G. 'Utilizing Deontic Operators in Information Systems Specification'. Requirements Engineering Journal, 1(4) (1996) 210-237 [22] Krogstie, J. 'Integrating the Understanding of Quality in Requirements Specification and Conceptual Modelling'. Software Engineering Notes 23(1) (1998) 86-91[23] Krogstie, J. & Sølvberg, A. (2000) Information Systems Engineering : Conceptual Modelling in a Quality Perspective, Draft of Book, Information Systems Groups, NTNU, Trondheim, Norway.[24] Krogstie, J., Berg, E. and Sandvold, Ø. 'Groupware Support for using Quality Function Deployment in Software Requirements Specifications in [4] 107-118[25] Krogstie, J. (1999). Pulling together the understanding of quality in requirements specifications and modelling. In Proceedings of Norsk Informatikkonferanse (NIK'99) November 15-17 (pp. 315-326), Trondheim, Norway. [26] Krogstie, J. and Sølvberg, A., Information Systems Engineering - Conceptual Modelling in a Quality Perspective, Draft of Book, Information Systems Groups, NTNU, Trondheim, Norway (1999)

All these papers are related to the "semiotic" quality framework as defined by John Krogstie. This evaluation framework could serve as a perfect quality evaluation framework to evaluate ADL's. Also these papers addresses the importants of the semiotic theory.

[27] McDonald, M. P., 'Quality Function Deployment - Introducing Product Development into the Systems Development Process', in Seventh Symposium on Quality Function Deployment, Novi , Michigan, June (1995)
Addressed shortly, no valuable material found, was refered by Krogstie

[28] Mylopoulos, J. , Chung, L. and Tu, E. 'From Object-oriented to Goal-oriented Requirements Analysis' Communications of the ACM. Vol. 42, No.1, January (1999) 31-37
Not read

[29] Sindre, G. and Krogstie, J., 'Process Heuristics to Achieve Requirements Specification of Feasible Quality', in Second International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ'95), Editors Pohl, K. and Peters. P. , Jyvälskylä, Finland, (1995) 92-103
Not read

[30] Achieving Quality in Natural Language Requirements, F.Fabbrini, M.Fusani,V.Gervais, S.Gnesi, S.Ruggieri
This paper was very interesting because it relates the semiotic theory, the semiotic ladder of stamper with the maturity of a language. This is suitable to make a statement about the maturity of a ADL.

[31] Towards a metric for method complexity, Graham McLeod, University of Cape Town, Private Bag, Rondebosch, 7700 South Africa mcleod@iafrica.com
This paper is relevant for expressing the complexity of modelling languages. The results of conducting a method point analysis are expressed in a value for method points. This number has no absolute value it is only comparable with others.

▪ 6 ▪ © 2004, MSc E-Technology, Cohort 2002-01

Adopting Archimate?

[32] Albrecht, A.J., 1979. Measuring Application Development Productivity, Proceedings IBM Share/Guide symposium, GUIDE International Corp., Chicago.
Not Read

[33] Veryard, R. (2004), Business-Driven SOA 2 – How business governs the SOA process, CBDI Journal, June 2004.
This paper was referred by [40]. This information is valuable for creating an evaluation baseline. Further investigation should point out that UML and Archimate support this point of view.

[34] Thesis Bart-Jan Hommes and Victor van Reijswoud "Analysing the Quality of a Business Modelling Technique" http://is.twi.tudelft.nl/~hommes/pubs.html
Some concepts are relevant to use in our research.

[35] Investigating the mapping of an enterprise description language into UML 2.0, Wiering, Bonsangue, Buuren, Groenewegen, Jonkers en Lankhorst. Telematica Instituut, Leiden university
This information is valuable for creating an evaluation baseline. If both languages can be mapped the relationships are clear.

[36] A UML-driven Enterprise Architecture Case Study Dr. Frank Armour, ArmourIT, LLC. Dr. Stephen Kaisler,U.S. Senate Jim Getter, Doug Pippin,U.S. Capitol Police
Interesting points of architectural challenges.

[37] Liu, K.. Semiotics in information systems engineering. Cambridge, England: Cambridge University Press (2000)
Background information of the semiotic theory

[38] Stamper, R. K. Organisational semiotics: Informatics without the computer? In K. Liu, R. J. Clarke, P. Bøgh Andersen, & R. K. Stamper (Eds.), Information, organisation and technology: Studies in organisational semiotics (pp. 115-171). Boston, MA: Kluwer Academic Publishers (2001)
Background information of the semiotic theory

[39] Mapping between ArchiMate and Standards ArchiMate Deliverable 2.2.3b, René van Buuren (TI), Luuk Groenewegen (LIACS), Henk Jonkers (TI), Peter Klaus (Ordina), Marc Lankhorst (TI), Martijn Wiering (LIACS).
This information is valuable for creating an evaluation baseline. If both languages can be mapped the relationships are clear.

[40] Viewpoints Functionality and Examples ArchiMate Deliverable 3.4.1a v2, Hugo ter Doest, Maria-Eugenia Iacob, Marc Lankhorst, Diederik van Leeuwen
This specification of the Archimate language could be used as

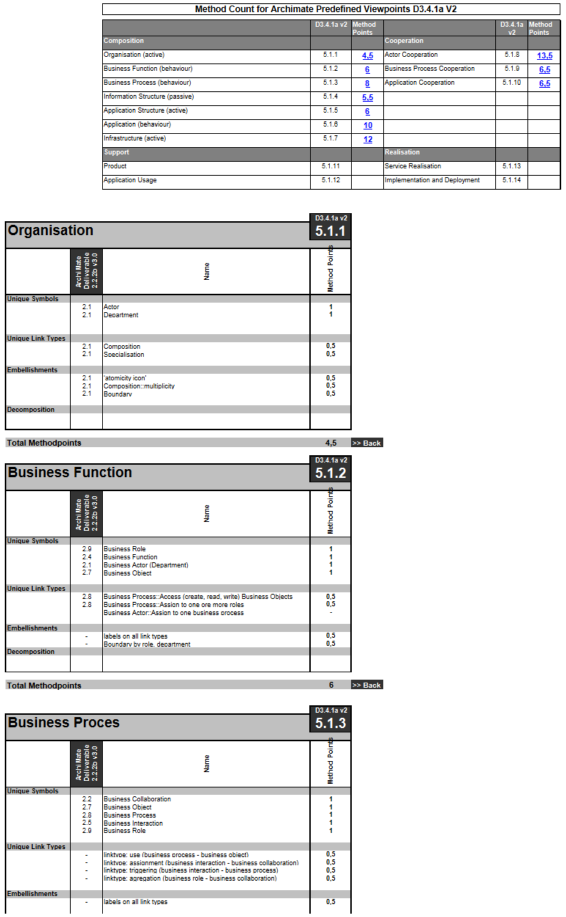
[41] OMG Unified Modelling Language Specification March 2003 Version 1.5 formal/03-03-01
Official UML specification to build our research while LogicaCMG's ADAM uses primarily UML as ADL

[42] Enterprise Architecture Analysis with XML, Frank de Boer, Marcello Bonsangue, Joost Jacob, Andries Stam, and Leendert van der Torre http://homepages.cwi.nl/~torre/papers.html
Interesting research abouth automatic proofing, transforming and metrics in ADL's when XML is used as persistency format. This could become a requirement.

[43] Arguments and points of attentions of meta models http://www.metamodel.com/staticpages/index.php?page=20021010225607569
Could serve as requirements for Meta Models in general

Roland Ettema ▪ 7 ▪

Appendix 2. Results of the method count evaluation (UML & ArchiMate)



Method Count for Archimate Predefined Viewpoints D3.4.1a V2

	D3.4.1a v2	Method Points		D3.4.1a v2	Method Points
Composition			**Cooperation**		
Organisation (active)	5.1.1	4,5	Actor Cooperation	5.1.8	13,5
Business Function (behaviour)	5.1.2	6	Business Process Cooperation	5.1.9	6,5
Business Process (behaviour)	5.1.3	8	Application Cooperation	5.1.10	6,5
Information Structure (passive)	5.1.4	5,5			
Application Structure (active)	5.1.5	6			
Application (behaviour)	5.1.6	10			
Infrastructure (active)	5.1.7	12			
Support			**Realisation**		
Product	5.1.11		Service Realisation	5.1.13	
Application Usage	5.1.12		Implementation and Deployment	5.1.14	

Organisation D3.4.1a v2 **5.1.1**

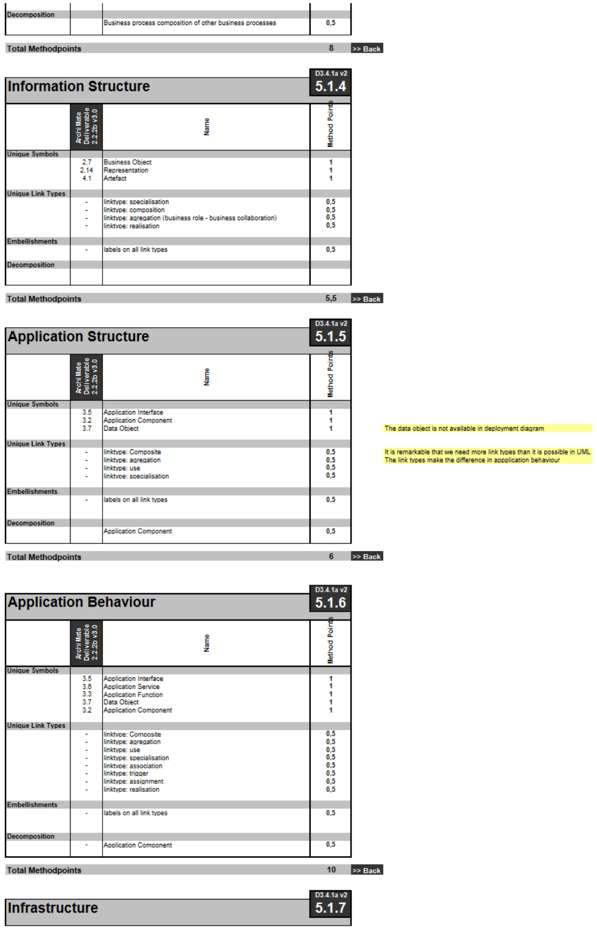
	Archi Mate Deliverable 2.2.2b v3.0	Name	Method Points
Unique Symbols			
	2.1	Actor	1
	2.1	Department	1
Unique Link Types			
	2.1	Composition	0,5
	2.1	Specialisation	0,5
Embellishments			
	2.1	'atomicity icon'	0,5
	2.1	Composition::multiplicity	0,5
	2.1	Boundary	0,5
Decomposition			
Total Methodpoints			4,5 >> Back

Business Function D3.4.1a v2 **5.1.2**

	Archi Mate Deliverable 2.2.2b v3.0	Name	Method Points
Unique Symbols			
	2.9	Business Role	1
	2.4	Business Function	1
	2.1	Business Actor (Department)	1
	2.7	Business Object	1
Unique Link Types			
	2.8	Business Process::Access (create, read, write) Business Objects	0,5
	2.8	Business Process::Assign to one ore more roles	0,5
		Business Actor::Assign to one business process	-
Embellishments			
	-	labels on all link types	0,5
	-	Boundary by role, department	0,5
Decomposition			
Total Methodpoints			6 >> Back

Business Proces D3.4.1a v2 **5.1.3**

	Archi Mate Deliverable 2.2.2b v3.0	Name	Method Points
Unique Symbols			
	2.2	Business Collaboration	1
	2.7	Business Object	1
	2.8	Business Process	1
	2.5	Business Interaction	1
	2.9	Business Role	1
Unique Link Types			
	-	linktype: use (business process - business object)	0,5
	-	linktype: assignment (business interaction - business collaboration)	0,5
	-	linktype: triggering (business interaction - business process)	0,5
	-	linktype: agregation (business role - business collaboration)	0,5
Embellishments			
	-	labels on all link types	0,5
Decomposition			
		Business process composition of other business processes	0,5
Total Methodpoints			8 >> Back



Information Structure D3.4.1a v2 **5.1.4**

	Archi Mate Deliverable 2.2.2b v3.0	Name	Method Points
Unique Symbols			
	2.7	Business Object	1
	2.14	Representation	1
	4.1	Artefact	1
Unique Link Types			
	-	linktype: specialisation	0,5
	-	linktype: composition	0,5
	-	linktype: agregation (business role - business collaboration)	0,5
	-	linktype: realisation	0,5
Embellishments			
	-	labels on all link types	0,5
Decomposition			
Total Methodpoints			5,5 >> Back

Application Structure D3.4.1a v2 **5.1.5**

	Archi Mate Deliverable 2.2.2b v3.0	Name	Method Points
Unique Symbols			
	3.5	Application Interface	1
	3.2	Application Component	1
	3.7	Data Object	1
Unique Link Types			
	-	linktype: Composite	0,5
	-	linktype: agregation	0,5
	-	linktype: use	0,5
	-	linktype: specialisation	0,5
Embellishments			
	-	labels on all link types	0,5
Decomposition			
		Application Component	0,5
Total Methodpoints			6 >> Back

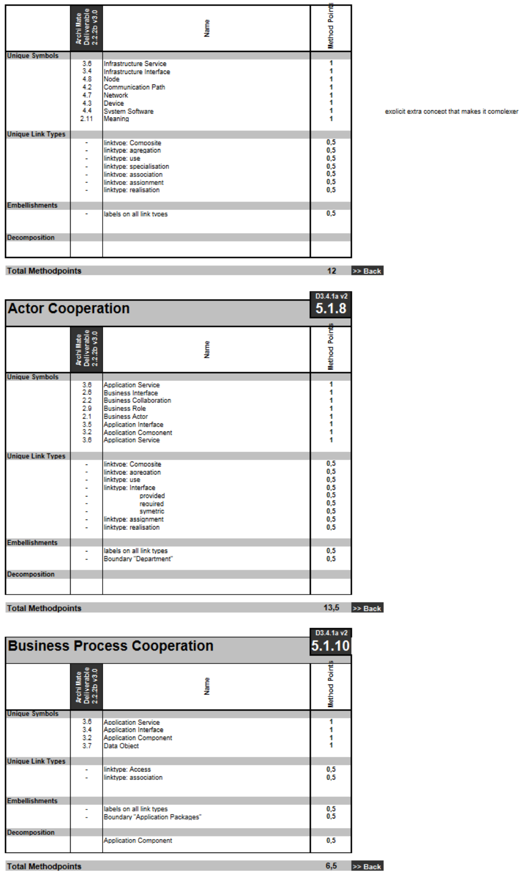
The data object is not available in deployment diagram

It is remarkable that we need more link types than it is possible in UML
The link types make the difference in appplication behaviour

Application Behaviour D3.4.1a v2 **5.1.6**

	Archi Mate Deliverable 2.2.2b v3.0	Name	Method Points
Unique Symbols			
	3.5	Application Interface	1
	3.6	Application Service	1
	3.3	Application Function	1
	3.7	Data Object	1
	3.2	Application Component	1
Unique Link Types			
	-	linktype: Composite	0,5
	-	linktype: agregation	0,5
	-	linktype: use	0,5
	-	linktype: specialisation	0,5
	-	linktype: association	0,5
	-	linktype: trigger	0,5
	-	linktype: assignment	0,5
	-	linktype: realisation	0,5
Embellishments			
	-	labels on all link types	0,5
Decomposition			
	-	Application Component	0,5
Total Methodpoints			10 >> Back

Infrastructure D3.4.1a v2 **5.1.7**



	Archi Mate Deliverable 2.2.2b v3.0	Name	Method Points
Unique Symbols			
	3.6	Infrastructure Service	1
	3.4	Infrastructure Interface	1
	4.8	Node	1
	4.2	Communication Path	1
	4.7	Network	1
	4.3	Device	1
	4.4	System Software	1
	2.11	Meaning	1
Unique Link Types			
	-	linktype: Composite	0,5
	-	linktype: agregation	0,5
	-	linktype: use	0,5
	-	linktype: specialisation	0,5
	-	linktype: association	0,5
	-	linktype: assignment	0,5
	-	linktype: realisation	0,5
Embellishments			
	-	labels on all link types	0,5
Decomposition			
Total Methodpoints			12

explicit extra concept that makes it complexer

>> Back

Actor Cooperation

D3.4.1a v2 **5.1.8**

	Archi Mate Deliverable 2.2.2b v3.0	Name	Method Points
Unique Symbols			
	3.6	Application Service	1
	2.6	Business Interface	1
	2.2	Business Collaboration	1
	2.9	Business Role	1
	2.1	Business Actor	1
	3.5	Application Interface	1
	3.2	Application Component	1
	3.6	Application Service	1
Unique Link Types			
	-	linktype: Composite	0,5
	-	linktype: agregation	0,5
	-	linktype: use	0,5
	-	linktype: Interface	0,5
	-	provided	0,5
	-	required	0,5
	-	symetric	0,5
	-	linktype: assignment	0,5
	-	linktype: realisation	0,5
Embellishments			
	-	labels on all link types	0,5
	-	Boundary "Department"	0,5
Decomposition			
Total Methodpoints			13,5

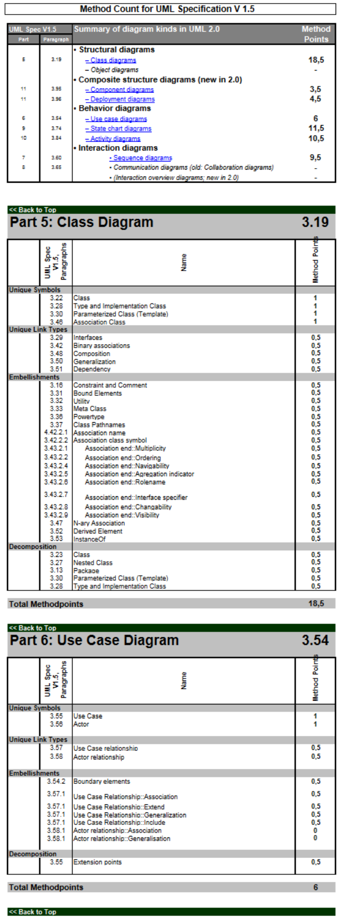
>> Back

Business Process Cooperation

D3.4.1a v2 **5.1.10**

	Archi Mate Deliverable 2.2.2b v3.0	Name	Method Points
Unique Symbols			
	3.6	Application Service	1
	3.4	Application Interface	1
	3.2	Application Component	1
	3.7	Data Object	1
Unique Link Types			
	-	linktype: Access	0,5
	-	linktype: association	0,5
Embellishments			
	-	labels on all link types	0,5
	-	Boundary "Application Packages"	0,5
Decomposition			
		Application Component	0,5
Total Methodpoints			6,5

>> Back



Method Count for UML Specification V 1.5

UML Spec V1.5		Summary of diagram kinds in UML 2.0	Method Points
Part	Paragraph		
		• **Structural diagrams**	
5	3.19	– Class diagrams	18,5
		– *Object diagrams*	-
		• **Composite structure diagrams (new in 2.0)**	
11	3.95	– Component diagrams	3,5
11	3.96	– Deployment diagrams	4,5
		• **Behavior diagrams**	
6	3.54	– Use case diagrams	6
9	3.74	– State chart diagrams	11,5
10	3.84	– Activity diagrams	10,5
		• **Interaction diagrams**	
7	3.60	• Sequence diagrams	9,5
8	3.65	• *Communication diagrams (old: Collaboration diagrams)*	-
		• *(Interaction overview diagrams; new in 2.0)*	-

<< Back to Top

Part 5: Class Diagram 3.19

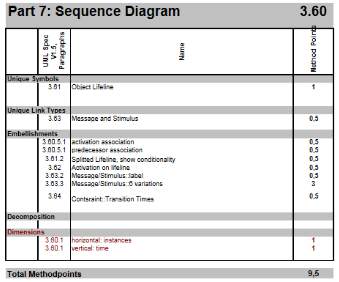
	UML Spec V1.5, Paragraphs	Name	Method Points
Unique Symbols			
	3.22	Class	1
	3.28	Type and Implementation Class	1
	3.30	Parameterized Class (Template)	1
	3.46	Association Class	1
Unique Link Types			
	3.29	Interfaces	0,5
	3.42	Binary associations	0,5
	3.48	Composition	0,5
	3.50	Generalization	0,5
	3.51	Dependency	0,5
Embellishments			
	3.16	Constraint and Comment	0,5
	3.31	Bound Elements	0,5
	3.32	Utility	0,5
	3.33	Meta Class	0,5
	3.36	Powertype	0,5
	3.37	Class Pathnames	0,5
	4.42.2.1	Association name	0,5
	3.42.2.2	Association class symbol	0,5
	3.43.2.1	Association end::Multiplicity	0,5
	3.43.2.2	Association end::Ordering	0,5
	3.43.2.4	Association end::Navigability	0,5
	3.43.2.5	Association end::Agregation indicator	0,5
	3.43.2.6	Association end::Rolename	0,5
	3.43.2.7	Association end::Interface specifier	0,5
	3.43.2.8	Association end::Changability	0,5
	3.43.2.9	Association end::Visibility	0,5
	3.47	N-ary Association	0,5
	3.52	Derived Element	0,5
	3.53	InstanceOf	0,5
Decomposition			
	3.23	Class	0,5
	3.27	Nested Class	0,5
	3.13	Package	0,5
	3.30	Parameterized Class (Template)	0,5
	3.28	Type and Implementation Class	0,5
Total Methodpoints			18,5

<< Back to Top

Part 6: Use Case Diagram 3.54

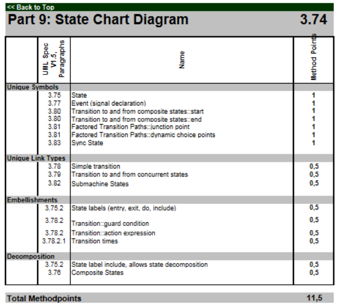
	UML Spec V1.5, Paragraphs	Name	Method Points
Unique Symbols			
	3.55	Use Case	1
	3.56	Actor	1
Unique Link Types			
	3.57	Use Case relationship	0,5
	3.58	Actor relationship	0,5
Embellishments			
	3.54.2	Boundary elements	0,5
	3.57.1	Use Case Relationship::Association	0,5
	3.57.1	Use Case Relationship::Extend	0,5
	3.57.1	Use Case Relationship::Generalization	0,5
	3.57.1	Use Case Relationship::Include	0,5
	3.58.1	Actor relationship::Association	0
	3.58.1	Actor relationship::Generalisation	0
Decomposition			
	3.55	Extension points	0,5
Total Methodpoints			6

<< Back to Top



Part 7: Sequence Diagram 3.60

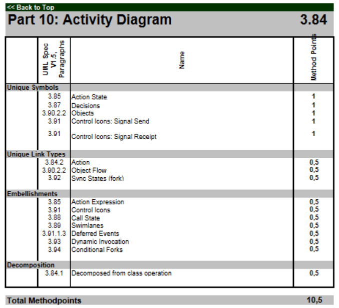
	UML Spec V1.5, Paragraphs	Name	Method Points
Unique Symbols			
	3.61	Object Lifeline	1
Unique Link Types			
	3.63	Message and Stimulus	0,5
Embellishments			
	3.60.5.1	activation association	0,5
	3.60.5.1	predecessor association	0,5
	3.61.2	Splitted Lifeline, show conditionality	0,5
	3.62	Activation on lifeline	0,5
	3.63.2	Message/Stimulus::label	0,5
	3.63.3	Message/Stimulus::6 variations	3
	3.64	Contsraint::Transition Times	0,5
Decomposition			
Dimensions			
	3.60.1	horizontal: instances	1
	3.60.1	vertical: time	1
Total Methodpoints			9,5



<< Back to Top

Part 9: State Chart Diagram 3.74

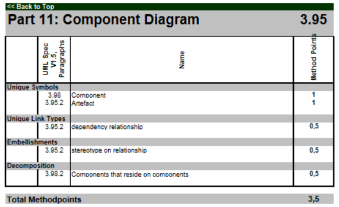
	UML Spec V1.5, Paragraphs	Name	Method Points
Unique Symbols			
	3.75	State	1
	3.77	Event (signal declaration)	1
	3.80	Transition to and from composite states::start	1
	3.80	Transition to and from composite states::end	1
	3.81	Factored Transition Paths::junction point	1
	3.81	Factored Transition Paths::dynamic choice points	1
	3.83	Sync State	1
Unique Link Types			
	3.78	Simple transition	0,5
	3.79	Transition to and from concurrent states	0,5
	3.82	Submachine States	0,5
Embellishments			
	3.75.2	State labels (entry, exit, do, include)	0,5
	3.78.2	Transition::guard condition	0,5
	3.78.2	Transition::action expression	0,5
	3.78.2.1	Transition times	0,5
Decomposition			
	3.75.2	State label include, allows state decomposition	0,5
	3.76	Composite States	0,5
Total Methodpoints			11,5



<< Back to Top

Part 10: Activity Diagram 3.84

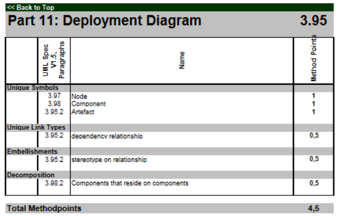
	UML Spec V1.5, Paragraphs	Name	Method Points
Unique Symbols			
	3.85	Action State	1
	3.87	Decisions	1
	3.90.2.2	Objects	1
	3.91	Control Icons: Signal Send	1
	3.91	Control Icons: Signal Receipt	1
Unique Link Types			
	3.84.2	Action	0,5
	3.90.2.2	Object Flow	0,5
	3.92	Sync States (fork)	0,5
Embellishments			
	3.85	Action Expression	0,5
	3.91	Control Icons	0,5
	3.88	Call State	0,5
	3.89	Swimlanes	0,5
	3.91.1.3	Deferred Events	0,5
	3.93	Dynamic Invocation	0,5
	3.94	Conditional Forks	0,5
Decomposition			
	3.84.1	Decomposed from class operation	0,5
Total Methodpoints			10,5



<< Back to Top

Part 11: Component Diagram 3.95

	UML Spec V1.5, Paragraphs	Name	Method Points
Unique Symbols			
	3.98	Component	1
	3.95.2	Artefact	1
Unique Link Types			
	3.95.2	dependency relationship	0,5
Embellishments			
	3.95.2	stereotype on relationship	0,5
Decomposition			
	3.98.2	Components that reside on components	0,5
Total Methodpoints			3,5

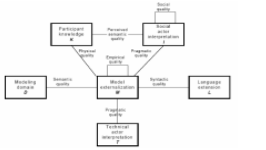


<< Back to Top

Part 11: Deployment Diagram 3.95

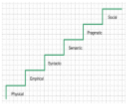
	UML Spec V1.5, Paragraphs	Name	Method Points
Unique Symbols			
	3.97	Node	1
	3.98	Component	1
	3.95.2	Artefact	1
Unique Link Types			
	3.95.2	dependency relationship	0,5
Embellishments			
	3.95.2	stereotype on relationship	0,5
Decomposition			
	3.98.2	Components that reside on components	0,5
Total Methodpoints			4,5

Appendix 3. Results of the Semiotic Evaluation



									K	L	M	D	I	T	UML (V1.5)		Archimate			
ow Nr	Language Properties	Empirical Quality	Pragmatic Quality	Physical Quality	Symantic Quality	Syntactic Quality	Social Quality	Percieved Sym. Quality	Participant Knowledge	Language Extension	Model Externalization	Modeling domain	Social Audience Interpretion	Technical Audience Interpretion	Sub	Total	Sub	Total	remarks	Improvements
	Domain appropriatness															160		221		
	The language should be independent of the business domair of the oroanisation														10		10		UML has a higher abstract level then Archimate.	
	The language should not allow expressiviness outside the enterprise architectural domain.														0		10		UML is to generic. It does not provide the necessary constraints so modelling outside the enterprise architecture is allowed. Archimate has fixed concepts for the architectural domain so modelling outside this domain is not supported.	
	The language should be able to express all architectural concepts in the following architectural dimensions																			
Business Strategy	Business Principles														0		0		is no concept available	A concept for this domain is essential. Think of increasement of the semantic for views in the business domain.
	Business Goals														0		0		is no concept available	
	Strategic Drivers														0		0		is no concept available	
Business	Business Structure (must be decomposable)														4		9		We can use class diagram to express organisational diagrams. However this diagram type is to complex	Detail level is unknown. What can we capture and what not
	Business Actor														10		10		both have the concept actor	
	Business Roles														3		9		UML could use a stereotype. Archimate delivers a discrete concept.	
	Business Collaboration														3		8		Archimate contains a concrete concept. It is questionable if it contains the correct detail.	
	Business Event														3		8		Archimate contains a concrete concept. It is questionable if it contains the correct detail.	
	Information (must be decomposable)														9		8		The class diagram is ideal to model information. Using the stereotypes wh can express the different sorts of information. Archimate has a concrete concept but misses the correct detailled level.	
Process	Business functions (must be decomposable)														6		9		The function support is in archimate on two levels available business, application	
	Proces and Workflow (must be decomposable)														7		8		UML and Archimate are capable to express processes and workflow. However the concept in archimate is better because the relationships with the environment (business, infrastructure) can be better expressed. Main question is also here: "How many details should be captured"	Make a statement how detailed it should be. What can we capture and what not.
	Products / Business Services (must be decomposable)														3		6		UML could express these concepts but only in a class diagram/use case diagram. Archimate is more expressive in this area. Archimate has a concept "value" which is important in trading.	Think of value networks value based governance "wieringa TU Twente"
Application	Application Function (must be decomposable)														5		7		Archimate documentation contains a viewpoint	
	Application Interface														4		5		Both have a suitable concept but the interface must be formulated in a specification	
	Application Service														4		6		Both have a suitable concept but the interface must be formulated in a specification	
	Application collaboration														8		9		UML (Activity Diagram) Archimate has a collaboration concept.	
Infrastructure	Infrastructure Service														3		9		In UML we have the deployment diagrams. It has the concepts as described here. The reason that the infrastructural service score 3 points i the mismatch with the interface concept in the deploymentdiagram. We have to use this symbol to express a service. The concepts confllict with eachother. Archimate has a open point if the details in every concept matches with the architectural needs	
	Device														8		9			
	network														9		9			
	platform														8		9			
	communication														9		9			
Architecture	Support for the concepts of IEEE 1471														4		7		I foccused particular on the concern aspect of IEEE1471. In Archimate the concurn can be expressed in discrete concepts. In UML it is a mixture of stereotype values and concerns. This is not interpretable and leads to inconsistency. (It is impossible to c	
	Architecture principles														2		2		Both languages uses notes to express architectural principles.	Introduce an explicit architecture principle concept.
	The language should be able to express the following seven general perspectives																			
	Structural perspective, the static structure (entities and relationships)														8		6		UML has more types of relationships between entities. We have to mentic that not all possibilities are relevant for the architectural domain.	
	Functional perspective, the processes, activities and transformations for the architectural domain														4		9		Archimate has a good fit with processes and services in the architectural domain. It scores however no 10 as result of the core granularity level. Th concepts should contain more emebellishments.	Add more embellishments in the archimate concepts.

Behavioral perspective, the states and transactions between them			■	■						■				7	7	UML has use cases, activity diagrams and state diagrams. Alle these diagrams are isolated views. Archimate offers a more integrated way to express behaviour. states and transactions	
Rule perspective, the rules for certain processes, activities..			■	■						■				2	0	UML has the activity diagram to express business rules. Archimate has nc business rule concepts. This is however no good centralised concept.	
Object perspective, the objects methods, attributes, processe and classes			■	■						■				10	6	UML is developped for this purpose. Archimate made a decission to support the enterprise architectural domain. Archimate has to few embelisshments in the concepts.	
Communication perspective, the language actions, meaning and agreement for the architectural domain			■	■						■				4	8	In UML a profile should contain all expressions and definitions. In Archimate the language specification contain all definitions	
Actor and role perspective, role, society and organisation in the architectural domain			■	■						■				3	9	UML has to many relationships available that cannot be appllied in the architectural domain.	
Participant language knowledge appropriateness		■	■						■	■				16	25		
The terms or concepts must be same as those for the organisation. At least all specfic organisational terms must be "mappable"		■	■						■	■				8	8	UML has stereotypes that can be used to express organisation specific concepts. Archimate has specific concepts that matches with 90% of the used concepts in organisation. Archimate has however no extention point like the stereotypes of UML that could be used for mapping	Implement extention points in archimate.
It must be easy to learn the language.		■	■						■	■				5	8	UML is not quite easy to learn. This becomes even more difficult when a architectural profile should be applied. Also the graphical UML symbols ar too uniform that the diagrams are hard to interprete. This makes UML no intuitive language. Archimate has more specific ICONS for all concepts.	
The external representation must be intuitive, meaning that the symbol represent the concept better then another symbol would.		■	■						■	■				3	9	These Icons are easy to interpete and makes Archimate a intuitive language.	
Knowledge externalisability appropriateness		■	■						■	■				18	16		
Structural data deliverables		■	■						■	■				10	6	The UML concept contains a XMI specification. Archimate has no official xmi it depends on tool implementations. Known formats for Archimate are propriatry reposity formats, eclipse modeling framework, XMI	Archimate should make an official specification of a xml based mode persistency format. This concept would be even stronger that XMI i the IEEE 1471 could be applied o the format by persisting views
Textual deliverables		■	■						■	■				0	0	Both languages have no textual deliverable specification	
Graphical deliverables		■	■						■	■				8	10	UML has a fixed set of diagram types that can be delivered by a user. The IEEE 1471 concept makes it possible to define own graphiacl deliverables This is a must have in the architectural domain while the stakeholders are unknown	
Comprehensibiliy appropriateness	■	■						■	■					31	49		
Easily distinguishable concepts	■	■						■	■					3	10	Archimate has for the architectururaI domain easy distinguishable concepts. To distinguish the concepts in UML is mainly based on the usag of stereotvpes.	
Reasonable number of concepts	■	■						■	■					3	10	For the architectural domain UML has unreasonable amount of concepts i is to generic	
Uniform usage of concepts	■	■						■	■					6	8	UML is for system development at a correct abstract level, At the abstract level a uniform usage is garanteed. But where UML scores badly is the semantic value of the diagram when stereotypes are used. (example: car does not relate to a bed) regard that whe evaluate both languages in the architectural domain.	
Flexible level of detail	■	■						■	■					9	4	decomposition of concepts in archimate	
Support for separation into areas of concern	■	■						■	■					6	9	UML knows diagram types for areas of concerns. Archimate supports IEE 1471	
Expressive economy (the most frequent and important statements are brief).	■	■						■	■					4	8	The expressiviness in UML has to occure with stereotypes. In the applied profile the meanings should be specificated. Archimate knows predefined architectural concepts that are uniquelly expressed with an icon.	
Technical actor appropriateness		■		■	■				■		■	■		23	7		
The process of making a more formal specification may reveal errors and ambiguities at an early stage in the develooment process														8	2	UML knowns manny concepts for consistency proofing, metrics and indicators. Together with a process methodology like RUP makes it possible to indicate errors in the specification in an early stage. Archimate is to young to fullfill this needs. UML Metrics is a mature scientific domain. Many implementations of metrics of the OO domain are available in modern UML tools. Architecture metrics is a niche scientific domain. (Torr described architecture analysis based on XML)	The unavailabity of an archimate xml based persitsency format ala XMI is a major lack of support in syntactically quality.
Support of formal metrics.														7	3		
Formal and automated proofing														8	2		
The remaining (or improvable) rules may be translated into executable constraints in some imperative language.														0	0		
UML V1.5	31	65	194	183	23												
Archimate	49	97	262	228	7												
UML V1.5 (relative)	0,388	0,401	0,425	0,445	0,767												
Archimate (relative)	0,613	0,599	0,575	0,555	0,233												
Relative values (UML - Archimate)	-0,23	-0,2	-0,15	-0,11	0,533	0	0										



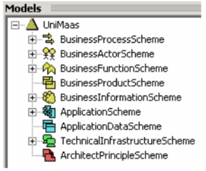
UML	Archimate
0	0
65	97 ←
183	228 ←
23 ←	7
31	49 ←
194	262 ←

Human Information Level (Social, Pragmatic, Semantic)	248	325
IT Information level (Syntactic, Empircal)	54	56
Physical	194	262

Quality types	Quality goals	Criteria	Activities/techniques
Syntactic	Syntactic validity	Lexicon correctness	Lexicon checking
		Graphical element selection	Graphical element checking
		Syntactic correctness	Syntax checking
		Structural correctness	Structure checking
Semantic	Semantic validity	Functional requirements compliance	Inspection, traceability analysis, data-flow analysis, O-O / requirements mapping analysis
		Quality requirements compliance	ISO/IEC-9126 inspection, complexity analysis, coupling & cohesion analysis, element counting
		Specific architectural requirements compliance	Inspection, element counting
	Semantic completeness	Domain-dependent completeness	Traceability analysis
		Domain-independent completeness	Traceability analysis, consistency checking
Pragmatic	Valid comprehension	Readability	Complexity analysis, coupling & cohesion analysis, readability formulae, language translation, joint reviews
	Complete comprehension	Navigability	Browsing, joint reviews

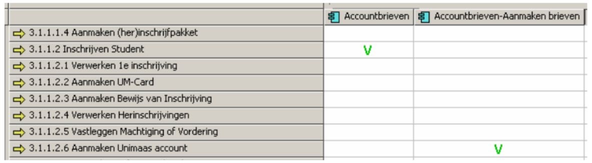
Appendix 4. The Landscape map as UM ICT policy tool

3 DE LANDSCHAPSKAART ALS BELEIDSINSTRUMENT



Zoals vermeld zijn er verschillende domeinen vertegenwoordigd in het architectuur domein. Als alle domein informatie in een repository is ingevoerd heeft men de beschikking over alle onderlinge samenhang is ingevoerd heeft men de beschikking over alle relatie's en entiteiten in een gestructureerde vorm. In de afbeelding aan uw rechterzijde is een afbeelding opgenomen van de architectuur repository van Bizzdesign Architect. Hierin zijn de verschillende architectuur domeinen goed te zien.

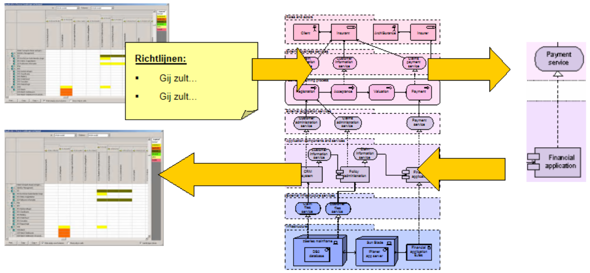
In het model liggen relaties tussen processen en ondersteunende applicaties maar ook bijvoorbeeld relaties tussen applicaties met de infrastructuur (servers en databases) of applicaties met de business objecten van de UM. Deze relaties zijn af te beelden in een matrix waarop deze relaties kunnen worden geprojecteerd. Dit is al een landschapskaart zoals wij het bedoelen in de architectuur context. Zie onderstaande afbeelding voor een indruk.



	Accountbrieven	Accountbrieven-Aanmaken brieven
3.1.1.1.4 Aanmaken (her)inschrijfpakket		
3.1.1.2 Inschrijven Student	V	
3.1.1.2.1 Verwerken 1e inschrijving		
3.1.1.2.2 Aanmaken UM-Card		
3.1.1.2.3 Aanmaken Bewijs van Inschrijving		
3.1.1.2.4 Verwerken Herinschrijvingen		
3.1.1.2.5 Vastleggen Machtiging of Vordering		
3.1.1.2.6 Aanmaken Unimaas account		V

Deze landschapskaart tussen processen en applicaties kan natuurlijk ook voor applicaties en infrastructuur worden gegenereerd. Dit project heeft een aantal landschapskaarten als project producten vastgelegd. (Zie 3). Om besluitvormers en beleidsmakers te ondersteunen is meer nodig dan een relatie matrix. De landschapskaarten moeten gericht zijn op de ICT Strategie waarop de veranderingen van de architectuur geprojecteerd kunnen worden. Wij introduceren hier het begrip Landschapsthema. Door een ICT thema te selecteren en deze te vertalen naar een eigenschap van een architectuur entiteit is men in staat om dit veld af te drukken op de landschapskaart waardoor de landschapskaart een thema krijgt. Een klein voorbeeld om dit toe te lichten:

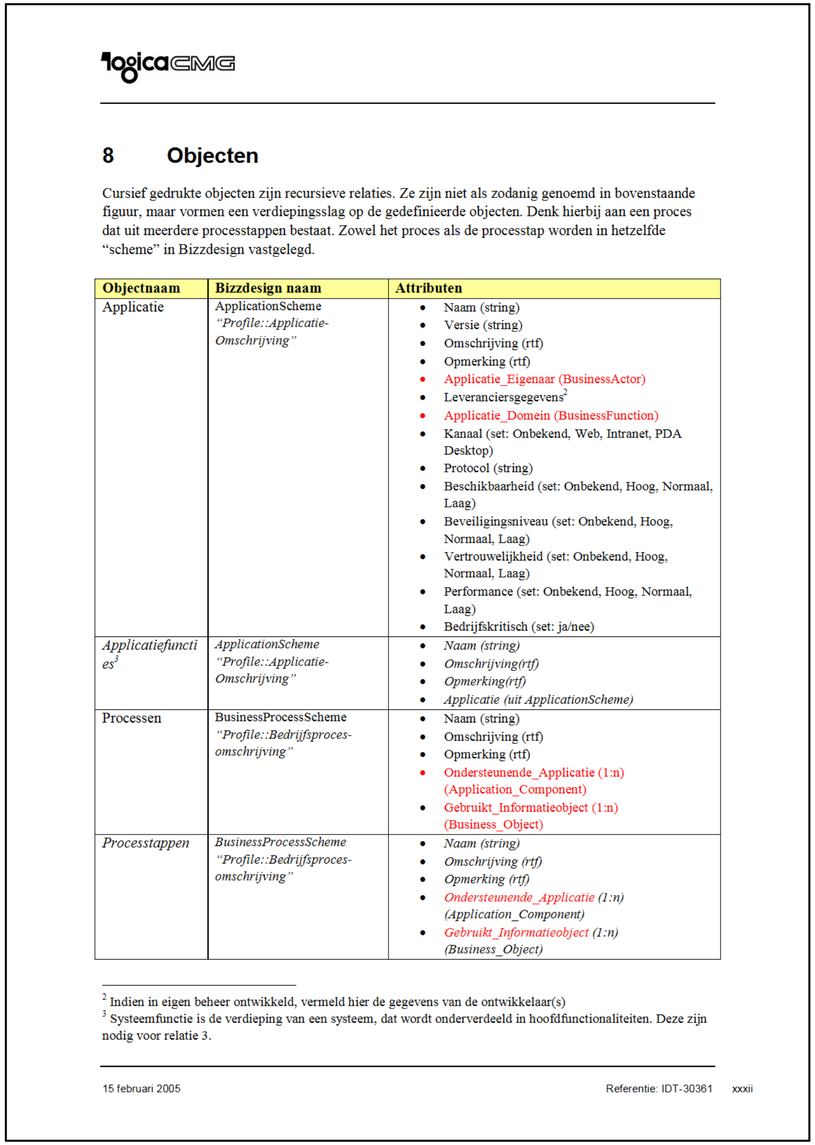
ICT Beleid - Een organisatie wil haar service verlening richting de klant verbeteren en de kosten in de backoffice reduceren.
ICT Initiatief – Er dient meer functionaliteit via het internet ontsloten te worden waardoor de klant 24x7 diensten kan afnemen en de Backoffice door 'Self Service' te ontlasten.
Indicator – Bij ieder applicatie wordt de eigenschap 'kanaal' toegevoegd waardoor wij van ieder applicatie, informatie rondom zijn beschikbaarheid verzamelen. Wij introduceren de eigenschap 'kanaal' met mogelijke waarden als (desktop, intranet, internet, Citrix)
De landschapskaarten zijn hiermee een uitvoeringsinstrument geworden voor de organisatie om het beleid te volgen en te monitoren. Zie voor een impressie de volgende afbeelding.



Doordat deze landschapskaarten gegenereerd worden vanuit een repository die actief wordt onderhouden of die bij audits gebruikt kunnen worden is de representatie levend en geven een levensecht beeld van de architectuur.

De eigenschappen die ieder architectuur entiteit meekrijgt zullen vanuit het ICT Beleid moeten worden doorvertaald. De tooling die voor dit traject is geselecteerd biedt maximale flexibiliteit om eigenschappen toe te voegen d.m.v. profielen. Voor de UM is een profiel opgesteld die na overdracht door de UM steeds opnieuw als basis kan worden gebruikt.

Appendix 5. The object contract of the UM inventory (Dutch)



8 Objecten

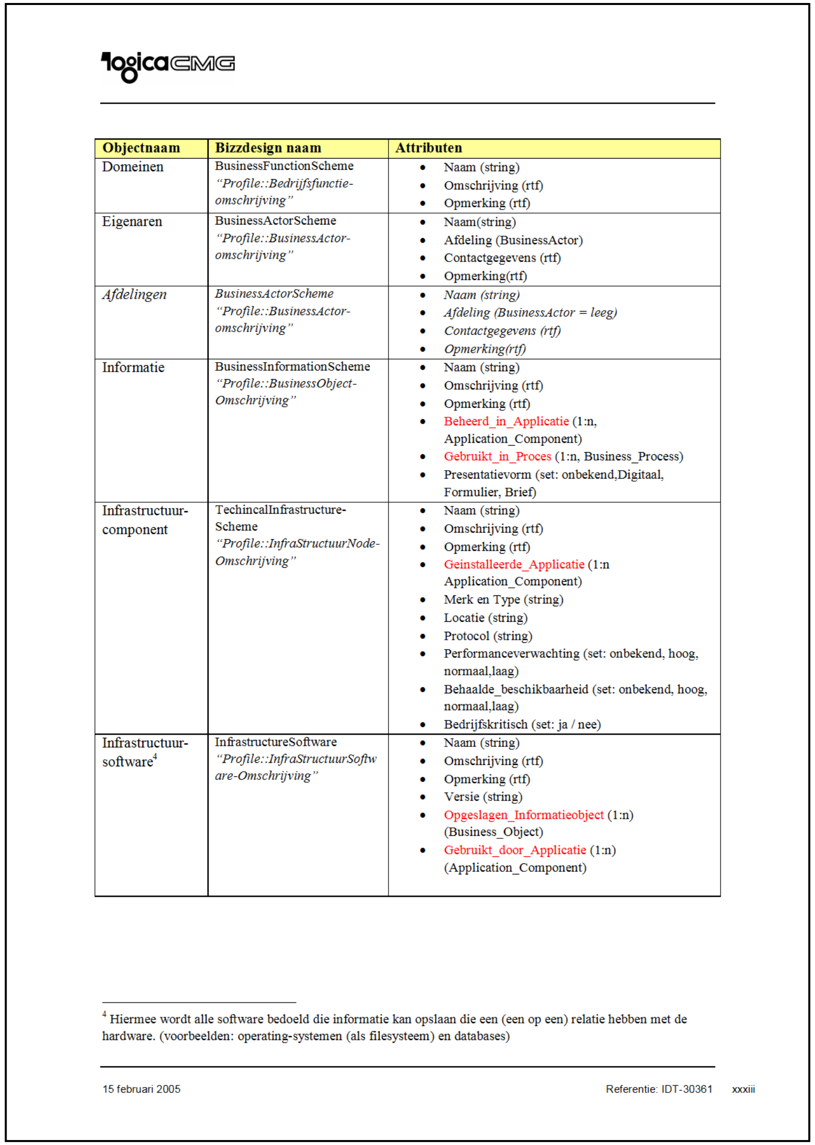
Cursief gedrukte objecten zijn recursieve relaties. Ze zijn niet als zodanig genoemd in bovenstaande figuur, maar vormen een verdiepingsslag op de gedefinieerde objecten. Denk hierbij aan een proces dat uit meerdere processtappen bestaat. Zowel het proces als de processtap worden in hetzelfde "scheme" in Bizzdesign vastgelegd.

Objectnaam	Bizzdesign naam	Attributen
Applicatie	ApplicationScheme *"Profile::Applicatie-Omschrijving"*	• Naam (string) • Versie (string) • Omschrijving (rtf) • Opmerking (rtf) • Applicatie_Eigenaar (BusinessActor) • Leveranciersgegevens[2] • Applicatie_Domein (BusinessFunction) • Kanaal (set: Onbekend, Web, Intranet, PDA Desktop) • Protocol (string) • Beschikbaarheid (set: Onbekend, Hoog, Normaal, Laag) • Beveiligingsniveau (set: Onbekend, Hoog, Normaal, Laag) • Vertrouwelijkheid (set: Onbekend, Hoog, Normaal, Laag) • Performance (set: Onbekend, Hoog, Normaal, Laag) • Bedrijfskritisch (set: ja/nee)
Applicatiefuncties[3]	*ApplicationScheme* *"Profile::Applicatie-Omschrijving"*	• *Naam (string)* • *Omschrijving(rtf)* • *Opmerking(rtf)* • *Applicatie (uit ApplicationScheme)*
Processen	BusinessProcessScheme *"Profile::Bedrijfsproces-omschrijving"*	• Naam (string) • Omschrijving (rtf) • Opmerking (rtf) • Ondersteunende_Applicatie (1:n) (Application_Component) • Gebruikt_Informatieobject (1:n) (Business_Object)
Processtappen	*BusinessProcessScheme* *"Profile::Bedrijfsproces-omschrijving"*	• *Naam (string)* • *Omschrijving (rtf)* • *Opmerking (rtf)* • *Ondersteunende_Applicatie (1:n) (Application_Component)* • *Gebruikt_Informatieobject (1:n) (Business_Object)*

[2] Indien in eigen beheer ontwikkeld, vermeld hier de gegevens van de ontwikkelaar(s)

[3] Systeemfunctie is de verdieping van een systeem, dat wordt onderverdeeld in hoofdfunctionaliteiten. Deze zijn nodig voor relatie 3.

15 februari 2005 Referentie: IDT-30361 xxxii

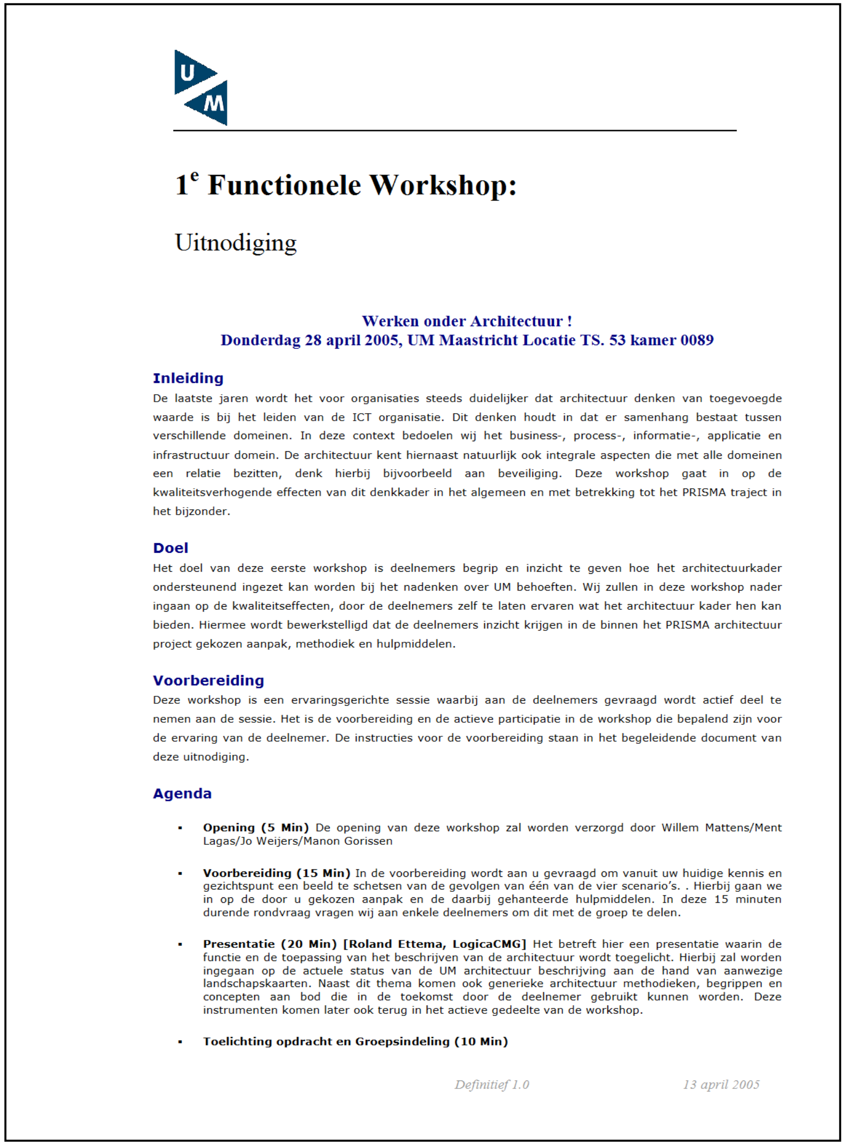


Objectnaam	Bizzdesign naam	Attributen
Domeinen	BusinessFunctionScheme *"Profile::Bedrijfsfunctie-omschrijving"*	• Naam (string) • Omschrijving (rtf) • Opmerking (rtf)
Eigenaren	BusinessActorScheme *"Profile::BusinessActor-omschrijving"*	• Naam(string) • Afdeling (BusinessActor) • Contactgegevens (rtf) • Opmerking(rtf)
Afdelingen	*BusinessActorScheme* *"Profile::BusinessActor-omschrijving"*	• *Naam (string)* • *Afdeling (BusinessActor = leeg)* • *Contactgegevens (rtf)* • *Opmerking(rtf)*
Informatie	BusinessInformationScheme *"Profile::BusinessObject-Omschrijving"*	• Naam (string) • Omschrijving (rtf) • Opmerking (rtf) • Beheerd_in_Applicatie (1:n, Application_Component) • Gebruikt_in_Proces (1:n, Business_Process) • Presentatievorm (set: onbekend,Digitaal, Formulier, Brief)
Infrastructuur-component	TechincalInfrastructure-Scheme *"Profile::InfraStructuurNode-Omschrijving"*	• Naam (string) • Omschrijving (rtf) • Opmerking (rtf) • Geinstalleerde_Applicatie (1:n Application_Component) • Merk en Type (string) • Locatie (string) • Protocol (string) • Performanceverwachting (set: onbekend, hoog, normaal,laag) • Behaalde_beschikbaarheid (set: onbekend, hoog, normaal,laag) • Bedrijfskritisch (set: ja / nee)
Infrastructuur-software[4]	InfrastructureSoftware *"Profile::InfraStructuurSoftware-Omschrijving"*	• Naam (string) • Omschrijving (rtf) • Opmerking (rtf) • Versie (string) • Opgeslagen_Informatieobject (1:n) (Business_Object) • Gebruikt_door_Applicatie (1:n) (Application_Component)

[4] Hiermee wordt alle software bedoeld die informatie kan opslaan die een (een op een) relatie hebben met de hardware. (voorbeelden: operating-systemen (als filesysteem) en databases)

15 februari 2005 Referentie: IDT-30361 xxxiii

Appendix 6. Workshop design & Instructions (Dutch)



1e Functionele Workshop:

Uitnodiging

Werken onder Architectuur !
Donderdag 28 april 2005, UM Maastricht Locatie TS. 53 kamer 0089

Inleiding

De laatste jaren wordt het voor organisaties steeds duidelijker dat architectuur denken van toegevoegde waarde is bij het leiden van de ICT organisatie. Dit denken houdt in dat er samenhang bestaat tussen verschillende domeinen. In deze context bedoelen wij het business-, process-, informatie-, applicatie en infrastructuur domein. De architectuur kent hiernaast natuurlijk ook integrale aspecten die met alle domeinen een relatie bezitten, denk hierbij bijvoorbeeld aan beveiliging. Deze workshop gaat in op de kwaliteitsverhogende effecten van dit denkkader in het algemeen en met betrekking tot het PRISMA traject in het bijzonder.

Doel

Het doel van deze eerste workshop is deelnemers begrip en inzicht te geven hoe het architectuurkader ondersteunend ingezet kan worden bij het nadenken over UM behoeften. Wij zullen in deze workshop nader ingaan op de kwaliteitseffecten, door de deelnemers zelf te laten ervaren wat het architectuur kader hen kan bieden. Hiermee wordt bewerkstelligd dat de deelnemers inzicht krijgen in de binnen het PRISMA architectuur project gekozen aanpak, methodiek en hulpmiddelen.

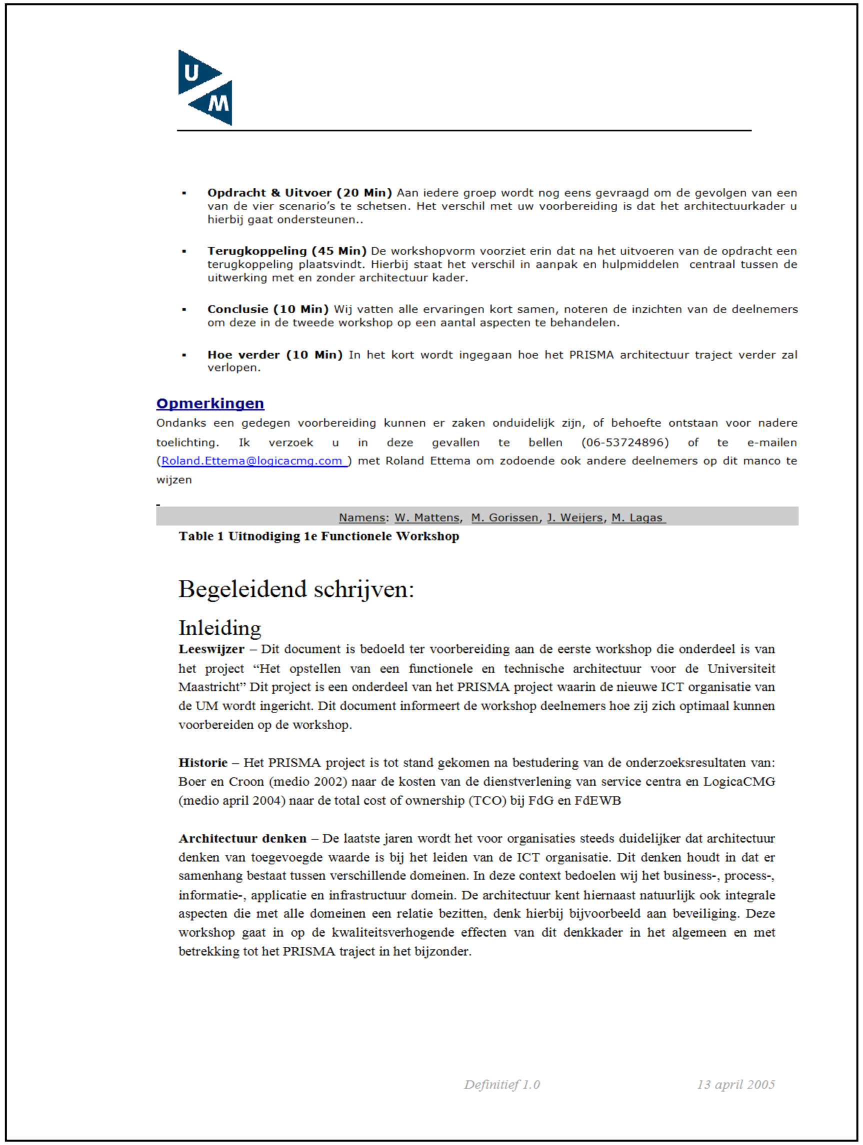
Voorbereiding

Deze workshop is een ervaringsgerichte sessie waarbij aan de deelnemers gevraagd wordt actief deel te nemen aan de sessie. Het is de voorbereiding en de actieve participatie in de workshop die bepalend zijn voor de ervaring van de deelnemer. De instructies voor de voorbereiding staan in het begeleidende document van deze uitnodiging.

Agenda

- **Opening (5 Min)** De opening van deze workshop zal worden verzorgd door Willem Mattens/Ment Lagas/Jo Weijers/Manon Gorissen
- **Voorbereiding (15 Min)** In de voorbereiding wordt aan u gevraagd om vanuit uw huidige kennis en gezichtspunt een beeld te schetsen van de gevolgen van één van de vier scenario's. . Hierbij gaan we in op de door u gekozen aanpak en de daarbij gehanteerde hulpmiddelen. In deze 15 minuten durende rondvraag vragen wij aan enkele deelnemers om dit met de groep te delen.
- **Presentatie (20 Min) [Roland Ettema, LogicaCMG]** Het betreft hier een presentatie waarin de functie en de toepassing van het beschrijven van de architectuur wordt toegelicht. Hierbij zal worden ingegaan op de actuele status van de UM architectuur beschrijving aan de hand van aanwezige landschapskaarten. Naast dit thema komen ook generieke architectuur methodieken, begrippen en concepten aan bod die in de toekomst door de deelnemer gebruikt kunnen worden. Deze instrumenten komen later ook terug in het actieve gedeelte van de workshop.
- **Toelichting opdracht en Groepsindeling (10 Min)**

Definitief 1.0 *13 april 2005*



- **Opdracht & Uitvoer (20 Min)** Aan iedere groep wordt nog eens gevraagd om de gevolgen van een van de vier scenario's te schetsen. Het verschil met uw voorbereiding is dat het architectuurkader u hierbij gaat ondersteunen..
- **Terugkoppeling (45 Min)** De workshopvorm voorziet erin dat na het uitvoeren van de opdracht een terugkoppeling plaatsvindt. Hierbij staat het verschil in aanpak en hulpmiddelen centraal tussen de uitwerking met en zonder architectuur kader.
- **Conclusie (10 Min)** Wij vatten alle ervaringen kort samen, noteren de inzichten van de deelnemers om deze in de tweede workshop op een aantal aspecten te behandelen.
- **Hoe verder (10 Min)** In het kort wordt ingegaan hoe het PRISMA architectuur traject verder zal verlopen.

Opmerkingen

Ondanks een gedegen voorbereiding kunnen er zaken onduidelijk zijn, of behoefte ontstaan voor nadere toelichting. Ik verzoek u in deze gevallen te bellen (06-53724896) of te e-mailen (Roland.Ettema@logicacmg.com) met Roland Ettema om zodoende ook andere deelnemers op dit manco te wijzen

Namens: W. Mattens, M. Gorissen, J. Weijers, M. Lagas

Table 1 Uitnodiging 1e Functionele Workshop

Begeleidend schrijven:

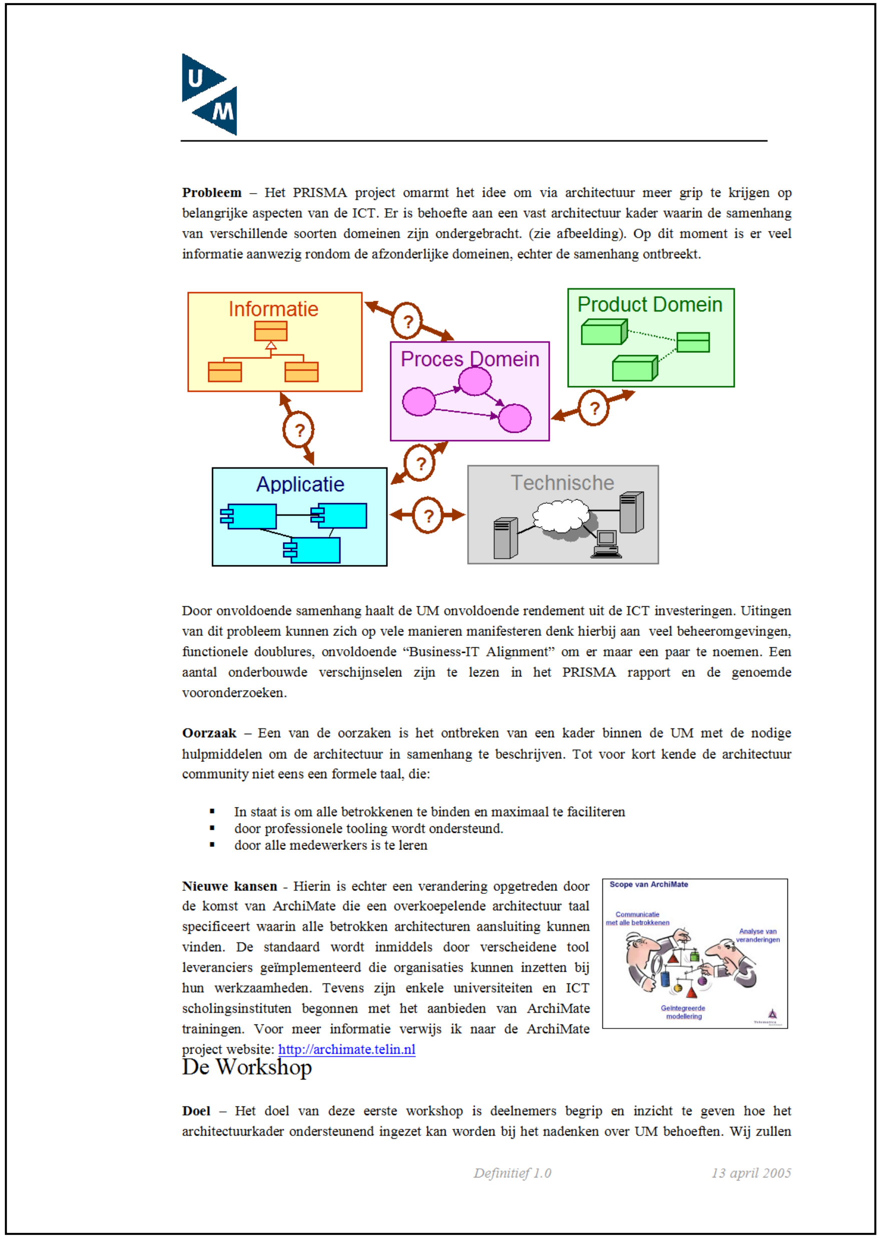
Inleiding

Leeswijzer – Dit document is bedoeld ter voorbereiding aan de eerste workshop die onderdeel is van het project "Het opstellen van een functionele en technische architectuur voor de Universiteit Maastricht" Dit project is een onderdeel van het PRISMA project waarin de nieuwe ICT organisatie van de UM wordt ingericht. Dit document informeert de workshop deelnemers hoe zij zich optimaal kunnen voorbereiden op de workshop.

Historie – Het PRISMA project is tot stand gekomen na bestudering van de onderzoeksresultaten van: Boer en Croon (medio 2002) naar de kosten van de dienstverlening van service centra en LogicaCMG (medio april 2004) naar de total cost of ownership (TCO) bij FdG en FdEWB

Architectuur denken – De laatste jaren wordt het voor organisaties steeds duidelijker dat architectuur denken van toegevoegde waarde is bij het leiden van de ICT organisatie. Dit denken houdt in dat er samenhang bestaat tussen verschillende domeinen. In deze context bedoelen wij het business-, process-, informatie-, applicatie en infrastructuur domein. De architectuur kent hiernaast natuurlijk ook integrale aspecten die met alle domeinen een relatie bezitten, denk hierbij bijvoorbeeld aan beveiliging. Deze workshop gaat in op de kwaliteitsverhogende effecten van dit denkkader in het algemeen en met betrekking tot het PRISMA traject in het bijzonder.

Definitief 1.0 *13 april 2005*



Probleem – Het PRISMA project omarmt het idee om via architectuur meer grip te krijgen op belangrijke aspecten van de ICT. Er is behoefte aan een vast architectuur kader waarin de samenhang van verschillende soorten domeinen zijn ondergebracht. (zie afbeelding). Op dit moment is er veel informatie aanwezig rondom de afzonderlijke domeinen, echter de samenhang ontbreekt.

Door onvoldoende samenhang haalt de UM onvoldoende rendement uit de ICT investeringen. Uitingen van dit probleem kunnen zich op vele manieren manifesteren denk hierbij aan veel beheeromgevingen, functionele doublures, onvoldoende "Business-IT Alignment" om er maar een paar te noemen. Een aantal onderbouwde verschijnselen zijn te lezen in het PRISMA rapport en de genoemde vooronderzoeken.

Oorzaak – Een van de oorzaken is het ontbreken van een kader binnen de UM met de nodige hulpmiddelen om de architectuur in samenhang te beschrijven. Tot voor kort kende de architectuur community niet eens een formele taal, die:

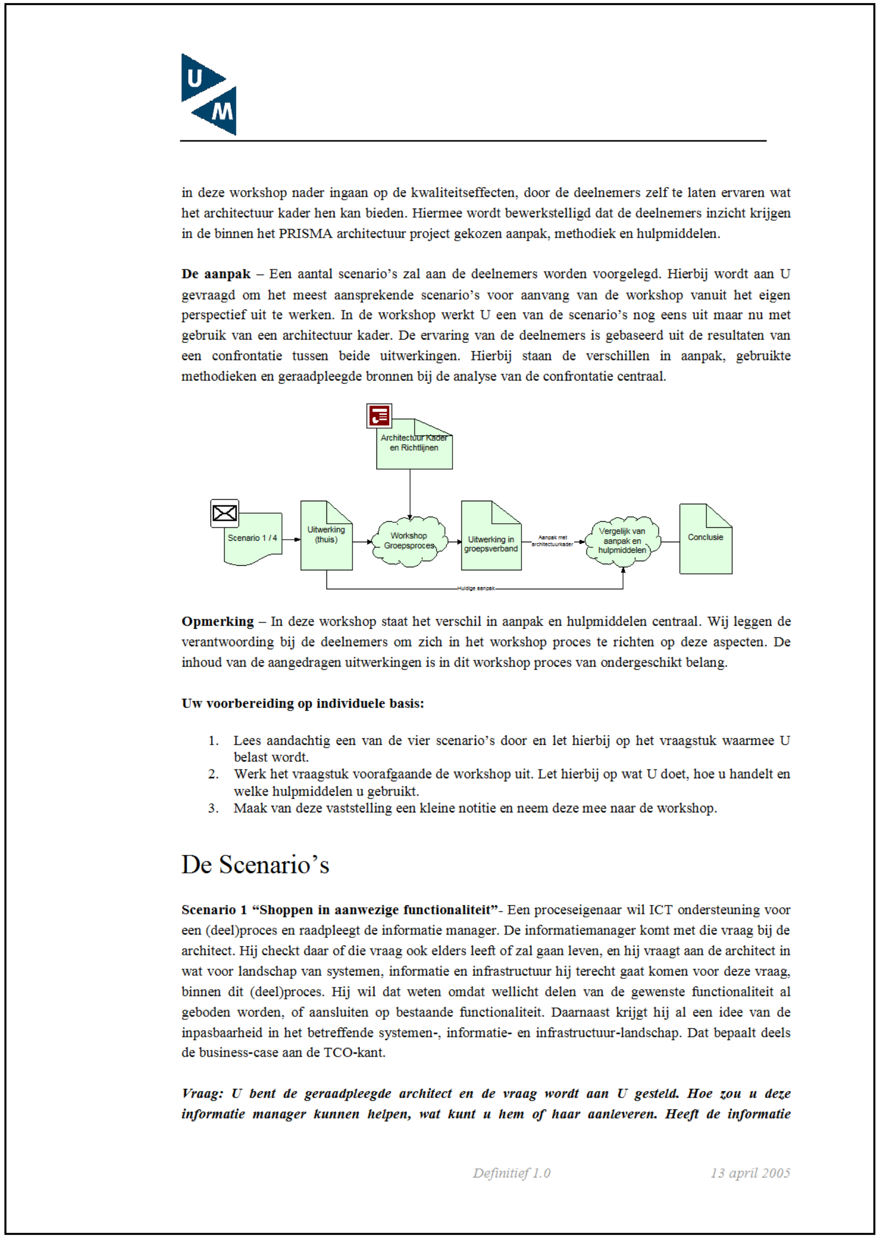
- In staat is om alle betrokkenen te binden en maximaal te faciliteren
- door professionele tooling wordt ondersteund.
- door alle medewerkers is te leren

Nieuwe kansen - Hierin is echter een verandering opgetreden door de komst van ArchiMate die een overkoepelende architectuur taal specificeert waarin alle betrokken architecturen aansluiting kunnen vinden. De standaard wordt inmiddels door verscheidene tool leveranciers geïmplementeerd die organisaties kunnen inzetten bij hun werkzaamheden. Tevens zijn enkele universiteiten en ICT scholingsinstituten begonnen met het aanbieden van ArchiMate trainingen. Voor meer informatie verwijs ik naar de ArchiMate project website: http://archimate.telin.nl

Scope van ArchiMate
Communicatie met alle betrokkenen
Analyse van veranderingen
Geïntegreerde modellering
Telematica

De Workshop

Doel – Het doel van deze eerste workshop is deelnemers begrip en inzicht te geven hoe het architectuurkader ondersteunend ingezet kan worden bij het nadenken over UM behoeften. Wij zullen



in deze workshop nader ingaan op de kwaliteitseffecten, door de deelnemers zelf te laten ervaren wat het architectuur kader hen kan bieden. Hiermee wordt bewerkstelligd dat de deelnemers inzicht krijgen in de binnen het PRISMA architectuur project gekozen aanpak, methodiek en hulpmiddelen.

De aanpak – Een aantal scenario's zal aan de deelnemers worden voorgelegd. Hierbij wordt aan U gevraagd om het meest aansprekende scenario's voor aanvang van de workshop vanuit het eigen perspectief uit te werken. In de workshop werkt U een van de scenario's nog eens uit maar nu met gebruik van een architectuur kader. De ervaring van de deelnemers is gebaseerd uit de resultaten van een confrontatie tussen beide uitwerkingen. Hierbij staan de verschillen in aanpak, gebruikte methodieken en geraadpleegde bronnen bij de analyse van de confrontatie centraal.

Architectuur Kader en Richtlijnen
Scenario 1 / 4
Uitwerking (thuis)
Workshop Groepsproces
Uitwerking in groepsverband
Aanpak met architectuurkader
Vergelijk van aanpak en hulpmiddelen
Conclusie
Huidige aanpak

Opmerking – In deze workshop staat het verschil in aanpak en hulpmiddelen centraal. Wij leggen de verantwoording bij de deelnemers om zich in het workshop proces te richten op deze aspecten. De inhoud van de aangedragen uitwerkingen is in dit workshop proces van ondergeschikt belang.

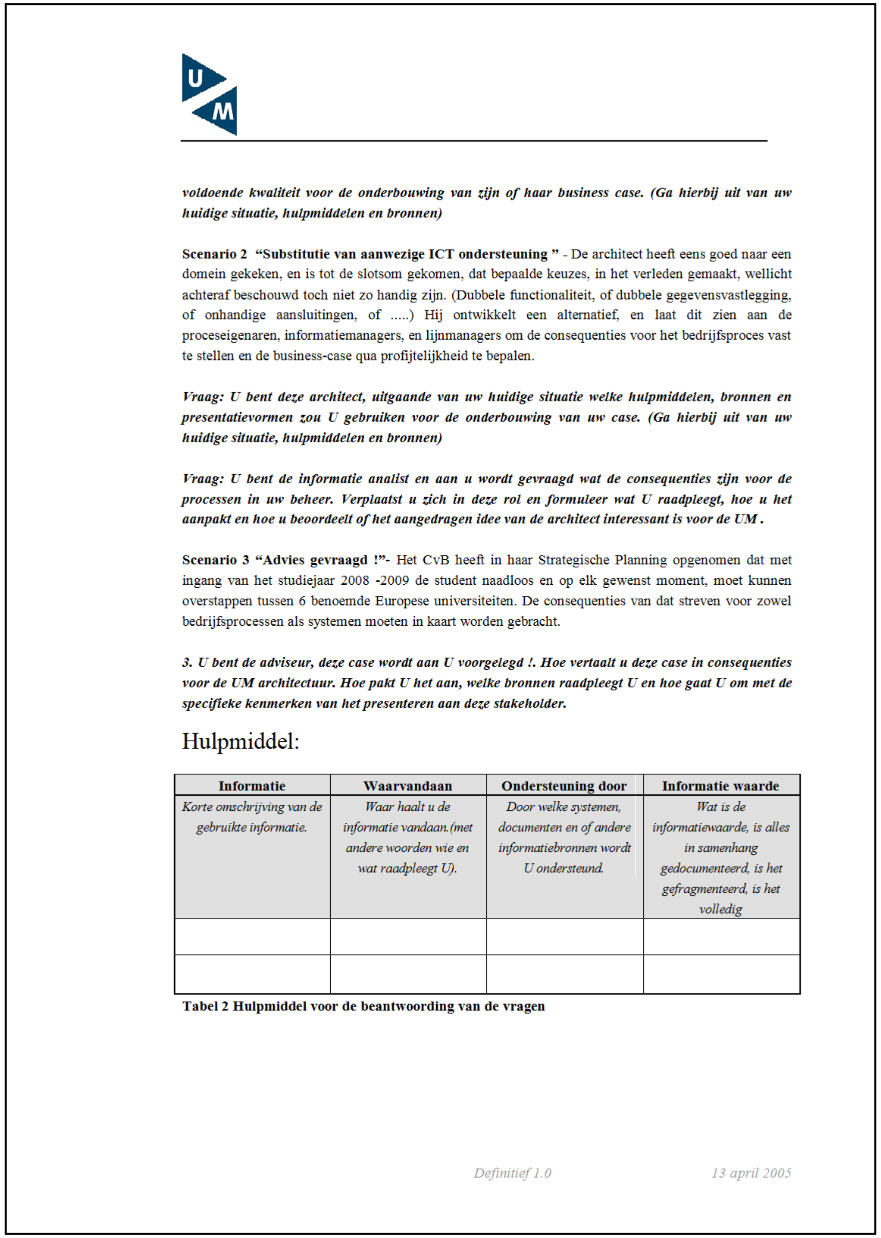
Uw voorbereiding op individuele basis:

1. Lees aandachtig een van de vier scenario's door en let hierbij op het vraagstuk waarmee U belast wordt.
2. Werk het vraagstuk voorafgaande de workshop uit. Let hierbij op wat U doet, hoe u handelt en welke hulpmiddelen u gebruikt.
3. Maak van deze vaststelling een kleine notitie en neem deze mee naar de workshop.

De Scenario's

Scenario 1 "Shoppen in aanwezige functionaliteit"- Een proceseigenaar wil ICT ondersteuning voor een (deel)proces en raadpleegt de informatie manager. De informatiemanager komt met die vraag bij de architect. Hij checkt daar of die vraag ook elders leeft of zal gaan leven, en hij vraagt aan de architect in wat voor landschap van systemen, informatie en infrastructuur hij terecht gaat komen voor deze vraag, binnen dit (deel)proces. Hij wil dat weten omdat wellicht delen van de gewenste functionaliteit al geboden worden, of aansluiten op bestaande functionaliteit. Daarnaast krijgt hij al een idee van de inpasbaarheid in het betreffende systemen-, informatie- en infrastructuur-landschap. Dat bepaalt deels de business-case aan de TCO-kant.

Vraag: U bent de geraadpleegde architect en de vraag wordt aan U gesteld. Hoe zou u deze informatie manager kunnen helpen, wat kunt u hem of haar aanleveren. Heeft de informatie



voldoende kwaliteit voor de onderbouwing van zijn of haar business case. (Ga hierbij uit van uw huidige situatie, hulpmiddelen en bronnen)

Scenario 2 "Substitutie van aanwezige ICT ondersteuning " - De architect heeft eens goed naar een domein gekeken, en is tot de slotsom gekomen, dat bepaalde keuzes, in het verleden gemaakt, wellicht achteraf beschouwd toch niet zo handig zijn. (Dubbele functionaliteit, of dubbele gegevensvastlegging, of onhandige aansluitingen, of .....) Hij ontwikkelt een alternatief, en laat dit zien aan de proceseigenaren, informatiemanagers, en lijnmanagers om de consequenties voor het bedrijfsproces vast te stellen en de business-case qua profijtelijkheid te bepalen.

Vraag: U bent deze architect, uitgaande van uw huidige situatie welke hulpmiddelen, bronnen en presentatievormen zou U gebruiken voor de onderbouwing van uw case. (Ga hierbij uit van uw huidige situatie, hulpmiddelen en bronnen)

Vraag: U bent de informatie analist en aan u wordt gevraagd wat de consequenties zijn voor de processen in uw beheer. Verplaatst u zich in deze rol en formuleer wat U raadpleegt, hoe u het aanpakt en hoe u beoordeelt of het aangedragen idee van de architect interessant is voor de UM .

Scenario 3 "Advies gevraagd !"- Het CvB heeft in haar Strategische Planning opgenomen dat met ingang van het studiejaar 2008 -2009 de student naadloos en op elk gewenst moment, moet kunnen overstappen tussen 6 benoemde Europese universiteiten. De consequenties van dat streven voor zowel bedrijfsprocessen als systemen moeten in kaart worden gebracht.

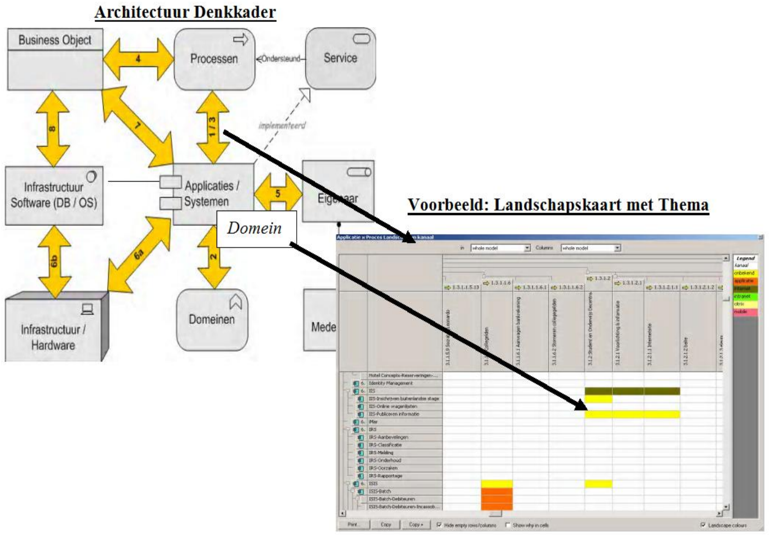
3. U bent de adviseur, deze case wordt aan U voorgelegd !. Hoe vertaalt u deze case in consequenties voor de UM architectuur. Hoe pakt U het aan, welke bronnen raadpleegt U en hoe gaat U om met de specifieke kenmerken van het presenteren aan deze stakeholder.

Hulpmiddel:

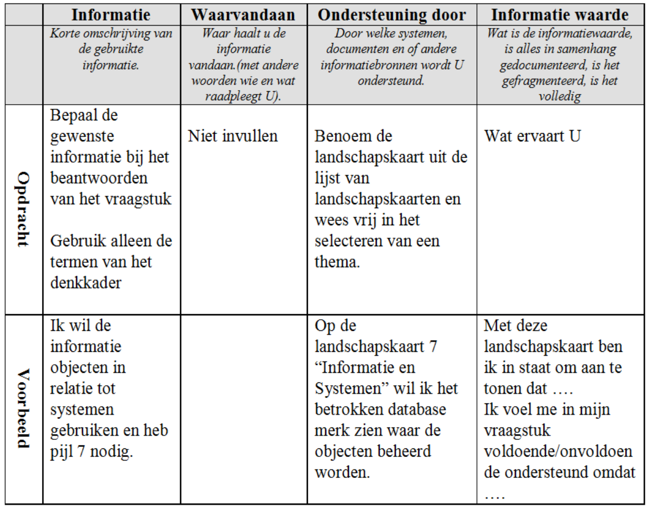
Informatie	Waarvandaan	Ondersteuning door	Informatie waarde
Korte omschrijving van de gebruikte informatie.	*Waar haalt u de informatie vandaan.(met andere woorden wie en wat raadpleegt U).*	*Door welke systemen, documenten en of andere informatiebronnen wordt U ondersteund.*	*Wat is de informatiewaarde, is alles in samenhang gedocumenteerd, is het gefragmenteerd, is het volledig*

Tabel 2 Hulpmiddel voor de beantwoording van de vragen

Instructieblad 1e Functionele Workshop

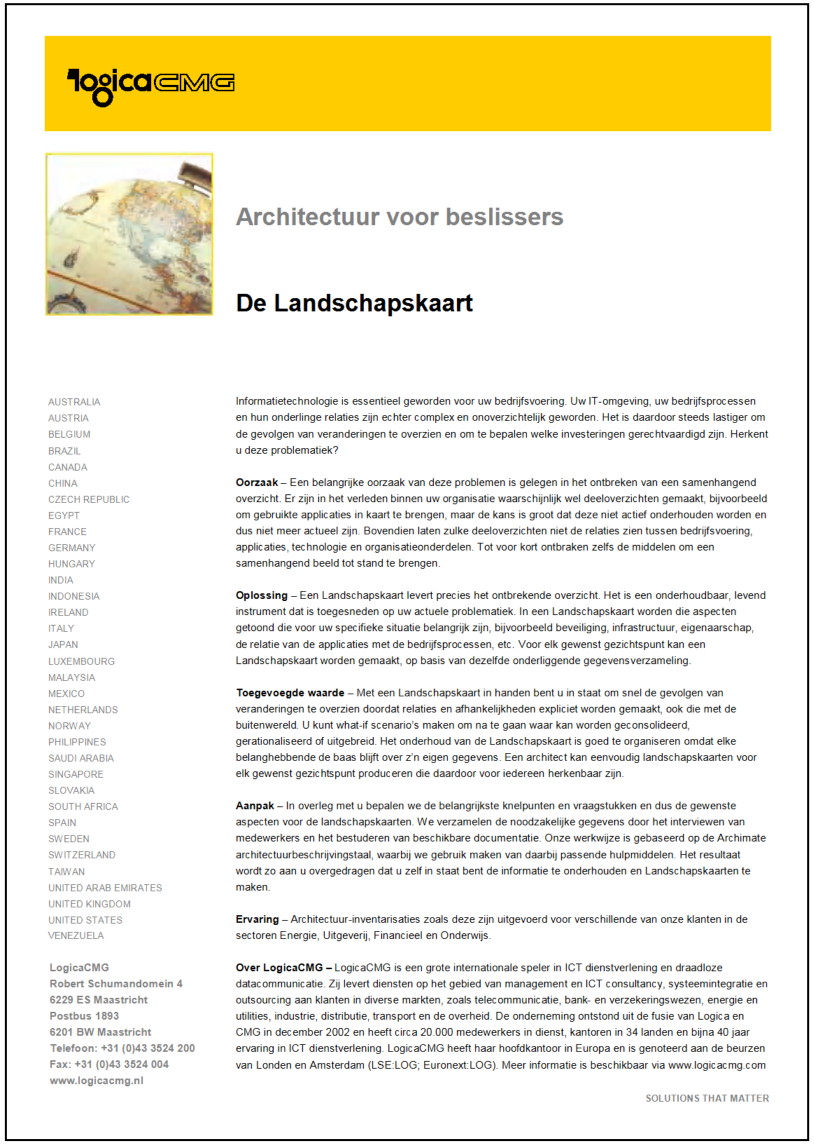


- **Landschapskaart 1 "Digitale ondersteuning bij processen"**
- **Landschapskaart 2 "Applicatieve coherentie"**
- **Landschapskaart 3a "Applicatiemodule bij de processtappen"**
- **Landschapskaart 3b "Applicatiefunctie bij de procedures"**
- **Landschapskaart 4 "Informatie en processen"**
- **Landschapskaart 5 "Systemen en eigenaar"**
- **Landschapskaart 6a "Systemen en infrastructuur" (systeem-server)**
- **Landschapskaart 6b "Database en infrastructuur" (db-server)**
- **Landschapskaart 7 "Informatie en systemen"**
- **Landschapskaart 8 "Informatie en infrastructuur" (db-informatie)**



	Informatie	Waarvandaan	Ondersteuning door	Informatie waarde
	Korte omschrijving van de gebruikte informatie.	*Waar haalt u de informatie vandaan.(met andere woorden wie en wat raadpleegt U).*	*Door welke systemen, documenten en of andere informatiebronnen wordt U ondersteund.*	*Wat is de informatiewaarde, is alles in samenhang gedocumenteerd, is het gefragmenteerd, is het volledig*
Opdracht	Bepaal de gewenste informatie bij het beantwoorden van het vraagstuk Gebruik alleen de termen van het denkkader	Niet invullen	Benoem de landschapskaart uit de lijst van landschapskaarten en wees vrij in het selecteren van een thema.	Wat ervaart U
Voorbeeld	Ik wil de informatie objecten in relatie tot systemen gebruiken en heb pijl 7 nodig.		Op de landschapskaart 7 “Informatie en Systemen” wil ik het betrokken database merk zien waar de objecten beheerd worden.	Met deze landschapskaart ben ik in staat om aan te tonen dat …. Ik voel me in mijn vraagstuk voldoende/onvoldoende ondersteund omdat ….

Appendix 7. Factsheet "The Landscape Map" (Dutch)



Architectuur voor beslissers

De Landschapskaart

AUSTRALIA
AUSTRIA
BELGIUM
BRAZIL
CANADA
CHINA
CZECH REPUBLIC
EGYPT
FRANCE
GERMANY
HUNGARY
INDIA
INDONESIA
IRELAND
ITALY
JAPAN
LUXEMBOURG
MALAYSIA
MEXICO
NETHERLANDS
NORWAY
PHILIPPINES
SAUDI ARABIA
SINGAPORE
SLOVAKIA
SOUTH AFRICA
SPAIN
SWEDEN
SWITZERLAND
TAIWAN
UNITED ARAB EMIRATES
UNITED KINGDOM
UNITED STATES
VENEZUELA

LogicaCMG
Robert Schumandomein 4
6229 ES Maastricht
Postbus 1893
6201 BW Maastricht
Telefoon: +31 (0)43 3524 200
Fax: +31 (0)43 3524 004
www.logicacmg.nl

Informatietechnologie is essentieel geworden voor uw bedrijfsvoering. Uw IT-omgeving, uw bedrijfsprocessen en hun onderlinge relaties zijn echter complex en onoverzichtelijk geworden. Het is daardoor steeds lastiger om de gevolgen van veranderingen te overzien en om te bepalen welke investeringen gerechtvaardigd zijn. Herkent u deze problematiek?

Oorzaak – Een belangrijke oorzaak van deze problemen is gelegen in het ontbreken van een samenhangend overzicht. Er zijn in het verleden binnen uw organisatie waarschijnlijk wel deeloverzichten gemaakt, bijvoorbeeld om gebruikte applicaties in kaart te brengen, maar de kans is groot dat deze niet actief onderhouden worden en dus niet meer actueel zijn. Bovendien laten zulke deeloverzichten niet de relaties zien tussen bedrijfsvoering, applicaties, technologie en organisatieonderdelen. Tot voor kort ontbraken zelfs de middelen om een samenhangend beeld tot stand te brengen.

Oplossing – Een Landschapskaart levert precies het ontbrekende overzicht. Het is een onderhoudbaar, levend instrument dat is toegesneden op uw actuele problematiek. In een Landschapskaart worden die aspecten getoond die voor uw specifieke situatie belangrijk zijn, bijvoorbeeld beveiliging, infrastructuur, eigenaarschap, de relatie van de applicaties met de bedrijfsprocessen, etc. Voor elk gewenst gezichtspunt kan een Landschapskaart worden gemaakt, op basis van dezelfde onderliggende gegevensverzameling.

Toegevoegde waarde – Met een Landschapskaart in handen bent u in staat om snel de gevolgen van veranderingen te overzien doordat relaties en afhankelijkheden expliciet worden gemaakt, ook die met de buitenwereld. U kunt what-if scenario's maken om na te gaan waar kan worden geconsolideerd, gerationaliseerd of uitgebreid. Het onderhoud van de Landschapskaart is goed te organiseren omdat elke belanghebbende de baas blijft over z'n eigen gegevens. Een architect kan eenvoudig landschapskaarten voor elk gewenst gezichtspunt produceren die daardoor voor iedereen herkenbaar zijn.

Aanpak – In overleg met u bepalen we de belangrijkste knelpunten en vraagstukken en dus de gewenste aspecten voor de landschapskaarten. We verzamelen de noodzakelijke gegevens door het interviewen van medewerkers en het bestuderen van beschikbare documentatie. Onze werkwijze is gebaseerd op de Archimate architectuurbeschrijvingstaal, waarbij we gebruik maken van daarbij passende hulpmiddelen. Het resultaat wordt zo aan u overgedragen dat u zelf in staat bent de informatie te onderhouden en Landschapskaarten te maken.

Ervaring – Architectuur-inventarisaties zoals deze zijn uitgevoerd voor verschillende van onze klanten in de sectoren Energie, Uitgeverij, Financieel en Onderwijs.

Over LogicaCMG – LogicaCMG is een grote internationale speler in ICT dienstverlening en draadloze datacommunicatie. Zij levert diensten op het gebied van management en ICT consultancy, systeemintegratie en outsourcing aan klanten in diverse markten, zoals telecommunicatie, bank- en verzekeringswezen, energie en utilities, industrie, distributie, transport en de overheid. De onderneming ontstond uit de fusie van Logica en CMG in december 2002 en heeft circa 20.000 medewerkers in dienst, kantoren in 34 landen en bijna 40 jaar ervaring in ICT dienstverlening. LogicaCMG heeft haar hoofdkantoor in Europa en is genoteerd aan de beurzen van Londen en Amsterdam (LSE:LOG; Euronext:LOG). Meer informatie is beschikbaar via www.logicacmg.com

SOLUTIONS THAT MATTER