Executable Enterprise Architecture Models: Enabling Business Analytics using the Monte Carlo Method

UNIVERSITY OF TWENTE.

Roland Smook

School of Management and Governance

University of Twente

This dissertation is submitted for the degree of

Master's of Industrial Engineering and Management

November 2014

Roland Smook

Student Number:	s1362275
e-mail:	rsmook@gmail.com
Degree:	Master of Science in Industrial Engineering and Management
Specialization:	Information Technology and Management
Faculty:	Management and Governance
	University of Twente
	Enschede
	The Netherlands

Examination Committee

1st Supervisor: Dr. Maria E. Iacob

e-mail:	m.e.iacob@utwente.nl
Department:	Industrial Engineering and Business Information Systems
Faculty:	Management and Governance
University:	University of Twente

2nd Supervisor: Ir. Henk Kroon

e-mail:	h.kroon@utwente.nl
Department:	Business Administration
Faculty:	Management and Governance
University:	University of Twente

External Supervisor: Dick Quartel

e-mail:	D.Quartel@bizzdesign.nl
Department:	R&D
Company:	BIZZdesign

To my parents, for all they have sacrificed without hesitation.
For all their love and support

God, for His unending grace and faithfulness.

Acknowledgements

My supervisors,	for all their guidance and insights.
My aunt Sieglinde,	for selflessly opening her home to me.
My friends Marta and Liezel,	for checking in, making time to listen, support in difficult times and laughter in the good times.

Abstract

Enterprise architectures (EA) are particularly useful at showing the various structural elements within an enterprises, and the complex net of connections between them. In support of them numerous frameworks exist which aid companies in designing, developing and managing EAs as well as the host of activities and standards present in those EA frameworks. However, EAs also have one stark shortcoming, namely their static nature prevents them from being used to predict the resultant dynamic behaviour of particular EA design choices. A limited umber of executable EA metamodels and approaches do exist however. In other words, models depicting all of the original elements of an EA model, but which are enriched with all the necessary properties for them to be used for simulation purposes.

Specifically focusing on business analytics, the research goal therefore is to *determine and demonstrate how enterprise architectures can be used as the foundation for business analytics.*

In light of this an artefact was developed which allows business analytics, which are based on the Monte Carlo method, to be executed, using EA models as their base.

To achieve this the Design Science Research Methodology was followed. Chapter 2 and 3 provide a literature study, setting the context within the EA and BA realms, motivating and identifying the exact problem the artefact had to answer.

Chapter 4 then defines the objectives of the artefact, which are to provide a method and means (implemented prototype) by which to execute Monte Carlo simulations within the EA modelling application Architect. Particular emphasis is given to user-friendly design which can be used generically with a host of different BAs.

Chapter 5 details the developed artefact, explaining how it is used by first integrating it with a users already existing analysis script, then specifying all random variables on the model by means of EA profiles. The third step considers the actual execution of the Monte Carlo simulation, before finally using various descriptive statistics and visual analyses to evaluate the results.

Two illustrative cases are presented in Chapter 6, both set within the banking

industry. The first considers a multi-period Net Present Value calculation, whilst the second considers the availability of an ATM service.

The evaluation of the artefact by means of three interviews is discussed in chapter 7. It was found that although there could be use for such an artefact, the entry-requirements are rather high as the user would have to have both the necessary know-how on how to use it, how to script his own analyses, know the Monte Carlo method, and lastly have the necessary information at hand to specify the random variables.

Chapter 8 provides a conclusion, specifically focusing on how the research objective was met, the limitations and contributions of the project. The project takes a new approach to executable EA models, which could be of use for further research.

Contents

Contents xi

List of Figures xv

List of Tables xvii

Nomenclature xix

1 Introduction **1**

1.1 Context & Motivation . . . 1
1.2 Research Goal and Objectives . . . 2
1.3 Research Methodology . . . 3
1.4 Scope & Assumptions . . . 5
1.5 Document Overview . . . 6

2 Enterprise Architecture Analysis **9**

2.1 Introduction . . . 9
2.2 Enterprise Architecture Fundamentals . . . 10
2.2.1 Purpose . . . 10
2.2.2 EA Today . . . 10
2.2.3 Beyond Static Models . . . 11
2.3 EA Analysis . . . 12
2.3.1 Classification Schemes . . . 12
2.3.2 Analytics Activities . . . 16
2.3.3 Relationships & Activity Types . . . 19
2.4 Executable Enterprise Architecture Models . . . 20
2.4.1 Definition . . . 20
2.4.2 EA Simulation . . . 21

2.5 Archimate 22
2.6 Conclusion to EA Analysis 22

3 Business Analytics 25
3.1 Introduction 25
3.2 Fundamentals 26
3.2.1 Analytics Competitors 28
3.2.2 Synergy between EA and BA 29
3.3 The Monte Carlo Method 30
3.3.1 Stochastic Models 30
3.3.2 Overview of Method 30
3.3.3 Histogram & Descriptive Statistics 31
3.3.4 Histogram 32
3.4 Conclusion to Business Analytics 33

4 Executable EA Models & the Monte Carlo Method 35
4.1 Introduction 35
4.2 Stakeholders 35
4.3 Stakeholder objectives 36
4.3.1 BIZZdesign 37
4.3.2 Advanced & Business Users 38
4.4 Stakeholder Requirements 38
4.5 Conclusion 39

5 The Artefact 41
5.1 Introduction 41
5.1.1 Satisfying Requirements 41
5.1.2 Modelling 42
5.1.3 Assumptions & Limitations 42
5.2 Develop & Integrate Analysis Script 43
5.2.1 Tagging Input Data 43
5.2.2 Indicating a ’Stop Event’ 44
5.2.3 Tagging Variables 45
5.2.4 Advanced 46
5.3 Specifying Random Variables 47
5.3.1 Specifying Random Variables 47

5.3.2 Discrete Random Variables . . . 49
5.4 Run Monte Carlo Simulation . . . 51
5.4.1 Overview . . . 51
5.4.2 Simulation Loop . . . 52
5.5 Analyse Results . . . 59
5.5.1 Histogram . . . 59
5.5.2 Other Graphs . . . 60
5.5.3 Export results . . . 60
5.6 Conclusion . . . 64

6 Artefact Demonstration **65**
6.1 Introduction . . . 65
6.2 Illustrative Case 1: Multi-period Financial Performance . . . 66
6.2.1 Case description . . . 66
6.2.2 Implementation . . . 70
6.2.3 Results Analysis . . . 71
6.3 Illustrative Case 2: Service Availability . . . 74
6.3.1 Case description . . . 74
6.3.2 Implementation . . . 75
6.3.3 Results analysis . . . 77
6.4 Conclusion . . . 78

7 Evaluation **81**
7.1 Introduction . . . 81
7.2 Method . . . 81
7.2.1 Evaluation Criteria . . . 81
7.2.2 Interviews . . . 83
7.3 Results . . . 84
7.3.1 Understandability & Effectiveness . . . 84
7.3.2 Ease of use, Simplicity, Elegance & Operationality . . . 84
7.3.3 Fidelity with Real World Phenomena & Generality . . . 85
7.4 Conclusion . . . 85

8 Conclusion **87**
8.1 Introduction . . . 87
8.2 Research Questions & Research Objective . . . 87

8.2.1 Research Question 1 . . . 87
8.2.2 Research Question 2 . . . 88
8.2.3 Research Question 3 . . . 89
8.2.4 Research Question 4 . . . 89
8.2.5 Research Objective . . . 89
8.3 Contributions . . . 90
8.3.1 Theoretical . . . 90
8.3.2 Practice . . . 90
8.4 Limitations . . . 90
8.5 Recommendations . . . 91
8.5.1 Theoretical . . . 91
8.5.2 Artefact . . . 92

References **95**

Appendix A Code **99**
A.1 Monte Carlo Script File . . . 99
A.2 Additions to User's Script . . . 111
A.3 Histogram . . . 112
A.4 Retrieve variable's simulation outcomes . . . 118
A.5 NPV Analysis . . . 120
A.6 Profile Definition (MonteCarlo.mpd) . . . 129
A.7 Translation file (MonteCarlo.mps) . . . 129

Appendix B Interview Questions **131**

Appendix C Interview Minutes **133**
C.1 Interview 1 . . . 133
C.2 Interview 2 . . . 134
C.3 Interview 3 . . . 134

List of Figures

1.1 Design Science Research Method by Peffers et al. (2007) 4
1.2 Document Overview 7

2.1 Analysis Dimensions (Jonkers et al., 2013) 13
2.2 Analysis and synthesis of EA with modelling Wan and Carlsson (2012). 17
2.3 The interrelationship of analysis activities according to Wan and Carlsson (2012). 20
2.4 ArchiMate metamodel as in Iacob and Jonkers (2006) 23

3.1 Nature of Decision Models according to Evans (2013) 27

4.1 Artefact Stakeholders 36

5.1 Executing the Analysis Method 42
5.2 Specifying a Random Variable Example 48
5.3 Random Variables Profile Attributes 48
5.4 Example use of the *Random Variables* profile 49
5.5 Random Variables Profile 50
5.6 Executing the Monte Carlo Simulation Action Diagram 53
5.7 Executing the Monte Carlo Simulation Context Level Data Flow Diagram 53
5.8 Executing the Monte Carlo Simulation Level 0 Data Flow Diagram . . 54
5.9 Process 4: Run Simulation Data Flow Diagram 57
5.10 Process 4.2: Generate, Assign & Save Stochastic Variables 58
5.11 Generate Histogram Activity Diagram 61
5.12 Generate Histogram Context Level Data Flow Diagram 62
5.13 Generate Histogram Level 0 Data Flow Diagram 63

6.1 Credit Card Service EA model 68
6.2 Bank Case EA Financials 71

6.3 Year 5 NPV Descriptive Statistics 72
6.4 NPV Histogram for Year 5 72
6.5 NPV Probability Chart for Year 5 72
6.6 Mean NPV per year 73
6.7 Yearly analysis 73
6.8 Mean Yearly Profit/Loss Descriptive Statistics 73
6.9 Mean Break-Even Analysis Statistics 74
6.10 Change in values for Year 5 NPV 74
6.11 ATM fault tree 75
6.12 Cash withdrawal service EA (Buschle et al., 2013). 76
6.13 Redundancy relationship 77
6.14 Cash withdrawal service availability 78

List of Tables

2.1 EA Analysis Classification Scheme according to Buckl et al. (2009a) . . 16

3.1 Common Applications of Business Analytics (Davenport, 2006) 27

6.1 Model assumptions for Credit Card Industry Business Case (Paisittanand and Olson, 2006). 69

6.2 Stochastic Nature of Model Variables for the Credit Card Service (Paisittanand and Olson, 2006). 70

7.1 Evaluation criteria for Design Science Research artefacts (March and Smith, 1995) 82

Nomenclature

Roman Symbols

$\overline{x}$ Sample mean

Greek Symbols

$\hat{\sigma}$ Sample standard deviation

Acronyms / Abbreviations

AD Activity Diagram

BA Business Analytics

DFD Data Flow Diagram

DSRM Design Science Research Methodology

EA Enterprise Architectures

IQR Interquartile Range

UML Unified Modelling Language

Chapter 1

Introduction

1.1 Context & Motivation

Every business needs to be properly managed in order to respond to shifting business environments and changing market demands in a manner that will ensure its own sustainability. However, even if the desired business behaviour is identified, the necessary changes required to achieve such behaviour can be difficult to identify. This difficulty can in part be ascribed to the complex behaviour resultant of the host of relationships that exist between the various component parts of a business. The inter-relationships cause complex behaviour to occur wherein it is difficult to distinguish between effect and cause. The enterprise architecture (EA) field has progressively advanced to aid managers in the design, decomposition and communication of enterprise architectures, but little has been done as far as developing tools to predict and evaluate the hypothetical behaviour of proposed architectures (Glazner, 2009).

Predicting and evaluating enterprise behaviour can be done in a number of ways. In the case of Glazner (2009) for example, an extensive hybrid simulation model was developed . However, other less involving methods also exists, such as those of business analytics (BA) . With business analytics the objective is to develop a decision model which will result in some measure of performance or behaviour, given certain input variables (Evans, 2013). Although business analytics are not limited to predictive analytics, a very similar argument can be made for their development as extensions to the existing field of enterprise architectures.

Developing and operationalizing decision models required by business analytics can be a complex, time-consuming affairs. Business users typically do not have the expertise to complete analytics by themselves. They either require an analytics expert, or at least an analytics application the business user can work with. When an analytics expert is being used, a lot of to and fro is required between the business user and the analytics expert, until the expert has produced the results usable to the business user. The business user after all, is typically not interested in the hard statistical data for instance, but requires the results in a format which he can use to make decisions. The analytics expert again, needs to decipher the needs of the business user, develop the necessary models and operationalize them. This to and fro takes up time, and is complicated by the complex inter-relationships existing within enterprises domains, which the models need to account for and within which they need to be operationalised. Getting the necessary results can therefore devour a lot of resources, and take longer to accomplish than what the business user has time for when making a decision. (Kohavi et al., 2002)

As discussed though, enterprise architectures already serve to make complex structures and relationships clear. An opportunity therefore exists for organisations who have already invested in an enterprise architecture, as they posses the ideal springboard from which to develop and operationalize business analysis efforts.

This research project therefore wishes to answer whether, and to what effect, business analytics can be enriched by developing them as an extension to enterprise architectures. If this question can be answered satisfactorily by means of a demonstrable artefact, the theoretical community and corporate practices will gain new insights into the possible ways through which to develop usable business analytics which take less time to formulate and see results from, as well as being easier to comprehend by business users.

1.2 Research Goal and Objectives

Given the previously described research opportunity, the following research goal is formalised:

> *Determine and demonstrate how enterprise architectures can be used as the foundation for business analytics.*

In order to fulfil the research goal, four primary research questions are identified. The first of these is primarily be answered by means of a literature study; the two sub-questions serving to fragment and pinpoint specific issues which need to be addressed. The remainder of the research questions correspond to different phases of the Design

Science Research Methodology discussed in the next section.

1. Can enterprise architecture models be used as the foundation for Monte Carlo simulation based business analytics, and if so what problem would this aim to solve?

    (a) What is understood under the term *enterprise architecture analytics*, and what is its purpose? Are there any existing standard guidelines or methods specifically designed to develop simulation models of enterprise architectures models?

    (b) What are business analytics? Is there a synergy between business analytics and enterprise architectures which would encourage and facilitate further investigation into this topic? How are Monte Carlo simulations used for business analytics?

2. Given these findings, what would be the requirements of a technical artefact, designed to enabled Monte Carlo simulations, based on using enterprise architecture models?

3. How could an artefact be developed an implemented to meet these requirements within the enterprise architecture modelling tool, Architect?

4. Does artefact tool fulfil its original purpose, is it user-friendly, and useful within the business realm?

1.3 Research Methodology

A significant part of the project will be dedicated to designing and developing a technical artefact. Each phase of the development of the artefact, from problem definition through to evaluation will be based on theoretical underpinnings. With this in mind the *Design Science Research Methodology* by Peffers et al. (2007) was identified as a suitable research methodology to guide the execution of this project. An outline of the phases can be seen in Figure 1.1, the execution of the phases are described in more detail next.

Identify Problem and Motivate The first research question serves to provide further background information and motivation for this particular research effort.

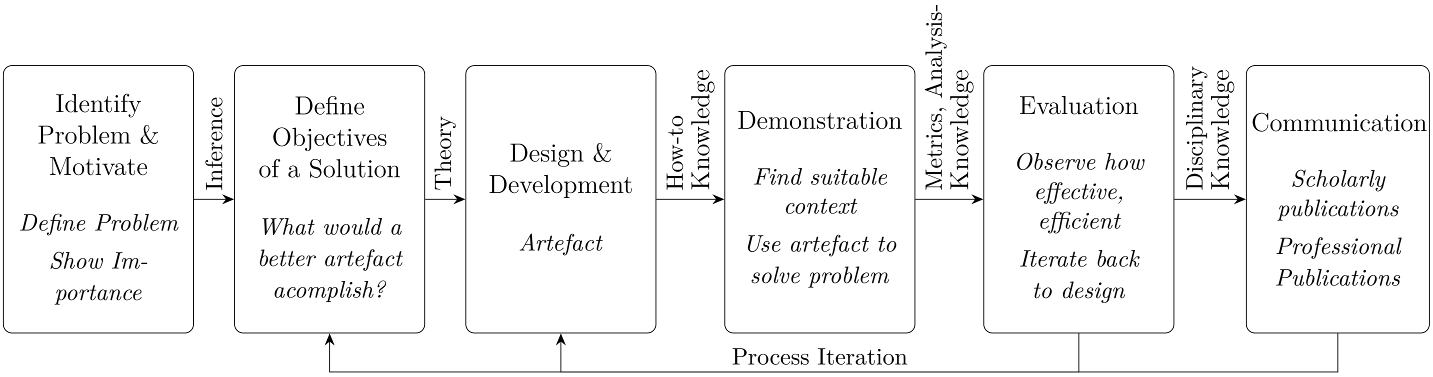


Fig. 1.1 Design Science Research Method by Peffers et al. (2007)

From the findings, a specific problem is identified which the artefact should solve. The principles used to solve this problem are also identified in this phase.

Define Objectives of a Solution The general objectives of this research project have already been identified. In conjunction to these, more specific objectives will be identified for the artefact, which should be fulfilled in order to solve the problem identified in the previous phase. These objectives are based on the needs of the stakeholders, for whom the artefact is developed. Based on these objectives, the required functionality of the artefact is identified. This phase then answers the second research question.

Design and Development The aim of the artefact is to meet the requirements set out in the previous phase, by following the theoretical principles identified in the first phase. In so doing the third research question is answered.

Demonstration Two illustrative cases are used to demonstrate the artefact, which are based on examples found in literature of cases in the real world. The objective here is to demonstrate that the artefact accomplishes its original objectives, and fulfils its requirements.

Evaluation The two illustrative cases are demonstrated to three individuals in order to evaluate the artefact. The objective here is to assess whether the artefact accomplishes its original objectives and fulfils its requirements in an easy-to-understand manner. This then answers the fourth and final research question.

Communication The thesis and colloquium will serve as the means of documenting and communicating the purpose, design and results of the artefact.

Seeing as this is an iterative approach, the different phases were each considered a few times, based on results attained from later phases. Particularly the requirements set out for the artefact, and its development were done in an agile fashion, dependent on the feed-back attained from early prototypes.

1.4 Scope & Assumptions

Although more assumptions and limitations are identified during the remainder of the chapters, the following points are important to understand from the beginning.

Firstly, even though a common set of business analytics do exist, it is by no means a fixed set. It is a broad and undefined range, the exact choice and operationalization of any analytics by an enterprise will differ from one industry to another, and even one company to another within the same industry. These choices depend on the exact objectives and capabilities of the enterprise in question (Davenport, 2006).

Certain types of analytics, will also most likely be executed in purpose built applications. Many business intelligence applications exist for instance, specifically to deal with descriptive analytics requiring the analysis of huge data sets. Other analytics though, such as those used for more prescriptive purposes, are far more likely to be solved in applications such as Microsoft Excel in which it is relatively easy to quickly develop new, one-off analysis models. Therefore the choice of analytics to be developed for this artefact is by no means an obvious one, and only covers a small range of analytics.

Secondly, during the completion of this project a varied number of sources were used to demonstrate the importance of business analytics and enterprise architecture analysis. Examples for business analytics include:

- The ample number of purpose built business analytics applications available on the market today, such as those provided by SAS [1], rapidminer [2] and Knime [3]. Then there are of course the solutions offered by big corporate players such as IBM's host of solutions for business analytics [4]. Of the numerous other players in this field, these are the four leaders therein according to Gartner's 2014 *Magic Quadrant for Advanced Analytics Platforms* report by Herschel et al. (2014).

[1] http://www.sas.com/
[2] http://rapidminer.com/
[3] https://www.knime.org/
[4] http://www.ibm.com/services/us/gbs/business-analytics/

- Predictions from corporations conducting research into business trends, regarding the growth, future importance and direction of the field. Examples include Gartner, IBM, KPMG and Accenture, to name but a few (Accenture, 2013; IBM, 2012; KPMG, 2013; Schulte et al., 2013).

- Academic research into the field, explaining its importance in establishing a competitive advantage (Davenport, 2006; Davenport and Harris, 2013; Sprongl, 2013).

The sources demonstrating the importance and potential of enterprise architecture analysis are mostly limited to academic research at this point in time, with those dealing with quantitative analysis (as in the case of business analytics) being even fewer. However, in answering the first research question the validity of this particular avenue of research is established. In the case of a positive evaluation of the artefact, it will serve as a further demonstration of the importance and potential of such work.

It is nonetheless, also recognised that not all ideas seen as worthwhile pursuits in the eyes of academic research will necessarily be end up being adopted in business practice. In order to maximise the possibility of acceptance, the objectives identified during step 2 of the design science research methodology, and the subsequent evaluation in step 5 will emphasise points which are important to business users (the stakeholders). Having made that statement of intent, it is nevertheless not the primary objective of this research to ensure that the artefact (or that which it represents) is immediately accepted by business users. It is after all not the case that every good, or worthwhile research outcome is implemented into practice. This does not however, diminish its importance, or its merit.

1.5 Document Overview

Sub-research questions *a* and *b* have a chapter dedicated to them each, serving to identify the primary pillars of theoretical knowledge, research directions and developments within this field.

Chapter 2 therefore briefly considers the basic foundations of enterprise architectures, before moving on to consider the various types of enterprise architecture analysis as these might provide some necessary insights when specifically considering enterprise architecture business analytics. In conclusion *RQ 1-a* is answered.

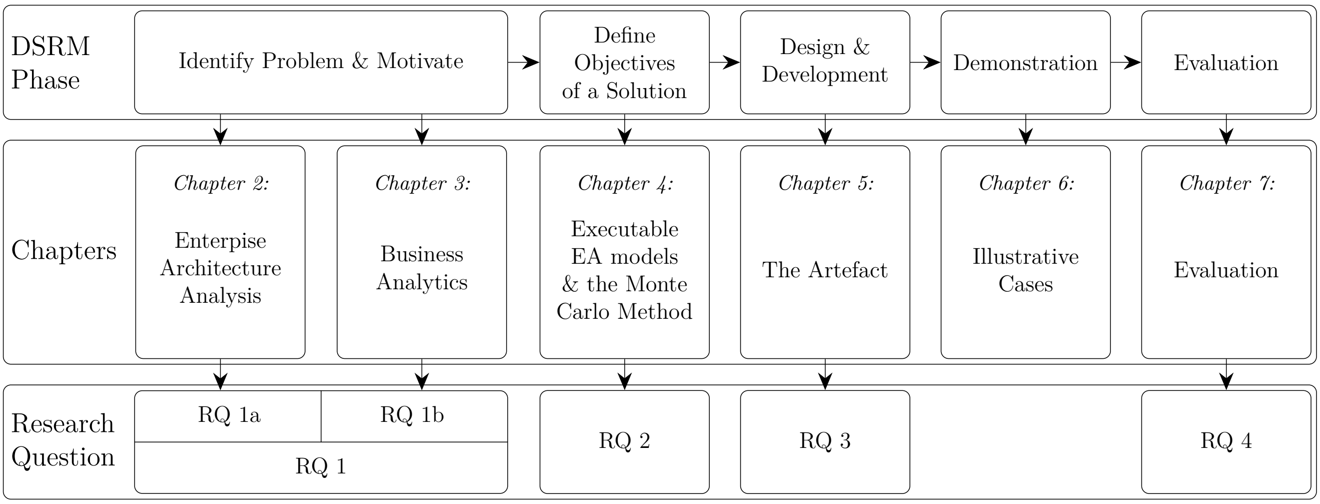


Fig. 1.2 Document Overview

Chapter 3 moves onto *RQ 1-b*, investigating the importance of business analytics to today's enterprise environment, and what synergies exist between these two fields which could be used to form the bedrock for this project from a theoretical perspective, and more practically speaking for the artefact. In its conclusion the first research question is answered as a whole.

Based on the findings of the first research question, chapter 4 serves to answer research question two. The purpose here is to be very specific about the objectives for the artefact, as inferred from theory and stakeholder analysis.

Having identified the objectives and subsequent requirements for the artefact, chapter 5 explains the design and development of the artefact, thereby answering the third research question.

Chapter 6 explains the context of two illustrative cases and how the artefact was used to solve the business problems in them. These cases are used to demonstrate the functionality provided by the artefact, which is used to for its evaluation summarised in chapter 7, thereby satisfying the fourth and final research objective.

Chapter 8 finally concludes the thesis, considering the extent to which the research goal has been fulfilled, the contributions this research project has made and suggestions for further work to be done.

Chapter 2

Enterprise Architecture Analysis

2.1 Introduction

In order to complete the first phase of the DSRM, namely to identify the problem and motivate its importance, a bit of background and contextual information is required first. This chapter specifically considers enterprise architectures (EAs), with the goal of answering research question *RQ 1-a* by means of a literature study. To do so, an investigation into the research progress and directions taken by other researchers specifically considering the topic of interest to this project, also has to be made. To reiterate, the first sub-research question reads:

> What is understood under the term *enterprise architecture analytics*, and what is its purpose? Are there any existing standard guidelines or methods specifically designed to develop simulation models of enterprise architectures models?

In the first section some of the fundamentals of enterprise architectures are discussed. Following this, section 2.3 discusses various EA analytics by means of three classification schemes. The first two discuss the various types of EA analytics which are possible, whilst the last classifies and interrelates six analysis activities which can take be undertaken. The classification schemes serve to introduce structure into the delineated the field of EA analytics. Following this, the concept of executable enterprise architecture models is introduced. Lastly, a conclusion is provided to the chapter, considering how the research question can be answered by the findings.

2.2 Enterprise Architecture Fundamentals

2.2.1 Purpose

A number of modelling methods and languages exist to model specific parts of an organisations' operations and structure. Business processes can for instance be modelled with Business Process Modelling Notation (BPMN); information systems again have a range of methods depending on what is being represented. The Unified Modelling Language alone consists of 13 different diagram types, designed to meet the requirements of different user groups, be it the coding expert, or information system's (IS) designer (Object Management Group, 2011, 2013). An enterprise's information and technology services, processes and infrastructure can therefore be modelled independently (Iacob and Jonkers, 2006; The Open Group, 2011).

Making and analysing the impact of changes on any one such part of the enterprise is facilitated by these models, allowing the architects and developers of these parts to design and optimise their local operations. However, an enterprise's performance can not be optimised at the global level merely by optimising the individual parts. Each part is related to, and dependent on the other. Therefore the design of any one part needs to take the requirements and functioning of the others into account if the organisation wants to be able to realise efficiency and effectiveness on a global level.

Therein lies the need for enterprise architectures. By capturing the complexity and relationships between the various parts of an enterprise, new organisational efficiencies can be attained by integrating processes into a coherent unit that is able to respond to change and deliver the business strategy. Enterprise architectures by no means attempt to replace the languages and methods used to describe the various parts in very fine detail. Instead it is designed to model these parts at an less detailed, abstracted level. The emphasis here, after all, is the global enterprise and the inter-relationships between its parts. (Jonkers et al., 2013; The Open Group, 2011)

2.2.2 EA Today

Although EAs were first introduced through the Zachman Framework for information system architecture in 1987, it has matured into a cross-disciplinary field important to every aspect of an organisation (Buckl et al., 2008; Zachman, 1987). Nowadays numerous EA frameworks exist to aid an organisations align their activities. Besides the Zachman Framework, numerous others have been developed. Examples include

TOGAF (The Open Group Architecture Framework), DODAF (U.S. Department of Defense Architecture Framework Working Group), FEAF (U.S. Federal Enterprise Architecture Framework), TEAF (U.S. Treasury Enterprise Architecture Framework), and more bespoke examples developed for specific enterprises (United States Department of Defense, 2010; U.S. Office of Management et al., 2007). (Many of these in turn have a number of variations designed for specific industries, or designed by specific groups.) These developments efforts were initiated from dissimilar starting positions, being driven by their own specific motivations and adopting different definitions of EA along the way.

Although no single, universally accepted definition of EA, or EA management subsists, for the purposes of this project the formal definition used by The Open Group Architecture Framework (TOGAF) is used (Buckl et al., 2008; Matthes, 2008). Namely, enterprise architecture is the methods, languages, frameworks and tools used to develop and represent the enterprise's architecture (The Open Group, 2011).

As the definition indicates, business areas such process management and also higher level concepts such as strategy management can now also form part EA initiatives (Wan and Carlsson, 2012). In addition to the basic frameworks, EAs typically also propose some form or another EA life-cycle management processes. (Wan and Carlsson, 2012)

Though a corporation, or an organisation as a whole is sometimes the enterprise in question, the term in this context also refers to anything from a single business domain to a collection of organisations in a cross-enterprise initiative. In short, as defined by The Open Group (2011, pg. 5), *"any collection of organisations with a common set of goals"* can be defined as an enterprise. Irrespectively of whether the enterprise in question constitutes the entire organisation, or a small subset thereof, it will involve multiple systems and functional groups.

2.2.3 Beyond Static Models

Although EA models provide enterprise architects with a holistic view of an enterprise and help their IT, business alignment and evolution efforts, static models are not sufficient means in themselves to provide decision makers with the information required to make informed business decisions . To assess the current as-is architecture, as well as anticipate the properties, behaviour and performance of any future architectures, requires methods of analyses which extend and enhance EAs beyond mere static representations. A number of such EA analyses models, methods and techniques exist for a variety of applications, however this is still a maturing field of research wherein a number of avenues remain to be explored. (Buckl et al., 2009b; Jonkers et al., 2013)

The purpose and various means of EA analysis are discussed next.

2.3 EA Analysis

A number of vendors provide tools with which architectures can be modelled in accordance to the guidelines set out in one or other architecture framework. These however, are mostly completely static, and are therefore unable to capture enterprise behaviour (Glazner, 2009). These models have typically only been used to model the structure and relationships of any enterprise. These models, tend to support the implementation of new systems, and can not describe what bearings the architecture of the enterprise will have on the actual performance of the enterprise.

According to Wan and Carlsson (2012) EA analysis is of critical importance if an organisation is to successfully leverage an EA:

1. Analytics brings new insights of an organisation and its architecture to light. By gathering and analysing enterprise data, any real or potential deficiencies can be identified. The typical EA artefacts such as models do not always show these weaknesses and gaps.

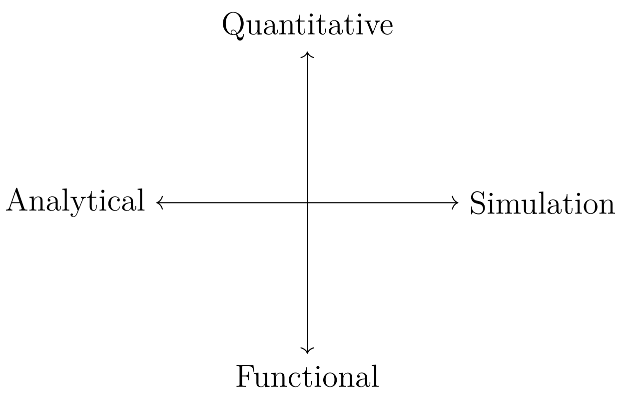
2. EA initiatives typically occur at a large scale and are inherently complex and risky. To reduce the uncertainty associated with architectural design decisions, in-depth analysis can provide decision makers with the necessary information to make informed choices.

2.3.1 Classification Schemes

Jonkers et al. (2013) classifies such analyses into four categories according to two dimensions. The first distinguishes the type of inputs/results of the analysis, and the second the technique used for analysis. When considering the inputs and results of an analysis, a distinction is made between functional and quantitative analysis.

Although the classification scheme above provides a useful basic introduction to the type and range of analysis that are possible, it does not quite reveal the full range and intricacies of EA analysis that are possible. For this purpose, the work of Buckl et al. (2009b) is considered next which both proposes a detailed classification scheme, and

Fig. 2.1 Analysis Dimensions (Jonkers et al., 2013)



the focus areas of most EA analysis in research. Whereas approaches to EA analyses are typically largely isolated from one-another, the classification scheme is presented as a stepping stone towards the development of a conceptual framework wherein the various analysis techniques can be integrated and eventually create a single approach to EA analysis. Whereas Jonkers et al. (2013) only refers to two classifiers, Buckl et al. (2009b) classifies EA analysis according to five different dimensions. An overview of the dimensions and their categories is provided in Figure 2.1, and described in further detail here:

Body of Analysis Where Jonkers et al. (2013) considers the input and results of analyses under one dimension, Buckl et al. (2009b) distinguishes between the *body of the analysis*, and the *analysis concern.* The body of analysis speaks to the type of result the analysis yields, whilst analysis concern speaks to whether or not the analysis concerns itself with the delivery of an organisational function. The body of analysis can either be of the *structure, behaviour statistics* or *dynamic behaviour.* Due to the structural complexity of an enterprise system, structural analyses are required to deal with this complexity, and this is where research on EA analysis has primarily focused. Such structurally complex system also exhibits complex behaviour. By measuring, aggregating and statistically analysing information from dynamic systems, insight can be gained into specific behavioural attributes dependent on those systems. On the other hand, the dynamic behaviour itself, such as a time-series of values, might be of interest in order to, for instance understand how one system event impacts another over time.

Time Reference An EA can either model a current enterprise, or that of a planned architecture. The advantage of *ex-post* analysis is that analyses rely on measurements taken from an active organisation and with certainty of structural composition. *Ex-ante* analyses, on the other hand, need to rely on predictions and account for the uncertainty of the final architecture.

When classifying EA analyses and the body of analysis is the structure of the EA, the question of time does not play a significant role as the method of analysis does not change. Similarly, when behavioural analyses are concerned, most are applicable in both time dimensions, however research in this field primarily focuses on demonstrating ex-post analysis in research, without explicitly considering the uncertainties introduced when considering ex-ante analyses.

Analysis Technique Where Jonkers et al. (2013) distinguishes between simulation and analytical approaches to analyses, Buckl et al. (2009b) distinguishes between expert-based, rule-based and indicator-based analysis. *Expert-based* analysis are the least formal, most flexible, but also most time-consuming of methods. By making use of EA views and relying on their experience and expertise, experts can provide anything ranging from concrete advice, to abstract suggestions and ideas. *Rule-based* analysis depends on formalised, pre-defined guidelines for choice architectural configurations prescribing both desirable and undesirable conventions. An analyses therefore either detects the manifestation or absence of select architectural patterns and can additionally be automated. *Indicator-based* analysis is the most formalised of techniques, delivering the most explicit and immediately interpretable of results. Indicators are extracted by quantitatively assessing properties of the architecture, typically computed from values of observable architectural properties (for example: complexity). Such indicators provide insight on very specific architectural attributes, but have to be interpreted with care as, for instance, they are always based on circumstantial assumptions.

As the rule-based and indicator-based analysis are very distinct concepts, identifying them in practice is easily accomplished, and no approaches simultaneously incorporate both. However, due to expert-based analysis' definition, it is not always easily delineated from other characteristics provided by an analysis' description.

The distinction between simulation and analytical techniques is not deemed important enough by Buckl et al. (2009b) to incorporate as a separate dimension into the classification scheme.

Analysis Concern As described in under *body of analysis*, Jonkers et al. (2013) distinguishes between functional and quantitative analysis.Buckl et al. (2009b) follows a very similar trend, however distinguishes between *functional* and more broadly defined *non-functional* analysis concern. Organisations exist, and are architected, to execute certain enterprise functions, such as marketing or production. Functional analysis then, aims to assess the enterprise system in its execution of these functional requirements as articulated in its architecture. In comparison an enterprise system's non-functional properties, such as throughput rate or processing time, are also of interest, especially if alternative architectures are being compared. Buckl et al. (2009b) argues that non-functional analysis do not-necessarily have to only be analysed quantitatively (for instance in the case of security matters), and that the term *quantitative analysis* is therefore inappropriate.

Non-functional, particularly economic concerns are the primary focus of EA analyses. This is likely due to the fact that functional facets of an EA are commonly difficult to define, which in turn makes highly structured analyses challenging. However some decidedly generic, mostly expert-based techniques can be applied to either of the concerns.

Self-Referentiality As previously discussed and outlined in figure **??**, the field of enterprise architecture not only considers the actual structure of the enterprise architecture, but a host of enabling and supporting methods, tools and theories. With this in mind, an enterprise's architecture does not merely contain the architecture landscapes, but also the blueprints and strategies guiding the design and management of those landscapes (including EA analysis efforts). As such EAs' are self-referential, in that one part refers to, and is dependent on another within the same architecture. In light of this architecture analysis can incorporate the analysis of one or more higher levels of referentiality contained in an EA. Most analysis are straight-forward and therefore consider *none* of the architectural components underlying EA management. In a *single-level* analysis, one level of self-referentiality would be considered, i.e the EA management activities contained within the architecture. Beyond this a *multi-level* analysis would then also incorporate meta-EA management activities (such as the governance of EA management activities) in its assessment. One of the possible benefits of including extended architectural components to analyse self-referentiality is to gain insight into the *emergent* behaviour of enterprise systems.

EA analysis have mostly not considered self-referentiality, one reason being that as described above, the function of EA management has been maturing itself still until rather recently. Before EA analyses can be developed to assess self-referentiality, an established function needs to exist, so as to know how to design and implement such an analysis. Related to the this subject of the development of the EA management field, are EA maturity models which have been receiving an increasing amount of attention in recent years.

Table 2.1 EA Analysis Classification Scheme according to Buckl et al. (2009a)

Classifier	Dimensions
Body of Analysis	structure, behaviour statistics, dynamic behaviour
Time Reference	ex-post, ex-ante
Analysis Technique	expert-based, rule-based, indicator-based
Analysis Concern	functional, non-functional
Self-Referentiality	none, single-level, multi-level

2.3.2 Analytics Activities

The understanding of the term *enterprise architecture analysis* differs in various contexts as it is associated with a variety of application areas. No single, clearly delineated definition has been formalised and most literature on the subject remain vague and ambiguous on the term. To demonstrate just how broad a range of understandings are associated with the term, Wan and Carlsson (2012) identifies, describes and relates a set of six analysis activities. By characterizing, classifying, and differentiate the activities, the field is explored.

Analysis are not only of interest in a wide range of different industries, organisations and contexts, but also in different stage of the EA process. In other words, analysis might be of interest as part of a business case for which no enterprise exists yet. Or on the opposite end of the spectrum, it could form part of a long standing organisations initiative to continuously monitor performance using various metrics and indicators. To clarify the concept of EA analysis, Wan and Carlsson (2012) identifies six EA analysis activities

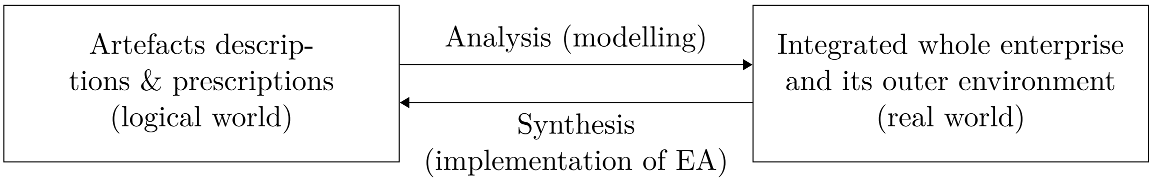


Fig. 2.2 Analysis and synthesis of EA with modelling Wan and Carlsson (2012).

I. System Thinking One of the primary purposes of EAs is to identify how the various components from different domains fit together and interoperate. Architects often know what the objective resultant behaviour is, and wish to determine how best to achieve this behaviour by considering the components involved in realising this behaviour. When considering EA from a systems perspective then, two types analysis are of interest. Firstly the emergent behaviour of the system might be of interest. The complexity in systems leads to a complex set of interactions, the outcome of which may be difficult to predict. Or given the behaviour of a system, the cause for that behaviour may be of interest, which requires that the components involved in producing the behaviour, and the interactions between them are studies in detail.

II. Modelling Another form of analysis required during the enterprise architectures development and evaluation process is that of assessing the enterprise system, as well as the external environment in which it is located in order to produce the necessary artefacts describing current as-is architectures, as well to-be design descriptions (see Figure 2.2). This process is not straight-forward though and requires a significant amount of work. The modellers need to make decisions in an attempt to reduce uncertainty. The design process attempts to clarify the state or to-be state of things, however details are often hard to come by, or no guidelines exists on how to best design solutions. Also, vast amounts of information which have no clear boundaries or inherent importance indicators need to be sifted through in order to determine which information is important to specific stakeholders, and how this information can be encapsulated and represented in a simplified form. The real-world can often be represented in multiple ways, requiring the modellers to rely on their experience, creativity and expert knowledge.

III. Measuring As with any change initiatives, EA initiatives are risky by nature.

In order to reduce this risk as far as possible, two types of measurement activities can take place. Firstly, tools and methods exist to analyse or predict EA properties and behaviour. These measurements from the real world, or based on models thereof, such as in the case of predictive measurements. These measurements for instance can serve to evaluate the EA. Secondly, the models used for these measurements, and the process used to develop them, can be also be assessed. This is necessary to ensure that the measurements, or predictions they yield are accurate representations of the properties of interest. Such analysis are for instance required for verification and validation purposes.

IV. Satisfying EA initiatives typically involve a number of stakeholders with different requirements. Although these requirements can at times counter each other, the design process ultimately needs to lead to an EA with which all stakeholders are these satisfied. EA design is therefore in effect a multi-criteria problem and the design process yields alternatives which architects need to analyse to check whether, and to what extent they satisfy stakeholders' various requirements. During this iterative design process three activities aid the architects in finding the most suitable of solutions. Firstly, during *sensitivity analysis* the affect of changing inputs on the resultant system properties and outputs are analysed. This analysis is performed to account for uncertainty inherent in any future estimates. Secondly, *trade-off analysis* is performed to compare how alternatives meet different requirements. Lastly, *revision and solutions formulation* are important considerations when attempting to find a satisfactory EA design. In order to reach a compromise between stakeholders, EA designs are revised until consensus is eventually reached. Then, looking beyond the design stage of EA, the chosen architecture also determines the enterprise changes that need to be made. The resultant change management plans may in turn influence the design of the EA.

V. Requirements Analysis Here both the requirements of the EA, and that of the analyses are of interest. Firstly, before any EA designs can be realised, the requirements for the EA need to be determined. These are usually derived after the vision, strategy and goals for the EA have been determined first and should account for the context within which it is being designed. Secondly, any analyses should only occur if a specific need exists. This need will be determinant in the choice of method and approach to the analysis.

VI. Comparing Alternatives Due to the multi-criteria nature of EA design, a number of alternatives are likely to be found which satisfy the requirements of the stakeholders. A final selection then needs to be made between these. In order to make an informed business decisions, the decision maker needs to compare the advantages and disadvantages of each alternative according to a number of dimensions. The modelling, measuring and satisfying activities can be used to form a complete assessment to use as a starting point for comparison. However, other methods, such as sensitivity analysis can be revisited to gain further insight.

2.3.3 Relationships & Activity Types

Based on the interrelationships between the six activities, they can be grouped into three levels. The first level is the *fundamental level* as the two activities therein, *system thinking* and *comparing with requirements*, form the elementary building blocks of EA analysis. Following this, the *main level*, includes the *modelling*, *measuring* and *satisfying* activities which are principal, or common to EA analysis. The ultimate end goal of any analysis is to provide useful insights which can be used in the decision making process. This is finally embodied in the *comparing alternatives* activity which and is therefore in a level of its own, the *decision-oriented level*.

Simplistically seen, the directions of the arrows in Figure 2.3 indicate which activities directly support others. This has two dimensions of significance. EA artefacts, such as models etc. are an integral part to any EA initiative, and so too artefacts are developed or used by each of the six activities. The artefacts produced by one activity are used as input by another, the arrows are therefore analogous to information flows. Information flows in turn are indicative of a temporal process. More specifically, the arrows indicate the temporal sequence of the activities which underlie a continuous EA management process supporting architecture evolution.

Although the six EA analyses activities discussed here are a robust depiction for any type of analysis, analysis in practice are usually specifically developed for very specific application scenarios. As such they do not propose methods which can used to apply their analysis to a range of different areas and contexts. These analyses therefore typically only incorporate a subset of the six activities discussed by Wan and Carlsson (2012). Although no attempts have been made as of yet to use these six activities and their mapping to develop a concrete EA analysis, they provide a useful starting point for such an attempt. It is sufficiently abstract and also comprehensive,

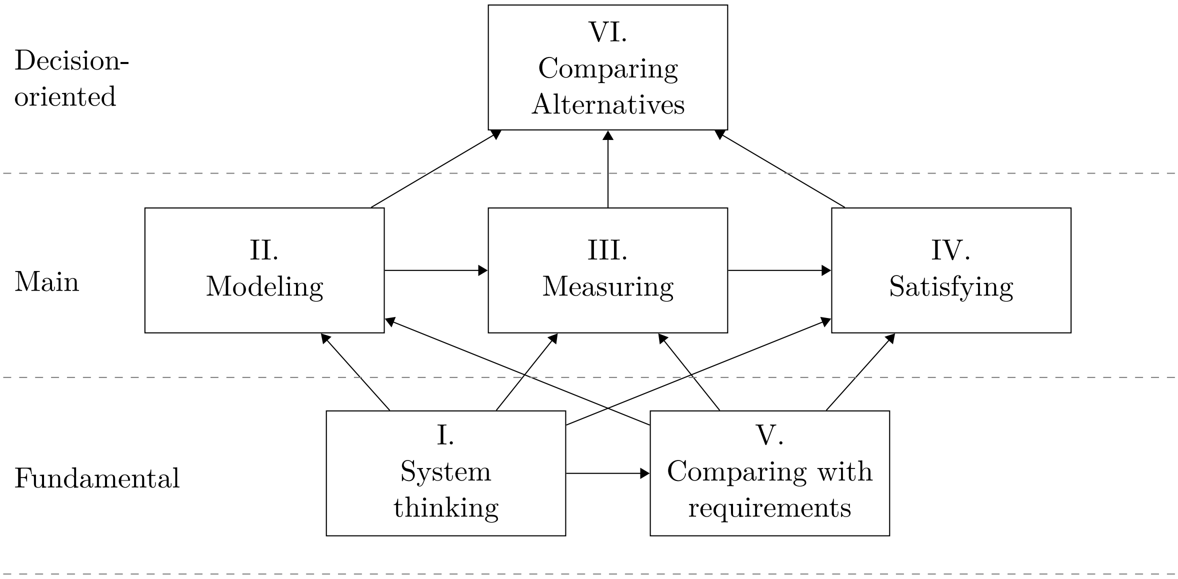


Fig. 2.3 The interrelationship of analysis activities according to Wan and Carlsson (2012).

including a broad range of concepts from EA practice, so as to encourage and support a theoretically vigorous development process.

2.4 Executable Enterprise Architecture Models

2.4.1 Definition

If we are interested in predicting (ex-post) the dynamic behaviour according to select indicators, various simulation techniques might be used to do so, which is where the concept of *executable EA models* can be introduced. To start, Bagnulo and Harvey (2008) concept of executable architectures (as cited in Ludwig et al. (2011)) is introduced (not specific to *enterprise* architectures). They identify them as *"the dynamic model of the behaviour of a system where the architecture elements are uniquely identified, consistently used and structured in a way to enable their simulation"*. In other words, executable architectures model dynamic systems, such that they can be simulated, but the elements of the original architectural model are uniquely identified. They therefore provide the link between architecture models to executable simulations.

When speaking of *executable enterprise architecture models* then, the architecture of interest is naturally that of an enterprise, and it is this EA model which is executable.

In other words, EA models which have been enriched with dynamic elements, which can be simulated in order to infer run-time characteristics of the enterprise (Banks et al. (2009) as cited in Manzur et al. (2012)).

2.4.2 EA Simulation

The development of such executable EA models has peaked the interest of a few researchers, some of these efforts are described here. Manzur et al. (2012) developed two metamodels, as part of a model-based platform in which EAs can be modelled (though only those elements of interest to the simulation) and subsequently simulated. The platform allows a variety of dynamic elements to be specified for the different structural elements of the EA. Following this, different scenarios can be defined, then executed in order to observe the run-time behaviour and calculate indicators.

Another foray into EA simulation metamodels is made by Ludwig et al. (2011). Their particular focus is to enable rapid prototyping, executing and analysing EA models in a loop. Both Manzur et al. (2012) and Ludwig et al. (2011) developed limited prototypes to show their metamodels in practice.

Buckl et al. (2008) does not go so far as to develop a prototype, but does provide an approach to develop EA simulations rooted in theoretical methods and principles such as application management, simulation techniques, metrics, etc.

Similar work has been done for UML type models, some of which can be adapted for the use with EA models. One such example is that of Buschle et al. (2013), who develop a tool to support the Predictive, Probabilistic Architecture Modelling Framework (Johnson et al., 2013) originally developed for UML models. It is used for probabilistic assessment and to predict system properties, or in this case, EA properties.

Glazner (2011) present a hybrid enterprise simulation model to predict the behaviour of an enterprise. Its approach is based on that of system dynamics, using agent-based simulation and discrete event simulation to model the behaviour of system. The hybrid approach allows different domains to be simulated in whatever manner most suiting to that particular domain and its stakeholders.

2.5 Archimate

As previously described, a number of EA frameworks exist. However, the architecture modelling language standard developed by The Open Group [1] is used for the purposes of this project and will briefly be described here.

ArchiMate provides the necessary concepts and techniques to model the structures and dependencies within the various business domains at a high-level of abstraction, as well as the relations between these domains. ArchiMate uses various visual identifiers, such as shapes and colours, to visually distinguish architectural concepts and so communicate to an architecture to both architecture experts, as well as business users. These models further lend themselves to analysis, particularly if a modelling tool such as BIZZdesign's Archimate is used for the modelling.

In figure 2.4 an informal metamodel of the ArchiMate language can be found, identifying the primary elements and the relations that may exist between them. The elements can be found in three different layers. The business layer depict concepts such as the organisational structure and business processes. Following this the application layer, which depict concepts such as application components and services. Finally, the technical infrastructure layer which are used to model items such as the hardware artefacts and networks. For each of these layers, ArchiMate specifically depicts structural, behavioural and informational aspects. (Iacob and Jonkers, 2006)

The architectural elements and relations of an EA can be augmented with quantitative data by adding custom properties, called *attributes* in ArchiMate (Iacob and Jonkers, 2006). The attributes can either be used as input variables for analyses, or be the output of the analyses themselves. The detailed description and specifications of the language can be found in The Open Group (2013).

2.6 Conclusion to EA Analysis

In the first step towards completing phase one of the DSRM (identifying the problem), this chapter considered the question:

> What is understood under the term *enterprise architecture analytics*, and what is its purpose? Are there any existing standard guidelines or methods specifically designed to develop simulation models of enterprise architectures models?

[1] http://www.opengroup.org/aboutus

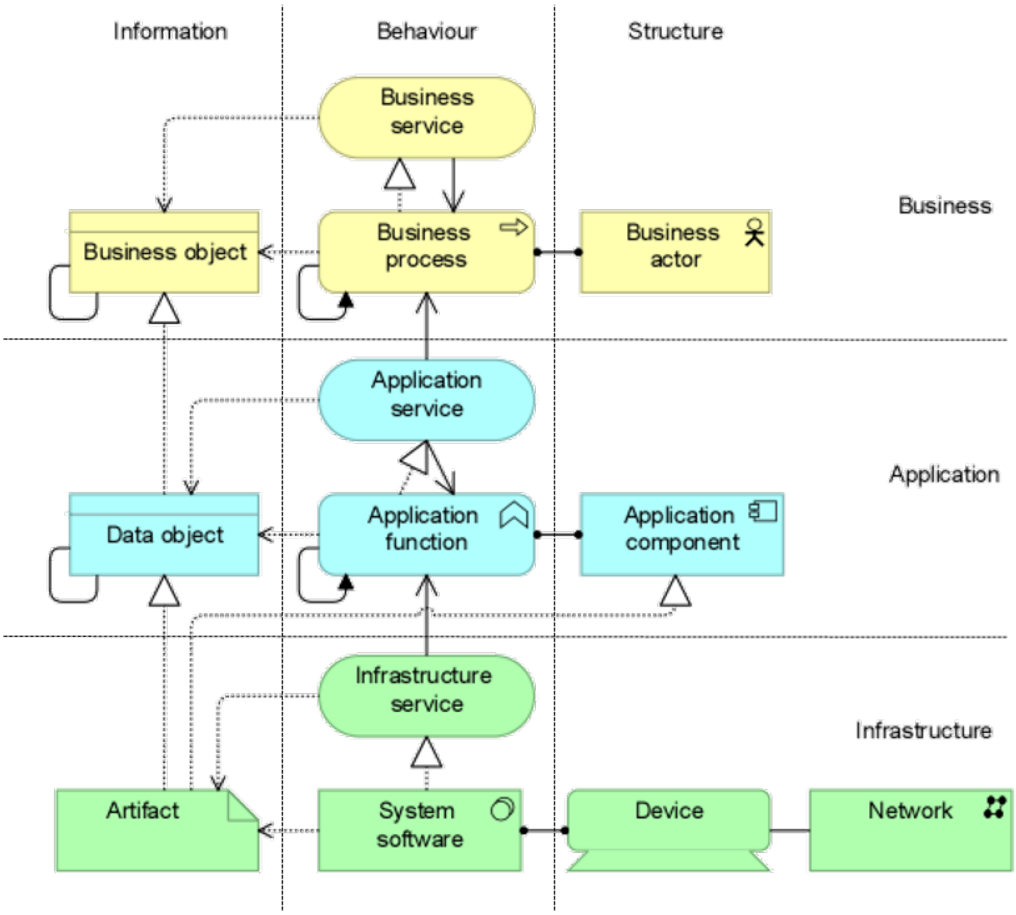


Fig. 2.4 ArchiMate metamodel as in Iacob and Jonkers (2006)

The basic premise, and purpose of EAs were considered first, before moving on to EA analytics in section 2.3, and finally executable EA models in section 2.4. The classification scheme presented by Buckl et al. (2009a), and the analysis activity types described by Wan and Carlsson (2012) provide a sufficient answer to the first part of the question.

Although the metamodels and the other sources discussed in section 2.4.2 might be useful, and one detailed approach to develop EA model based simulations is discussed by Buckl et al. (2008), none of the methods can be said to be standard, i.e. widely used and accepted. Most of the work in this field is directed at creating metamodels, or otherwise distinct models which can be used for simulation purposes. However none of these are far developed yet, and as such neither are their supporting methods.

In the next chapter business analytics are considered, particularly to investigate the synergies between EAs and analytics.

Chapter 3

Business Analytics

"In God we trust, all others bring data."

- William Edwards Deming

3.1 Introduction

The purpose of this chapter is first and foremost to answer research question *1-b*, reading:

> What are business analytics? Is there a synergy between business analytics and enterprise architectures which would encourage and facilitate further investigation into this topic? How are Monte Carlo simulations used for business analytics?

After this, conclusions can be drawn to answer research question *1*:

> Can enterprise architecture models be used as the foundation for Monte Carlo simulation based business analytics, and if so what problem would this aim to solve?

This will then also complete phase 1 of the DSRM, and as such will identify the primary problem the artefact is developed to solve.

The chapter starts with a discussion on the purpose and possible influence that business analytics can have on a business, as well as what businesses should do to achieve the best possible results with business analytics. After this, the Monte Carlo method, commonly used for various business analytics, is discussed.

3.2 Fundamentals

Firms are always looking for new ways in which to gain a competitive advantage, however as any core difference between products and the technologies behind them becoming increasingly negligible in many markets, new avenues are being sought through which to gain real competitive advantage. One remaining differentiator is the business processes delivering those products and determining all internal workings of a firm. This is where business analytics are being used by numerous companies in various capacities to increase the value of those business processes. However it is only used by a few to its full potential, which is to become a cornerstone in a business's strategic reach for competitive advantage. (Davenport, 2006)

Today firms can access a host of data sources to find information regarding what customers are buying, what internal employee turnover rates and compensation costs are and when to their inventory levels are low. However, firms having with a realised the full potential of business analytics can also know how much their customers are willing to pay and how many times they will re-purchase items in a lifetime; what each individual employee contributes to the bottom line and how salaries relate to performance levels; how to keep inventory levels low with ideal re-orders given predicted changes in demand levels and supply chains limitations.

These are but some typical examples of the wealth of information analytics can make available to organizations, enabling them to hone their business processes, making informed decisions based on real, data-driven evidence. Further examples of common BAs can be found in table 3.1.

Although various formal definitions and classification of analytics exist, Evans (2013) describes Business Analytics as:

> "The use of data, Information Technology, statistical analysis, quantitative methods, and mathematical or computer-based models to help managers gain improved insight about their business operations and make better, fact-based decisions."

The basic nature of models used for business analytics can be seen in figure 3.1, the basic premise being that some input data is fed into a decision model, which produces an output in the form of some metric.

Table 3.1 Common Applications of Business Analytics (Davenport, 2006)

Function	**Description**
Supply Chain	Simulate and optimize supply chain streams; reduce inventory levels and stock-out occurrences
Customer selection, loyalty and service	Differentiate customers with greatest profit prospects; Increase the product offering's desirability to customers. Maintain customer loyalty.
Pricing	Ascertain the price point that will maximize sale numbers, or profit.
Human Capital	Identify potential employees most suited to the specific assignments, at particular salary levels.
Product and service quality	Promptly detect quality issues and reduce error occurrences.
Financial performance	Gain clear insight into financial performance drivers and the influence of non-financial factors
Research and development	Improve product quality, value, and if applicable, safety.

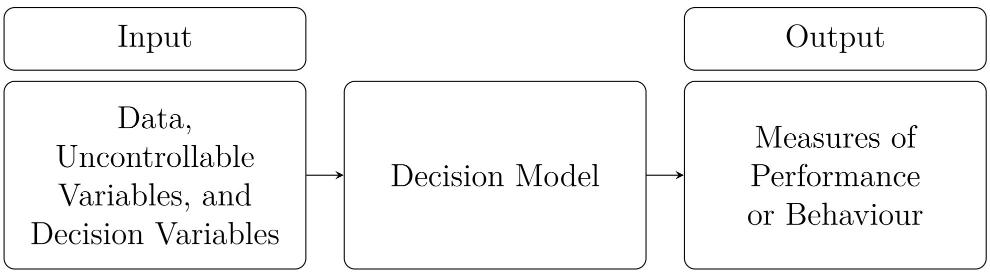


Fig. 3.1 Nature of Decision Models according to Evans (2013)

3.2.1 Analytics Competitors

Davenport (2006) identified a number of corporations which have been using analytics to as a strategic mechanism to gain dominant market positions. A number of common attributes exist among these corporations, which have enabled them to become, what Davenport has labelled, *analytics competitors*:

Widespread use of modelling and optimization As already mentioned, a wide range of analytics exist, such as average order size per purchase, to the far more complex predictive analysis. Corporations wishing to contend at the same level as analytics competitors need to explore the full potential range analytics have on offer. By exploiting data generated internally, and far more effectively exploiting data available externally than their less analytically focused competitors, they can gain far greater business insights. They can explore the which customers give them the most business, which ones are most likely to do so in future, and which ones are likely to turn to competitors instead. By evaluating their supply chains they can not only work to optimize it, but pre-emptively identify potential constrains, simulate alternatives and find effective, efficient solutions. They can price their products in real time to gain the greatest potential revenues from customers. Complex financial models relate operational costs to their financial position. Instead of relying on intuition, leaders in the field use sophisticated business experiments to analyse the impact of strategies, using the results to improve their analytics for similar future initiatives.

An enterprise approach Using analytics throughout an organisation means that potentially every department's operations can benefit from them. Customer management can use insights on customer attrition to pay closer attention to those customers at risk of switching to competitors. Even the marketing department, which typically relies on intuition rather than advanced quantitative techniques can gain new insights, learning which of its methods show the the greatest results and other characteristics of its target audience. Different analytics therefore serve the needs of the various business functions. The full value of analytics for an enterprise, however, can only be realised if such initiatives are the result of single, coherent enterprise-wide endeavour. If individual business units independently make use of analytics, it inadvertently leads to duplicate efforts, often based on inconsistent and partial data. Each department has a different view-point, and therefore adheres to its own definition for terms, influencing the models used for, and the data required for analytics. Data quality also often suffers as departments often end up working with individual spreadsheets, which often contain incorrect data. To remedy this and ensure al efforts are geared towards fulfilling strategic goals, analytics initiatives need to centrally managed across all business units, using common technology , methods and tools. This ensures that data and resources can be well managed and shared across business functions as these conform to consistent formats, definitions and standards.

Senior executive advocates Making analytics a widely used, and core part of an enterprise necessitates wide ranging changes in an organisation's culture, processes, behaviour and skill set. Although such efforts can start in individual business or functional units, the effect of such efforts remain contained as the leaders of these units do not necessarily possess the necessary authority, or even perspective, to realise meaningful results. Like any other major reform effort it instead requires the support, and leadership from top-level executives. These senior advocates do not necessarily have to be statistical experts themselves, but they need to value the power of analytics, whilst also understanding its underlying principles and limitations. The company needs to be willing to invest in the required resources to develop its own capacity, and when needed retain necessary the experts capable of relating complex quantitative methods to the business's operations and the needs.

3.2.2 Synergy between EA and BA

Instead of considering the enterprise architecture first, and how it can be used as the foundation for the prediction of enterprise behaviour, here business analytics are considered in terms of the impact they can have on an enterprise, and what is required of the enterprise in order to achieve such analytical capabilities. A number of the requirements identified by Davenport (2006), can be related to what EAs already provide.

A synergy can therefore be identified, between that of business analytics used to gain serious competitive advantage, and the realm of enterprises architectures:

- The first and greatest of synergies can be found in the need for an enterprise approach for business analytics. Even though *enterprise* in this instance is supposed to be indicative of the entire corporation, and not merely a specific enterprising venture as is possible when discussing EAs, the synergy stems from the

EAs cross-functional approach. Architectures intend to capture all connections that exist between the various building blocks, independent of the domain they belong to. BA can therefore possibly gain advantage by relying on the practices, or being based on the artefacts produced by EA's when their own models are being developed.

- The previous point is also closely related to next. Widespread use of modelling and optimisation can be achieved far more easily if an EA is used to support the effort. As EAs span boundaries within enterprises, it is far easier to identify where business analytics should be used throughout an organisation (as opposed to ones view being limited by the dividers between business units) and then making the necessary arrangements to facilitate the necessary modelling and optimisation efforts.

3.3 The Monte Carlo Method

3.3.1 Stochastic Models

As seen in figure 3.1, business analytics requires some input variables in order to calculate an output measure. If considering past events, this is easy to accomplish if you have the necessary data on hand. However, once you start considering the future for instance, things start to look different.

Due to randomness, uncertainty and risk is introduced into the model. This randomness can be specifying input variables as appropriate probability distributions . Some famous continuous distributions include the Uniform, Normal, Triangular and Exponential distribution (Allen, 2011). However, many more exist of course. Once a model contains a random variable, it is a stochastic, or probabilistic model in itself. (Evans, 2013)

3.3.2 Overview of Method

The Monte Carlo method, quite simply, is the process of generating a random variable for all of the stochastic input variables, calculating the random output, and repeating the process many times in order to understand the output distribution of the model's metric.(Allen, 2011; Evans, 2013)

Once a sufficient number of runs have been completed, histograms and other descriptive statistics can be used to describe the output of the model, in order to garner information about its distribution. This information, having taken the uncertainties and risks of the inputs into account, is then used to make a decision as with other BAs.

3.3.3 Histogram & Descriptive Statistics

The descriptive statistics typically useful for the analysis of the results of a Monte Carlo simulation are as follows (Evans, 2013):

Sample mean

the mean if a sample of n observations, $x_1, x_2, \ldots, x_n$ referred to as $\overline{x}$, is calculated as:

$$\overline{x}(n) = \frac{1}{n}\sum_{j=1}^{n} x_j \tag{3.1}$$

Sample standard deviation

$$\hat{\sigma} = \sqrt{\frac{\sum\limits_{j=1}^{n}(x_j - \overline{x}_n)^2}{n-1}} \tag{3.2}$$

Seeing

Kurtosis

$$g_2 = \frac{m_4}{m_2^2} - 3 = \frac{\frac{1}{n}\sum\limits_{i=1}^{n}(x_1 - \overline{x})^4}{\left(\frac{1}{n}\sum\limits_{i=1}^{n}(x_i - \overline{x})^2\right)^2} - 3 \tag{3.3}$$

m_4 is the fourth sample moment about the mean and m_2 the second sample moment about the mean (or sample variance)

Estimator of Population Skewness

$$g_1 = \frac{m_3}{\hat{\sigma}}^3 = \frac{\frac{1}{n}\sum\limits_{i=1}^{n}(x_i - \overline{x})^3}{\left[\frac{1}{n-1}\sum\limits_{i=1}^{n}(x_i - \overline{x})^2\right]^{3/2}} \tag{3.4}$$

m_3 is the sample third central moment.

Median

The median is either the middle number in an ordered list of sample values, if there are an odd number of n values. Or the average of the two middle values in the list if there are an even number of n values.

Interquartile range

To calculate the Interquartile range (IQR), the first Q_1 and third quartile Q_3 need to be calculated. If there are an odd number of samples, the central value, or median, is excluded from the upper and lower half of the sorted list of values. As in the case of the median, if these halves contain an even number of samples, the average of middle denote the first quartile, and similarly, the middle two value of the upper half of the list denote the third quartile. If, however, there are an odd number of values in each of the two halves, the middle value in each the lower and higher list are taken as the first and third quartile respectively. The IQR is the calculated as:

$$IQR = Q_3 - Q_1 \tag{3.5}$$

3.3.4 Histogram

In order to analyse the results of a Monte Carlo simulation visually, a histogram can be created. Histograms are useful to visually inspect the spread of the sample data over its range; to visually inspect some of the quantitative descriptive statistics described above. The number of bins to use for the histogram is up to the individual conducting the analysis, however a number of formulas exist to help in this his choice (Scott, 1992):

Square-root choice

$$k = \sqrt{n} \tag{3.6}$$

Sturge's formula

$$k = \lceil \log_2 n + 1 \rceil \tag{3.7}$$

Doane's formula

$$k = \log_2(n) + \log_2\left(1 + \frac{\mid g_1 \mid}{\sigma_{g1}}\right) \tag{3.8}$$

where

$$\sigma_{g1} = \sqrt{\frac{6(n-2)}{(n+1)(n+3)}}$$

Scott's reference rule

$$h = \frac{3.5\hat{\sigma}}{n^{1/3}} \tag{3.9}$$

Freedman-Diacois' choice

$$h = 2\frac{IQR(x)}{n^{1/3}} \tag{3.10}$$

here IQR is the interquartile range, which is less sensitive to outliers in sample.

After the user has selected the number of bins h, the bin width k can be calculated as

$$k = \left\lceil \frac{\max x - \min x}{h} \right\rceil$$

after which the number of samples fitting into each bin can be determined.

3.4 Conclusion to Business Analytics

Coming back to the research questions, *RQ 1-b* asks:

> What are business analytics? Is there a synergy between business analytics and enterprise architectures which would encourage and facilitate further investigation into this topic? How are Monte Carlo simulations used for business analytics?

By considering some of the basic principles of BAs, and some common BAs, the first part of that research question could be answered. Moving on, section 3.2.2 considers the synergies that exist between EAs and BAs, finding that there could potentially be numerous benefits if these synergies were to be exploited. Finally the Monte Carlo method (or simulations) are briefly considered, explaining the concept of variability, and how the Monte Carlo method can be used in conjunction with descriptive statistics.

Having answered both *RQ 1-a* and *RQ 1-b, 1* can be considered:

> Can enterprise architecture models be used as the foundation for Monte Carlo simulation based business analytics, and if so what problem would this aim to solve?

In chapter 2 it was seen that EAs can be extended, or used as the foundation for various types of simulation studies. This is a useful pursuit as it is important to predict and analyse resultant dynamic behaviour of a static EA model. The Monte Carlo method is also a method quite commonly used for various types of business analytics, allowing uncertainty and risk to be incorporated into a model through appropriate stochastic variables. The use the Monte Carlo method as a simulation method, being based on EA models, therefore seems to hold great promise.

In answer to the last part of the *RQ-1*, the following problem statement can then be formalised:

> How to develop an executable enterprise architecture model which can enable a diverse range of business analytics, which are founded on the Monte Carlo method, and can be both executed and evaluated?

This then is the problem identified for phase 1 of the DSRM.

Chapter 4

Executable EA Models & the Monte Carlo Method

4.1 Introduction

In the previous chapter, the problem, and subsequent aim of the to-be artefact was identified. In short, the artefact should enable a diverse range of business analytics, which are founded on the Monte Carlo method, to be based upon EA models and subsequently executed and evaluated.

Following this, in the second phase of the DSRM, objectives are identified for the solution to the identified problem. This is the subject matter of this chapter, for which the second research question was identified:

> Given these findings, what would be the requirements of a technical artefact designed to enabled Monte Carlo simulations, based on using enterprise architecture models?

To answer this question, the different stakeholders are identified first, before considering their respective objectives, and lastly the requirements that need to be fulfilled by the artefact in order to meet those objectives.

4.2 Stakeholders

Three primary sets of stakeholders can be identified whose interests are of primary concern for the artefact (see figure 4.1). The first stakeholder is the organization

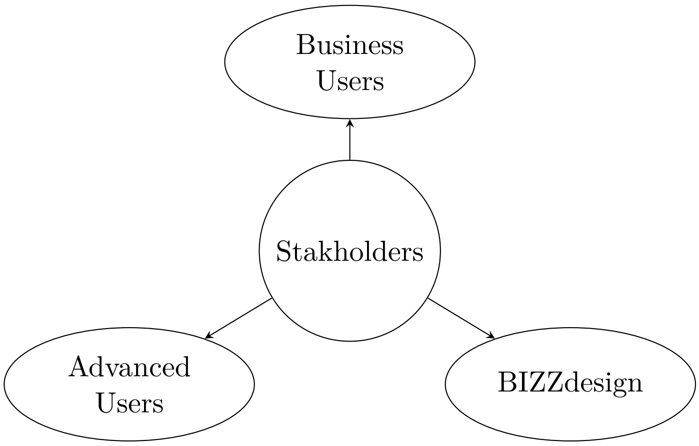


Fig. 4.1 Artefact Stakeholders

wishing to make the use of the artefact available to end-users. In this case it would therefore be BIZZdesign. However, the objectives and requirements identified below in the following sections will not necessarily be specific to BIZZdesign, but more general to any organisation wishing to develop an artefact, as the one described in this project.

Following this, the end-users whom the use of the artefact is being offered to, are of interest. These could either be internal to, external to the first stakeholder. In this specific case that means consultants from BIZZdesign itself, or clients using BIZZdesign's Architect application. Two types of end-users are subsequently identified again. Firstly there are the users who know and understand the scripting environment and can write analysis scripts themselves. Then there are the users who would use the analysis capabilities provided (whether built-in or custom made by the first group of users), only to execute analyses.

The first set of users is henceforth described as the *advanced users*, given their advanced knowledge of scripting. The second group of users are the *business users*, who are primarily concerned with getting results from the artefact.

4.3 Stakeholder objectives

The objectives and subsequent requirements were identified in direct connection to the artefact being developed.

4.3.1 BIZZdesign

Considering BIZZdesign's objectives', the first is the business need to develop products that their clients will be using. Over a period of several years BIZZdesign has developed process analysis functionality in their BIZZdesigner [1] tool. However, contrary to the purported business needs for analysis described by literature, these analyses tools are typically not being by their clients. It is therefore questionable whether EA analysis tools will then be used by their clients using Architect. Although the reason for this lack of use of analytical tools is out of the scope of this study, and could be due to any number of reasons, the finding is perhaps not wholly unexpected. Although a clear, and growing need for analytics exists within the business environment, in order to be truly successful and consequential, any such efforts have to be driven from within the top echelons of a company, as was described in chapter 3. Although the use of analytics for competitive advantage is being used successfully in a number of corporations, such extensive use is certainly not wide-spread, and therefore the tools enabling such analysis not widely used either. However, due to the growing trend in business analytics, and the very clear need for them, further study and development within the field is certainly required, even if the business uptake is not immediate.

This first objective then makes for an important assumption, without which this artefact would not have any immediate purpose. Before specific requirements for the artefact can be deduced from it, it is subdivided into three sub-objectives, designed to ensure that clients who to have the need for such an artefact, will in fact make use of it:

1. With this first objective, the intention is to focus purely on providing a range functions to users, which will enable them to run their analysis, in a Monte Carlo fashion.

2. The ease-of-use of an artefact is also equally important if it is to find wide spread use.

3. Lastly, the artefact should also be applicable to a wide range of cases. In other words, a wide range of analytics should be made available through it, which can be applied to wide range of models.

[1] http://www.bizzdesign.com/tools/bizzdesigner/

4.3.2 Advanced & Business Users

The advanced user's objective is quite simply to enable their counterparts, the business users, to easily develop and execute specific Monte Carlo based analytics, which they, the advanced user's, have made possible via provided scripts.

The business user's objective is to develop, implement and ultimately use the results of a Monte Carlo analysis to make informed business decisions.

4.4 Stakeholder Requirements

Given the five objectives identified above, the following combined list of requirements was identified:

Req. 1: Standard Approach
In order for the analyses method enabled by the artefact to be generic in as far as possible, a standard method needs to be provided through which a range of different Monte Carlo analytics can be developed.

Req. 2: Integrating Analyses
The artefact is aimed at users who want to develop their own custom scripts for some form of business analytics, which require the use of the Monte Carlo method. These users need a simplistic way of incorporating their scripts, with that of the Monte Carlo simulation's. The aim here then be for the user to make a minimal number of changes and additions to his script which are easy to understand and accomplished, and still provide the necessary level of integration.

Req. 3: Defining Various Random Variables
As previously described, the ArchiMate language specifies the use of profiles through which to assign attributes to elements. In the case of the Monte Carlo method, an unspecified number of random variables will have to be to be assigned to elements of the EA model. These would have to work in conjunction with the profiles already used by the user's analysis script. It would also have to provide the means by which a sufficient number of parameters can be entered for each random variable, specifying variables distribution.

Req. 4: Straight-forward Execution
Although the user will have to have a basic understanding of the principles of Monte Carlo simulation, the execution thereof should occur with minimal user effort required to provide the necessary input. With this in mind the user should be presented with relevant options during the use of the tool-set, and not have to search for, or make customizations to the script in order to successfully run a simulation.

Req. 5: Analysis of Results After the analysis has been completed by the user, he should have been provided with relevant, and a sufficient amount information to have gained a clearer insight. In the case of Monte Carlo simulations, a range of descriptive statistics as well as graphical representations can be provided, as previously described. These results should be easy to attain and understand.

Req. 6: Minimal Opportunity for User Error
To make it as simplistic as possible, minimal opportunities should exist for the introduction of user error and the use of the tool-set should be as self-explanatory in as far as possible.

4.5 Conclusion

In this chapter, the second research question was answered, as well as corresponding second phase of the DSRM. The requirements for the development of a technical artefact were identified, by first considering the stakeholders, and then their respective objectives for such an artefact. In line with the findings of chapters 2 and 3, the artefact is meant to enable the use of Monte Carlo analytics, based on enterprise architecture models. Six primary requirements were identified, which need to be fulfilled during the development and implementation of this artefact, which is considered in the following chapter.

Chapter 5

The Artefact

5.1 Introduction

This chapter documents third phase of the DSRM, namely the *Design & Development* of the artefact as it was implemented in Architect. Through it, users are enabled to develop their own analyses which can then be simulated in a Monte Carlo fashion and the results analysed with the aid of various descriptive statistics and graphical representations of the results. This is done in answer of the third research question:

> How could an artefact be developed an implemented to meet these requirements within the enterprise architecture modelling tool, Architect?

The description of the artefact is divided into four parts, following the four phases required to successfully make use of the artefact, as can be seen in figure 5.1. Section 5.2 details the steps required to integrate an analysis script with those developed for the Monte Carlo simulation. Section 5.3 describes how random variables are to be specified on the model. The development of the script used to execute the Monte Carlo simulation is described in section 5.4, whilst section 5.5 considers the set of tools developed to analyse the results of a simulation.

5.1.1 Satisfying Requirements

The five objectives, and six requirements detailed in the previous chapter are the primary drivers influencing the design choices that were made. Particularly, the four phases as they are identified above, and the rigorous set of steps set out for each of these in turn, are in answer to the first requirement, namely to provide a standard

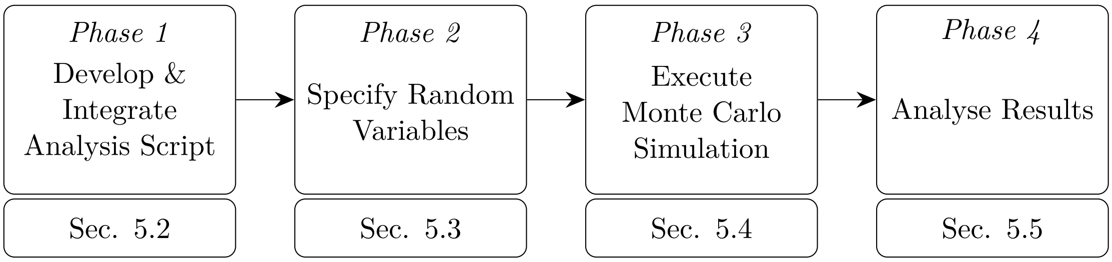


Fig. 5.1 Executing the Analysis Method

method by which to use the artefact. Where specific design decisions were made to satisfy further requirements, these are explicitly mentioned throughout the chapter. A general remark on this point is also made in the conclusion (of this chapter).

5.1.2 Modelling

Two types of modelling diagrams from the set of Unified Modelling Languages (UML) were used for the purposes of this chapter. Firstly, Activity Diagrams (AD) were used specifically to convey the interactions occurring between the user and the artefact. Following this Data Flow Diagrams (DFD) were specifically used to convey how data is retrieved and where it is stored. This is particularly important as the artefact is built around the concept of profiles, used to assign values to elements.

5.1.3 Assumptions & Limitations

As already mentioned, the artefact was developed and implemented using BIZZdesign's Architect application for EA modelling. Although its scripting capabilities are not unique (the scripting is done in the common programming language C) and as such the artefact as it is described here, both in method and technical implementation, should be translatable to other similar EA modelling applications.

Besides Architect, the artefact also uses the Component Object Model (COM) interface to make use of Microsoft Excel's worksheet functions. Although these are only being used for a limited number of calculations (finding the inverse for the Normal, Lognormal and Beta distributions), the application is opened during the initialization phase of the Monte Carlo simulation, and therefore needs to be available on the computer being used (specifically Excel 2010 or later).

Further, it is assumed that the user will implement his analysis script through the use of profiles on the EA within Architect. In other words, that the user will define custom profiles with which he can attach supplementary information to an EA models' elements in order to perform his analysis. This is described further in section 5.3.

Although the processes developed to provide the given functionality are discussed in quite extensive detail in this chapter, numerous specifics are also skipped regarding its implementation within the scripting environment.

5.2 Develop & Integrate Analysis Script

The method used to develop the analysis model itself, and subsequently implement this model by means of a script is not part of the scope of this project. Interested readers are referred to sources such as Evans (2013), describing such methods. It should be noted that although they are not discussed in detail here, their importance should not be underestimated as the results of the simulation will only be as useful as the this model is accurate. Any results from such a simulation should always also be considered in light of the assumptions and decisions made during the development of such a model and the selection of its input data.

This phase, it should be noted, was developed in response to the second requirement identified in the previous chapter, namely to integrate the user's analysis script's with those of the artefact, with a minimal number of changes.

After the user's analysis script has been developed, the three partial scripts described in this section should be added to it, as required. They integrate the script with those of the artefacts. Non of these additions should hinder the script from being run as a stand-alone analysis, in other words without starting a Monte Carlo analysis. With exception of the last script, the first two will not necessarily be used if the user's script does not require them.

5.2.1 Tagging Input Data

If the user's analysis script requires some type of input from the user during its execution, this should only be asked during the very first simulation run, i.e. the first time the analysis query is executed. As it is not possible to conceive and define all input requirements an analysis might possibly require within the Monte Carlo script, the programmer needs to account for this within his portion of the script.

This can be accomplished by including (part of) algorithm 1 in the users script. It describes by means of an example how to get input from a user, which is required for the analysis, and tag this data to the model. In the example the script asks the user which part of the model to calculate costs for (line 7), and assigns this value to *modelAndObjects*. This in turn is added to the structure, *QueryInsG*, under the name *'modelAndObjects'* (line 12). In this manner any number of variables can be added to *QueryInsG*, which is then tagged to the model as textit'QueryIns'.

Algorithm 1 Attain, store & re-use initial input values

```
1: QueryInsG ← Structure()
2: if model has tagged value QueryIns then
3:    QueryInsG ← tagged value QueryIns on model
4:    modelAndObjects ← value for "modelAndObjects" in QueryInsG
5:    FirstRun ← false
6: else
7:    modelAndObjects ← ask("Select that (part of the model that you want to
      calculate total costs for")
8:    if modelAndObjects is undefined then
9:       Set tag RunStopped on model as true
10:      return undefined
11:   end if
12:   Add modelAndObjects to QueryInsG as "modelAndObjects"
13:   Set tag QueryIns on model as QueryInsG
14:   FirstRun ← true
15: end if
```

In line 2 the it is checked whether the model has already been tagged with *QueryIns*, in other words, whether or not this is the first time the script is being called by the Monte Carlo simulation or not. If it is not, it recalls the tag and assigns it to *QueryInsG* (line 3), which in turn can be used to retrieve any data the script may require, such as the variable *modelAndObjects*" in the example (line 4).

5.2.2 Indicating a 'Stop Event'

In the example above, the model is also tagged with the boolean *RunStopped* (line 9 & algorithm 2) if the user does not provide a selection. This tag indicates to the Monte Carlo script to discontinue the simulation. This can be used in exactly the same manner elsewhere by the user's script to inform the Monte Carlo simulation to discontinue if an error is encountered.

The boolean, *FirstRun* is set to true only for the first run (line 5 & 14), and can be used by the user's script as required. For example, the script might check with the

Algorithm 2 Stop Script Tag

Set tag *RunStopped* on model as *true*

user as to whether or not he is sure that he would like to overwrite some existing data. Such a check should only be completed during the first iteration of the simulation though, for which $FirstRun$ can be used.

5.2.3 Tagging Variables

If the boolean $FirstRun$ is not convenient, the Monte Carlo simulation provides the user with another means by which to check whether it is the first iteration of the simulation. The model is namely tagged with a boolean of the same name at the start of the first iteration, and subsequently removed again at the start of the second. This allows the user's script to use the 'hasTaggedValue' method to check whether or not it is the first iteration of the simulation. Its use is illustrated in algorithm **??**, which must be included in the user's script to indicate all attributes which are variables calculated by user's analysis script, and are of interest for later analysis. Although the Monte Carlo simulation will run to completion without this having been done, none of the results assigned to these variables will be stored for later analysis.

Algorithm 3 Tag Attribute as Variable

if model has tagged value $FirstRun$ **then**
  type ← "result"
  call TagObject($object, attribute, type$)
end if

The user has to provide the function *TagObject* [1] with three parameters. The first, $object$, is the element on which the attribute is located. The second, $attribute$, is the attribute which is the random variable of interest. Lastly, *type* indicates what type of variable it is. In the case of a calculated variable, this should be assigned the string value *'result'*, as in the example above.

The *TagObject* function, which should be called separately for each attribute, creates a list containing the names of all the attributes identified as being variables of interest. This is done separately for each element and it attributes, and the list is then tagged to the element to which the attributes are assigned. In the case of resultant variables, the name of the list is *"ObjectWRes"*.

[1] Located in ArchiMate.MonteCarlo.SetStochVar

The variable *type* can also be assigned the string value *'variable'*, in which case the script can be used to indicate which attributes are used as input variables. In this case the name of the list tagged to the element is *"ObjectWVar"*. This can be useful if the analysis only uses a subset of all of the attributes indicated as random through the use of the *Random Variables* profile. Per default the Monte Carlo simulation will calculate all variables specified as random, which can incur a time penalty of a portion of these are not required for the particular analysis.

As illustrated, the random variables are only tagged during the first simulation run (i.e. when *FirstRun* exists). The *TagObject* function could be called with each iteration of the simulation, being sent the same attributes without causing problems, however this is unnecessary as the tags remain on the model until the application is closed (or they are explicitly removed). By only calling the function during the first iteration, processing time can be saved which can become significant if a large number of iterations are completed.

5.2.4 Advanced

The simplistic addition should not prove to be too troublesome for an advanced user to include. Nonetheless, for users finding this insufficient, a point to note: although any name other than *QueryIns* could conceivably be used, or any other number of tagged values, this would not be supported by the Monte Carlo script which looks for, and erases the tagged value *QueryIns* at the start of each simulation. An advanced user using additional tags can work around this in one of two ways. In either instance, the programmer needs to ensure that the parameter, *persistent*, for the *setTaggedValue* method is set to *false*[2]. This ensures the tagged values are not stored along with the model when it is closed. The user can then simply close and re-open the model if he needs to re-set the initial input values of the selected query. The second method requires the programmer to alter the Monte Carlo script itself in order to include commands that will clear the additional tags defined by him. Caution should be practised though, to ensure that the Monte Carlo script remains functional with other analysis scripts too.

A copy of the scripts can be found in Appendix A.2.

[2]See Architect's scripting reference guide for more details under *Scripting language → Methods → Miscellaneous→ Tagged Values*

5.3 Specifying Random Variables

As previously described, the ArchiMate language specifies the use of profiles in order to be able to append data to elements within the EA model. This feature, and its implementation within Architect is leveraged by the artefact in order to specify random variables, to be used as input values to the analysis. As stated in section 5.1.3, it is assumed that the user's analysis script makes use of such profile attributes in order to input the variables of the analysis model on the EA model, and subsequently also store the results of the analysis on profile attributes.

5.3.1 Specifying Random Variables

Once variables have been defined on user's profiles', it should be possible to specify them as being random, and provide the necessary information describing the beliefs held about the range of possible values the variable could assume. This can be done via another set of attributes on another profile, which is the purpose of the *Random Variables* profile, provided as part of the artefact. The first set of structured attributes on this profile are explained next, corresponding to the example in figure 5.2 and figure 5.3.

Variable Name The first field on this profile which has to be filled out by the user is the 'Variable Name'. The string value entered here should be name of the attribute which the user wishes to be specified as a random value. The name should be an exact copy of the name as it is shown in the user interface (UI), as can be seen in the example in figure 5.2.

Attribute (auto-filled) As indicated, this field does not have to filled out by the user as it is automatically filled out during the execution of the Monte Carlo simulation

Distribution Here one of five continuous distribution can be selected, namely the Normal, Lognormal, Uniform, Triangular or Beta distribution (although other options could easily be added, the selection was limited for the purposes of this artefact to keep the remainder of profile as simplistic as possible).

Parameters Depending on the chosen distribution, between two and four different parameters need to be provided by the user. Which of these need to be filled out, and what parameter is assigned to which field, is indicated by the numbers

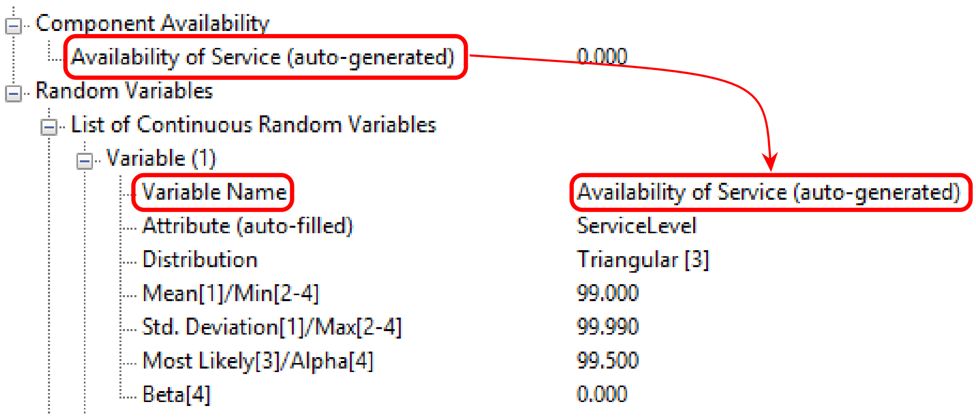


Fig. 5.2 Specifying a Random Variable Example

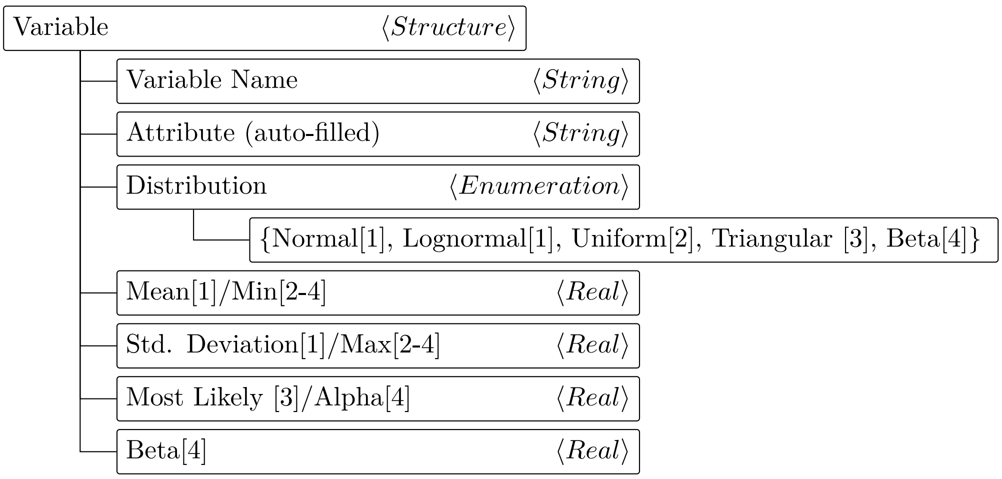


Fig. 5.3 Random Variables Profile Attributes

in square brackets, which correspond to the numbers in square brackets behind the names of the distributions as they are provided in the UI.

As it is not known how many random variables will exist on each element, and one cannot assign more than one Random Variables profile to a single element, the attributes have to be structured in such a way that an unspecified number of variables can be specified via the single profile. As such the structure as described above is contained within an attribute of type *list*. This allows the user to add any number of attribute names to the list.

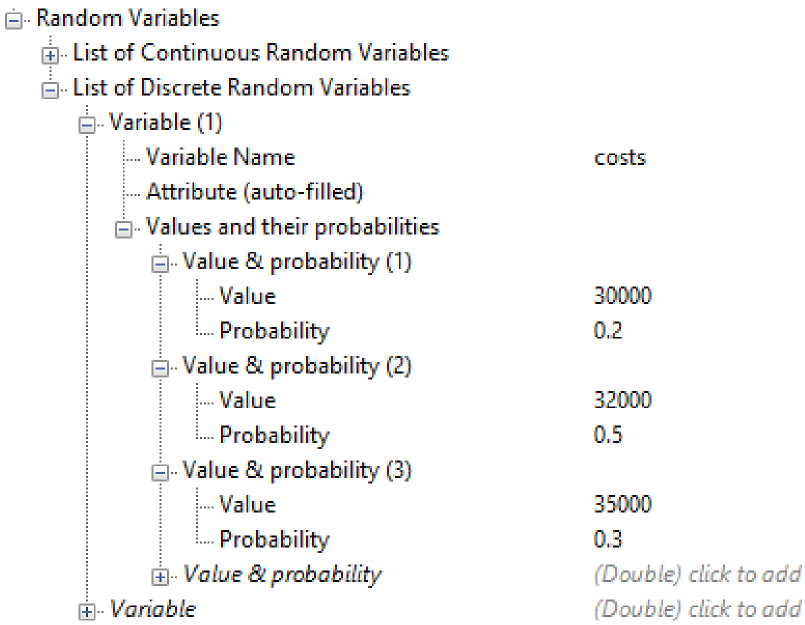


Fig. 5.4 Example use of the *Random Variables* profile

5.3.2 Discrete Random Variables

Besides providing the ability to specify attributes as continuous random variables, it is also possible to specify them as discrete variables. A second structured set of attributes is provided for this purpose on the Random Variable profile. As before, the name of the attribute has to be provided to 'Variable Name', but instead of specifying a distribution, the user has to provide a list of values and their corresponding probabilities (where the sum of the probabilities for a single variable must equal 1). The profile, as can be seen in Architect, can be seen in the example in figure 5.4.

The complete structure of the attributes on the Random Variables profile, along with their data types, can be seen in figure 5.5. A copy of the profile definition and its translation file can be found in Appendix A.6 and A.7 respectively. Note that the names provided in the figure are as they are seen in the UI of Artefact. The names do not necessarily correspond to the attribute type, but rather were chosen to help the user make sense of it. The variables n_r, n_{dr} and $n_{i,vp}$ indicate that multiple instances of the random variable, discrete random variable, and their values & probabilities can exist.

The complete profile, as represented here, allows the artefact to meet the third requirement identified for its development in 4.4, namely to be able to specify any number of attributes on any given element as random, including all necessary param-

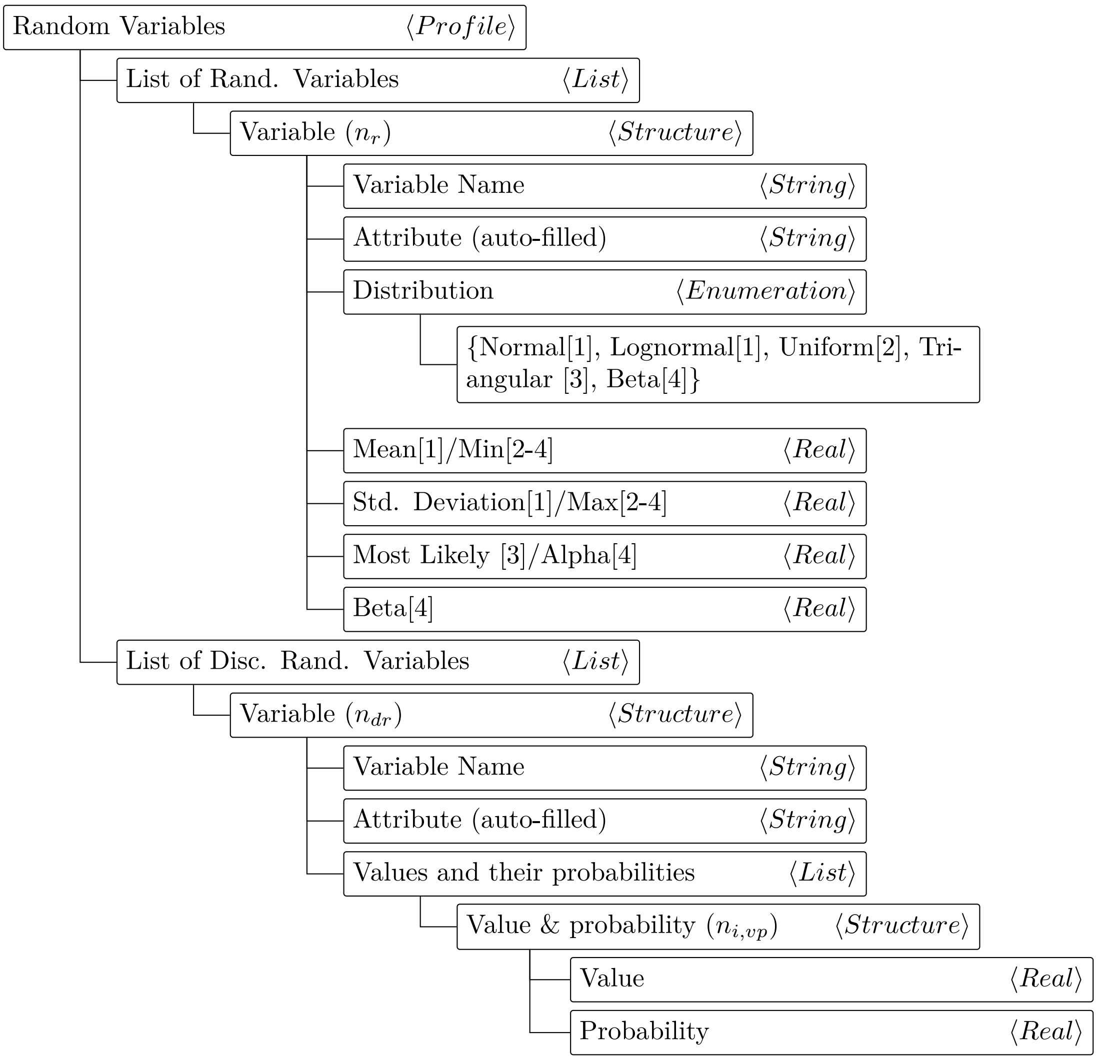


Fig. 5.5 Random Variables Profile

eters.

5.4 Run Monte Carlo Simulation

Once the necessary additions have been made to the user's script and the relevant variables have been defined as random, the user can proceed to execute the Monte Carlo simulation. This is done according to the activity diagram in figure 5.6 and the corresponding level 0 data flow diagram (DFD) in figure 5.8. For a context level DFD of the simulation, indicating what important information the user has to provide and what is provided to him, see figure 5.7.

5.4.1 Overview

An overview of simulation processes are described here, corresponding to the processes in the DFD of figure 5.8. As can be seen from the Activity Diagram in figure 5.6, the number of steps the user is required to complete in order to execute the Monte Carlo simulation is quite limited. This was developed to meet the fourth requirement from section 4.4 above, namely to provide a straight-forward means by which to execute the Monte Carlo simulation.

Process 1 To initiate the simulation , the user has to select the Monte Carlo simulation query, located in the Viewpoints toolpane within Architect. He is subsequently asked which analysis (or viewpoint) should be executed. After selecting an analysis script developed and integrates according to the requirements set out in section 5.2, he is asked how many times the simulation should be iterated.

Process 2 After a valid choice has been entered, the values of all of the attributes specified as random variables via the Random Variables profile are tagged to the respective elements to which they are assigned via a tag. The name of these tags corresponds to the name of the attributes whose value is being stored. These tags are used again in phase 5 to restore the original values to the attributes, thereby restoring input data on the EA model to its original state. This process is achieved by considering each item on the list of random variables and list of discrete random variables on the *Random Variables* profile (see figure 5.5). Each entry's user provided variable name is used to find the attribute being indicated on the element. If the attribute, cannot be found, or more than one attribute exists with that name as it is displayed in the UI, an error message is generated and the simulation cannot continue. If the attribute is found, its name, as used by the system, is assigned to the *Attribute (auto-filled)* field, to be used later on by the simulation.

Process 3 To initialize the simulation, the script removes the *QueryIns, RunStopped, ObjectWVar* and *ObjectWRes* tags from the model and all the elements. (Their use is described in section 5.2).

Process 4 The steps involved in executing the simulation loop are described in section 5.4.2 below. From the users point of view, the only further input he might have to provide at this point in time is if his selected analysis script requires some. In order to avoid having to re-input the same data for each run, the user's analysis script has to store these data to a tag on the model, also described in 5.2 above.

Process 5 See the description for process 2 above.

Process 6 After the simulation has completed, the user is informed of this. At this point in time no further information is provided to the user, as it is not known which of the variables the user wishes to analyse, which is the subject matter of section 5.5.

5.4.2 Simulation Loop

Here the primary processes involved in executing the simulation loop (process 4 in figure 5.8) are discussed. Although the exact implementation thereof is not discussed in extensive detail (processes 4.1 and 4.2 could again be subdivided and described in their own right), it gives an overview of the different processes required, how they fit together and how the requirements for this artefact were achieved.

Process 4.1 This process only occurs during the first two loops of the simulation to identify the list of elements on which random variables exist. During the first simulation loop, the list is simply populated by all elements which have the *Random Variable* profile assigned to it. As such, all attributes indicated as random variables are generated during the first iteration loop. If the user as specified which particular random variables are used by the analysis script through the *TagObject* method described in section 5.2, then the simulation only considers the list of these elements from the second simulation run onwards (the user's query is only called for the first time in 4.3). If this has not been

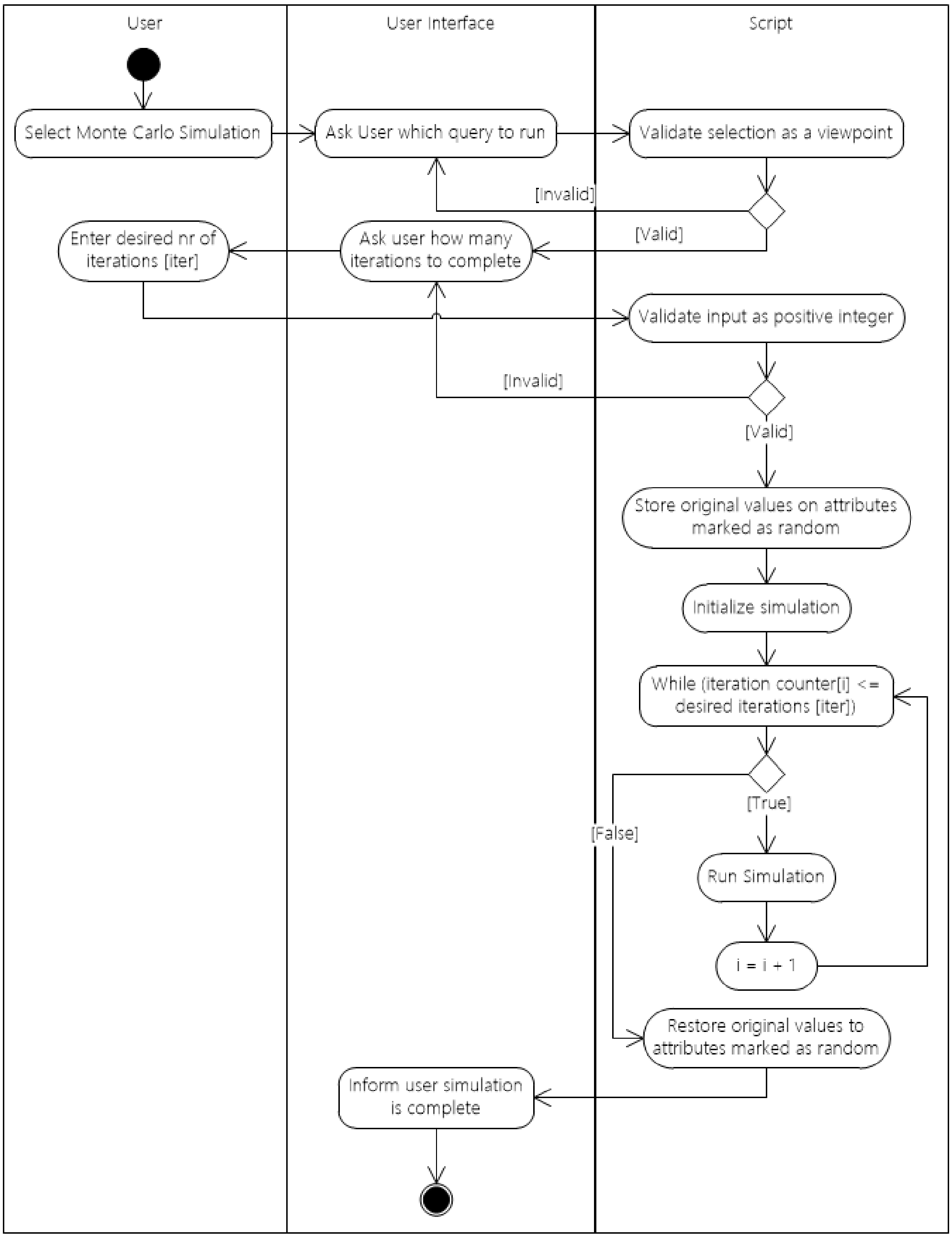


Fig. 5.6 Executing the Monte Carlo Simulation Action Diagram

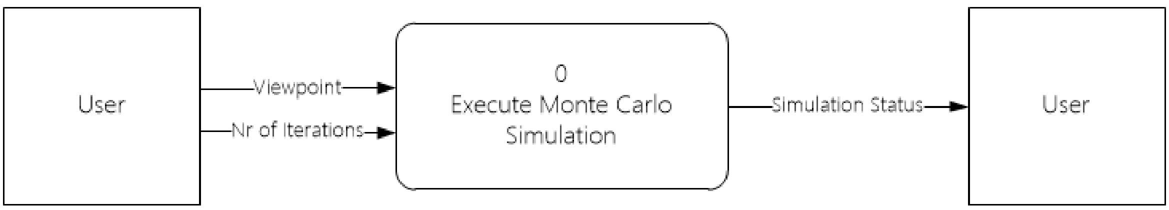


Fig. 5.7 Executing the Monte Carlo Simulation Context Level Data Flow Diagram

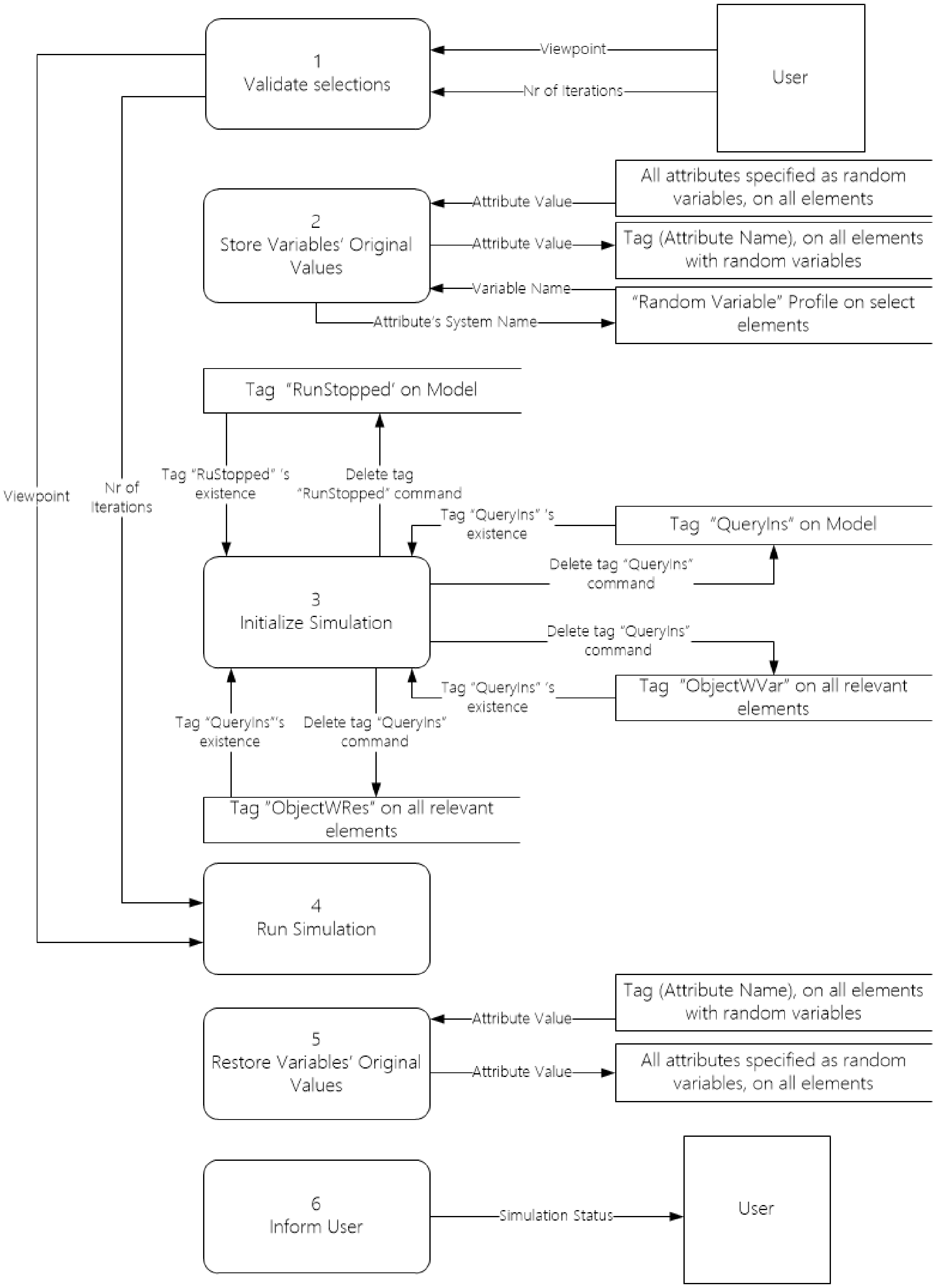


Fig. 5.8 Executing the Monte Carlo Simulation Level 0 Data Flow Diagram

incorporated into the user's script, the list of elements identified during the first iteration is used instead for the remainder of the iterations.

Process 4.2 For each element on the list identified in the previous step, the following sub-steps are completed, as seen in figure 5.10:

Process 4.2.1 For each entry on the list of random variables and list of discrete random variables (on the element's *Random Variables* profile), the attribute name and all parameters regarding its distribution are attained.

Process 4.2.2 All of these parameters are then used to find an inverse of the user specified distribution, using a random number generated with the built in *rand()* method as the probability. When required, excel's worksheet function is called to accomplish this step.

Process 4.2.3 This value generated from the inverse distribution is then assigned to the attribute for which the distribution was originally specified. (This, in turn, will then be used by the user's analysis script for its calculations.)

Process 4.2.4 With each iteration of the simulation, the value that was is generated for a specific attribute is added to a list containing the results of all previous iterations of that simulation. This list, tagged to the element on which the attribute is located, has the same name as that of the attribute's system name, appended with an "L" (for *List*).

Process 4.3 After all of the random variables have been generated and assigned to their respective attributes, the user selected analysis script is called.

Process 4.4 The purpose and functioning of this process has already been described in section 5.2.

Process 4.5 After the completion of the user's script, the artefact checks whether the model has been tagged with the boolean *RunStopped*. If it has,the original values found on the attributes specified as random are restored (process 5 from figure 5.8), the user informed of the error and the simulation ended.

Process 4.6 This processes is the same as that of 4.2.3, except that it saves the resultant values on the attributes identified by the user's script via process 4.4.

Process 4.7 This processes sends a message to Architect's interface in increments of 10%, indicating what percentage the total number of required iterations have been completed.

Process 4.8 The last step in the simulation loop, simply $i = i + 1$. It is worth mentioning that although the loop counter i is not a permanent data store, it is used by numerous processes and it was therefore useful to explicitly indicate it at this level.

The reader interested in more details is referred to the code located in appendix A.1, which has been provided with extensive commenting.

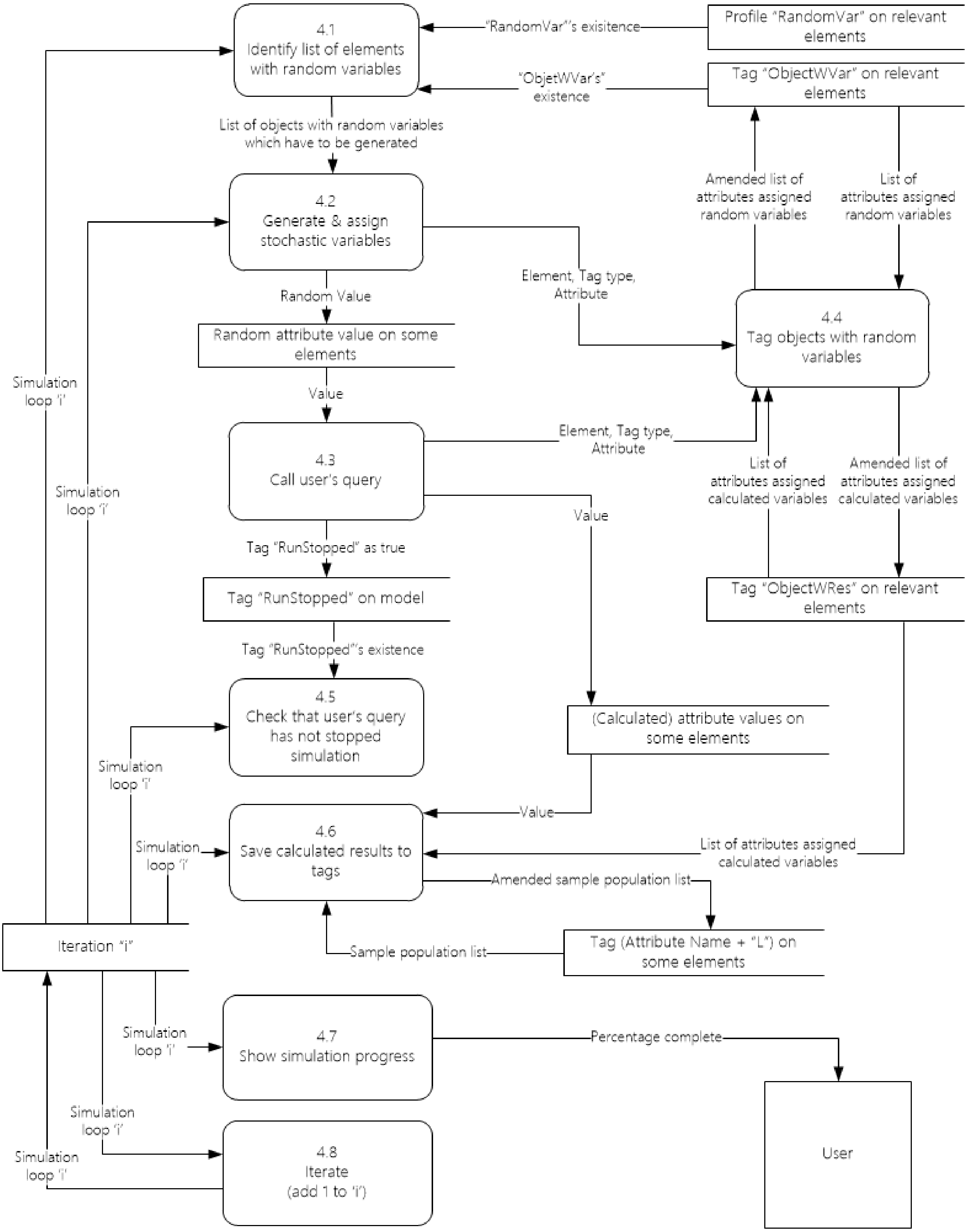


Fig. 5.9 Process 4: Run Simulation Data Flow Diagram

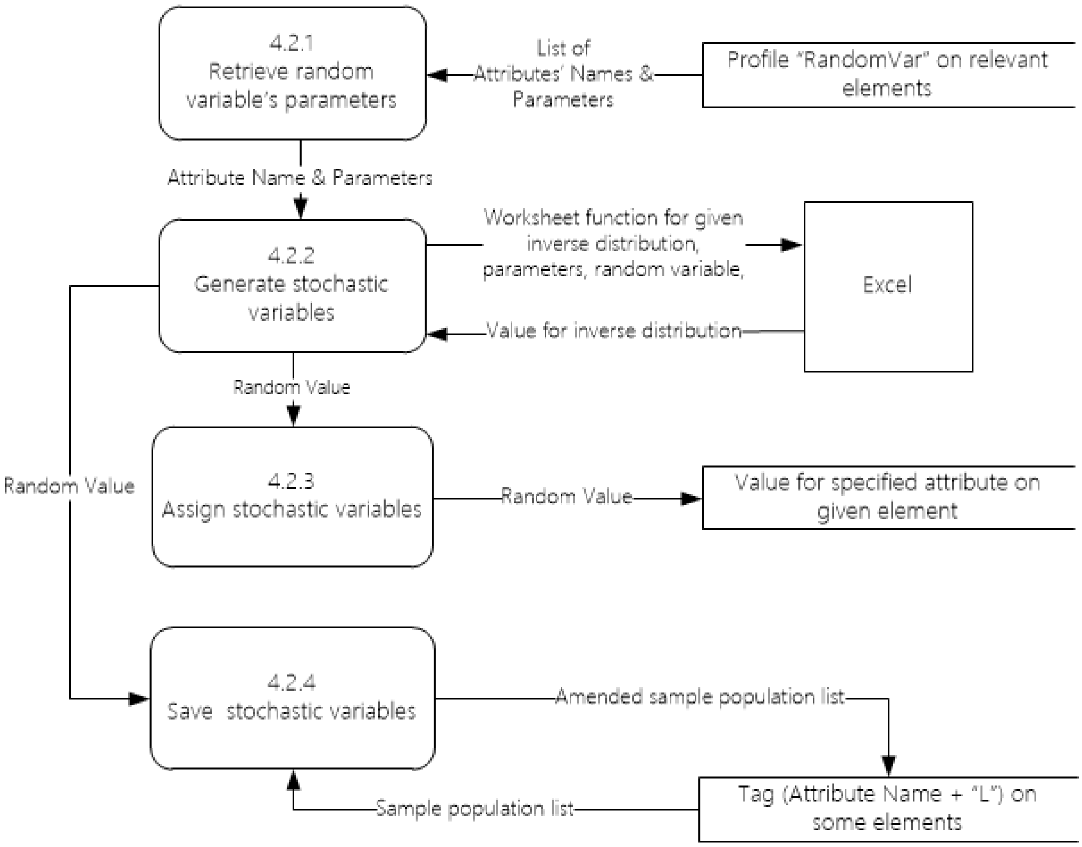


Fig. 5.10 Process 4.2: Generate, Assign & Save Stochastic Variables

5.5 Analyse Results

After the Monte Carlo simulation has completed, the final phase of the Monte Carlo analysis method can be completed, namely to analyse the results. Results produced by the simulation for variables of interest need to be examined in detail. This is achieved by providing the user with a range of descriptive statistics and the means to construct a range of possible graphical representations of the data. The unit of analysis are the population sample produced by each run of the Monte Carlo simulation for a specific variable of choice. Although the user's analysis script should pass a list of variables which the user's script calculates, not all of them will necessarily be of interest for further analysis. In particular, not all will be considered simultaneously, especially if there are a large number of them over multiple dimensions. With this in mind the analysis takes place on a variable by variable basis.

5.5.1 Histogram

In order to produce the histogram, the user has to insert a new bar chart in Architect (long press left click and select bar chart from the list of charts). Following this, the viewpoint labelled *Histogram* should be selected as the data source for the chart, from the Monte Carlo Simulation group on the viewpoints toolpane.

Process 1 After having selected the *Histogram* viewpoint, the user is asked to select the element to which the (calculated) random variable of interest to the analysis is assigned. The script then searches for the *ObjectWVar* and *ObjectWRes* tags on the element, and provides the user with a list of variables assigned to the element, for which sample populations of random variables were produced/-collected by the most recent Monte Carlo simulation. Each of these variables is assigned a number, which in turn is used by the user to select which of the variables he would like to analyse.

Process 2 Following the user's selection, the variables sample is retrieved.

Process 3 A range of descriptive statistics is calculated for the sample population of results from the simulation.

Process 4 These, along with a number of suggestions for the number of bins the histogram should contain, are provided to the user.

Process 5 The script then asks the user to select the number bins the histogram should contain (hence the list of suggestions) and uses this to calculate the bin width.

Process 6 The script also asks the user whether the histogram should be a normal frequency distribution, or a cumulative frequency distribution. Following the user's selection, the histogram is generated, indicating the percentage of results (cumulative or otherwise) contained within each bin, and each bin's midpoint. The user is also provided with a list of the bins upper and lower bounds.

The different graphical representations are useful for a range of analysis. The normal frequency distribution for instance can be useful to ascertain where the mode lies, what type of distribution could fit the data, whether the data is skewed, its spread, etc. The cumulative frequency distribution on the other hand is likely useful to ascertain what the probability is of an outcome being below a certain threshold, or visa versa.

The code used to calculate these descriptive statistics and produce histogram can be found in A.3.

5.5.2 Other Graphs

Besides the histogram, a range of other chart types can also be created from a variable's population sample. The illustrative case in the next section demonstrates a number of these.

5.5.3 Export results

The analysis of stochastic variables was designed to be straight-forward, whilst simultaneously providing a sufficient range of analysis dimensions and various options by which the user can make the analysis relevant to his needs. It is however, also recognised that other applications exist which can facilitate far more nuanced analyses, such as Microsoft Excel. To facilitate the export of data to such a program, an adapted version of the histogram script was is provided which produces a single column table, listing the results of Monte Carlo simulation for a chosen variable. The script is labelled *List Simulation Results for Variable* and is also grouped along the other viewpoints in the Monte Carlo Simulation viewpoints folder. See A.4 for the a copy of the code.

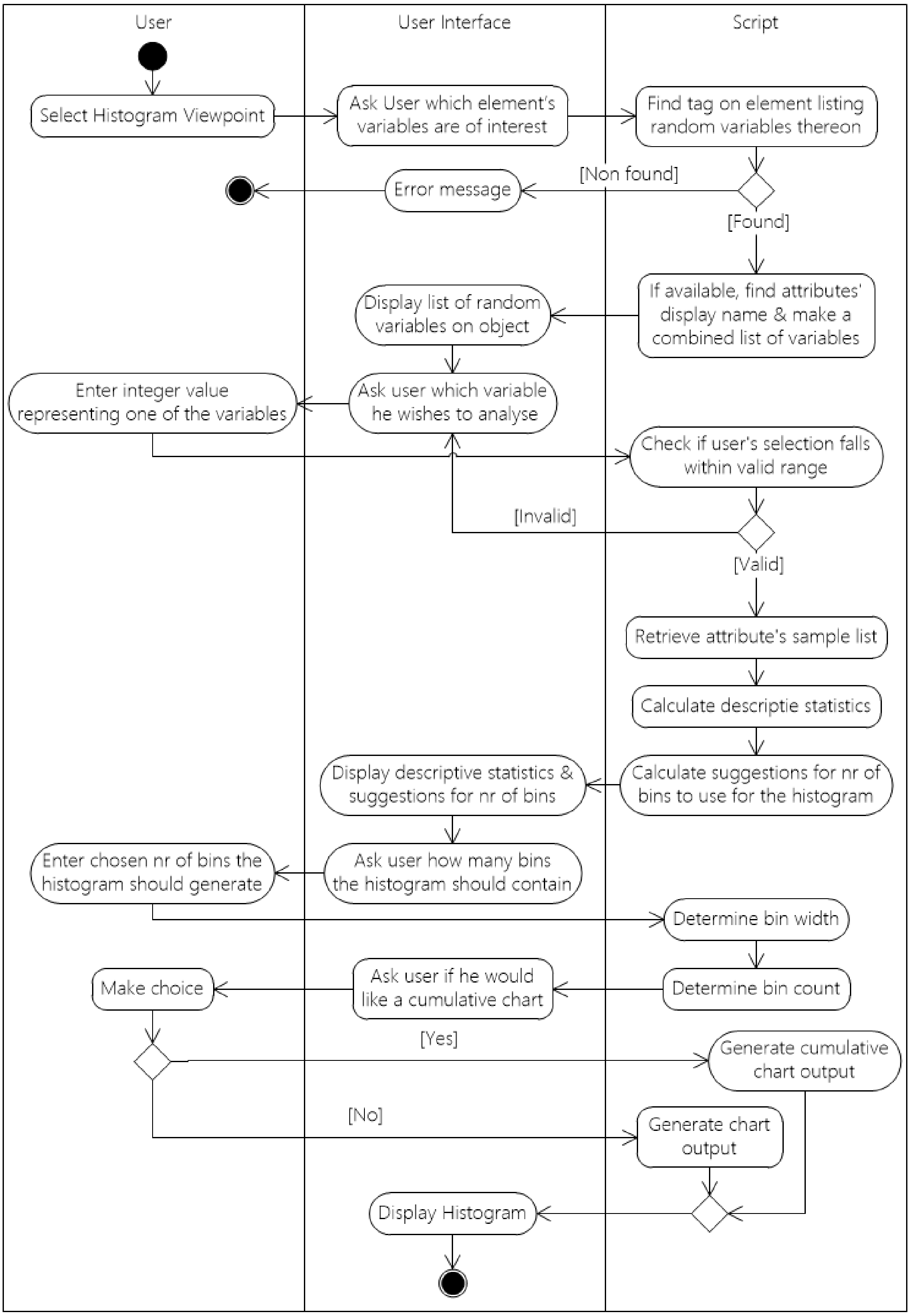


Fig. 5.11 Generate Histogram Activity Diagram

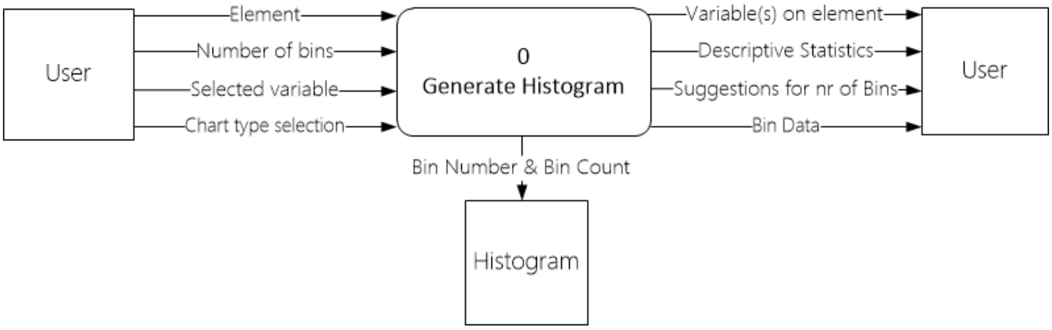


Fig. 5.12 Generate Histogram Context Level Data Flow Diagram

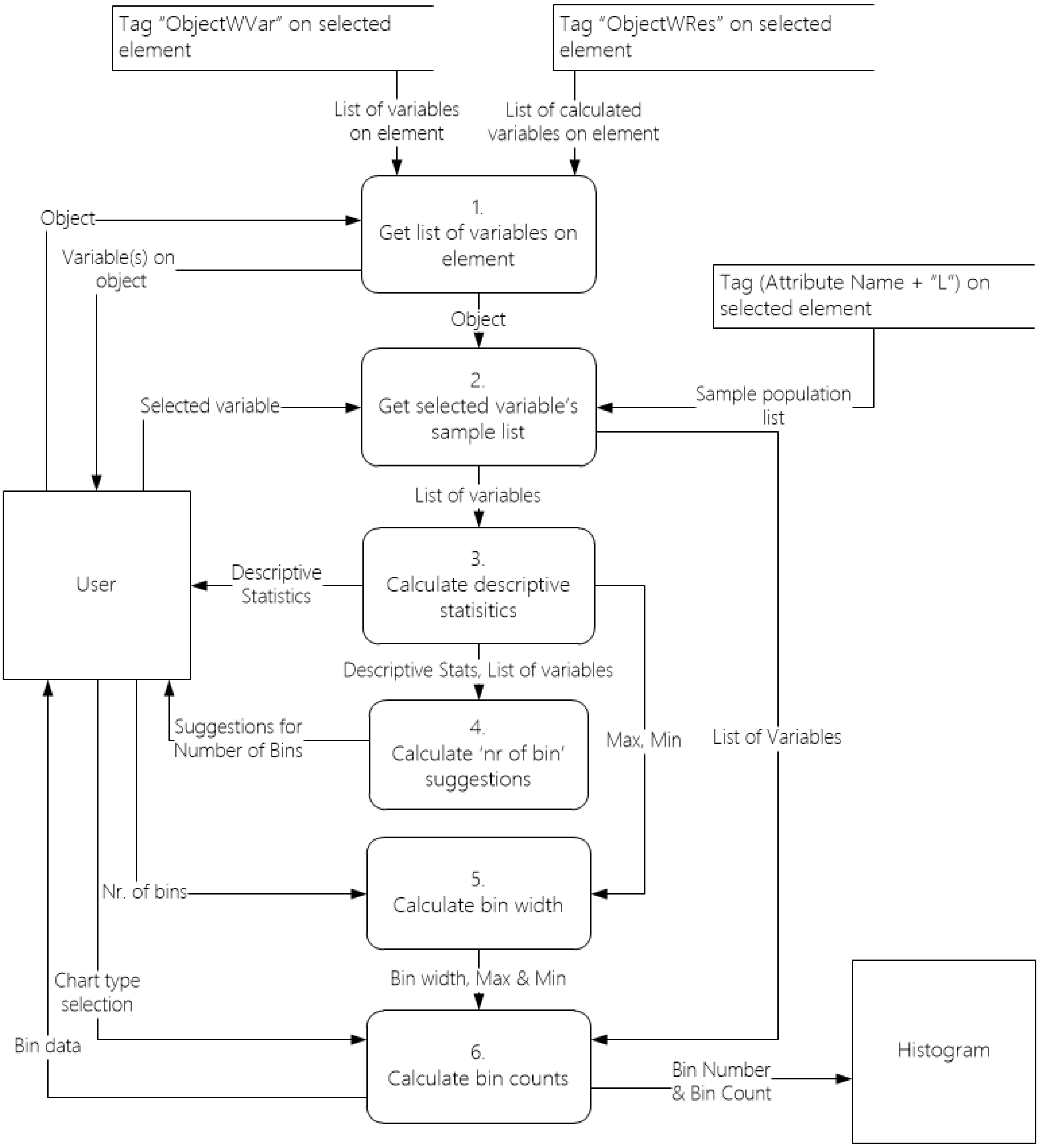


Fig. 5.13 Generate Histogram Level 0 Data Flow Diagram

5.6 Conclusion

This chapter described the artefact in terms of the four phases required for its implementation through to the analysis of results. In so doing, the third research question is successfully answered, as it was developed and implemented within Architect. This allows the artefact to be demonstrated by means of two illustrative cases in the following chapter.

Chapter 6

Artefact Demonstration

6.1 Introduction

The purpose of this chapter is to demonstrate the artefact described in the previous chapter. This is also done in preparation of fulfilling the third research objective (to evaluate the tool), which is described in the following chapter.

One primary reasons for developing an artefact specifically to enable Monte Carlo simulations was that the method can be used in a wide variety of contexts. Although the two case studies described here are both set within the baking industry, their respective objectives are entirely different. The first case study revolves around a Thai bank considering the introduction of a credit card service for the public. As part of the business case a financial analysis is required to consider the pay-back period and the net-present value (NPV) of the project over a five year span. A number of uncertainties exist regarding the projected costs and revenues and is ideally suited for a simulation study. As already inferred, the second case is also set within the banking industry. However this case does not primarily concern itself with any financial matters. Instead the availability of the banks' automated teller machines (ATM) are investigated, where the availability of the subsystems required by the ATM are uncertain.

6.2 Illustrative Case 1: Multi-period Financial Performance

6.2.1 Case description

Paisittanand and Olson (2006) describes a real case involving a bank in Thailand considering outsourcing a portion of the IT services required by a proposed expansion into the credit card service industry. The bank, referred to as the PPP-Bank, estimated that in-sourcing the required IT would require an investment of close to $ 40 million and require a significant increase in the IT departments staff for operational support. This required investments were deemed to high, leading them to consider other alternatives. Outsourcing all processing services and related operations would significantly reduce the required upfront investment, whilst allowing them to concentrate on all marketing related functions. The primary operations to outsource are as follows:

1. *Credit card issuing*, which involves receiving applications, storing data, verifying details and the physical issuing of the card.

2. *Credit card operations* to manage the account settlement and providing statements.

3. *Debt collections,* intensive process requiring multiple notifications and direct customer contact before finally initiating legal action and finalising each case individually.

4. *Credit Card System*, the back-end system supporting the service, including the required infrastructure, its applications, its installation and ongoing support.

PPP-Bank used its current customer base, information garnered from the industry and potential service providers to develop their financial model for a project feasibility study. The assumptions for the financial model are briefly discussed here:

1. *Number of cardholders*- PPP-Bank wants to roll out the service to 300 000 of its existing customers within the first year, growing that number by 20% per year.

2. *Active users*- Based on survey results, 46% customers said they would use the credit card, but PPP-Bank suspects only 70% of those 46% would be active users.

3. *Utilization volume*- The monthly transaction volume on credit cards is in Thailand is Bt2200. PPP-Bank suspects that the volume of transactions on their new credit cards will be 70% of the industry average. 80% of this being used for purchases, and 20% for cash withdrawals.

4. *Revolving volume* - Customers do not necessarily balance their credit cards at the end of the month. When considering the entire base of credit card users, the total volume of outstanding funds over a period of time is known as a revolving loan. It is estimated that 70% of the purchasing volume will end up as a revolving volume. The interest rate charged on the revolving volume is 18%.

5. *Non-performing loan (NPL)*- The industry sees about 24% of non-asset backed loans (such as n the case of credit cards loans) not being paid back on time, or *non-performing*. Of these it is estimated that 2% will have to be written off as bad debt in the first year after the debt collectors have gone trough their process.

6. *Operating costs*

   (a) *Funding costs*- PPP-Bank's cost of funds for the revolving volume is 2.8% per year.

   (b) *Outsourcing costs*- The selected service providers charges the following costs:

      i. Bt200 for each new card issued to customers
      ii. Bt10 per card per month to print and mail statement.
      iii. Bt480 per card per year for the operation of the back-end system
      iv. Debt collectors take a 20% cut of debts recovered.

7. *Marketing costs*- These are expected to reach 25% of the total revenue within the first year, decreasing to 20%, 15%, 15% and 10% per year the following years.

8. *Miscellaneous costs*- Other costs incurred to develop the service, and for other miscellaneous tasks are estimated to consume 10% and 5% of total revenue respectively.

9. *Credit card cost* - A credit card service provider (such as Visa, or Mastercard) charge the following amounts

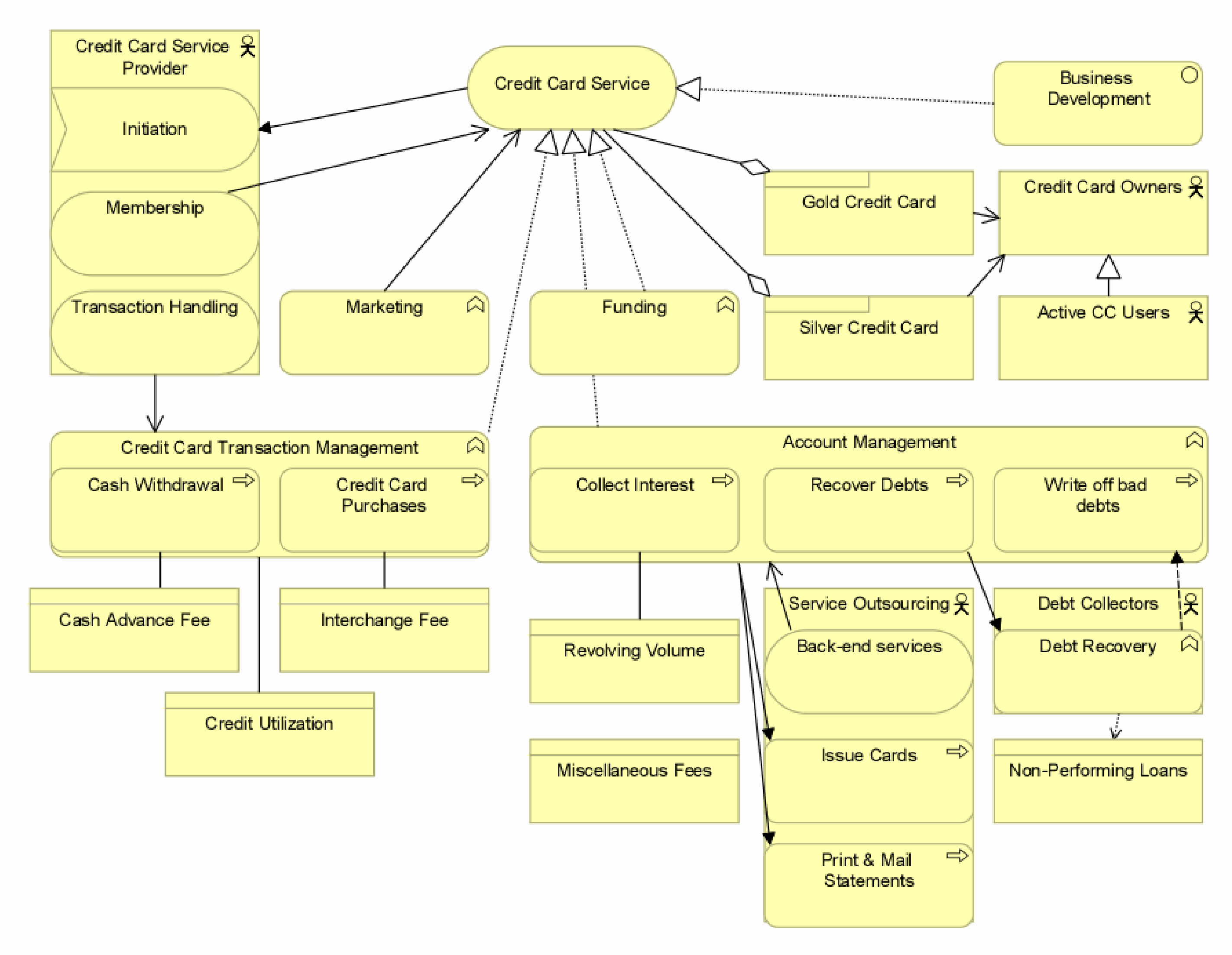
   (a) Bt8.6 million once-off initiation fee

(b) Bt41.67 membership fee per card per year

(c) Bt1.00 transaction fee

A summary of these values can be found in 6.1, in addition to a few other, self-explanatory items. An EA of the the new credit card service can be found in 6.1.

Fig. 6.1 Credit Card Service EA model



Due to the large number of uncertainties involved in the financial model, PPP-Bank's board of directors were dissatisfied with the initial financial analysis as based on the values above. To capture the uncertainties with respect to the future benefits and the impact of outsourcing IT services, it was decided to conduct a financial simulation

Table 6.1 Model assumptions for Credit Card Industry Business Case (Paisittanand and Olson, 2006).

Financial Model Element	Value	Description	EA Element
Assumptions			
Number of card holders	300,000	Accounts	Credit Card Owners
Gold card account holders	20%	Of card holders	Gold Credit Card
Silver card holders	80%	Of card holders	Silver Credit Card
Active card holders	46%	Of card holders	Active CC Users
Industry average for credit card usage	2200	Bt/month	Credit Utilization
Utilization volume (%)	70%	Of no. of active cards in 1st year, 80% in 2nd year and 100% from 3rd to 5th year	Credit Card Transaction Management
Purchasing volume	80%	Of utilization volume	Credit Card Purchases
Cash advance volume	20%	Of utilization volume	Cash Withdrawal
Revolving volume	70%	Of purchasing volume	Revolving Volume
Non-performing loans (NPL)	24%	Of revolving Volume	Non-Performing Loans
Discount Rate	6.25%		Discount Rate
Revenues			
Annual fee (gold card) (Bt)	1000		Gold Credit Card
Annual fee (silver card) (Bt)	600		Silver Credit Card
Revolving interest	18%	Of revolving volume/month	Collect Interest
Cash advance fee	3%	Of cash advance volume	Cash Advance Fee
Interchange fee from acquirer	1.20%	Of purchasing volume	Interchange Fee
PPP-Bank card operation cost			
Initiation-fee	8,600,000 Bt	One-off payment	Initiation
Member-fee	41.67 Bt	per card/year	Membership
Transaction-fee	1 Bt	Per transaction	Transaction Handling
Bad debt	2%	2% of NPL in 1st year. 3% in 2nd year, 5.0% in 3rd year, 7.5% in 4th year, and 10% in 5th year	Write off bad debts
Funding costs of revolving volume (%)	2.80%	Average per year	Funding
Marketing (%)	25%	25% of revenue in 1st year. 20% in 2nd year, 15% in 3rd year, 15% in 4th year, and 10% in 5th year	Marketing
Outsourcing cost			
Card issuing (new card)	200 Bt	Per card	Issue Cards
Statement printing and mailing	10 Bt	Per card/month	Print & Mail Statements
System	480 Bt	Per year/card	Back-end services
Debt collection fee	20%	Of recovered debt	Debt Recovery
Other expenses			
Business development fee	10%	Of revenue	Business Development
Miscellaneous	5%	Of revenue	Miscellaneous Fees

of the model using the Monte Carlo method. The choice of distributions, and their parameters are recreated in 6.2.

The exact results of the analysis found in Paisittanand and Olson (2006) differ

Table 6.2 Stochastic Nature of Model Variables for the Credit Card Service (Paisittanand and Olson, 2006).

Items	Distribution types	Parameters	EA Element
No. of active cardholder	Normal distribution	Mean 46%, SD 5%	Active CC Users
% Credit usage	Normal distribution in 1st year, lognormal in 2nd–5th	Mean 70%, SD 7% in 1st year Mean 80%, SD 8% in 2nd year Mean 100%, SD 10% in 3rd year Mean 100%, SD 10% in 4th year Mean 100%, SD 10% in 5th year	% Credit Utilization
Revolving interest	Normal distribution	Mean 18%, SD 2%	Collect Interest
Cash advance fee	Normal distribution	Mean 3%, SD 0%	Cash Advance Fee
Debt recovery rate	Normal distribution	Mean 30%, SD 3%	Debt Recovery
Interchange fee from acquirer	Normal distribution	Mean 1.2%, SD 0.1%	Interchange Fee
Member fee	Normal distribution	Mean 1, SD 0.1	Membership
Bad debt	Triangular distribution	Min. 2%, likeliest 5%, Max. 10%	Write off bad debts
Funding cost	Normal distribution	Mean 2.8%, SD 0.3%	Funding
Marketing cost	Normal distribution in 1st year, lognormal in 2nd–5th	Mean 25%, SD 2.5% in 1st year Mean 20%, SD 2% in 2nd year Mean 15%, SD 1.5% in 3rd year Mean 15%, SD 1.5% in 4th year Mean 10%, SD 1.1% in 5th year	Marketing
Card issue	Normal distribution	Mean 200, SD 20	Issue Cards
Statement and printing	Normal distribution	Mean 10, SD 1	Print & Mail Statements
System	Normal distribution	Mean 480, SD 48	Back-end services
Debt collection fee	Normal distribution	Mean 20%, SD 2%	Debt Recovery
Business development fee	Normal distribution	Mean 10, SD 1	Business Development
Miscellaneous	Normal distribution	Mean 5, SD 1	Miscellaneous Fees

from the ones discussed in the following section as insufficient information was provided regarding the underlying model for the cost analysis. To resolve this certain simplifications had to be made (but could easily be introduced if the missing information were made available). Although only the most pertinent information required for this case study were recreated here, Paisittanand and Olson (2006) provides further background information and other discussion points for the interested reader.

6.2.2 Implementation

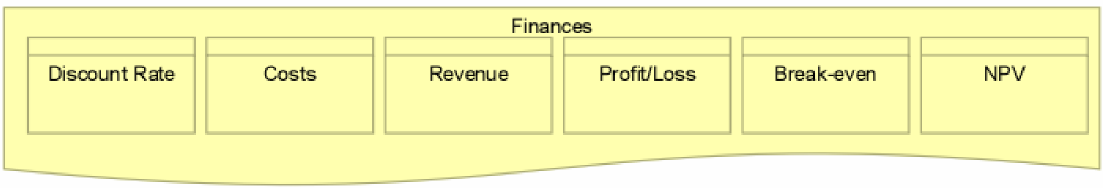
In order to capture the financial mode, each of the financial parameter discussed above had to be assigned to an object within the EA. Although more than one parameter could have been assigned to any one object, this would have negatively impacted the clarity of the model, made it difficult to change parameters (for further analysis), and required a complex set of profiles within Architect with which to assign each of the parameters to objects.

Three profiles were developed within Architect for the purposes of this analysis.

Two of these to capture all of the required parameters of the financial model, and one to capture the results of the simulation runs.

The first profile, labelled *NPV (simple)* is used to capture one single parameter, used in cases where the parameter is a constant. The next profile, *NPV (extended)*, is used to store parameters that change on a yearly basis. Along with the *RandomVvariables* profile introduced in the previous chapter, the stochastic financial model can be captured in the EA model. The last profile, *NPV (results)* is used to store the results of each calculation step. The *NPV (extended)* and *NPV (results)* profile each have six attributes, to store the parameters for year 0 through to year 5 used for the financial planning horizon in the model. In addition to the profile used to capture the results, in most cases only one of the other two profiles (the simple or extended NPV profile) had to be assigned to each of the objects in the EA model. The exceptions are in cases such as that of the *Credit Utilization* object where both the industry average for credit card usage, and the yearly projected utilization volume (as a % of the industry average) was captured. In cases where more financial model data could be assigned to a single EA object, the modeller has to decide how to split these over more than one object.

Fig. 6.2 Bank Case EA Financials



As can be seen in figure 6.2, a set of EA business objects was also created to capture all of the intermediate outcomes in the sequence of financial calculations.

The calculations were hard coded into a script, a copy of which can be found in Appendix A.5. The simulation was run 160 times, after which the results were analysed, as discussed in the following section.

6.2.3 Results Analysis

Although a host of financial and statistical metrics can be considered, only the most important outcomes are briefly discussed here which would be of use for PPP-Bank in deciding whether or not to pursue the business case.

The first financial metric to be considered here is the projected net present value of the venture over a five year period. A list of descriptive statistics can be seen in figure 6.3. From it it can be seen that the mean of teh NPV is around Bt233 million. However the standard deviation is also significant at Bt191 million. From figure 6.5 in turn it can be seen that there is a 90% probability that the NPV over the first five years will be greater than zero.

Fig. 6.3 Year 5 NPV Descriptive Statistics

Sample Mean: 232 743 569, Sample Standard Deviation: 191 252 408, Skewness estimator: 0.277
Min: - 276 736 822, First Quartile: 99 651 562, Sample Median: 233 546 663, Third Quartile: 349 795 421, Max: 782 984 655
Interquartile range: 250 143 859, Range: 1 059 721 478

Fig. 6.4 NPV Histogram for Year 5

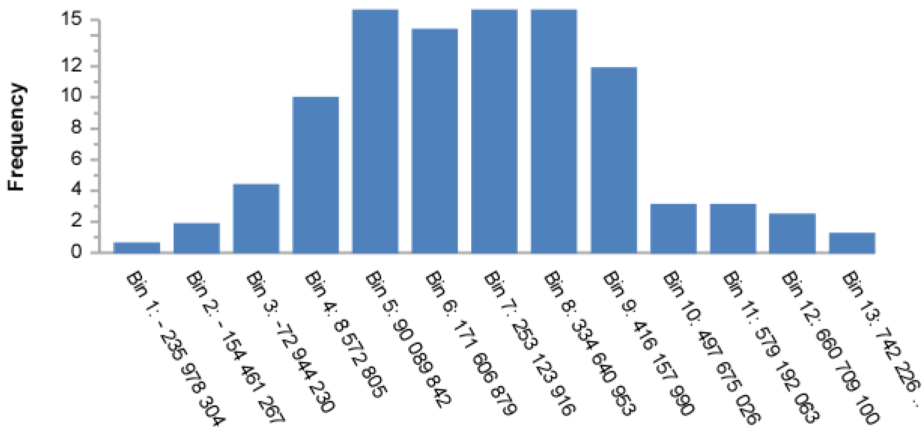


Fig. 6.5 NPV Probability Chart for Year 5

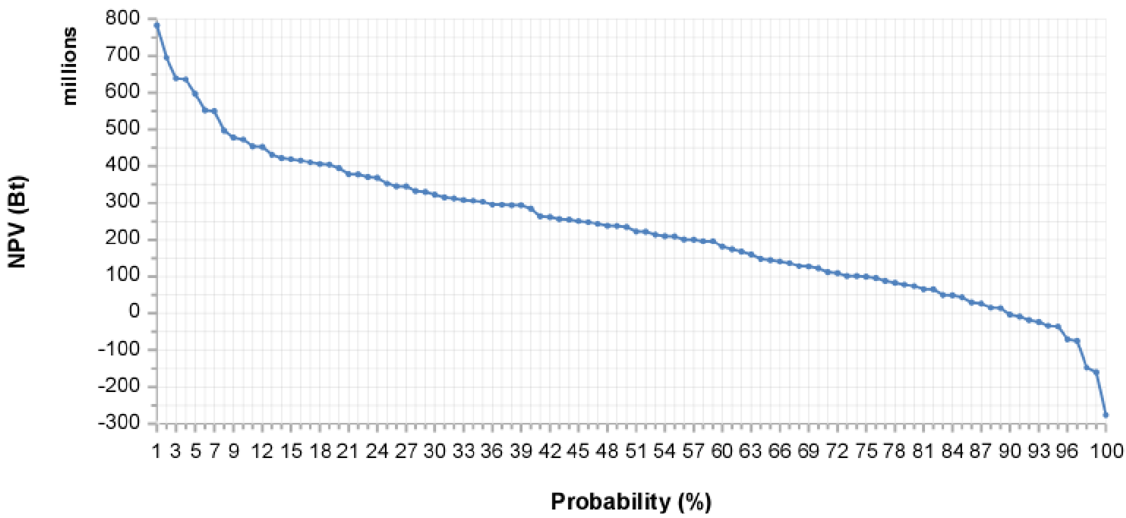
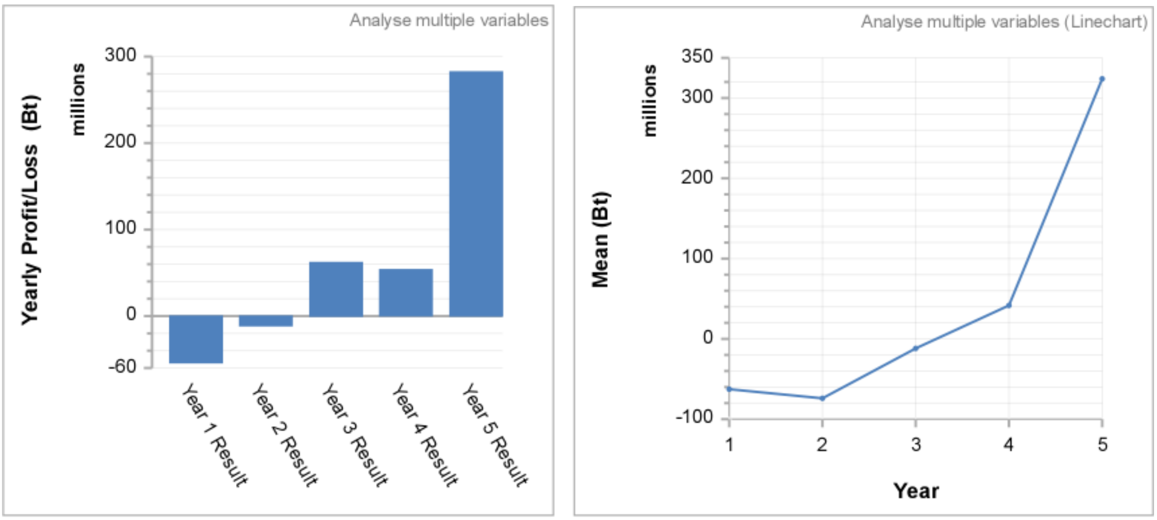


Fig. 6.6 Mean NPV per year

Year 1 Result. Sample Mean: -59 504 376, Sample Standard Deviation: 30 402 918
Year 2 Result. Sample Mean: -69 543 776, Sample Standard Deviation: 120 357 845
Year 3 Result. Sample Mean: -17 702 705, Sample Standard Deviation: 142 145 150
Year 4 Result. Sample Mean: 24 274 940, Sample Standard Deviation: 180 532 211
Year 5 Result. Sample Mean: 232 743 569, Sample Standard Deviation: 191 252 408

From figure 6.7 it can be seen that the mean of the profit estimates becomes positive in year three of the project. The cumulative profit, or break-even point is in year four. The standard deviation of these values could be of interest, and can be found in figure 6.8 and 6.9. As can be seen the standard deviation for these two points is Bt 123 million and Bt 217 million in turn. For further analysis the interested party might wish to generate probability charts for these two years to evaluate whether the predicted values fall within in acceptable ranges.



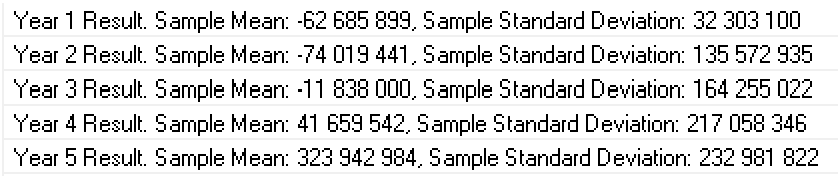
(a) Mean Yearly Profit/Loss (b) Mean Break-Even Analysis

Fig. 6.7 Yearly analysis

Fig. 6.8 Mean Yearly Profit/Loss Descriptive Statistics

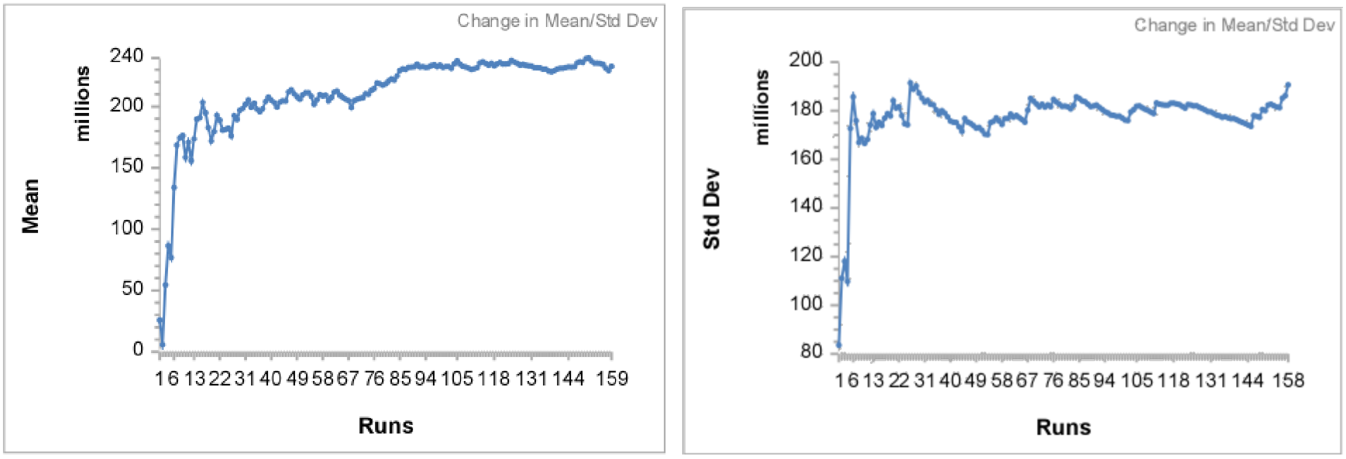
Year 1 Result. Sample Mean: -54 085 899, Sample Standard Deviation: 32 303 100
Year 2 Result. Sample Mean: -11 333 541, Sample Standard Deviation: 134 858 110
Year 3 Result. Sample Mean: 62 181 440, Sample Standard Deviation: 123 246 683
Year 4 Result. Sample Mean: 53 497 543, Sample Standard Deviation: 149 917 393
Year 5 Result. Sample Mean: 282 283 441, Sample Standard Deviation: 83 016 772

Fig. 6.9 Mean Break-Even Analysis Statistics



Year 1 Result. Sample Mean: -62 685 899, Sample Standard Deviation: 32 303 100
Year 2 Result. Sample Mean: -74 019 441, Sample Standard Deviation: 135 572 935
Year 3 Result. Sample Mean: -11 838 000, Sample Standard Deviation: 164 255 022
Year 4 Result. Sample Mean: 41 659 542, Sample Standard Deviation: 217 058 346
Year 5 Result. Sample Mean: 323 942 984, Sample Standard Deviation: 232 981 822

The last point of interest discussed for this case study are the changing mean and standard deviation over the simulation runs. Their values have been plotted in figure 6.10a and 6.10b below. It can be seen that the mean reaches a steady state. However, the standard deviation still exhibits large changes, and consequently it should be considered to re-run the simulation with a greater number of repetitions.



(a) Change in Mean

(b) Change in Standard Deviation

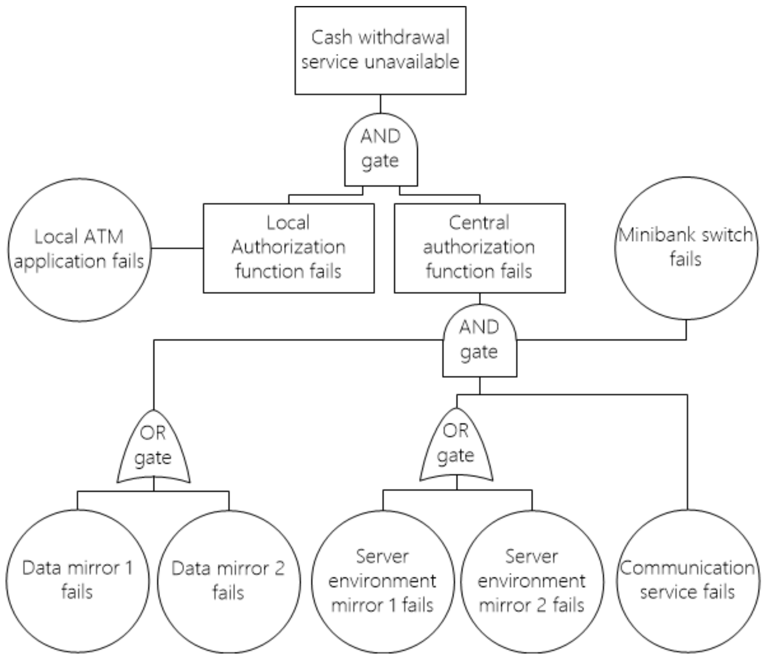
Fig. 6.10 Change in values for Year 5 NPV

6.3 Illustrative Case 2: Service Availability

6.3.1 Case description

The second case considered has been adapted from Buschle et al. (2013) and considers the availability of a bank's cash withdrawal service. It too is based on an actual case, this time a Nordic bank, whose service in question is available to some 10 million customers. A basic fault tree for the system can be found in figure 6.11, whilst its EA model is seen in figure 6.14.

Fig. 6.11 ATM fault tree

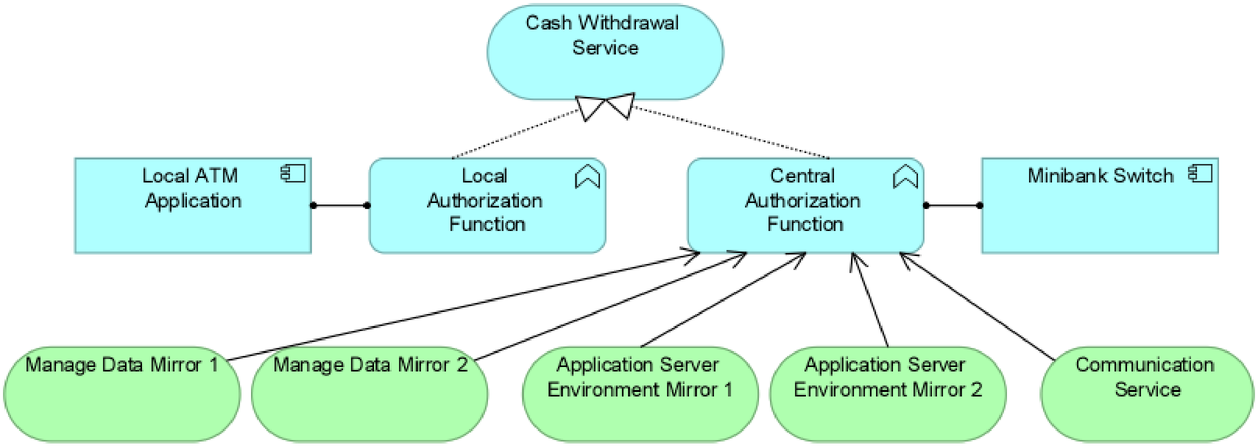


Besides relying on the local application, each ATM machine is also connected to a distributed mainframe consisting of various hardware, network and application components providing various degrees of redundancy. These are not modelled in detail though, instead a high level overview is provided particularly concerned with this redundancy, which is why a fault tree is an appropriate model to use, making it easy to identify such them. The structure of the fault tree, along with the input variables have to be incorporated on the EA somehow, which is discussed next.

6.3.2 Implementation

Two profiles were created within Architect for the purposes of this analysis. The *Availability Analysis* contains a single attribute labelled *Availability of Service*, which also has to be provided as the name of the random variable in the *Random Variable* profile. In order to model the dependencies between the objects as represented in the failure tree, a means of indicating how an objects availability is dependent on others' is required. Here one should also be able to indicate whether any redundancy exists or not. In other the sequence of *OR-gate* and *AND-gate*'s. For this purpose another profile is introduced which can be linked to relationships modelled on the EA. The profile, *Component Availability Dependency* as two attributes, *Redundancy available* &

Fig. 6.12 Cash withdrawal service EA (Buschle et al., 2013).

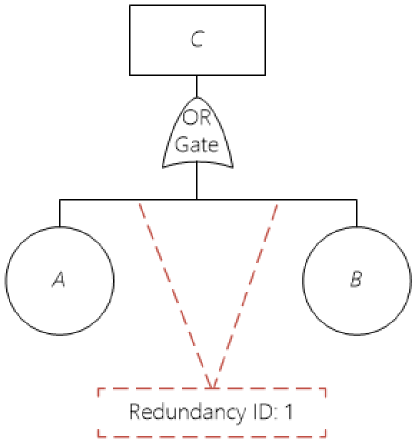


Unique Redundancy ID. By linking the profile to a relationship, it is indicated that the object is dependent on the object from the which the relationship originates. By setting the boolean attribute *Redundancy available* to true, it indicated that a redundancy exists (in other words, an *OR-gate*). To indicate which objects provide redundancy, the user as to provide a unique ID (any string of values) to the *Unique Redundancy ID* attribute, and duplicate this ID on the relationship of the second object providing the redundancy. In other words, if two objects, A & B, are connected *to* another object, C, and these relationships have the same *Redundancy ID*, either A & B have to be available for C, as illustrated in figure 6.13.

Although Buschle et al. (2013) use Metropolis-Hastings sampling to calculate the service's predicted availability, this can also be done via the Monte Carlo method. Unfortunately the parameters used for the calculations were not made available in Buschle et al. (2013), as such a fictitious range of parameters were used. To simplify this example, the usual exponential or Weibull functions used to model time to failure (MTTF) and time to repair (MTR) where not used. Instead if was decided to focus the final result of interest, i.e. the expected availability of the service. With this in mind it was assumed that the expected variability of the availability of the components could be represented with triangular distributions.

The model as such uses a crude, or 'direct' approach to system availability prediction (Korver, 1994). It is however, sufficient for the purposes of this demonstration. The script ascertains the availability of each object by considering each of the relationships going to that particular object. If the relationship indicates the object on

Fig. 6.13 Redundancy relationship

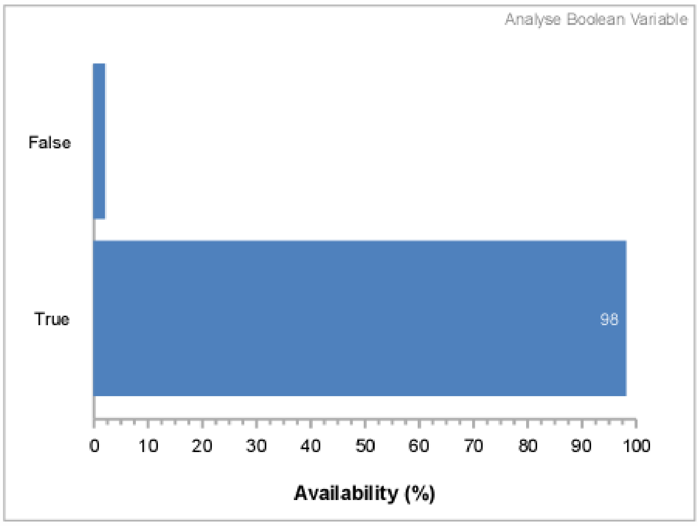


the other end influences the availability of the first object, that objects availability is ascertained first. If a sufficient combination of objects is available (i.e. functioning), to support the functioning of the first object in question, then the object's own availability probability is considered (if it has one). This is done in a purely boolean fashion, in other words an object is either available (true), or not. This boolean value is determined by means of a randomly generated number. If the random number is smaller than availability probability for that object (which in turn can also be a stochastic variable determined by an inverse distribution), then the object is available for that simulation run (true), or else not (false). The availability of any one object is then determined by considering the total proportion of simulation runs for which it was available (true).

6.3.3 Results analysis

As mentioned above, this particular analysis only returns boolean results for the availability of objects for each iteration of the simulation. For this particular case study, the availability of the cash service was of interest. Having run a Monte Carlo simulation of the model for 250 iterations, it was discovered that the availability of the service was expected to be 98% (see figure 6.14). The relevant decision makers can then use such information to determine whether or not this is sufficient. In the case of a bank, if any ATM is out of service, it could have a significant impact on their customers' satisfaction levels, and even the amount of revenue they generate from any fees they charge for the service. The EA model of the service can then be altered to investigate how the addition of any more redundancy would increase the expected availability of the service.

Fig. 6.14 Cash withdrawal service availability



Although not currently included in the model, it could be extended to include the availability of communication channels in the analysis with a few additions to the analysis script, and linking the already available *Availability Analysis* profile to the relationships of interest (Such as the use-by-relationship, in the cash withdrawal service model).

6.4 Conclusion

This chapter described two case studies to demonstrate how the artefact can be used to analyse the behaviour of an enterprise captured by an enterprise architecture model. As previously explained, the Monte Carlo method allows architects to incorporate uncertainties into quantitative analysis, that exist when a new venture is undertaken.

By incorporating the means to analyse different enterprises (or in some cases more accurately said systems, such as with the second case) within the EA modelling tool, it removes the need to re-create the models in an external application. Additionally, any changes made to the architecture (whether as a result of the analysis or otherwise), can then quickly be re-analysed. Naturally the speed and ease with which this can be done depends on how the analysis script and its data was incorporated into the EA model.

One caveat regarding the first case study has to be mentioned. Aspects of the model, such as the profiles used to provide starting input parameters and store multi-period values, can generically be used by other NPV financial models. However inter-dependence between the variables and sequence of calculations were hard coded into an analysis script. Therefore and changes to the existing model parameters can easily be analysed, but any changes to the structure of the model also have to be incorporated into the script. As the purpose of the these demonstrations was to demonstrate the artefact, and not specific analyses, this was not deemed as a problem for the purposes of this project. This point is discussed further in the concluding chapter.

Although untested, the author thinks the structure of the model used for the NPV calculations used for the first case study can be constructed by a script. It would require two additions to the EA model within Architect. Firstly, the existing NPV profiles have to be extended with attribute which can be used to specify the value types on the objects. The analysis script needs to known whether the values are fractions, costs, revenues or some other indicator. Additionally another profile needs to be introduced which can be added to the relationships between the objects on a model to capture the type of calculations which need to occur between the attributes of the two objects. In other words, whether the values on two respective objects need to be added, subtracted, multiplied, etc. In a similar fashion to what the script for the second case study does, the analysis model can then be formulated by the script by investigating the EA model and the user specified attributes on its various objects.

Another limitation introduced in Architect are the limited options currently available for the graphs. This is generally important in the business realm, and particularly so in analytics as the graphics are often used to convey necessary information to decision makers. In it's current form some graphs can be difficult to interpret and cannot adequately be used to convey granular data.

Chapter 7

Evaluation

7.1 Introduction

This chapter discusses the method used to, and results of the evaluation of the artefact. This is done to complete phase five of the Design Science Research Method, and to fulfil the third research objective, namely to *evaluate the tool according to its purpose, its usability, and usefulness in the business realm.*

The evaluation method is based on the evaluation criteria set out in March and Smith (1995) and particularly focused on the investigating whether the objectives and requirements set-out in Chapter 3 ad 4 for the artefact, were fulfilled.

In this chapter then, the evaluation method is discussed first, followed by its results and a discussion of its implications.

7.2 Method

7.2.1 Evaluation Criteria

The framework and evaluation criteria presented in March and Smith (1995) were developed to promote IT research that is both relevant and effective. Due to the particular focus in this project on the usefulness and ease-of-use of the artefact for business users, it was deemed as good starting point for the evaluation of artefact. The 14 criteria do not apply to the entire spectrum of the artefact though, of which March and Smith (1995) identifies four (see figure 7.1):

Constructs - underpin the theoretical foundation of the domain, conceptualizing the key concepts to describe the problem and specify solutions.

Model - expresses the relationships that exist between the constructs in a set of propositions or statements.

Method - based on the constructs and model, a set of steps required to execute a task.

Instantiation - the realised artefact in its environment.

Table 7.1 Evaluation criteria for Design Science Research artefacts (March and Smith, 1995)

	Construct	Model	Method	Instantiation
Completeness	■	■		
Ease of use	■		■	
Effectiveness				■
Efficiency			■	■
Elegance	■			
Fidelity with real world phenomena		■		
Generality			■	
Impact on the environment and on the artefact's users				■
Internal consistency		■		
Level of detail		■		
Operationality			■	
Robustness		■		
Simplicity	■			
Understandability	■			

As the artefact in this project is only a prototype demonstrated by means of two illustrative scenarios, only a subset of these criteria were found to be relevant for the purposes of this evaluation (Peffers et al., 2012). Further, the artefact was evaluated by means of qualitative interviews with personnel from BIZZdesign, further limiting the range of criteria which can possibly be evaluated, and to what extent.

Completeness of the constructs and model, were not deemed of primary importance as the interviewees were not Monte Carlo experts. *Ease of use, simplicity* and *understandability* were considered important so as to ensure that it was clear how the Monte Carlo simulation could be implemented and executed via the artefact. The *effectiveness* of the realised artefact was only evaluated in relation to the original objective of this research study, in other words whether or not the artefact accomplishes its goal of making EAs executable. The *efficiency* of the realised artefact was not evaluated as this would require some kind of performance comparison with other tools able to provide the same functionality, but the interviewees were asked about the perceived efficiency of the method. In other words the steps required to set-up, execute, and analyse results of a Monte Carlo simulation via the artefact. *Elegance* was only considered at a very high level of abstraction, again because the interviewees were not domain experts. *Fidelity with real world phenomena* is important as the artefact's purpose is to meet a real business need. *Generality* is an important metric as one of the objectives for the artefact was that it could support a wide range of different analytics. However, as is the case with *fidelity with real world phenomena*, this could only be evaluated as it was perceived by the interviewees. *Impact on the evironment and on the artefact's users* could not be assessed as it was not demonstrated in a real business setting. *internal consistency* and *level of detail* was not deemed important as the Monte Carlo method itself as a model is well established. The consistency and level of detail of the particular analysis developed for the illustrative examples was not of importance. *Operationality* was considered together with *ease of use*. Although similarly important to *generality*, *robustness* was not evaluated as the limited illustrative scenarios did ot allow this.

The final set of criteria used for this evaluation were again reduced by grouping similar criteria together. This resulted in the following list of criteria: *(a) Understandability & effectiveness, (b) ease of use, simplicity, elegance & operationality,(c) fidelity with real world phenomena & generality.*

7.2.2 Interviews

In order to demonstrate how the artefact would be used in a business setting, the required set-up, execution steps and analyse means were described to the interviewees (the three primary phases of the method). The demonstrations were done by means of the two illustrative cases discussed in the previous chapter.

The three interviews were conducted with personnel from BIZZdesign, all of whom had some familiar with the Monte Carlo method, and necessarily know the EA field. In preparation for the interview a list of questions was compiled designed to address each of the areas identified as important to the evaluation, which can be found in Appendix B. Due to the nature of qualitative interviews, these questions were primary used as a guideline, and not answered on a one-by-one basis.

The first interviewee, now working at BIZZdesign's sisters company had just completed her masters in business administration, and had made use of the Monte Carlo method for her thesis too. Her experience in using the method was therefore valuable to evaluate how the artefact allowed users to make use of it. The second interviewee is a senior research consultant at BiZZdesign, specialised in ArchiMate as well as business models and performance analysis. As the artefact could potentially find great use in the latter two fields, his input was invaluable. The third interviewee is the product manager at BIZZdesign. His perspective on the artefact, particularly from the customers point of view was therefore invaluable, particularly to predict if it would have any uptake.

7.3 Results

7.3.1 Understandability & Effectiveness

The purpose and of the artefact, i.e. to make EA models executable in order to simulate and predict the behaviour of the EA was understood and appreciated by all of the interviewees. As they all already knew the basic principles of the Monte Carlo method, it was easy to explain how the artefact functioned and how the results were attained. The interviewees generally conceded that the artefact could be used in real world business settings, but that the number of businesses who would be able and willing to do so would probably be rather small. Although the virtuous and use of analytics have been explained in Chapter 3 of this document, the use of the more advanced analytics seem to be still rather limited to particular domains, or to business with necessary capacity knowledge and resources to develop them. Not only that, but although the scripting capabilities within Architect are commonly available, they are not commonly used by clients either.

The interviewees all had the opinion that the artefact would have far more use if it was developed for specific applications. This would allow business users to set-up, execute a simulation and analyse without having to develop any scripts themselves, which would significantly lower the barrier to entry required to use the artefact.

7.3.2 Ease of use, Simplicity, Elegance & Operationality

The consensus amongst the interviewees was that even though the steps required to set-up, execute and analyse the simulation results were quite clear, it was by no means the easiest of tasks to complete. On the one hand the set-up of the model via the

Random Variable profile could be rather intimidating due its complex structure and the intensive involvement required by the user. The remaining tasks to execute and analyse the results are also rather user intensive, in themselves requiring a significant amount of set-up via numerous steps (such as placing the charts, selecting the objects, then the variables of interest, etc.).

Once the tasks were completed tough, the range of analyses capabilities provided to the user were found to be useful, providing a range of necessary statistics and graphics. Here interviewees were of the opinion that a few more options would be useful, in order to be able to analyse the results on a more granular level. For example, to be able to find out exactly what the probability is that the NPV is greater than a certain amount x, in the the first illustrative case.

7.3.3 Fidelity with Real World Phenomena & Generality

The ability to use the Monte Carlo method in a wide range of settings was generally recognised, but in general particular emphasis was always placed on its use with respect to financial analyses as this is a common business need. As such, even though it is recognised that it can be used in a wide range of applications, its greatest use will probably be found in this area.

7.4 Conclusion

By means of qualitative interviews, the artefact was evaluated according to three dimensions: Its *(a) Understandability & effectiveness, (b) ease of use, simplicity, elegance & operationality,(c) fidelity with real world phenomena & generality.* Given the requirements for the artefact, the potential usefulness of the artefact to business users was of particular concern.

The results of the evaluation were generally positive as the purpose and potential business applications could clearly be identified. Where the artefact came short was on its ease-of-use as it requires numerous user intensive steps to implement and execute. The results of the analyses, as presented via the two example cases were found to be useful for further decision making too. The most significant caveat of the artefact (as opposed to advantage) is that it still requires the user to write scripts himself. Although the initial purpose of the artefact was to enable a wide range analytics, it might have been more useful as a tool enabling a few select set of analytics instead.

Hereby the fifth phase of the Design Science Research Method was completed, fulfilling the third research objective, namely to *evaluate the tool according to its purpose, its usability, and usefulness in the business realm.*

Chapter 8

Conclusion

8.1 Introduction

In this final chapter, the results of the previous chapters will briefly be considered in light of the original research objective, which was to:

> Determine and demonstrate how enterprise architectures can be used as the foundation for business analytics.

The contributions this research project has made to theory and the practice will also be considered, along with its limitations and recommendations for further research and practice.

8.2 Research Questions & Research Objective

The research questions were answered at the end of each of the respective chapters. Here follows an account of primary outcomes from each of the research questions, specifically as related the DSRM, and finally serve to demonstrate how the research objective was met.

8.2.1 Research Question 1

- Can enterprise architecture models be used as the foundation for Monte Carlo simulation based business analytics, and if so what problem would this aim to solve?

1. What is understood under the term *enterprise architecture analytics*, and what is its purpose? Are there any existing standard guidelines or methods specifically designed to develop simulation models of enterprise architectures models?

2. What are business analytics? Is there a synergy between business analytics and enterprise architectures which would encourage and facilitate further investigation into this topic? How are Monte Carlo simulations used for business analytics?

With the first research question, the goal was to identify if EAs models, which are static, can feasibly be used as the foundation for business analytics and how best to build upon what had been done. It was found that a various number of analytics can be performed on EAs, however little had been done in the way of using EA models to conduct quantitative analysis. Furthermore, when considering executable EA models, or in other words, models which have been enriched with the necessary properties such that they can be used for simulations, a few distinct research efforts have been under way. However, all of them were rather complex, typically using further metamodels to develop executable EA models. An opportunity was seen to develop an artefact which would enable business analytics, without the need of a complex metamodel, making use of the Monte Carlo method.

8.2.2 Research Question 2

> Given these findings, what would be the requirements of a technical artefact, designed to enabled Monte Carlo simulations, based on using enterprise architecture models?

Given the opportunity identified from the first research question, a set of requirements had to be identified, which the artefact had to meet in order to be functional and usable. In essence the requirements specified that the artefact should enable a four phase method, so that analysis could be integrated, set-up, executed and lastly analysed. This should be done in a user friendly manner, limiting the number of unintentional errors.

8.2.3 Research Question 3

> How could an artefact be developed an implemented to meet these requirements within the enterprise architecture modelling tool, Architect?

Following this, the artefact was developed for the ArchiMate modelling, and specifically implemented in the EA modelling tool Architect by means of one extensive profile definition which can be used with any stochastic model, and numerous scripts required to execute the actual calculations of the simulation and the analysis of results.

8.2.4 Research Question 4

> Does artefact tool fulfil its original purpose, is it user-friendly, and useful within the business realm?

From the very beginning, one of the primary goals for the artefact was to make it as useful as possible. Naturally this is difficult to do with a prototype, and the artefact certainly has some problems with its usability as it require the use of a slightly convoluted profile to specify the random variables, and numerous steps to execute and analyse the results. The tools potential usefulness was recognised, however it was generally thought that a range of specific analyses would be more useful to the end-user, as they typically do not possess the know-how to develop analysis.

8.2.5 Research Objective

Finally, the research objective can be considered:

> Determine and demonstrate how enterprise architectures can be used as the foundation for business analytics.

By developing and implementing an artefact that enables a range of business analytics, all without leaving the front-end user environment within Architect, it was demonstrated that EA models have great potential when acting as the foundation for BAs. Given the synergies that exist between the two fields identified in *RQ 1*, and the success of the artefact, the research objective has been fulfilled.

8.3 Contributions

8.3.1 Theoretical

Although a few other researchers have used EA models as the bedrock for simulation studies, or developing simulation capable artefacts, these are typically done in purpose built applications, using simulation specific modelling languages, or at least a purposefully extended version of an EA modelling language. Although analyses require the user to develop an analysis model, this model should be in the form of an EA model, allowing the EA model to be used as the basis for the simulation directly. By providing this capability, the number of steps required to develop and implement a Monte Carlo simulation, based on an EA model, is reduced significantly. These savings in effort not only occur during the development, simulation and analysis of the EA model, but also when changes are to the model, as changes can easily, or near automatically, be incorporated into the analysis model too. This is allows different design alternatives, and their projected behaviours, to be created quickly.

The focus of this project was on the technical artefact itself, leaving the development and implementation of the models up to the user. Although this is could be considered crude, it fits well with the nature of the Monte Carlo method, which can be used and applied in a variety of ways.

8.3.2 Practice

By enabling Monte Carlo simulations to be done within an EA modelling tool, the extra effort required to develop a simulation model from existing EA model in another application is removed. By showing one way in which this can be accomplished, it lay groundwork for the development of similar artefacts.

8.4 Limitations

A number of theoretical limitations can also be identified here. These are mostly related to the fact that the primary purpose and focus of this project were on the development of a technical artefact.

Although the artefact can successfully be used to simulate EA models, its approach to achieving this was not compared with other methods that have been developed to date which aim to solve the same problem, such as that of Buschle et al. (2013). As their methods make use of external applications, and models abstracted from the EA, their approaches might be far more robust than the one presented here in this project.

This research project was primarily focused on realising a technical artefact, in order to fulfil the research objective by the means of a practical demonstration. In so doing, the EA was not considered from a semantic point of view at all, to discover how business analytic's models could be superimposed, or otherwise related to EA from a theoretical point of view. Although the current approach yields usable results, the use of artefact in many ways is no different to the use of other tools used for simulations, in that the analysis model and EA model are fitted to work together on an a rather ad-hoc basis. In other words, it does not consider how models for Monte Carlo simulations should be developed within the constraints of EA modelling capabilities.

Although this field is theoretically interesting, it seems only a very limited number of business users will potentially actually make use of such a tool. This point is discussed further under section 8.5.1 below.

8.5 Recommendations

Given the authors first hand experience with the artefact and the results of the evaluation, a few recommendations for future research efforts in the field, and the further development of the artefact, are considered.

8.5.1 Theoretical

Two primary recommendations are made from a theoretical point of view. Firstly, given the feedback from the interviewees, there seems to be a gap between the theoretical usefulness of executable EA models, and the potential for actual wide-spread adoption thereof. It would be useful then to construct a link between these two by identifying who the stakeholders are, that would benefit most from executable EA models. The specific needs of these stakeholders should then be identified, and proposals made for how best to meet them.

The potential benefits of the pervasive use of business analytics were discussed in this thesis. However, their actual usage at this point in time, and more specifically that of simulation modelling for these purposes should also be investigated as a related topic. The overlap between this group of businesses, and those using EA modelling can then be investigated to identify the potential group of users who would readily use an artefact such as the one presented here.

The second recommendation is to investigate if the outcomes of this artefact can, or should be linked with related research efforts, such as that of Veneberg (2014), which considers how EA can be enriched with operational data. This would perhaps partially answer a question left unanswered by this thesis, namely how to identify the input data required by the analysis models.

8.5.2 Artefact

If BIZZdesign, were to pursue the continued development of this artefact, a number of improvements can readily be identified which could be implemented first. The artefact in its current form only serves as a prototype ans is as such rather crude, but a number of lessons can be drawn from it:

Interface The general usability of the tool could be improved greatly by developing a purpose-built user interface for the artefact through which to specify variables as random, provide the necessary information for the distributions and select all of the other options available to the user. This would be particularly useful in cases where the same simulations, or analysis have to be re-run. Currently the user has to re-select the analysis query to run, the number of iterations to complete etc. These options should be saved by the tool, and only changed when the user explicitly indicates he wishes to do so.

Distributions The artefact provides the user with a number distributions from which to choose. Although these were specifically selected based on their common use and discussion within literature, there a number of distributions which might still be added to the artefact to cover a greater number of use cases. Fortunately this can be easily done through the addition of a few attributes to the *Random Variable* profile in Architect and subsequently incorporating them in the Monte Carlo script, using MS Excel to calculate the draw from the distribution (MS Excel provides a large range worksheet functions to calculate the inverse of various distributions. See the Excel 2013 object model developer reference for the WorksheetFunction Object [1])

[1] http://msdn.microsoft.com/en-us/library/office/ff834434(v=office.15).aspx

The necessity of user profiles Currently only attributes already existing within other attributes on the object can be defied as random. The tool-set was designed based on the assumption that analysis scripts, and therefore the profiles containing the attributes used by the scripts would already have been implemented as stand-alone analyses. However this assumption does not hold as it is possible that a user would develop a script purely for the use with the Monte Carlo method. In such a case it would be far easier if the user could use the Random Variables profile to define variables, independently of another profile. This would require the addition of more attribute fields for each list, and complicate the code, but might be worth considering if the tool-set is developed in future.

Changing application language When the user changes the application language within Architect, the display names of all of the variables change (in as far as they are provided in the translation files). Currently however the user has to provide the display name of the attributes defined as random, and the tool-set uses this name to search for the relevant attribute. All of these provided names for the random variables will therefore have to be changed by the user, to match the new names provided by the translation file of the new language. Ideally this would naturally happen automatically, especially when the number of attributes defined as random becomes large.

Lastly, as was stated by a the interviewees, the development of a a few specific analyses (based on using Monte Carlo simulations in this case), which can be applied in a generic fashion (i.e with minimal user input, perhaps only using profiles), would be of more worth to a wider variety of customers. This, in juxtaposition to the current focus being a robust artefact, enabling users to develop a wide range of analysis based on a back-end system enabling Monte Carlo simulations.

References

Accenture (2013). Accenture technology vision 2013. Technical report, Accenture.

Allen, T. T. (2011). Probability theory and monte carlo. In *Introduction to Discrete Event Simulation and Agent-based Modeling*, pages 9–27. Springer London, London.

Bagnulo, A. and Harvey, G. (2008). Executable architecture–survey and suitability assessment study of systems engineering standards and methodologies. *Defence R&D Canada Report*.

Banks, J., II, S. Carson, J., Nelson, L., B., and Nicol, M., D. (2009). *Discrete-event System Simulation*. Prentice Hall.

Buckl, S., Franke, U., Holschke, O., Matthes, F., Schweda, C., Sommestad, T., and Ullberg, J. (2009a). A pattern-based approach to quantitative enterprise architecture analysis. *AMCIS 2009 Proceedings*.

Buckl, S., Matthes, F., Renz, W., Schweda, C., and Sudeikat, J. (2008). Towards simulation-supported enterprise architecture management. In *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI)*, volume P-141, pages 131–145.

Buckl, S., Matthes, F., and Schweda, C. M. (2009b). Classifying enterprise architecture analysis approaches. In Poler, R., Sinderen, M. v., and Sanchis, R., editors, *Enterprise Interoperability*, number 38 in Lecture Notes in Business Information Processing, pages 66–79. Springer Berlin Heidelberg.

Buschle, M., Johnson, P., and Shahzad, K. (2013). The enterprise architecture analysis tool – support for the predictive, probabilistic architecture modeling framework. *AMCIS 2013 Proceedings*.

Davenport, T. (2006). Competing on analytics. *Harvard Business Review*, 84(1):98–107+134.

Davenport, T. H. and Harris, J. G. (2013). *Competing on Analytics: The New Science of Winning*. Harvard Business Press.

Evans, J. R. (2013). *Business analytics: methods, models, and decisions*. Pearson, Boston, 1st ed edition.

Glazner, C. G. (2009). *Understanding Enterprise Behavior Using Hybrid Simulation of Enterprise Architecture*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA. AAI0822455.

Glazner, C. G. (2011). Enterprise transformation using a simulation of enterprise architecture. *Journal of Enterprise Transformation*, 1(3):231–260.

Herschel, G., Linden, A., and Kart, L. (2014). Magic quadrant for advanced analytics platforms. Technical Report G00258011, Gartner.

Iacob, M.-E. and Jonkers, H. (2006). Quantitative analysis of enterprise architectures. In Konstantas, D., Bourrières, J.-P., Léonard, M., and Boudjlida, N., editors, *Interoperability of Enterprise Software and Applications*, pages 239–252. Springer London.

IBM (2012). Fast track to the future: The 2012 IBM tech trends report. Technical report, IBM Center for Applied Insights.

Johnson, P., Ullberg, J., Buschle, M., Franke, U., and Shahzad, K. (2013). P2amf predictive, probabilistic architecture modeling framework. In Sinderen, M. v., Luttighuis, P. O., Folmer, E., and Bosems, S., editors, *Enterprise Interoperability*, number 144 in Lecture Notes in Business Information Processing, pages 104–117. Springer Berlin Heidelberg.

Jonkers, H., Iacob, M.-E., Van Der Torre, L., de Boer, F.S, and Bonsangue, M (2013). Architecture analysis. In *Enterprise Architecture at Work*, The Enterprise Engineering Series, pages 189–220. Springer Berlin Heidelberg.

Kohavi, R., Rothleder, N., and Simoudis, E. (2002). Emerging trends in business analytics. *Communications of the ACM*, 45(8):45–48.

Korver, B. (1994). The monte carlo method and software reliability theory.

KPMG (2013). Big data & analytics: turning conceptual thinking into powerful results. Technical report, KPMG.

Ludwig, M., Farcet, N., Babau, J.-P., Champeau, J., and others (2011). Organizational configurations in executable enterprise architecture models. *proceedings of Complex Systems Design and Management 2011*.

Manzur, L., Santa, J., Sánchez, M., and Villalobos, J. (2012). Experimentation in executable enterprise architecture models. In Bajec, M. and Eder, J., editors, *Advanced Information Systems Engineering Workshops*, number 112 in Lecture Notes in Business Information Processing, pages 455–469. Springer Berlin Heidelberg.

March, S. T. and Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4):251–266.

Matthes, F. (2008). *Enterprise architecture management tool survey 2008*. Technische Universität München, München.

Object Management Group (2011). Business process model and notation (BPMN) version 2.

Object Management Group (2013). OMG unfified modeling language (OMG UML) version 2.5.

Paisittanand, S. and Olson, D. L. (2006). A simulation study of IT outsourcing in the credit card business. *European Journal of Operational Research*, 175(2):1248–1261.

Peffers, K., Rothenberger, M., Tuunanen, T., and Vaezi, R. (2012). Design science research evaluation. In Peffers, K., Rothenberger, M., and Kuechler, B., editors, *Design Science Research in Information Systems. Advances in Theory and Practice*, number 7286 in Lecture Notes in Computer Science, pages 398–410. Springer Berlin Heidelberg.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77.

Schulte, W. R., Chandler, N., Herschel, G., Laney, D., Sallam, R. L., Tapadinhas, J., and Sommer, D. (2013). Predicts 2014: Business intelligence and analytics will remain CIO's top technology priority. Technical Report G00258063, Gartner.

Scott, D. W. (1992). Histograms: Theory and practice. In *Multivariate Density Estimation*, pages 47–94. John Wiley & Sons, Inc.

Sprongl, P. (2013). Gaining competitive advantage through business analytics. *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, 61(7):2779–2785.

The Open Group, editor (2011). *TOGAF Version 9.1*. Van Haren Publishing, 10th edition.

The Open Group (2013). *ArchiMate 2.1 specification: Open Group Standard*. Van Haren Publishing, Zaltbommel.

United States Department of Defense (2010). The DoDAF architecture framework version 2.02.

U.S. Office of Management, U.S. Federal Chief Information Officers Council, U.S. Budget's Office of E-Government and Information Technology, and U.S. General Services Administration (2007). FEA consolidated reference model document version 2.3.

Veneberg, R. K. M. (2014). *Combining Enterprise Architecture & Operational Data To Better Support Decision-Making*.

Wan, H. and Carlsson, S. (2012). Towards an understanding of enterprise architecture analysis activities. pages 334–341.

Zachman, J. (1987). A framework for information systems architecture. *IBM Systems Journal*, 26(3):276–292.

Appendix A

Code

A.1 Monte Carlo Script File

This code below was located in ArchiMate/MonteCarlo/SetStochVar.script and contains the primary function (MCsimulation) as well as the supporting functions used by the Monte Carlo simulation. The code contains numerous comments defining the variables that are used and explaining the functioning of the code.

```
using SetStochVar for ArchiMate.MonteCarlo.SetStochVar;


////////////////////////////////////////////////////////////////////////////////
//function MCsimulation is the main body of the simulation
  //QueryIns: User inputs required by the analysis, received & tagged during the first run
  //RunStopped: in case of an error, the analysis can tag the model with 'RunStopped' to cancel the run
  //ObjectWVar, ObjectWRes: these tags indicate which objects contain a random variable (used for input or
       calculated)
  //AttsTagged: Indicates whether the user's script indicates which random variables are used for his analysis
  //i: Loop counter for simulation
  //FirstRun: Tag indicating whether this is the first loop of the simulation, can be used by user's script
  //j: used by script indicating the progress of the simulation run to the user through messages
////////////////////////////////////////////////////////////////////////////////

function MCsimulation() {
  // Any changes to the model will be undone in case an error occurs:
  session "rollback on error";

  // Ask the user which script he would like to run as a Monte Carlo simulation:
  selection = "";
  VP = false;
  while (VP == false) {
    selection = ask("MM_Object", "Select the analysis you wish to run");
    if (selection == undefined)
      return undefined;
    else if (selection.type().toString() != "Viewpoint")
```

```
        answer = messageBox ( "Please select a viewpoint", "Select  Viewpoint", "ok",  "error");
      else
        VP = true;
    }

    selectionS  = selection.toString();

    //Ask user how many iterations the monte carlo simulation should be run for:
    iter  = ask("integer",  "How many times should the simulation be iterated?");
    if  ((iter  == "") ||  (iter  == undefined) || (iter  == 0))
      return undefined;

    //Remove the various tags from the model if it  was used by a previous simulation run:
    if  (model.hasTaggedValue("QueryIns"))
      model.removeTaggedValue("QueryIns");

    if  (model.hasTaggedValue("RunStopped"))
      model.removeTaggedValue("RunStopped");

    forall  object  in  model {
      if  (object.hasTaggedValue("ObjectWVar"))
        object.removeTaggedValue("ObjectWVar");
      if  (object.hasTaggedValue("ObjectWRes"))
        object.removeTaggedValue("ObjectWRes");
    }

    // Initialise  values:
    AttsTagged = false;
    i  = 1;
    model.setTaggedValue("FirstRun",true, false);
    jump = true;

    //Save the original  values  of  the  attributes  specified  as random by the user:
    message "Storing original  values";
    run = "SaveOrigVar";
    f_result  = "";
    forall  object  in  model {
      f_result  = SetStochVar:FindRandomVar(object, run, i, "empty", AttsTagged);
    }

    // The function FindRandomVar will return 'failed'  if  any of the  attributes  cannot be found of have not been
          specified:
    if  (f_result  == "failed") {
      error  "Could not save original  values";
      return undefined;
    }

    //Open MS Excel:
    message "Opening Excel";
    excel  = ExternalObject("Excel.Application");
    //excel. Visible  = false;

    message "Starting Simulation Run...";
```

```
//Primary section: the simulation loop
while (i <= iter) {

  //Part I: Specify objects
    // The user can tag each object with random variable(s) with "ObjectWVar".
    // This is beneficial if the model has a lot of random variables which are not important to the particular
         analysis.
    // The first iteration will go through the entire model, as the analysis script will not have tagged any
         objects yet.
    // In the second iteration, the entire model is checked for tagged objects
    // If any tagged objects are found, only those objects' random variables are calculated for the remainder
         of the iterations

  if (i == 1) {
    mymodel = List();
    forall object in model {
      if (object.hasProfile("RandomVar") )
        mymodel.add(object);
    }
  } else if (i == 2) {
    model.removeTaggedValue("FirstRun");
    mymodel2 = List();
    forall object in model {
      if (object.hasTaggedValue("ObjectWVar"))
        mymodel2.add(object);
    }
    if (!(mymodel2.empty())) {
      AttsTagged = true;
      mymodel = mymodel2;
    }
  }

  // Part II: check specified objects for random variables & randomize variables
  run = "StochVar";
  forall object in mymodel {
    f_result = SetStochVar:FindRandomVar (object, run, i, excel, AttsTagged);
  }

  // Part III: Call script, check for errors, show progress:
    //Call the user defined script (has to conform to the requirements set out in the manual)
  QuRet = query(selectionS);

  // If the user selected script returns 'RunStopped', it was stopped for some reason, and so the simulation is
       cancelled too. All original values are restored, and Excel closed:
  if (i == 1) {
    if (model.hasTaggedValue("RunStopped")) {
      f_result = SetStochVar:RestoreToOrig(excel);
      answer = messageBox ( "Your script cancelled the run", "Simulation cancelled", "ok", "error");
      return undefined;
    }
  }

```

```
//Save variables user's script indicates as being calculated variables (required if these are to be analysed)
if (i == 1) {
  mymodel3 = List();
  forall object in model {
    if (object.hasTaggedValue("ObjectWRes") ) {
      mymodel3.add(object);
      ResList = object.getTaggedValue("ObjectWRes");
      nct = 1;
      VarAttr = "";
      VarAttrL = "";
      while (nct<= ResList.size()) {
        VarAttr = ResList[nct];
        VarAttrL = VarAttr + "L";
        if (object.hasTaggedValue(VarAttrL))
          object.removeTaggedValue(VarAttrL);
        if ((object.attrValue(VarAttr)).isMoney())
          VarList = List((object.attrValue(VarAttr)).amount);
        else
          VarList = List(object.attrValue(VarAttr));
        object.setTaggedValue(VarAttrL, VarList, true);
        nct = nct + 1;
      }
    }
  }
} else {
  if (!(mymodel3.empty())) {
    forall object in mymodel3 {
      ResList = object.getTaggedValue("ObjectWRes");
      nct = 1;
      VarAttr = "";
      VarAttrL = "";
      while (nct<=ResList.size()) {
        VarAttr = ResList[nct];
        VarAttrL = VarAttr + "L";
        VarList = object.getTaggedValue(VarAttrL);
        if ((object.attrValue(VarAttr)).isMoney())
          VarList.add((object.attrValue(VarAttr)).amount);
        else
          VarList.add(object.attrValue(VarAttr));
        object.setTaggedValue(VarAttrL, VarList, true);
        nct = nct + 1;
      }
    }
  }
}


// 'Percentage complete' messages show simulation run progress to user:
PerComp = ((i*100)/iter);
if (((jump) && (PerComp == 10 || PerComp == 30 || PerComp == 50 ||PerComp == 70 ||PerComp == 90
    )) || ( !(jump) && (PerComp == 20 || PerComp == 40 || PerComp == 60 || PerComp == 80 || PerComp
    == 99)) ) {
  if (jump) {
```

```
        jump = false;
      } else {
        jump = true;
      }
      PerCompM = "Percentage Complete: ";
      PerCompM.add(PerComp.toString());
      PerCompM.add("%");
      message PerCompM;
    }
    i = i + 1;                    //Iterate loop
  }

  // return model to original state:
  message "Finalising Run...";
  f_result = SetStochVar:RestoreToOrig(excel);

  // Inform user about completion of the calculation:
  answer = messageBox ( "Simulation complete", "Monte Carlo Simulation", "ok", "information");
  return undefined;
}


////////////////////////////////////////////////////////////////////////////////////
// The FindRandomVar function searches for all user defined random variables (defined via the 'Random Variables' profile) on an object
  // ListVars: list of user defined random variables
  // ListSize: Number of discrete variables
  // VarS: the characteristics of the variable, stored within a structure
  // VarAttr: attribute name of random variable
  // VarAttr_D: name of attribute (as displayed in the user interface) specified as a random variable by the user.
  // ListVP: the list of discrete values and their probabilities
  // ListVP_S: the individual values and their probabilities, stored within a structure
  // ListVP_Size: Number of discrete values assigned to variable
  // StochVar: The stochastic variable, randomly found
  // Disc_Val: Discrete value associated with a given probability
  // Disc_Prob: Probability associated with Disc_Val
  // f_result: Result returned from a called function
  // VarDist: type of distribution
  // VarMean: Distribution mean (or maximum, in case of uniform distribution)
  // VarStdD: Distribution standard deviation,(or minimum, in case of uniform distribution)
  // VarDistS = The distribution type, as type string
////////////////////////////////////////////////////////////////////////////////////

function FindRandomVar (object, run, i, excel, AttsTagged) {

  f_result = undefined;

  // Check if the "RandomVar" profile has been assigned to the object:
  if (object.hasProfile("RandomVar") ) {

    // Part I: Check if any discrete random variables have been defined:
    if (object.hasUserValue("List_Dis_var")) {
      ListVars = List();
```

```
      ListVars = object.attrValue("List_Dis_var");
      ListSize = ListVars.size();
      if (ListSize > 0) {
        ct = 1;
        VarS = Structure();
        ListVP = List();

        // Loop through list of discrete variables:
        while (ct <= ListSize) {
          VarS = ListVars[ct];
          VarAttr = VarS.selct_attr_disc_PA;
          VarAttr_D = VarS.selct_attr_disc;

          //If this run is used to randomize the variables:
          if (run == "StochVar") {
            ListVP = VarS.Disc_R_Vals;
            ListVP_S = Structure();
            ListVP_Size = ListVP.size();
            r = rand();
            StochVar = 0;
            Disc_ProbT = 0;
            ct2 = 1;

            // Go through list of probabilities until the cumulative probability is larger than a random number:
            while (ct2 <= ListVP_Size) {
              ListVP_S = ListVP[ct2];
              Disc_Val = ListVP_S.Discrete_Value;
              Disc_Prob = ListVP_S.Discrete_Prob;
              Disc_ProbT = Disc_ProbT + Disc_Prob;
              if (r<= Disc_ProbT) {
                StochVar = Disc_Val;
                break;
              }
              ct2 = ct2 +1;
            }
            // Save the variable to a tagged list & assign to attribute:
            f_result = AssignSave(VarAttr, object, StochVar, i, AttsTagged);

          //If this run is used to Save the original variables:
          } else if (run == "SaveOrigVar") {
            type = "DiscreteR";

            // Check that the provided attribute names match an attribute on the object, tag the original value
                of the variables to the object:
            f_result = SetStochVar:SaveOrigVar(object, VarAttr_D, ListVars, ct, ListSize, type);
            ListVars = f_result[2];
            f_result = f_result[1];
            if (f_result == "failed")
              return f_result;

          //If this run is used to restore the original variables to the object's attributes
          } else if (run == "RestoreOrigVar") {
            f_result = SetStochVar:RestoreOrigVar(object, VarAttr);
```

```
          } else {
            error "Run not recognised (function FindRandomVar)";
            f_result = "failed";
            stop;
          }
          ct = ct + 1;
        }
      }
    }

    //Part II: random variables
    if (object.hasUserValue("List_Vars")) {
      ListVars = List();
      ListVars = object.attrValue("List_Vars");
      ListSize = ListVars.size();
      if (ListSize > 0 ) {
        VarS = Structure();
        VarAttr = "";
        VarDist = "";
        ct = 1;

        // Loop through list of variables:
        while (ct <= ListSize) {
          VarS = ListVars[ct];
          VarAttr_D = VarS.selct_attr;
          VarAttr = VarS.selct_attr_PA;

          //If this run is used to randomize the variables:
          if (run == "StochVar") {
            VarDist = VarS.Dist_Type;
            VarMean = VarS.dist_mean;
            VarStdD = VarS.dist_std_dev;
            VarLike = VarS.dist_m_likely;
            VarBeta = VarS.dist_beta;
            r = rand();
            StochVar = 0;

            if (i == 1) {
              if ((VarMean == undefined) || (VarStdD == undefined)) {
                error "Mean/Min or Std. Deviation/Max not specified on object: " + object.name() + ", variable:
                    " + VarAttr_D;
                f_result = "failed";
              }
            }

            VarDistS = VarDist.toString();
            if(VarDistS == "Uniform [2]") {
              StochVar = (VarStdD - VarMean)*r + VarMean;
            }
            else if(VarDistS == "Normal [1]") {
              StochVar = excel.WorksheetFunction.Norm_Inv(r, VarMean, VarStdD);
            }
            else if(VarDistS == "Lognormal [1]") {
```

```
          StochVar = excel.WorksheetFunction.LogNorm_Inv(r, VarMean, VarStdD);
        }
        else if(VarDistS == "Triangular [3]") {
          if (i == 1) {
            if (VarLike == undefined) {
              error "Most likely value not specified on object: " + object.name() + ", variable: " + VarAttr_D;
              f_result = "failed";
            }
          }

          if (r < ((VarLike - VarMean)/(VarStdD-VarMean)))
            StochVar = VarMean + sqrt(r*(VarStdD-VarMean)*(VarLike- VarMean));
          else
            StochVar = VarStdD - sqrt((1-r)*(VarStdD-VarMean)*(VarStdD-VarLike));
        }
        else if(VarDistS == "Beta [4]") {
          if (i == 1) {
            if (VarLike == undefined) {
              error "Alpha not specified on object: " + object.name() + ", variable: " + VarAttr_D;
              f_result = "failed";
            }
            if (dist_beta == undefined) {
              error "Beta not specified on object: " + object.name() + ", variable: " + VarAttr_D;
              f_result = "failed";
            }
          }
          StochVar = excel.WorksheetFunction.Beta_Inv(r, VarLike, VarBeta, VarMean, VarStdD);
        }
        else {
          error "Script Error: Distribution not recognised on object: " + object.name() + ", variable: " + VarAttr_D;
          f_result = "failed";
        }
        // Save the variable to a tagged list & assign to attribute:
        f_result = AssignSave(VarAttr, object, StochVar, i, AttsTagged);

      //If this run is used to Save the original variables:
      } else if (run == "SaveOrigVar") {
        type = "Random";

        // Check that the provided attribute names match an attribute on the object, tag the original value of the variables to the object:
        f_result = SetStochVar:SaveOrigVar(object, VarAttr_D, ListVars, ct, ListSize, type);
        ListVars = f_result[2];
        f_result = f_result[1];
        if (f_result == "failed")
          return f_result;

      //If this run is used to restore the original variables to the object's attributes
      } else if (run == "RestoreOrigVar") {
        f_result = SetStochVar:RestoreOrigVar(object, VarAttr);
      } else {
```

```
            error "Run not recognised (function FindRandomVar)";
            f_result = "failed";
            stop;
          }
          ct = ct + 1;
        }
      }
    }
  }
  return f_result;
}


////////////////////////////////////////////////////////////////////////////////////
// The AssignSave function assigns the stochastic variable to the object's attribute and adds it to the tagged
     list of the attributes random variables generated during the monte carlo simulation.
  // VarAttr: The attribute being considered
  // Var: The value of the attribute
  // VarList: The list of random variables generated during each simulation run for the attribute and tagged to
       the object
  // VarAttrL: The name of the list
  // type: the type of attribute, whether a random variable used as input to the calculation, or a calculated
       result
////////////////////////////////////////////////////////////////////////////////////

function AssignSave(VarAttr, object, StochVar, i, AttsTagged) {
  // Assign the random variable to attribute
  Var = object.attrValue(VarAttr);
  if (Var.isMoney() ) {
    object.setAttrValue(VarAttr,Money(StochVar));
  } else {
    object.setAttrValue(VarAttr,StochVar);
  }

  // If this is the first run, check for a previously tagged list (from a previous simulation), and remove it
       before creating a new list
  VarAttrL = VarAttr + "L";
  if (i == 1) {
    if (object.hasTaggedValue(VarAttrL))
      object.removeTaggedValue(VarAttrL);
    VarList = List(StochVar);
  // After first iteration, retrieve tagged list of attributes from object, add the new value, and re-tag the
       list:
  } else {
    VarList = object.getTaggedValue(VarAttrL);
    VarList.add(StochVar);
  }

  // If the user's analysis script does not specify which random variables the analysis uses, this is done for
       all random variables on model (as it is required for later analysis):
  if ((i==2) && (!AttsTagged)) {
    type = "variable";
    f_result = SetStochVar:TagObject(object, VarAttr, type);
```

```
  }

  // Tag list of random variables to object:
  object.setTaggedValue(VarAttrL, VarList, true);
  return "tagged";
}


/////////////////////////////////////////////////////////////////////////////////////
// The SaveOrigVar function tags the original value of a variable to the object
  // AttributeL: A list of all attributes on the object
  // AttributeL_S: Size of AttributeL
  // found: number of attributes found on object which have the same name in the user interface as the name provided by the user
  // VarAttr_PA: Attribute name
  //
/////////////////////////////////////////////////////////////////////////////////////

function SaveOrigVar (object, VarAttr_D, ListVars, ct, ListSize, type) {

  // Different attribute and list names for discrete and continuous variables:
  if (type == "DiscreteR") {
    AttributeType = "selct_attr_disc_PA";
    ListType = "List_Dis_var";
  } else {
    AttributeType = "selct_attr_PA";
    ListType = "List_Vars";
  }

  VarS = Structure();
  VarS = ListVars[ct];
  AttributeL = object.attrs();
  AttributeL_S = AttributeL.size();

  // Check that an attribute name has been provided:
  if (VarAttr_D == undefined) {
    errorM = "Random variable profile selected, but one or more attributes are not specified. Object: " +
        object.name();
    error errorM;
    return " failed ";
  }

  ct3 = 1;
  found = 0;
  VarAttr_PA = '';

  //Go through list of attributes on object, compare to user provided name
  while (ct3 <= AttributeL_S) {
    if (VarAttr_D == AttributeL[ct3].first.toString() ) {
      VarAttr_PA = AttributeL[ct3].first.name();
      found = found + 1;
    }
    ct3 = ct3 + 1;
```

```
    }

    //If no match was found:
    if (found == 0) {
      errorM = "The attribute name '" + VarAttr_D + "', on object '" + object.name() + "', does not match any of
            the existing attributes on the profile.";
      error errorM;
      return "failed";

    //If multiple matched were found:
    } else if (found > 1) {
      errorM = "A provided attribute name matches more than one attribute on profile. Attribute: " + VarAttr_D +
            ". Object: " + object.name();
      error errorM;
      return "failed";

    //If one match was found, tag original value of attribute to object and set the name of the desired attribute
         to attribute 'selct_attr_disc_PA' or 'selct_attr_PA' on the 'random variable' profile.
    } else {
      VarS.add(AttributeType,VarAttr_PA);
      NewListVars = List();
      ct5 = 1;
      while (ct5 <= ListSize) {
        if (ct5 == ct)
          NewListVars.add(VarS);
        else
          NewListVars.add(ListVars[ct5]);
        ct5 = ct5 + 1;
      }
      object.setAttrValue(ListType, NewListVars);
      ListVars = NewListVars;
      object.setTaggedValue(VarAttr_PA, (object.attrValue(VarAttr_PA)), false);
    }

    return List("saved",ListVars);
}

//////////////////////////////////////////////////////////////////////////////////////
// The RestoreOrigVar function restores all original values saved in tags. See SaveOrigVar function.
//////////////////////////////////////////////////////////////////////////////////////

function RestoreOrigVar (object, VarAttr) {

    Var = object.getTaggedValue(VarAttr);
    object.setAttrValue(VarAttr, Var);

    return "restored";
}


//////////////////////////////////////////////////////////////////////////////////////
// In case of a fault, the RestoreOrigVar function restores all original values saved as tags and closes excel.
//////////////////////////////////////////////////////////////////////////////////////
```

```
function RestoreToOrig(excel) {

  message "Closing Excel";
  excel.Quit();

  //Restore the original values of all random attributes:
  message "Restoring original values";
  run = "RestoreOrigVar";
  i = 1;
  forall object in model {
    f_result = SetStochVar:FindRandomVar(object, run, i, "empty", false);
  }

  return "restored";
}

/////////////////////////////////////////////////////////////////////////////////
// function TagObject is to be used by an analysis script to specify the objects' random variables or (and)
     objects' attributes calculated by an analysis
  // type: the type of attribute, whether a random variable used as input to the calculation, or a calculated
       result
  // VarL: List of random variables on object
  // TagType:
    // ObjectWVar: list of random variables on object, used as input
    // ObjectWRes: list of calculated random variables on object.
  // added: If the random variable is already contained on the list
/////////////////////////////////////////////////////////////////////////////////

function TagObject(object, VarName, type) {
  if (type == "variable") {
    TagType = "ObjectWVar";
  } else if (type == "result") {
    TagType = "ObjectWRes";
  } else {
    error "Invalid tag type specified", type;
    stop;
  }
  VarL = List();

  // check if the object already has a previously tagged list of variables
  if (object.hasTaggedValue(TagType)) {
    VarL = object.getTaggedValue(TagType);
    ct = 1;
    added = false;

    // Check if the random variable is already contained within the list of variables
    while (ct<= VarL.size()) {
      if (VarName == VarL[ct]) {
        added = true;
        break;
      }
      ct = ct + 1;
```

```
      }

      //If the variable has not been added to the list yet, do so and tag the list to the object
      if (!(added)) {
        VarL.add(VarName);
        object.setTaggedValue(TagType, VarL, true);
        return 0;
      } else {
        return 0;
      }

    // If no tagged list exists yet, create one:
    } else {
      VarL.add(VarName);
      object.setTaggedValue(TagType, VarL, true);
      return 0;
    }
}
```

A.2 Additions to User's Script

"Tag User Input"

```
QueryInsG = Structure();
if (model.hasTaggedValue("QueryIns")) {
  QueryInsG = model.getTaggedValue("QueryIns");
  modelAndObjects = QueryInsG.valueFor("modelAndObjects");
  FirstRun = false;
} else {
  modelAndObjects = ask("MM_Object", "Select that (part of the) model that you want to calculate total costs
      for.");
  if ( modelAndObjects == undefined) {
    model.setTaggedValue("RunStopped", true, false);
    return undefined;
  }
  QueryInsG.add("modelAndObjects", modelAndObjects);
  model.setTaggedValue("QueryIns", QueryInsG, false);
  FirstRun = true;
}
```

"Stop Script Tag"

```
model.setTaggedValue("RunStopped", true, false);
```

"Tag Attribute as Variable"

```
if (model.hasTaggedValue("FirstRun")) {
  type = "result";
  f_result = SetStochVar:TagObject(concept, attribute, type);
}
```

SetStochVar is short for the path: ArchiMate.MonteCarlo.SetStochVar.

A.3 Histogram

```
////////////////////////////////////////////////////////////////////////////////
// The shortInt changes a real value into a string and removes the remaining zeros and decimal
////////////////////////////////////////////////////////////////////////////////

function shortInt(Int) {
  OldString = Int.toString();
  NewString = OldString.substring(1,OldString.find(".") - 1);
  return NewString;
}

////////////////////////////////////////////////////////////////////////////////
// The spacing function takes a string and inserts spaces between sets of 3 digits
////////////////////////////////////////////////////////////////////////////////

function spacing(str){
  strS = str.size();
  if (strS > 3)
    NewString = spacing(NewString.substring(1, strS - 3)) + " " + str.substring(strS - 2, 3);
  else
    NewString = str;
  return NewString;
}

////////////////////////////////////////////////////////////////////////////////
// The toLen function removes any superfluous digits after a decimal and reuturns a string
////////////////////////////////////////////////////////////////////////////////

function toLen(Int) {
  OldString = Int.toString();
  if (OldString.size() > 5) {
    if (OldString.find(".") > 4) {
      NewString = OldString.substring(1, OldString.find(".") - 1);
      if (NewString.size() > 3)
        NewString = spacing(NewString);
    } else if (OldString.find(".") > 2) {
      NewString = OldString.substring(1, 5);
    } else {
      NewString = OldString;
    }
  } else {
    NewString = OldString;
  }
  return NewString;
}

////////////////////////////////////////////////////////////////////////////////
```

```
//The histogram function uses the 'ObjectWVar' and 'ObjectWRes' tags generated by the Monte Carlo simulation
      to provide the user a list of attributes for which he can generate a histogram
  // ObjectWVar: Tag listing the (input) variables on the object
  // ObjectWRes: Tag listing the resultant (calculated) variables on the object
  // found_URV: boolean indicating whether any user defined random variables are present
  // found_CRV: boolean indicating whether any calculated random variables are present
  // VarList_u: List of variables available on object
  // VarAttr: The user selected variable
  // VarList: The list of historic data generated during the last simulation run for that attribute
  // VarAttrL: The name of the tag containing VarList
  // VarListS: Size of VarList
  // Sturge: Sturges' formula, suggesting number of bins
  // Scott: Scott's normal reference rule, suggesting number of bins
  // SqrRtCh: Square-root choice
  // Bin_count: Number of bins
  // Bin_width: The width of each bin
  // Min_val: Smallest value in VarList
  // Max_val: Largest value in VarList
  // STCM = sample third central moment
  // Skew = estimator of the population skewness
////////////////////////////////////////////////////////////////////////////////

//Part I: Find Attribute
//Ask user to select relevant object

selection = ask("MM_Object", "Select the object where the attribute exists for which you wish to create a
      histogram");
if (selection == undefined)
  stop;

found_URV = false;
Msg_URV = "User defined random variable(s) on object: ";
cnt_u = 1;
VarList_u = List();

// Go through list of input variables (if any), assign counter (cnt_u), add to VarList_u and construct message
      to display to user (Msg_URV)
cnt_t = 0;
if (selection.hasTaggedValue("ObjectWVar")){
  found_URV = true;
  VarList = selection.getTaggedValue("ObjectWVar");
  VarListS = VarList.size();
  i = 1;
  while (i<=VarListS) {
    if (i < VarListS)
      Msg_URV = Msg_URV + "[" + cnt_u.toString() + "] " + selection.attr(VarList[i]).first.toString() + ", ";
    else
      Msg_URV = Msg_URV + "[" + cnt_u.toString() + "] " + selection.attr(VarList[i]).first.toString() + ".";
    cnt_t = cnt_u;
    cnt_u = cnt_u + 1;
    VarList_u.add(VarList[i]);
    i = i + 1;
  }
```

```
}

// Go through list of calculated variables (if any), assign counter (cnt_u), add to VarList_u and construct
      message to display to user (Msg_CRV)
found_CRV = false;
Msg_CRV = "Calculated random variable(s) on object: ";
if (selection.hasTaggedValue("ObjectWRes")) {
  found_CRV = true;
  VarList = selection.getTaggedValue("ObjectWRes");
  VarListS = VarList.size();
  i = 1;
  while (i<=VarListS) {
    if (i < VarListS)
      Msg_CRV = Msg_CRV + "[" + cnt_u.toString() + "] " + selection.attr(VarList[i]).first.toString() + ", ";
    else
      Msg_CRV = Msg_CRV + "[" + cnt_u.toString() + "] " + selection.attr(VarList[i]).first.toString() + ".";
    cnt_u = cnt_u + 1;
    VarList_u.add(VarList[i]);
    i = i + 1;
  }
}

// Check if any variables were found. If not: show error message
if ((!found_URV) && (!found_CRV)) {
  answer = messageBox ( "Object does not contain any attributes with historic data from a Monte Carlo
        simulation", "Invalid Selection", "ok", "error");

// If any were found, list them for the user (Msg_F).
  } else {
  Msg_F = "";
  if (found_URV)
    Msg_F = Msg_URV + " ";
  if (found_CRV)
    Msg_F = Msg_F + Msg_CRV;
  message Msg_F;

  //Ask the user for which of the variables he would like to develop a histogram (& ensure his selection is valid)
  proper = true;
  selected_a = 0;
  while (proper) {
    selected_a = ask("integer", "Select variable (see message window for options)");
    if (selected_a == undefined)
      stop;
    else if (selected_a >= cnt_u)
      answer = messageBox ("Your selection is out of range. See the message window for options.","Invalid
           selection","ok", "error");
    else
      proper = false;
  }

  //Part II: Generate Histogram
  VarAttr = VarList_u[selected_a];
  VarAttrL = VarAttr + "L";
```

```
VarList = selection.getTaggedValue(VarAttrL);
VarListS = VarList.size();

//get VarListS as real number
VarListSr = (VarListS/10.0)*10.0;

if (selected_a > cnt_t) {
  if (selection.attr(VarAttr).second.isMoney())
    Mon = "€ ";
  else
    Mon = "";
} else
  Mon = "";

//Find the mean:
Total = 0.0;
forall item in VarList
  Total = Total + item;
Avg = Total / VarListSr;

// Find the standard deviation:
Sum = 0.0;
forall item in VarList {
  Sum = Sum + sqr(item - Avg);
}
Var = (Sum / (VarListSr-1.0));
StDev = sqrt(Var);

// Median & Quartiles
VarList_Sort = VarList.sortedAsc();
Low_half = (VarListS/2);
if (mod(VarListS , 2) == 0) {
  Upper = VarList_Sort[Low_half];
  Lower = VarList_Sort[(Low_half+1)];
  Median = (Upper + Lower)/2.0;
  odd = 0;
} else{
  Median = VarList_Sort[Low_half+1];
  odd = 1;
}

Low_quarter = (Low_half/2);
if (mod(Low_half , 2) == 0) {
  FQuartileL = VarList_Sort[Low_quarter];
  FQuartileU = VarList_Sort[Low_quarter + 1 ];
  FQuartile = (FQuartileL + FQuartileU)/2.0;

  Low_Tquarter = Low_half + Low_quarter + odd;
  TQuartileL = VarList_Sort[Low_Tquarter];
  TQuartileU = VarList_Sort[Low_Tquarter + 1];
  TQuartile = (TQuartileL + TQuartileU)/2.0;
} else {
  FQuartile = VarList_Sort[Low_quarter+1];
```

```
   TQuartile = VarList_Sort[Low_half + Low_quarter + 1 + odd];

}
IQR = TQuartile − FQuartile;

// Skewness
Sum = 0.0;
 forall item in VarList {
   Sum = Sum + pow((item − Avg),3);
}
Skew = (Sum/VarListSr)/pow(StDev, 3);


// Nr of Bins Guidelines
Scott = ceil((StDev *3.5)/(pow(VarListSr, 1.0/3.0)));
SqrRtCh = ceil(sqrt(VarListSr));
Sturge = ceil((log(VarListSr) / log(2.0)) + 1.0);

//Doane's formula
sigmag1 = sqrt((6*(VarListSr − 2.0))/((VarListSr + 1.0)*(VarListSr + 3.0)));
Doane = ceil(1.0 + log(VarListSr)/log(2.0) + log(1.0 + (abs(Skew)/sigmag1))/log(2.0));

//Freedman–Diaconis' formula
FreedDiac = (2*IQR)/(pow(VarListSr, 1.0/3.0));

Min_val = min(VarList);
Max_val = max(VarList);
Range_val = Max_val − Min_val;

message "Sample size: " + VarListS.toString();
message "Sample Mean: " + Mon + toLen(Avg) + ", Sample Standard Deviation: " + Mon + toLen(StDev) + ",
      Skewness estimator: " + toLen(Skew);
message "Min: " + Mon + toLen(Min_val) + ", First Quartile: " + Mon + toLen(FQuartile) + ", Sample Median:
      " + Mon + toLen(Median) + ", Third Quartile: " + Mon + toLen(TQuartile) +", Max: " + Mon +
      toLen(Max_val);
message "Interquartile range: " + Mon + toLen(IQR) + ", Range: " + Mon + toLen(Range_val);
message "Suggested Number of bins:";
message "Sqare Root Choice: " + shortInt(SqrRtCh);
message "Sturges rule: " + shortInt(Sturge);
message "Doane's formula: " + shortInt(Doane);
message "Scotts Normal Reference Rule: " + shortInt(Scott);
message "Freedman Diaconis' Rule: " + shortInt(FreedDiac);

// Ask user how many bins should be used to generate the histogram
Bin_count = ask("integer", "Select number of bins (see message window for suggestions)");

 if (Bin_count == undefined)
   stop;

Bin_width = Range_val/Bin_count;
// Initialise the indexed set, Bins. ct = bin number

Bins = Index();
```

```
ct = 1;
while (ct <= Bin_count) {
  Bins.add(ct, 0);
  ct = ct + 1;
}

// Find the appropriate bin for each value in VarList
ct = 1;
forall item in VarList {
  Binned = false;
  Bin_ct = 0;
  Bin_current = Min_val;
  while (!Binned) {
    Bin_current = Bin_current + Bin_width;
    Bin_ct = Bin_ct + 1;
    if (item < Bin_current) {
      Binned = true;
      temp = Bins.valueFor(Bin_ct);
      Bins.add(Bin_ct, (temp + 1));
    } else if ((Bin_ct == Bin_count) && (item == Max_val)) {
      Binned = true;
      temp = Bins.valueFor(Bin_ct);
      Bins.add(Bin_ct, (temp + 1));
    }
  }
}

Bin_mid_point = Min_val + 0.5*Bin_width;
Bin_current = Min_val;

ChartTypeC = messageBox( "Would you like a cumulative chart?", "Chart Type", "yesno", "information");

  Cumulative = 0;

//Provide the output required by the chart
forall ct, size in Bins{
  Cumulative = Cumulative + size;
  message "Bin " + ct.toString() + ": " + Mon + toLen(Bin_current) + " to " + Mon + toLen(Bin_current +
        Bin_width) +". Frequency: " + ((size/VarListSr)*100).toString() + "%, Cumulative Frequency: " +
        ((Cumulative/VarListSr)*100).toString() + "%.";
  Bin_current = Bin_current + Bin_width;
  s = Structure();
  bin = "Bin " + ct.toString() + ": " + Mon + toLen(Bin_mid_point);
  Bin_mid_point = Bin_mid_point + Bin_width;
  s.add("object", bin);
  s.add("property1", "Frequency");
  //s.add("property2", "Frequency 2");
  if (ChartTypeC == "yes")
    s.add("value1", ((Cumulative/VarListSr)*100));
  else
    s.add("value1", ((size/VarListSr)*100));
  output s;
}
```

```
}
```

A.4 Retrieve variable's simulation outcomes

```
header "value";

 selection  = ask("MM_Object", "Select the object where the attribute exists for which you wish to create a
       histogram");
 if  ( selection  == undefined)
stop;

found_URV = false;
Msg_URV = "User defined random variable(s) on object: ";
cnt_u = 1;
VarList_u = List();

// Go through list of input variables (if any), assign counter (cnt_u), add to VarList_u and construct message
       to display to user (Msg_URV)
cnt_t = 0;
 if  ( selection .hasTaggedValue("ObjectWVar")){
   found_URV = true;
   VarList = selection .getTaggedValue("ObjectWVar");
   VarListS = VarList.size() ;
   i  = 1;
   while  (i<=VarListS) {
      if  (i  < VarListS)
      Msg_URV = Msg_URV + "[" + cnt_u.toString() + "] " + selection.attr(VarList[i]).first.toString() + ", ";
      else
      Msg_URV = Msg_URV + "[" + cnt_u.toString() + "] " + selection.attr(VarList[i]).first.toString() + ".";
      cnt_t = cnt_u;
      cnt_u = cnt_u + 1;
      VarList_u.add(VarList[i]);
      i  = i + 1;
   }
}

// Go through list of calculated variables (if any), assign counter (cnt_u), add to VarList_u and construct
       message to display to user (Msg_CRV)
found_CRV = false;
Msg_CRV = "Calculated random variable(s) on object: ";
 if  ( selection .hasTaggedValue("ObjectWRes")) {
   found_CRV = true;
   VarList = selection .getTaggedValue("ObjectWRes");
   VarListS = VarList.size() ;
   i  = 1;
   while  (i<=VarListS) {
      if  (i  < VarListS)
      Msg_CRV = Msg2_CRV + "[" + cnt_u.toString() + "] " + selection.attr(VarList[i]).first.toString() + ", ";
      else
      Msg_CRV = Msg_CRV + "[" + cnt_u.toString() + "] " + selection.attr(VarList[i]).first.toString() + ".";
```

```
    cnt_u = cnt_u + 1;
    VarList_u.add(VarList[i]);
    i = i + 1;
  }
}

// Check if any variables were found. If not: show error message
if ((!found_URV) && (!found_CRV)) {
  answer = messageBox ( "Object does not contain any attributes with historic data from a Monte Carlo
       simulation", "Invalid Selection", "ok", "error");

  // If any were found, list them for the user (Msg_F).
} else {
Msg_F = "";
if (found_URV)
Msg_F = Msg_URV + " ";
if (found_CRV)
Msg_F = Msg_F + Msg_CRV;
message Msg_F;

//Ask the user for which of the variables he would like to develop a histogram (& ensure his selection is valid)
proper = true;
selected_a = 0;
while (proper) {
  selected_a = ask("integer", "Select variable (see message window for options)");
  if (selected_a == undefined)
  stop;
  else if (selected_a >= cnt_u)
  answer = messageBox ("Your selection is out of range. See the message window for options.","Invalid
       selection","ok", "error");
  else
  proper = false;
}

//Part II: Generate Histogram
VarAttr = VarList_u[selected_a];
VarAttrL = VarAttr + "L";
VarList = selection.getTaggedValue(VarAttrL);
VarListS = VarList.size();

output "Object:";
output selection;

output "Attribute:";
output selection.attr(VarAttr).first.toString();

output "Sample:";
forall item in VarList
output item;
}
```

A.5 NPV Analysis

```
using NPVscript for ArchiMate.standardAnalysis.NPV;
using SetStochVar for ArchiMate.MonteCarlo.SetStochVar;

function GetYearListVals(concept, Years, type) {
  yList = List();
  i = 1;
  while (i <= Years) {
    aName = "Value_Y" + i.toString();
    Val = concept.attrValue(aName);
    if (type == "percentage") {
      if (Val > 100.0) {
        Val = 100.0;
      }
    }
    if (Val < 0 )
      Val = 0.0;
    yList.add(Val);
    i = i + 1;
  }
  return yList;
}

function SetYearListVals(object, Years, ValArr) {
  i = 1;
  while (i <= Years) {
    aName = "Value_Y" + i.toString() + "_R";
    object.setAttrValue(aName,ValArr[i]);
    i = i + 1;
  }

  if (model.hasTaggedValue("FirstRun")) {
    i = 1;
    type = "result";
    aName = "";
    while (i <= Years) {
      aName = "Value_Y" + i.toString() + "_R";
      f_result = SetStochVar:TagObject(object, aName, type);
      i = i + 1;
    }
  }
  return 0;
}

function NPVanalysis() {
  ////////////////////////////////////////////////
  CardHolders = List();
  CardHolders_growth = 0.0;
  Years = 5;

  obj = model.getObject("Credit Card Owners");

```

```
CardHolders.add(obj.attrValue("Value_YA"));
CardHolders_growth = obj.attrValue("percentage_growth");

i = 2;
while (i <= Years) {
  CardHolders.add((CardHolders[i−1]* (1+CardHolders_growth/100.0)));
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,CardHolders);


////////////////////////////////////////////
ActiveCardHolders = List();
ActiveCardHoldersP = List();

obj = model.getObject("Active CC Users");

ActiveCardHoldersP = GetYearListVals(obj, Years, "percentage");

i = 1;
while (i <= Years) {
  ActiveCardHolders.add(round((ActiveCardHoldersP[i]/100.0)*CardHolders[i]));
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,ActiveCardHolders);

////////////////////////////////////////////

Utilization = List();
UtilizationP = List();
obj = model.getObject("Credit Utilization");
UVolumePM = obj.attrValue("Value_YA");
UtilizationP = GetYearListVals(obj, Years, "percentage");

i = 1;
while (i <= Years) {
  Utilization.add((UtilizationP[i]/100.0)*(UVolumePM*12.0));
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,Utilization);
////////////////////////////////////////////

GoldCardRevenue = List();
GoldCardFees = List();
obj = model.getObject("Gold Credit Card");
GoldCardP = obj.attrValue("Value_YA");
GoldCardFees = GetYearListVals(obj, Years, "");

i = 1;
while (i <= Years) {
```

```
  GoldCardRevenue.add(CardHolders[i]*(GoldCardP/100.0)*GoldCardFees[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,GoldCardRevenue);
////////////////////////////////////////////

SilverCardRevenue = List();
SilverCardFees = List();
obj = model.getObject("Silver Credit Card");
SilverCardP = obj.attrValue("Value_YA");
SilverCardFees = GetYearListVals(obj, Years, "");

i = 1;
while (i <= Years) {
  SilverCardRevenue.add(CardHolders[i]*(SilverCardP/100.0)*SilverCardFees[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,SilverCardRevenue);
////////////////////////////////////////////

TransactionVolume = List();
obj = model.getObject("Credit Card Transaction Management");

i = 1;
while (i <= Years) {
  TransactionVolume.add(ActiveCardHolders[i]*Utilization[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,TransactionVolume);

////////////////////////////////////////////

CWithdrawals = List();
obj = model.getObject("Cash Withdrawal");
CWithdrawalsP = obj.attrValue("Value_YA");

i = 1;
while (i <= Years) {
  CWithdrawals.add((CWithdrawalsP/100.0)*TransactionVolume[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,CWithdrawals);

////////////////////////////////////////////

CAdvanceFee = List();
obj = model.getObject("Cash Advance Fee");
CAdvanceFeeP = obj.attrValue("Value_YA");

```

```
i = 1;
while (i <= Years) {
  CAdvanceFee.add((CAdvanceFeeP/100.0)*CWithdrawals[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,CAdvanceFee);
//////////////////////////////////////////////

CCPurchases = List();
obj = model.getObject("Credit Card Purchases");
CCPurchasesP = obj.attrValue("Value_YA");

i = 1;
while (i <= Years) {
  CCPurchases.add((CCPurchasesP/100.0)*TransactionVolume[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,CCPurchases);

//////////////////////////////////////////////

TransactionCosts = List();
obj = model.getObject("Transaction Handling");
TransactionsPY = GetYearListVals(obj, Years, "");
TransactionsFee = obj.attrValue("Value_YA");

i = 1;
while (i <= Years) {
  TransactionCosts.add(TransactionsPY[i]*ActiveCardHolders[i]*TransactionsFee);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,TransactionCosts);
//////////////////////////////////////////////

InterchangeFee = List();
obj = model.getObject("Interchange Fee");
InterchangeFeeP = obj.attrValue("Value_YA");

i = 1;
while (i <= Years) {
  InterchangeFee.add((InterchangeFeeP/100.0)*CCPurchases[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,InterchangeFee);
//////////////////////////////////////////////

CardIssuingFeeC = List();
obj = model.getObject("Issue Cards");
CardIssuingFee = GetYearListVals(obj, Years, "");
```

```

CardIssuingFeeC.add(CardHolders[1]*CardIssuingFee[1]);
i = 2;
while (i <= Years) {
  CardIssuingFeeC.add((CardHolders[i] - CardHolders[i-1]) *CardIssuingFee[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,CardIssuingFeeC);

////////////////////////////////////////////

StatementFeeC = List();

obj = model.getObject("Print & Mail Statements");

StatementFeePM = GetYearListVals(obj, Years, "");

i = 1;
while (i <= Years) {
  StatementFeeC.add(StatementFeePM[i]*12*CardHolders[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,StatementFeeC);

////////////////////////////////////////////

SystemFee = List();
obj = model.getObject("Back-end services");
SystemFeeC = GetYearListVals(obj, Years, "");

i = 1;
while (i <= Years) {
  SystemFee.add(SystemFeeC[i]*CardHolders[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,SystemFee);

////////////////////////////////////////////

RevVolume = List();
obj = model.getObject("Revolving Volume");
RevVolumeP = obj.attrValue("Value_YA");

i = 1;
while (i <= Years) {
  RevVolume.add((RevVolumeP/100.0)* CCPurchases[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,RevVolume);
```

```
////////////////////////////////////////////

 Interest  = List();
obj = model.getObject("Collect Interest");
InterestP  = GetYearListVals(obj, Years, "percentage");

i  = 1;
while  (i  <= Years) {
   Interest .add(((RevVolume[i]/12.0)*(InterestP[i]/100.0))*12.0);
   i  = i + 1;
}

 result  = NPVscript:SetYearListVals(obj,Years,Interest);
////////////////////////////////////////////

NPL = List();
obj = model.getObject("Non-Performing Loans");
NPLP = obj.attrValue("Value_YA");

i  = 1;
while  (i  <= Years) {
  NPL.add((NPLP/100.0)* RevVolume[i]);
  i  = i + 1;
}

 result  = NPVscript:SetYearListVals(obj,Years,NPL);
////////////////////////////////////////////

DebtsRecovered = List();
obj = model.getObject("Recover Debts");
DebtsRecoveredP = GetYearListVals(obj, Years, "percentage");

i  = 1;
while  (i  <= Years) {
  DebtsRecovered.add((DebtsRecoveredP[i]/100.0)* NPL[i]);
  i  = i + 1;
}

 result  = NPVscript:SetYearListVals(obj,Years,DebtsRecovered);
////////////////////////////////////////////

DebtRecoveryFee = List();
obj = model.getObject("Debt Recovery");
DebtRecoveryFeeP = GetYearListVals(obj, Years, "percentage");

i  = 1;
while  (i  <= Years) {
  DebtRecoveryFee.add((DebtRecoveryFeeP[i]/100.0)* DebtsRecovered[i]);
  i  = i + 1;
}

 result  = NPVscript:SetYearListVals(obj,Years,DebtRecoveryFee);
////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////////////
Revenue = List();
obj = model.getObject("Revenue");

i = 1;
while (i<=Years){
  Revenue.add(Interest[i] + CAdvanceFee[i] + InterchangeFee[i] + GoldCardRevenue[i] + SilverCardRevenue[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,Revenue);
////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////

Marketing = List();
obj = model.getObject("Marketing");
MarketingP = GetYearListVals(obj, Years, "percentage");

i = 1;
while (i <= Years) {
  Marketing.add((MarketingP[i]/100.0)*Revenue[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,Marketing);

////////////////////////////////////////////

BusDevelopFee = List();
obj = model.getObject("Business Development");
BusDevelopFeeP = GetYearListVals(obj, Years, "percentage");

i = 1;
while (i <= Years) {
  BusDevelopFee.add((BusDevelopFeeP[i]/100.0)*Revenue[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,BusDevelopFee);
////////////////////////////////////////////

BadDebtsWO = List();
obj = model.getObject("Write off bad debts");
BadDebtsWOP = GetYearListVals(obj, Years, "percentage");

i = 1;
while (i <= Years) {
  BadDebtsWO.add((BadDebtsWOP[i]/100.0)*NPL[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,BadDebtsWO);
////////////////////////////////////////////
```

```
MiscFees = List();
obj = model.getObject("Miscellaneous Fees");
MiscFeesP = GetYearListVals(obj, Years, "percentage");

i = 1;
while (i <= Years) {
  MiscFees.add((MiscFeesP[i]/100.0)*Revenue[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,MiscFees);
/////////////////////////////////////////////

MembershipF = List();
obj = model.getObject("Membership");
MembershipFPC = GetYearListVals(obj, Years, "");

i = 1;
while (i <= Years) {
  MembershipF.add(MembershipFPC[i]*CardHolders[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,MembershipF);
/////////////////////////////////////////////

Funding = List();
obj = model.getObject("Funding");
FundingP = GetYearListVals(obj, Years, "percentage");

i = 1;
while (i <= Years) {
  Funding.add((FundingP[i]/100.0)*RevVolume[i]);
  i = i + 1;
}

result = NPVscript:SetYearListVals(obj,Years,Funding);
/////////////////////////////////////////////

obj = model.getObject("Initiation");
Launch = obj.attrValue("Value_YA");
obj = model.getObject("Discount Rate");
DRate = (obj.attrValue("Value_YA"))/100.0;

//Costs:
obj = model.getObject("Costs");
Costs = List();
i = 1;
while (i<=Years){
  Costs.add(MembershipF[i] + StatementFeeC[i] + SystemFee[i] + CardIssuingFeeC[i] + DebtRecoveryFee[i] +
       BadDebtsWO[i] + BusDevelopFee[i] + Marketing[i] + MiscFees[i] + Funding[i]+ TransactionCosts[i]);
  i = i + 1;
```

```
}
aName = "Value_Y0_R";
obj.setAttrValue(aName,−Launch);
result = NPVscript:SetYearListVals(obj,Years,Costs);

////////////////////////////////////////////////
//Yearly Profits/Losses
obj = model.getObject("Profit/Loss");
Profit = List();
i = 1;
aName = "Value_Y0_R";
obj.setAttrValue(aName,−Launch);

while (i<=Years){
  Profit.add(Revenue[i] − Costs[i]);
  i = i + 1;
}
result = NPVscript:SetYearListVals(obj,Years,Profit);

////////////////////////////////////////////////
//Yearly Profits/Losses
obj = model.getObject("Break−even");
BE = List();

aName = "Value_Y0_R";
obj.setAttrValue(aName,−Launch);
BE.add(Revenue[1] − Costs[1] − Launch);
i = 2;
while (i<=Years){
  BE.add(BE[i−1] + Revenue[i] − Costs[i]);
  i = i + 1;
}
result = NPVscript:SetYearListVals(obj,Years,BE);

////////////////////////////////////////////////
//NPV
obj = model.getObject("NPV");
NPV = List();
NPV.add(−Launch);
i = 1;
while (i<=Years) {
  NPV.add(NPV[i] + Profit[i]/pow((1+DRate),i));
  i = i + 1;
}

i = 0;
while (i <= Years) {
  aName = "Value_Y" + i.toString() + "_R";
  obj.setAttrValue(aName,NPV[i+1]);
  i = i + 1;
}

if (model.hasTaggedValue("FirstRun")) {
```

```
    i = 1;
    while (i <= Years) {
      VAttr = "Value_Y" + i.toString() + "_R";
      type = "result";
      f_result = SetStochVar:TagObject(obj, VAttr, type);
      i = i + 1;
    }
  }
  return 0;
}
```

A.6 Profile Definition (MonteCarlo.mpd)

```
type StatDistribution= enum{
  NormalD,
  LognormalD,
  UniformD };

type Var_struc = struct {string selct_attr; string selct_attr_PA; StatDistribution Dist_Type; real dist_mean;
     real dist_std_dev; };
type Var_list = list Var_struc;

type Disc_Rand_Vals = struct {real Discrete_Value; real Discrete_Prob; };
type Disc_R_Val_List = list Disc_Rand_Vals;
type Disc_var_struc = struct {string selct_attr_disc; string selct_attr_disc_PA; Disc_R_Val_List
     Disc_R_Vals; };
type Disc_var_list = list Disc_var_struc;

PROFILE RandomVar {
  assignable to AbstractCompound;
  Var_list       List_Vars;
  Disc_var_list   List_Dis_var; };
```

A.7 Translation file (MonteCarlo.mps)

For the purposes of this project, only an English translation file was developed.

```
language = en

RandomVar = Random Variables

//The lists for the two types of random variables
RandomVar.List_Vars = List of Random Variables
RandomVar.List_Dis_var = List of Discrete Random Variables

// Two types of random variables
Var_list.Var_struc = Variable
Disc_var_list.Disc_var_struc = Variable
```

```

// Types for the random variable
Var_struc.selct_attr = Variable Name
Var_struc.selct_attr_PA = Attribute (auto−filled)
Var_struc.Dist_Type = Distribution
Var_struc.dist_mean = Mean/Min
Var_struc.dist_std_dev = Standard Deviation/Max

// Types for the discrete random variable
Disc_var_struc.selct_attr_disc = Variable Name
Disc_var_struc.selct_attr_disc_PA = Attribute (auto−filled)
Disc_var_struc.Disc_R_Vals = Values and their probabilities
Disc_R_Val_List.Disc_Rand_Vals = Value & probability
Disc_Rand_Vals.Discrete_Value = Value
Disc_Rand_Vals.Discrete_Prob = Probability

// Types of distributions
StatDistribution.NormalD = Normal
StatDistribution.LognormalD = Lognormal
StatDistribution.UniformD = Uniform
```

Appendix B

Interview Questions

Usability

1. Is it (easily) understood how the random variable profile works (i.e how to define a variable as random?)
2. Is it clear how to initiate & execute the Monte Carlo simulation?
3. Is it clear how to go about completing the analysis?

Intent

1. Is the basic concept and intent of the Monte Carlo method understood?
2. Is the intention of the provided means of analysis clear (for example the Histogram)?

Results

1. Are the results easily understood?
2. Are the means of representation adequate?

Applicability

1. Is the purpose of the artefact understood? I.e. to make enterprise architecture executable?

2. Does the artefact accomplish its intent in this regard?
3. Would business users benefit from this artefact?
   (a) In comparison to other means to execute such a simulation.
   (b) Easy to set-up, execute, analyse.
   (c) Easy to understand.
   (d) Will the artefact be useful in a wide range of applications?
4. Can you identify a better approach to solving the same problem?

General

1. Is any desired functionality missing from the results?
2. Any other general impressions on usability
3. Any recommendations?
4. What should the primary focus be for future developments?

Appendix C

Interview Minutes

C.1 Interview 1

Position:	Young Professional at InnoValor[1] (BIZZdesign sister company) and recent graduate from the University of Twente with a Master of Science in business administration.
Date:	30/10/2014

Usability

The method requires a large number of steps. Executing all of these steps either requires more guidance (such as in the form of a user manual), or a significant amount of user experience.

Intent

The intent of the various facets of the artefact are clearly understood, but the user has to have a certain amount of foreknowledge of the Monte Carlo method and the various means by which to apply it and analyse its results.

Results

The results seem useful, however some further means of analysis would be beneficial, such as to be able to zoom in on specific values (for example, the precise probability for net present value equalling zero).

Applicability

Although the basic intent of the artefact is understood, the exact advantages of the artefact, as opposed to some analysis application, was difficult to see. In other words, the exact business value of the artefact was questioned.

C.2 Interview 2

Position: Senior Research Consultant at BiZZdesign
Date: 31/10/2014

Usability

- In order to set-up, and execute the method requires a large number of 'clicks', from the set-up of the profile through to the input required by all of the scripts.
- The various set of structures and lists makes it difficult to find, and define the random variables.

Intent

The intent of the artefact is understood.

Results

- Business users are interested in results, other analysis, such as that considering the change in mean and standard deviation, are interesting, but less useful.
- Including more standard analysis options, for example to find present values for financial analysis, would be useful.

Applicability

The artefact could be useful to business users, but definitely requires the users to have the necessary know-how.

General

If the script could stop once the mean and/or standard deviation have reached a steady state.

C.3 Interview 3

Position: Product Manager at BIZZdesign
Date: 06/11/2014

Usability

Only a very limited number of business users have the necessary knowledge to develop their own analysis scripts. Either providing business users with some 'out of the box' analyses tools, or providing it as part of a consultation purpose could alleviate this problem.

Intent

The intent of the various facets of the artefact are clearly understood, but the user has to have a certain amount of foreknowledge of the Monte Carlo method and the various means by which to apply it and analyse its results.

Results

The graphical representations are comprehensible, but do require some explanation (somebody who has not been explained the results will not know which variable's results he is looking at).

Applicability

- EAs are typically administered by the IT departments. They do not necessarily have access right data and information required to develop analytics.
- Business need to have mature EA models, as well as data collection and processing capabilities before they can consider using the artefact extensively.

General

- Identifying the correct input data can be very difficult and cumbersome task.
- Users will need necessary background knowledge to make the Monte Carlo method work.