Meta Modeling for Business Model Design:
Designing a Meta model for E3 value model based on MOF

Akshay Rasiwasia (830130-6174)

DSV Master Thesis
Supervisor: Paul Johansson

Department of Computer and Systems Sciences
Royal Institute of Technology
Table of Contents

Abstract .......................................................................................................................... V

1 Introduction .................................................................................................................. 1
  1.1 Background ........................................................................................................... 1
  1.2 Problem ............................................................................................................... 2
  1.3 Research question ............................................................................................... 2
  1.4 Intended Audience .............................................................................................. 3
  1.5 Disposition ........................................................................................................... 3

2 Extended background ................................................................................................ 4
  2.1 Business Modeling: ............................................................................................ 4
  2.2 E3 Value Model: ................................................................................................ 5
    2.2.1 Viewpoints .................................................................................................... 5
    2.2.2 Important concepts ...................................................................................... 5
    2.2.3 Use Case Maps ............................................................................................ 6
    2.2.4 Shortcomings ............................................................................................... 7
    2.2.5 Advantages .................................................................................................. 7
    2.2.6 Motivation to adopt e3 value model: ............................................................. 7
  2.3 Other Major Business Modeling Methods ............................................................ 7
    2.3.1 Resource Event Agent (REA) ....................................................................... 7
    2.3.2 Business Model Ontology (BMO) ................................................................. 8
  2.4 Model Driven Approach (MDA) ........................................................................... 9
    2.4.1 Object Management Group (OMG) ................................................................. 9
    2.4.2 Unified Modeling Language (UML) ............................................................... 9
    2.4.3 Model Driven Approach (MDA) ................................................................ 9
    2.4.4 Meta Object Facility (MOF) ......................................................................... 10
    2.4.5 XML Metadata Interchange (XMI) ............................................................... 13
  2.5 Eclipse Metadata Interchange (EMF) .................................................................. 14
    2.5.1 Ecore Meta-Meta-Model ............................................................................ 14
    2.6 Visualization and Modeling Software Development Kit (VMSDK) ..................... 16

3 Method ........................................................................................................................ 18
  3.1 Design Science ..................................................................................................... 18
    3.1.1 Design Science Canvas ................................................................................ 19
  3.2 Explicate Problem ............................................................................................... 20
    3.2.1 Choice of Method ....................................................................................... 20
    3.2.2 Application of Method ................................................................................ 21
  3.3 Define Requirements: ......................................................................................... 24
    3.3.1 Choice of Method ....................................................................................... 24
    3.3.2 Application of Method ................................................................................ 24
  3.4 Develop Artifact: ................................................................................................. 26
    3.4.1 Choice of Method ....................................................................................... 26
    3.4.2 Design the meta-model ............................................................................... 28
    3.4.3 Implementation in EMF ............................................................................. 31
    3.4.4 Implementation in Microsoft VMSDK ......................................................... 39
3.5 Demonstrate Artifact: .................................................................43
  3.5.1 Choice of Method: .................................................................43
  3.5.2 E3 value model tool: ..............................................................44
  3.5.3 E3 value model editor in Eclipse ........................................49

4 Case Study: EU Supply Holding AB .........................56
  4.1 Observation ..................................................................................57
  4.2 Documents ....................................................................................58
  4.3 Interviews: ...................................................................................59
  4.4 Qualitative Analysis of Case Study Data ..............................60
  4.5 Results of Analysis .........................................................................61
  4.6 Case Study Report ..........................................................................63

5 Result ..................................................................................65
  5.1 E3 value Metamodel ........................................................................65
  5.2 E3 value modeling tool: ..................................................................68

6 Analysis: ................................................................................72
  6.1 Choice of Method ...........................................................................72
  6.2 Application of Method ....................................................................73
    6.2.1 Requirements on the metamodel: ...........................................73
    6.2.2 Requirements on the graphical tool: .......................................74
    6.2.3 Reflection on analysis ................................................................75

7 Conclusion .............................................................................75

References .....................................................................................i

Appendix ..................................................................................... vi
  7.1 Design Science Canvas .....................................................................vi
  7.2 EU Supply Holding AB ....................................................................xiii
  7.3 Observation Checklist ....................................................................xiii
  7.4 Document Checklist ........................................................................xiii
  7.5 Interview Questions .........................................................................xiii
  7.6 Observation Notes ...........................................................................xiv
  7.7 Interview Notes ...............................................................................xvi
  7.8 XML Serialization of the e3 value model .................................... xviii
List of Figures

Figure 2-1: EMOF model at the package level ................................................................. 11
Figure 2-2: UML class diagram of EMOF ................................................................. 12
Figure 2-3: Four Layered Architecture of MOF ....................................................... 13
Figure 2-4: Ecore Meta-metamodel .............................................................................. 1
Figure 2-5: A screenshot of VMSDK editor ............................................................... 17
Figure 3-1: e3 value metamodel .................................................................................. 1
Figure 3-2: EMF Uml editor with e3 value metamodel ................................................ 32
Figure 3-3: e3 value metamodel in ecore (first version) ............................................. 33
Figure 3-4: EMF generator model .............................................................................. 34
Figure 3-5: EMF editor based on e3 value metamodel ............................................... 35
Figure 3-6: e3 value metamodel without value offering in ecore ................................ 36
Figure 3-7: e3 value model in ecore (final version) ................................................... 38
Figure 3-8: e3 value metamodel implementation in VMSDK ....................................... 42
Figure 3-9: e3 value model of Airtel Teleservices .................................................... 45
Figure 3-10: Serialized e3 value model of Airtel Teleservices ................................. 46
Figure 3-11: e3 value modeling tool with validate option ........................................... 47
Figure 3-12: context menu on value interface ............................................................ 48
Figure 3-13: validation error in e3 modeling tool ....................................................... 48
Figure 3-14: Print option in e3 value modeling tool .................................................. 49
Figure 3-15: Eclipse based e3 value modeling editor ................................................ 50
Figure 3-16: Context menu on elements in eclipse e3 value editor ......................... 51
Figure 3-17: Validations in eclipse e3 value editor .................................................... 53
Figure 3-18: Serialized e3 value model in eclipse editor ............................................ 54
Figure 5-1: e3 value metamodel final version ........................................................... 67
Figure 5-2: E3 value model editor ............................................................................ 68
Figure 5-3: EU Supply Holding AB e3 value model ................................................... 69

List of Tables

Table 3-1: Design Science Canvas .............................................................................. 20
Table 3-2: Comparison summary of UML profiles, UML extension, MOF based metamodel ....................................................................................... 25
Table 3-3: Requirement analysis criteria .................................................................... 26
Table 4-1: EU Supply Holding business transactions .................................................. 62
Table 4-2: EU Supply Holding value activities ........................................................... 63
Table 5-1: E3 value model editor toolbox items ........................................................ 69

List of Abbreviation
<table>
<thead>
<tr>
<th>BM</th>
<th>Business Model/Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMO</td>
<td>The Business Model Ontology</td>
</tr>
<tr>
<td>CIM</td>
<td>Computation Independent Model</td>
</tr>
<tr>
<td>CMOF</td>
<td>Complete Metaobject Facility</td>
</tr>
<tr>
<td>CORBA</td>
<td>common object request broker architecture</td>
</tr>
<tr>
<td>CWM</td>
<td>common warehouse metamodel</td>
</tr>
<tr>
<td>DSC</td>
<td>Design Science Canvas</td>
</tr>
<tr>
<td>DSL</td>
<td>Domain Specific Language</td>
</tr>
<tr>
<td>EMF</td>
<td>Eclipse Modeling Framework</td>
</tr>
<tr>
<td>EMOF</td>
<td>Essential Metaobject Facility</td>
</tr>
<tr>
<td>ICT</td>
<td>Information and Communication Technologies</td>
</tr>
<tr>
<td>IS</td>
<td>Information Systems</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>MDA</td>
<td>Model Driven Approach</td>
</tr>
<tr>
<td>MOF</td>
<td>Meta-Object Facility</td>
</tr>
<tr>
<td>OCL</td>
<td>Object Constraint Language</td>
</tr>
<tr>
<td>OMG</td>
<td>Object Management Group</td>
</tr>
<tr>
<td>OOP</td>
<td>Object-oriented programming</td>
</tr>
<tr>
<td>PIM</td>
<td>Platform Independent Model</td>
</tr>
<tr>
<td>PSM</td>
<td>Platform Specific Model</td>
</tr>
<tr>
<td>REA</td>
<td>Resource-Event-Agent</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>VMSDK</td>
<td>Visualization and Modeling Software Development Kit</td>
</tr>
<tr>
<td>XMI</td>
<td>XML Metadata Interchange</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
Abstract

E3 value model is a conceptual tool to express the business logic of a firm from a holistic perspective, focusing on the value proposition that the firm offers and the architecture of its network of partners. It can be used as a tool of alignment between the business strategy and business process. However, the lack of concrete modeling principles and relationship between the components of a business makes extraction of knowledge difficult from e3 value models. It remains isolated from the rest of the business design and IS architecture. Software architects face difficulty in extracting knowledge from business models to design information systems. On the other hand, business analysts find hardship in communicating and controlling system design changes as the business model changes over time. To solve these problems, the thesis proposes the design and implementation of a well-defined e3 value meta-model such that the e3 models then created are compatible with UML and could be serialized as XML. The meta-model designed clearly defines the relationship between each e3 value model components and also adds a classification of value objects as an extension to the meta-model. To illustrate the usefulness of this meta-model, and its inherent validation properties to create a correct e3 value model, a graphical editor tool for end user is created. A case study is then conducted at a software company, and its business model is then implemented to demonstrate the efficiency of both the meta-model and the tool. The research work closely follows the design science guidelines and principles to achieve the results.
1 Introduction

1.1 Background

Information and Communication Technologies (ICT) have developed rapidly in recent years. The role of ICT in business has moved forward from being merely a support tool to meet business ends, and to provide new ways of doing business. Swift growth of E-Business, ‘electronic way of doing business’ has put focus on altering the conventional business models to better utilize the opportunities provided by ICT, and has provided a multitude of possible business configuration over tradition business practices for value creation and developing relationship with the customer and suppliers [1]. These increasing flexibility of choices and configuration, continuously evolving ICT sector, and increased competition with little margin for error have made the design and implementation of business model a very complicated task.

Despite the growing importance of a sound business model for the success of the business organization, the concept of business model is still vague and ambiguous[2]. Moreover, the concept of business modeling (BM) has its origins in diverse subjects such as Information Systems (IS), business management, technology, economics, business strategy, and e-Commerce. These varied sources of BM inspiration have resulted in a lack of cohesive and a solid understanding of BM definition and its components[2]. There is little reconciliation among the scholars on the definition, fundamental concepts, components and taxonomy of business model[3]. The place and role of BM in an organization is also debatable. The distinction between a business model and a business strategy is vague[4][2].

Different authors may denote dissimilar things when referring to a business model: it can be described as an abstract concept that defines all real world business or a set of BM each one describing a particular class of business domain or it may be a conceptualization of a real world business model of an organization[4]. According to [4], a business model(BM) is a conceptual tool to express the core business logic of an enterprise; describing the set of values that it offers to its customers, and the network of its partners to create, sell and deliver those values in a profitable and sustainable way. According to [1] [5], a business model is an efficient way to understand, evaluate, manage and convey business framework among both the business and technology stakeholders. According to [3], it also serves as the basis on which the information system is designed to support the BM of the enterprise.

According to[1], research within BM is classified in eight sub-domains a) BM Definitions, scope, b) fundamental constructs and components c) taxonomies d) conceptual models and relationship between the BM components e) Design method and tools, f) adoption factor g) evaluation models and h) change methodologies for a BM. According to[2], a comprehensive answer is required to the fundamental issues in BM. It has recognized four fundamental issues related to the 1) constituents of BM concepts; 2) modeling principles, features and guidelines for BM; 3) role and position of the BM concept within the organization; 4) significance and effectiveness of BM.

There exist several methods to capture and design the business model of a business enterprise [6]. The three major business modeling ontologies are: E3 value model ontology, Business Model Ontology (BMO) and Resource-Event-Agent (REA) ontology. The e3 value ontology is a conceptual business modeling tool that offers graphical representation of its components to depict the creation and distribution of economic value objects in a joint network of enterprise [7]. E3 model assumes a global perspective while defining the network
of business enterprise. The BMO[8] focuses on a single actor and his position in the business network and how he can generate revenue. Another popular approach, REA ontology, which specifies the involved economic agents (A), the resources exchanged (R), and the economic events (E) that trigger the exchange of resources from an economic viewpoint and thus justifies the business collaborations among the agents[9].

The thesis work is focused on e3 value models in the area of theoretical model and relationship between the BM elements, and the design methods and tools, which are the two research sub-domains defined in [1].

1.2 Problem

E3 value model is a light-weight and a semi-formal approach to represent business models consisting of multiple business entities and their business value transactions. The semi-formal approach, lack of concrete meta-model and well-defined relationship between the components of e3 value model makes structuring and codifying the knowledge within e3 value model difficult [4][3].

At present, e3 value models remains isolated from the rest of the business design and IS architecture. It is not integrated with other aspects of business design such as business process modeling and system architecture of business information systems [1][2]. It is not compatible with UML- the most popular modeling standard in software engineering. The lack of integration with software design and modeling frameworks poses difficulty in utilizing the knowledge stored in e3 value models while designing business information systems using the model-driven approach.

The e3 value modeling tool also suffers from many limitations. It lacks a validation framework to validate e3 value models, or guide users in creating a correct model. The e3 value model components lack attributes to hold additional information (e.g. descriptions). The serialized model as stored by the tool is complex and specific to the tool. It is not saved as an XML, which is the most popular standard for data transfer.

In a typical scenario, a business analyst designs the e3 value business model, which holds sufficient knowledge about the business of an organization, but requirement engineers and software architects find it difficult to extract relevant knowledge to design the information system. Much of the knowledge lies under-utilized due to inability to transfer it further. On the other hand, business analysts find difficulty in communicating and controlling system design changes as the business model changes over time.

The contribution of e3 value model to business systems design is rather limited and not realized to its full potential.

1.3 Research question

Based on the problem description in 1.2, following research question has been formulated:

Design a metamodel for ‘e3 value’ business modeling framework, so that:

- Relationships and constraints between the e3 value model elements are well-defined.
- E3 value business models are compatible with UML.
- E3 value models can be serialized as XML.
- Graphical editor tool for end users based on metamodel can be implemented.
1.4  **Intended Audience**

It is expected that this research work would help business analysts, requirement engineers, software architects and system developers to understand e3 value models from an object-oriented perspective and aid in the process of business information systems development using e3 value models in a model driven approach.

1.5  **Disposition**

The master thesis is structured in seven chapters followed by a reference list and appendix.

Chapter 1 essentially provides the background of the problem area, the specific problem which the thesis will address, the research question and objectives that the thesis work intends to achieve and the audience that may benefit from the result.

Chapter 2 gives a detailed description of all the important concepts involved or referred to in this thesis. It describes the origin and concepts of business models, the business model definition adopted for the thesis, core concepts of e3 value model, its strength and weakness, short description of other prominent business models. It also describes the fundamental concepts of model driven approach, Object Management Group (OMG) standards such as meta-object-facility (MOF), unified modeling language (UML), and xml metadata interchange (XMI). Further, it gives a brief description of development software used - Eclipse Modeling Framework (EMF) and Microsoft’s Visualization and Modeling Software Development Kit (VMSDK).

Chapter 3 describes the method and its application for creation of e3 value metamodel and the graphical tool. It has five subsections. Section 3.1 explains the concept and application of Design Science methodology for this research; the subsequent sections 3.2, 3.3, 3.4, 3.5 are the steps followed in this research based on design science - explicate problem, define requirements, develop artifact and demonstrate artifact respectively. Each section describes the choice of the method, its application and result.

Chapter 4 describes the case study done to understand its business model at a software company.

Chapter 5 presents the important contribution of the thesis - the e3 value metamodel and the graphical editor tool. The business model from the case study in chapter 4 is implemented here to showcase the tool.

Chapter 6 presents the analysis of the result or the evaluation of the artifact, the final step in the design science approach.

Chapter 7 is a brief discussion on the overall research work, its contribution, its relation to previous research, limitations and possible future work.
2 Extended background

2.1 Business Modeling:

BM has its origin from multiple disciplines such as e-Business, strategy, information systems, management and economics [1]. The concept of BM is fairly new which started gaining attention since late 1990s[4]. Moreover, business modeling is increasingly being applied in new technological ventures with inter-organizational collaborations and continuously evolving relationship between stakeholders[4][2], but the definition of a BM is still under debate. Different researchers have defined the concept of BM in different ways, which overlap with each other, but hard to reconcile to a single definition [1][4][2][3]. These researchers have done extensive review of the business model definition given by different researchers in different domains since the early days of the BM concept.

According to[1], a BM is defined either as a purely business concept describing the business logic of a firm or as a link between strategy, business process and information system of the firm It concludes that a BM is not a purely management level concept, but with the rise of networked organizations and e-business, BM also includes the concept of inter-organizational elements.

According to[2], a BM has four fundamental dimensions: value proposition, value architecture, value network and value finance. BM is a conceptual framework that provides a comprehensive but abstract view of a firm’s business logic. It is dynamic and can be adapted for multiple purposes at various levels. It also concludes that BM acts as an intermediary between business strategy and business process modeling leading to IS architecture.

According to[4], BM can be described as a three level hierarchy consisting of an abstract over-arching concept at the top leading to classification of BM based on set of business having common characteristic and finally concrete conceptualization of real world business. It concludes that BM is a holistic concept that embraces nine major building blocks: value proposition, customer, distribution channel, relationship, value configuration, core competency, partner network, and cost structure and revenue model. It regards BM as a conceptual link between business strategy, business process and IS.

According to[3], the discussion on the concept of BM has centered around four themes: a) as a new unit of business analysis that covers that firm and its network b) a systematic and holistic approach towards business logic that involves not only what a business does but also how it is done, c) it conceptualizes not only the activities performed by the firm but also its partners, suppliers and even customers; d) value creation is the main emphasis of all the business models.

Based on the above discussions, a definition embracing the common concepts for BM has been adopted for this research work. A business model is defined as follows:

A conceptual tool to express the business logic of a firm from a holistic perspective, focusing on the value proposition that the firm offers and the architecture of its network of partners with which it collaborates to create, distribute and market the values in an economically sustainable way. It also acts as the conceptual tool of alignment between the business strategy and business process leading to business information system.
2.2 E3 Value Model:

E3 value modeling is a light-weight, semi-formal, conceptual and multi-viewpoint approach drawing inspiration from business science, requirements engineering and conceptual modeling techniques from software engineering to model e-business ideas using graphical notations [7]. The main focus of e3 value modeling is on business value viewpoint - how things of economic values are created and exchanged in an inter-organizational network[7].

E3 model was designed to provide a replacement for ad-hoc representation of business models with a modeling approach that combined the rigor of IT system analysis with economic perspective of value from business science. It was intended to be the first step in requirement analysis of a multi-enterprise partnership and e-business operations requirement both qualitatively and quantitatively[10]. It aimed to fulfill the gap between business oriented approach which were not specific or detailed enough for IT system development and conventional requirement engineering approach in IT which lacked commercial perspective. An e3 value model essentially represents the exchange of things of economic value between actors unlike UML activity diagram which represents the flow of event. E3 value model also provides a way to evaluate the feasibility and profitability of different business value models using profit sheets and evolutionary scenarios[11].

Based on[10] and [7], a short description of e3 value ontology and use case maps are as follows:

2.2.1 Viewpoints

E3 value ontology is structured along three business sub-viewpoint:

1. Global actor viewpoint – which shows the actors involved and the cause and effect of economic reciprocity between the actors.
2. Detailed viewpoint – shows the collaboration between actors to offer or request values jointly.
3. Value activity viewpoint which shows the assignment of value creation to actors.

2.2.2 Important concepts

Some of the most important concepts in the e3 value ontology are:

Actor: an economically independent and sometimes a legal entity involved in increasing its value or making profit by participating in value creation or exchange activities. A business model is only viable if all actors make profit. An actor has a name that represents its role.

Value Object: product or service or money, which actors value and exchange among themselves. Actor value objects based on their preferences and the difference in value perception leads to exchange. There are basically four different types of value object: goods, money, services and information.[12]. Value object represents the type of object exchanged and not their individual instances.

Value Port: is used to request/provide value objects from/to actor to the external environment. It abstracts the internal working of an actor and focuses on how external actors and other components are connected to the actor. It is directional (in-port or out-port).

Value Interface: Value interfaces group together individual value ports. It represents the atomic level of value exchange between the actors. Either value exchange occurs at all the ports within the value interface or none at all. An actor can have multiple value interfaces.
Value Exchange: A value exchange represents the possible trade of values between two value ports. It connects a value port from one actor to the value port of another actor.

Value Offering: It is a set of equally directed value ports which implies that the actor requests/offers value objects in combination. Either value exchange occurs through all the ports in a value offering or none at all.

Value transaction: It is used to denote a sequence of value exchange that must occur between two or more actors as a consequence of how value objects are connected to one another. Within value transactions, the rules of value offering and value interface must not be violated.

Market segment: It is used to denote a large number of actors who assign equal economic value to value objects, but actors are independent individuals, and they may have value interfaces other than that represented in the market segment. A market segment does not actually group the actor themselves but groups the value interfaces which exchange objects valued equally by all participating actors.

Composite actor: A composite actor is used to denote the partnership between actors who decide to represent themselves as a virtual enterprise to the external environment, requesting or offering value objects together. A composite actor only groups the value interface of the participating actors but not the actors themselves. An actor can also offer some of the service independently from the composite actor.

Value activity: A set of operational activities that can be assigned as a whole to an actor is represented by a value activity. A value activity has value interfaces, and it is performed by one elementary actor. A value activity is only performed if it increases economic value for the performing actor.

2.2.3 Use Case Maps

E3 value ontology incorporates Use case maps (UCM) to represent operational scenarios to capture parts of e-business value model and also for better understanding of the model. UCM is graphical notations to understand system behavior at a high level. Ontology of UCM used in the context of e3 value modeling is as follows:

Path element: It is used to connect value interfaces to represent cause and effect relationship when values are exchanged at one of the interfaces.

Stimulus element: A use case map starts with at least one start stimuli and ends with an end stimuli. Start stimuli represents an event that triggers the UCM while end stimuli represents the end of the UCM.

Connection: They connect individual path segments. ‘AND’ join mergers several sub-paths into one while ‘AND’ fork splits one path into two or more sub-paths. Similarly, ‘OR’ join merges two or more sub-paths into one, while ‘OR’ split offers the choice of choosing one sub-path among the available, multiple sub-paths. Direct connections connect two individual path segments. ‘AND’ and ‘OR’.

Scenario path: A scenario path represents a specific route in the use case map. It has at least one start and end stimuli and a set of path elements. The path elements may be connected by ‘AND’ or ‘OR’ join or fork. Each path segment can have only one successor and predecessor.

Stub element: A map may be plugged to another map using a stub. A stub hides the details of maps to be attached and offers connectors with the same start and end stimuli of the attached map to other maps..
2.2.4 Shortcomings

Business is dynamic and with time the value model evolves and the interactions between actors change[11]. Moreover, profitability analysis is only a subjective way to analyze depending on the perceptions of economic value of different objects by the designer/stakeholder. Also, it does not account for the technical feasibility of a business model. One more drawback of e3 model is the lack of integration with business process models/viewpoint and IS model/viewpoint. The viewpoints are inter-dependent, and changes in one viewpoint necessitate changes in others. Thus, it is a cyclic process and to be effective e3 must be integrated with other viewpoints[11].

2.2.5 Advantages

According to[8], the three most well-known business modeling techniques are e3 value model, Resource-Event-Agent (REA) and Business Modeling Ontology (BMO). In [13], a classification of most important methodologies and concepts in B2B according to ‘Open-edi reference model’, classifies e3 value model as one of the major business modeling technique along with BMO. REA model is based on the concept that all multi-actor scenarios can be decomposed and represented as binary transactions [14]. One advantage of REA is that it can be easily represented in UML diagrams. However, BMO specifically focuses on the business model from one actor’s perspective whereas REA becomes too complex and unmanageable when handling multiple-actor scenarios [14]. In [14], it has been shown that concepts from e3 value model can be mapped to REA since the concepts in e3 have been adapted from original REA model, but the graphical representation of e3 model makes it easy to use and understand than REA. E3 value model is a lightweight graphical approach and is effective in establishing a common understanding between different stakeholders about the e-commerce system[15].

2.2.6 Motivation to adopt e3 value model:

Based on the above considerations, it can be concluded that e3 value model is one of the major business modeling technique which is flexible, scalable and easy to use. It is better equipped to represent multiple-actor e-business model than other major BM techniques like REA and BMO. There is a possibility to enhance the benefits derived from e3 modeling if it can be brought closer to software modeling and development. It would help to specify the business motivation for the design of the information system from the IT point of view.

2.3 Other Major Business Modeling Methods

2.3.1 Resource Event Agent (REA)

REA ontology was initially developed to conceptualize the accounting framework of an organization to analyze the economic exchanges within and across the organization. Gradually, its concepts and granularity were extended, and new components added to realize it as domain ontology [16].

The core focus of the REA model is the economic exchanges or conversions that take place within the organization and also with external business entities. These exchanges/conversions follow a pattern where a thing of economic value (resource) is given out by an agent A (economic unit) to another agent B in an economic transaction (event); this decrement event is always paired with a similar economic increment event where agent A gains a resource of greater economic value as perceived by its organization. This is the underlying concept of the REA model where it is assumed that all business entities follow a similar pattern
where things of economic values are given out to other business entities in a series of exchange or consumption process to gain things of greater economic value and thus, generate profit [16].

The major strengths of REA are that since it is based on accounting discipline, all its concepts are well grounded in the economic theories [17]. Additionally, it has been developed through the continuous use of abstraction mechanisms like ‘aggregation’, ‘classification’, ‘generalization’ which means that all its concepts follow the semantic representations of ontology. Moreover, concepts in REA model have been tried and tested in multiple disciplines such as accounting, micro-economics, computer science and knowledge representation [16].

Some of the major limitations of REA highlighted by [17] are that the relationships among the REA concepts have some ambiguities which lead to diverging interpretations. For instance, the multiplicity of relationship between core concepts such as event and resource is not well defined. It is not clear if the increment/decrement of resource occurs due to multiple events or a single event. There exists no modeling language to specify exactly the relationship between REA concepts and to implement compliance of REA models to stated axioms. Moreover, the complexity of the model increases exponentially as the number of process or the number of business entities increases marginally. There is no dedicated REA modeling notation to address this problem.

2.3.2 Business Model Ontology (BMO)

BMO consists of four main components [8]:

1) Products: This represents the ‘value proposition’ that an organization offers to its ‘customer segments’ and its ‘capabilities’ to deliver those values.

2) Infrastructure: This represents the organization’s activities or processes, its internal resources and network of partners or inter-organizational ties to assist in delivering value.

3) Relationship with Customers: This represents the strategies with which the organization maintains and improves its customer relationship. The important elements are the a) ‘information strategy’ with which it gathers, maintains and exploits customer information b) Distribution Strategy, the multiple channels through which it reaches the customer c) Customer Loyalty, methods to establish customer trust in the organization and to achieve customer satisfaction.

4) Finance: This includes all the financial components of the business such as the ‘revenue model’ which describes the cost and liabilities of the organization and profit generated from the value offered to its customers.

The advantages [17] of BMO are that it is easy to use for domain experts and focuses on the categorization of the important aspects to create value. It encourages creativity and places very little restriction on the business model design. However, it has many limitations [17] considering that it offers only a single organization viewpoint of the multiple-entity business environment. It offers no conceptualization of the business concepts which makes it difficult to analyze or check for compliance programmatically.
2.4 Model Driven Approach (MDA)

2.4.1 Object Management Group (OMG)

OMG is consortium established in 1989 that defines and maintains the standards for interoperable enterprise software working in heterogeneous and distributed environment. Some of the important OMG specifications include UML, CORBA (common object request broker architecture) and CWM (common warehouse metamodel)[18]. OMG does not develop any software on its own but develops and distributes the computer industry standards and guidelines which are developed by an open community of its members.

2.4.2 Unified Modeling Language (UML)

UML is an industry-wide standard visual modeling language for software architects, engineers or developers to design and document the software systems using both text and graphical notations. It can be used across different application domains as well as technology platforms as an all-purpose modeling language[19].

UML consists of different types of diagrams used to specify different features of a software system. Class diagrams[20] describe the static structure of a system and the relationship between different elements of the system. Activity diagrams[20] or sequence diagrams[20] are used to capture the dynamic behavior and the interaction between the different elements of the system. Use case diagrams[20] are used to store information about the system requirements and workflow.

The UML specification is maintained by the OMG group. OMG has defined the UML specification using a meta-modeling approach where the UML meta-model is described based on Meta Object Facility (MOF). UML specification is organized in two volumes as UML Infrastructure and UML Superstructure. UML Infrastructure consists of two packages – Infrastructure Library and Primitive type. Infrastructure Library consists of ‘Core’ and ‘Profile’ packages. The ‘core’ package is at the center of MDA. MOF, CWF and UML all depend on this core package, and each of them either imports it or specialize its meta-classes[19].

Since the graphical notation cannot describe all the information about the system, UML specification contains a ‘Comments’ package that specifies the ability to annotate elements with additional textual information. OMG also defines standards for a declarative language – ‘Object Constraint Language’ (OCL) [21] which is used to describes rules and constraints attached to elements of a UML model.

2.4.3 Model Driven Approach (MDA)

Model Driven Architecture is an architectural framework proposed by OMG which positions models at the center of information systems development. It aims to use modeling artifacts to manage the abstract information about the business aspects, functional aspects and implementation requirements of the information systems and then generate executable code using the models. According to MDA, models represent the formal specifications of a system at different levels of abstraction using both textual and graphical notations[20].

MDA defines two major types of models [20]:
Platform Independent Model (PIM): These models capture information about the systems and the business processes in a platform independent way.
Platform Specific Model (PSM): These models provide the implementation details of systems with respect to a particular platform.

The MDA approach consists of using a number of models at various level of abstraction to represent the system rather than a single super model. These various models are linked to one another through a well-defined relationship and mapping among them. These models can be based on different metamodels and can consist of their own set of views [20].

MDA proposes three levels of abstraction for complex systems. The Computation Independent Model (CIM) as the highest level of abstraction which represents the business model supported by the underlying information system. CIM captures the major elements of the business or the domain it represents. CIM is mapped to PIM which represents the computational aspects of the system and the business processes. Finally, the system designed as PIM and is mapped to PSM [20].

2.4.4 Meta Object Facility (MOF)

MOF is a meta-meta-model defined and maintained by OMG. It is used to define metamodels for models based on object oriented principles. The metamodel of UML and XMI is defined using MOF. A model is a schema that represents the general knowledge about a domain, and a metamodel is a schema that represents the general knowledge about the model in the domain[22].

Metadata is information about the data. It is used to explain the significance and structure of the data. The exchange of information among different applications is rendered limited due to incompatible metamodels. MOF provides a framework and a set of services to develop metamodels which are interoperable. MOF has also contributed towards the basic principles of MDA by introducing the concept of formal metamodels, platform independent models (PIM) of metadata and their mapping to specific platforms. Standards such as UML, CWM are examples of such metamodels. MOF helps in maintaining the relationship between different models as all the metamodels reference MOF. MOF manages all the models in a way that their respective metamodels are MOF complaint, which in turn facilitates interoperability between models and different modeling tools[20]. The first specification of MOF was released in Nov 1997 while MOF2 is the latest version. MOF2 core and MOF 2 XMI will be used for the thesis work. MOF2 core is the foundation for designing metamodels within the family of MDA languages. MOF2 XMI is the basis for transforming MOF based model to XML format[18].

MOF2 has been designed to share the core modeling concepts between UML2, MOF2 and other OMG metamodels. Moreover, the UML2 infrastructure library has been reused in MOF2. As a result, metadata modeling using MOF2 requires the knowledge of only a subset of UML class modeling concepts without any additional constructs or notations. Most of the UML modeling tools can also be used to create metamodels in MOF2. MOF2 can be used to define metamodels based on object oriented or non-object-oriented principles[18].

MOF2 has been designed to be very modular, reusable and platform-independent. Thus, it facilitates mapping to many technology platforms and MOF implementations can be exchanged between these platforms.
Modular design of MOF makes it possible to use the parts of specification separately. Packages are used to group together related elements. These packages can be easily imported to other packages and it allows using the elements from the imported package in the importing package directly. This approach allows us to reuse the concepts and also extend a metamodel as and when needed[18].

MOF2 model consists of two main packages – Essential MOF (EMOF)[18] and Complete MOF (CMOF)[18]. Other packages that are used to extend metamodels are the Identifiers[18], Reflection[18], Extension[18] and Common[18]. Package EMOF is used to design simple metamodels based on the most basic concepts which can be easily extended for more complex metamodels using CMOF. All the MOF packages are defined using UML models. EMOF uses only the most basic concepts from UML along with some constraints on them to create metamodels. Some of the concrete metaclasses used from the UML’s kernel in EMOF are – class, association, data type, enumeration, generalization, operation, package, parameter, literal string and primitive types[18]. Following figure shows an overview of the EMOF model at the package level.

![EMOF model at the package level.](image)

A simplified uml class diagram of the EMOF classes, types and data-types is shown in the figure 2.2.
The main modeling concept in MOF is the concept of classifiers/class and it instances/objects and the ability to navigate from instance/object to its metaobject/class. Thus, MOF requires a minimum of 2 layers to
navigate between metaobject and its object while the maximum number of layers can be as many as the user defines [18].

For the purpose of this thesis, a four layered approach has been adopted consisting of M0, M1, M2 and M3 layers. M3 layer represents the MOF (Meta) classes, associations and other MOF elements that will be used to define the meta-models. An M3 layer is referred to as meta-meta-model. M2 layer represents the meta-models such as the UML metamodel defined on MOF meta-meta-model. M1 layer is a model, an instance of the meta-model such as a UML model. M0 layer is an instance of the UML model defined in M1. M0 model represents the real word entities and concepts. The following diagram displays the four layered architecture used in the thesis.

![Four Layered Architecture of MOF](image)

Figure 2-3: Four Layered Architecture of MOF.

### 2.4.5 XML Metadata Interchange (XMI)

A model defines entities and the relationships among them, but it does not provide any specifications on how to represent physically those concepts. As a result, different modeling tools using the same modeling language may store data and physically represent the models in a different format. It is an OMG standard to serialize model data in XML for models which has its metamodel based on MOF schema. Using XMI, the model instances between different modeling tools can be exchanged. The instances of MOF schema are serialized to XML representation by applying a set of rules to the entities, attributes and relationships in the instances [22].
2.5 Eclipse Modeling Framework (EMF)

EMF is an open source, low cost, modeling framework based on java and distributed under eclipse framework [23]. It provides the functionality to create structured models either as XML or java interfaces or as UML diagrams and then transform from one form to another or generate implementation classes[23]. EMF takes modeling a step further in the model driven development context such that a model not only represents a high level abstraction of the domain but modeling becomes equivalent to programming. Models can generate implementation code and vice versa using EMF[23].

The meta-model in EMF that is used to represent models is called Ecore. Since ecore is itself an EMF model, it can also be called a meta-meta-model[23]. Ecore is very similar to the EMOF specification provided by OMG except for a few naming differences. EMF can easily read and write serializations based on EMOF. In fact, EMF started as an implementation of MOF specification and gradually developed as an API for java implementation of model-driven development [24]. XMI is the official representation of the Ecore model in a serialized format[23].

EMF consists of two major frameworks the EMF core framework and the EMF.Edit. The core provides facilities for the basic model creation and generation of implementation classes while EMF.Edit framework extends these functionalities to create an eclipse plugin to provide a basic model editor. The EMF.Edit can be used to build a model editor on top of the Ecore metamodel to create model instances [24].

2.5.1 Ecore Meta-Meta-Model

A simplified hierarchical class structure of ecore meta-meta model with key attributes and associations based on the description in [25] is described in the following diagram.
Figure 2-4: Ecore Meta-metamodel.
The class names mentioned in italics’ are abstract classes. EObject is the base class from which every model element inherits. ENamedElement defines one attribute ‘Name’ and most of the ecore classes extend this super-class to inherit this attribute. ETypedElement contains an attribute eType which is also inherited by many ecore classes [25].

EClass denotes a modeled class which has a name, zero or more attribute and references. EDT dataType signifies the data types of the attributes which can either be a primitive type (such as integer, Boolean, string) or object types (book). EClass and EDT dataType inherit from the EClassifier. EClassifier has attributes such as ‘instanceClassName’ which defines the name of the class or interface; ‘defaultValue’ attribute defines the default value of the class or the data type. EClass has a reference to EStructuralFeature to access all the structural features (EAttributes and EReferences) belonging to the class. EClass has attributes ‘abstract’ and ‘interface’ to denote if the class is abstract or an interface respectively. EDT dataType models a single data type, which can be a primitive data type, a class, an array or an interface. It contains one attribute ‘serializable’ which denotes whether a data type can be serialized [25].

EAttribute denotes modeled attribute that has a name and type. It contains an attribute iD which is used to identify the instance of the containing class. EReference denotes one end of a binary association between classes. It includes a flag to denote if it is a composite relation, a name and reference type – which is another class. EAttribute and EReference inherit from a common base class EStructuralFeature [25].

EOperation is used to define the behavioral features of EClass. It contains zero or more EParameters and has bi-directional association with EClass [25].

Enumerated type is a data type which can take a value only from an explicit list of values called ‘literals’. EEnum denotes zero or many EEnumLiterals using a bidirectional association with EEnumLiteral class. EEnumLiteral class inherits attribute ‘name’ from ENamedElement and has another attribute called ‘value’ which is an integer type [25].

EPackage models packages that are used to group similar data type and classes. EFactory models factories which creates instances of classes and data types in the package. EPackage has two attributes nsURI and nsPrefix; nsURI contains the URI that uniquely identifies the package and nsPrefix contains the corresponding namespace prefix. EFactory inherits only from EModelElement which is the base class for every ecore element. EFactory does not inherit from ENamedElement and so it does not have a name attribute [25].

EAnnotation models annotations; annotations are used to add extra information to objects in ecore model[25].

2.6 Visualization and Modeling Software Development Kit (VMSDK)

VMSDK is used to create model-based tool for development and can be integrated in visual studio. It allows creating a user-defined meta-model definition, which can then be used to create models, in code generation, or combined with other tools to create a customized toolset. It has an editor to define the meta-model and graphical notations associated with the meta-model elements. Once the meta-model and graphical notations have been defined, VMSDK creates a graphical tool to create models. It also provides other facilities such as
tree based explorer for models, model serialization in XML, code generation using templates. All these features can be customized and extended[26].

The main components in the VMSDK are [27]:

- Toolbox: it contains domain elements and domain relationships.
- Domain Model Design Canvas: the meta-model is created here using domain classes and the three types of domain relationships - aggregation, reference and inheritance.
- Graphical Notation: It consists of ‘Geometry Shape’, ‘Image Shape’, ‘Port Shape’, ‘Compartment Shape’, ‘Connector’. Geometry shapes represent geometrical figures like rectangles, circles, ellipses; Image shape represents domain concepts as images; Port shapes are rectangle or circular shapes that get attached on the edge of the other geometrical shapes. Connectors represent relationship connections – aggregation, association and inheritance between the domain classes. ‘Diagram Element Map’ provides relationship connection between shapes and domain classes.

It has similar features as in EMF, but the tools developed on it can be used only on Windows operating systems. The meta-metamodel used in VMSDK is a proprietary of Microsoft and is not available publicly. However, it seems to be similar to MOF except for the existence of attributes on the relationship. The XML serialization of the models is also Microsoft proprietary format but similar to XMI standards.

Figure 2-5: A screenshot of VMSDK editor.
3 Method

According to [28], there are two paradigms to acquire knowledge in IS domain – behavioral science and design science. Behavioral science, which is based on natural science, deals with theories that seek to explain the people-technology interaction under the influence of organizational environment, and Thus, impact the design decisions. On the other hand, design science based on engineering is primarily a problem solving paradigm that aims to improve the overall process of IS development, its management and use, by applying innovation and critical thinking to develop practically applicable solutions and tools which not only solves the problem at hand but also extend the boundaries of human knowledge and organizational capabilities.

The aim of this research is to solve an existing practical problem in the realm of business modeling by creating an artifact based on the existing theories but applying creativity, innovation and problem solving capabilities from the researcher. The created artifact would then be practically applied to bridge the gap between business modeling and design of IS to support the business. Thus, the research clearly lies in the domain of design science research in information systems.

Ethical considerations were kept in mind during the entire research work and have been discussed as and where applicable.

3.1 Design Science

According to [28], Design Science is a problem solving paradigm based on the principles of science and engineering where knowledge of the problem and its solution is achieved by construction and application of the designed artifact. Within the context of IS research, the artifact which can be a construct, a method, a model or an instantiation, is produced by applying or extending existing theories based on the researcher’s innovation, experience and problem solving skills.

The conceptual framework of design science as described in [28] includes the environment where the problem of interest lies. The environment is composed of people, technology and organization. The other component is the knowledge base which is the foundation for the research work. Once the problem based on the business needs is identified, design science proceeds with the research by building and evaluating the artifact in an iterative loop; feedback from the evaluation is used to improve the artifact until the artifact is efficient enough to address the problem.

The important guidelines provided by [28] for a design science research in IS domain are:

1) The research should result in a practically useful IT artifact
2) It must provide a technical solution to an important and complex business problem
3) The produced artifact must be rigorously demonstrated and evaluated for quality and effectiveness.
4) It must contribute to the existing knowledge base
5) The research must be based on scientifically acceptable research and evaluation methods.
6) The research is an iterative search process where available resources are used to achieve desired goals while maintaining the environmental constraints.
7) The research must be communicated effectively to both technical and management people.

A common design science methodology based on the above guidelines and other prominent research on design science has been constructed in [29] which identifies six major activities:
Activity 1: Define and motivate a research problem using the knowledge of the problem domain and the significance of its solution. It is recommended to breakdown the problem to the atomic level to understand its complexity.
Activity 2: Define the solution objectives which are feasible and acceptable based on the identified problem, available resources and effectiveness of any existing solutions.
Activity 3: Construct the actual artifact. It includes the design and development of the artifact.
Activity 4: Demonstration of the artifact by using it over one or more problem instances. Demonstration can include experimentation, case study or any method that effectively demonstrates the usefulness and usability of artifact.
Activity 5: The artifact is evaluated to measure its degree of achievement of the objectives. The actual results are compared with the desired objectives by using relevant metrics and analysis method. Analysis can be both qualitative and quantitative. This activity may also lead to iteration to activity 3 to improve the developed artifact or researcher can move to the next step and leave the improvements as future work.
Activity 6: The final activity includes communication of the research work to the relevant audience and other researchers. Communication can be in the form of scholarly research articles.

The above activities though mentioned sequentially, a researcher can start from any activity depending on the scenario. As for an example, if the research focus is to solve a problem, activity 1 will be the starting point; if the focus is utilizing an existing artifact to different problem domain, activity 3 can be a starting point.

3.1.1 Design Science Canvas

In [28], seven main design science research guidelines are provided which every design science project must fulfill. These guidelines are well-reflected in the ‘Design Science Canvas’ (DSC) from the book [30]. The ‘Design Science Canvas’ is a very compact but comprehensive representation of the processes in a design science project. It is concise, user-friendly, and practical. Therefore, it has been adopted as a framework to conduct this research.

The DSC is a multi-purpose tool which has been used extensively throughout the project. It has been used to prepare the outline of the project and then updated iteratively to improve the project definition and fill in precise details for each phase of the project along the course of the project. It was also used as a tool for communication and better understanding of the project with all the stakeholders.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Artifact</th>
<th>Knowledge Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>Describe the problem being tackled in an accurate and concise way. Describe the significance of solving the problem. List down all the possible stakeholders impacted by the problem and who would</td>
<td>Describe the type and details of artifact produced and how it addresses the problem.</td>
<td>Describe the existing important theories, artifacts, knowledge, best practices used as the basis of the work and how these knowledge was utilized.</td>
</tr>
</tbody>
</table>

benefit from its resolution.

### Scope
Discuss the scope of the developed artifact, its context of use, and its limitations.

### Requirements
Describe the specific functional requirements on the artifact and non-functional requirements pertaining to environment, construction. Justify the requirements fulfill the resolution of the problem.

### Constructs
Describe the major concepts, key words used in the project.

<table>
<thead>
<tr>
<th>Explicate Problem</th>
<th>Define Requirements</th>
<th>Develop Artifact</th>
<th>Demonstrate Artifact</th>
<th>Evaluate Artifact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify the problem.</td>
<td>Decide the artifact to construct.</td>
<td>Create the artifact based on the collected requirements.</td>
<td>Demonstrate the practical use of the created artifact on the given problem.</td>
<td>Evaluate the effectiveness of the artifact.</td>
</tr>
<tr>
<td>Formulate the problem in clear and concise manner.</td>
<td>Define each requirement in clear and concise manner.</td>
<td>Clearly describe the development methods and tools used.</td>
<td>Clearly describe and justify the case chosen for demonstration.</td>
<td>Describe the method used to evaluate the artifact.</td>
</tr>
<tr>
<td>Ensure the problem has wider interest.</td>
<td>Ensure the requirements are proper, consistent and achievable.</td>
<td>Discuss the design decisions involved in the development.</td>
<td>Discuss how the features of the artifact were used in demonstration.</td>
<td>Justify if the artifact fulfills the requirements and solves the identified problem.</td>
</tr>
<tr>
<td>Ensure problem is solvable.</td>
<td>Clarify the process how the requirements were created.</td>
<td>Justify the originality of the artifact.</td>
<td>Discuss the features of the artifact.</td>
<td></td>
</tr>
<tr>
<td>Identify the source of problem.</td>
<td>Justify the requirements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Describe the process of how the problem was identified</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Construction
Document the design rational used in the construction of the artifact. The design rational can be one of the significant contributions of the project. It can also be reused.

### Function
Document the functions and features of the artifact. Document how these features fulfill the requirements and solve the problem.

### Usability
Discuss the usability of the artifact. Guidelines and best practices to use the artifact. Ways to improve the usability.

### Effects
Discuss the advantages and disadvantages of the artifact. Discuss its social and ethical effects. Discuss the effectiveness of the artifact in solving the problem.

<table>
<thead>
<tr>
<th>Table 3-1: Design Science Canvas</th>
</tr>
</thead>
</table>

The blue boxes on the top describe the different aspects of artifact being constructed. They describe the problem statement, requirements, type and features of the artifact and knowledge base used to create the artifact. The green boxes in middle describe the five main activities in the design science project. It starts from problem explication on the left and continues to artifact evaluation on the right. The bottom red boxes describe the outcome of the project classified as the design rational followed during construction, functions of the developed artifact, usability perspective the overall effectiveness of the artifact[30].

Based on DSC 5 green activity box, the project is divided into five main activities: ‘Explicate Problem’, ‘Define Requirements’, ‘Develop Artifact’, ‘Demonstrate Artifact’ and ‘Evaluate Artifact’. The design science canvas created for the research work here has been attached in appendix.

### 3.2 Explicate Problem

#### 3.2.1 Choice of Method

Problem explication involved investigating the problem scenario to identify the problem, its cause and motivation to solve. The process was iterative, and the problem was refined as the new knowledge and scenarios were discovered. The problem was broken into a number of sub-problems. This gave a clear and concise view.
There were many choices of research method for data collection in this step. Observation[31] could have been used to observe how managers and engineers use and perceive business models, but it would have required a lot of time, access to the physical environment and establishing the relationship with the subject of observation. Interviews[32] with experts on business modeling could have generated useful information but getting access to many experts and managing to get interviews were difficult. Questionnaires[33] would not have been useful as the focus of the research was creating a design artifact and required greater involvement from the correspondents. The research method used for this activity was ‘Literature Study’.

Literature review is a significant ingredient of any research. It helps to ensure that the researcher has consulted relevant literature and present work in that domain [34]. According to[35], the primary source of relevant literature is the university or the state library along with their online extensions. Internet is also a rich source of information, but since most of the content is unchecked and lack proper validation, they may not be very reliable.

There are two prominent ways of searching either keyword searching or browsing through references and library catalogs. Online search usually results in a large number of articles. They have to be limited in some ways to narrow down to most relevant articles. Keyword search with specific terms limiting the articles to last fifteen or ten years is a way to narrow down. Online databases can be searched for journal articles based on keywords, year of publication, author, publishers and many other criteria [34][36].

Relevance and reliability of an article must be assessed to establish the article’s importance. To assess relevance - for books check its index, introduction, summary of a few of the chapters, bibliography for relevant titles; in case of a journal article, go through its abstract or further with its introduction and conclusion or bibliography. To ensure the reliability, make certain that the publisher is reputable and reliable, the published source has been peer-reviewed, the author is a prominent scholar in her field, the website is sponsored by a reputable organization, the source is contemporary and up-to-date, and the number of citations is high [35] [36].

Literature study is similar to observation where documents are studied to collect knowledge and data about the past and present [34]. Literature study as data collection method is very relevant in this research. It would not only give a complete overview of the current practice in business modeling but also most of the expert opinions and researches are available today as scholarly documents. Moreover, it is easier to gain access to online journals and books through the university library.

3.2.2 Application of Method

The broad outline of the problem given by the supervisor was taken as the starting point. Research articles were searched with the different combination of following keywords:

- e3 value model,
- business model(ing),
- meta-model,
- value model(ing),
- ebusiness,
- integration of models,
- REA,
- BMO,
- UML,
- MDA,
- OMG,
Business process model.

‘Google Scholar’ (http://scholar.google.se/) and KTH online library were searched with the given keywords. On the university’s online library ‘MetaSearch’ page, 39 databases for online e-journal were given based on the category as ‘general’ and sub-category as ‘E-journal’. Out of these 39 databases, following databases were selected to search for relevant literature:

- ACM Digital Library,
- IEEE Xplore,
- Science Direct,
- SpringerLink – Journals,

These e-journal databases publish journals on both business and computer science categories. They are sponsored and managed by well-known organizations. Their database contains a very large number of articles. All the articles on these databases are peer-reviewed.

The search produced articles in the range of few hundred to thousands. To narrow down the search different combination of following strategies were used:

- keywords were searched as part of the title,
- publication date was limited to last ten years,
- ranking of the articles (if available) from the database was considered,
- the number of citations of the article,

Approximately 40 research articles were selected. The abstract of each of the article was read carefully to further filter around 20 most relevant articles. These selected articles were studied in detail. The reference list of these articles led to other similar articles which were again analyzed in the similar manner. This process continued until I could not find anything new or substantially different from what I had discovered on the chosen topic.

Websites of important organizations related to modeling and design were also searched for relevant information, standards and specifications-

- World Wide Web Consortium (W3C) [37],
- Object Management Group (OMG) [38],
- E3 value model research group [39],

Some of the related modeling tools were also considered for the important features, strengths and limitations to gain more insight into the problem. The tools considered were

- E3 modeling tool: a graphical tool to create e3 value business model as provided by the E3 value research group [39].
- IBM Rational Software Architect: an integrated modeling and development tool for model driven development by IBM [40].
- Adocus Metamodel Agent: an extension to modeling tool IBM Rational Software Architect to design metamodels for UML diagrams [41].
- Protégé: an open source ontology editor tool to model ontologies [42].

The major problems identified at a high level with business modeling were:
Rapid development in ICT and swift growth of E-Business has put focus on altering the conventional business models to better utilize the opportunities provided by ICT, and has provided a multitude of possible business configuration for value creation by forming inter-organization networks[1].

The design and implementation of business model is a very complicated task. Despite the growing importance of a sound business model for the success of the business organization, the concept of business model is still vague and ambiguous due to following reasons[2][3][4][1]:

- Lack of well-defined modeling principles to construct a BM.
- Lack of well-established relationship between the components of a business model makes extraction of knowledge from BM difficult.
- Difficult to integrate BM with other aspects of business design such as a business strategy, business process modeling, and architecture of IS to support the business.

E3 value modeling was then studied from the perspective of these high level problems. The high level problems were transformed as questions and the important articles on e3 value modeling concepts[5], [7], [11], [12], [14], [15], [43] were reviewed:

- Are all the relationships between the ‘e3 value model’ components defined clearly?

The relationships between the e3 model components have been described in a natural language. There did not exist any concrete meta-model defining the relationships between the components in a clear and precise way. Some of the relationships were ambiguous such as the relationship between the actor, composite actor and market. Composite actor was defined as a group of actors, but again these elementary actors could appear both independently and as a composite actor in the same business model. Market is defined as a large group of actors, but it does not specify if those actors could exist independently or if they could have value activities.

- Is e3 model integrated with the modeling and development of business information systems?

The e3 value model is isolated from the software design and development. The models are not compatible with UML- the de facto modeling language in software engineering. The models are not saved as XML, which is most popular format to exchange information between software systems and tools. Moreover, there is no information available on how to extract knowledge from the e3 value models and used it in the process of software design.

Lastly the usability[44] of e3 value modeling tool was tested from an end user perspective. It was found that:

- The tool lacked any sort of validation framework to validate the created e3 value models.
- Many of the components such as value objects, value transactions did not have a graphical representation and lacked attributes (such as name or description) to hold useful information about the business model.

The articles referred during the literature review have been taken only from reputed publishers and each of these articles has been peer reviewed. All the articles are contemporary in their field and they have considerable number of citations. The websites referred are sponsored and maintained by reputed organizations and are the direct owners of the website content. The content of the website are up-to-date. Therefore, the information available though these journal articles, websites and books is reliable. The process of selecting the articles was based on the relevant list of keywords that were used both individually and in multiple combinations to get a large number of articles. The articles were then selected based on relevant title,
abstract and bibliography. The selected articles were then qualitatively analyzed. Moreover two other sources of information: important websites and tools were also used to gather information. The problem was explicated gradually from high level problems to problems specific to e3 value model. Therefore, the method of problem explication is both reliable and valid.

During the literature study ethical practices related to plagiarism and giving proper credit to authors were followed. All the information collected from different journals and books have been clearly cited. Only the journals and articles to which I had a valid access have been used in the research work.

3.3 Define Requirements:

This activity involved a further investigation of the explicated problem but from the perspective of proposing a solution. The outcome of this activity was the outline of the proposed solution artifact and the important requirements on this artifact. These requirements define the functions of the artifact, its construction method and constraints and also the artifact’s relationship with the environment and any non-functional requirements on the artifact.

3.3.1 Choice of Method

The research strategies and methods for eliciting requirements are many. Surveys[45] can be used for getting requirements directly from stakeholders but has the limitation that stakeholders may be biased, miss important requirements and do not analyze enough. Case studies [46] can help in deeper investigation of stakeholder’s requirements but require a lot of time, resources and competence from the researcher. It is difficult to generalize requirements from a single case study. Theoretical analysis [30] is used when the artifact is novel, or it is not possible to do empirical studies. In this research, theoretical analysis involves a careful study of the explicated problem to bring out requirements. Though this method lacks empirical support, it is suitable in the given time constraints and resources. The research method employed here is document studies [30] in which literature was reviewed with the objective of eliciting requirements.

3.3.2 Application of Method

The problem statement was used to derive high level requirements. These high level requirements expressed in natural language were very broad and generic. The high level requirements were:

- Establish well-defined relationship between the various components of e3 value model.
- E3 value models should be compatible with UML modeling standards.
- E3 value models must be serialized as XML based on standards for model exchange.

Based on the high level requirements, it was evident that a UML compatible meta-model for e3 value modeling must be constructed. There are three major alternatives available to create such a metamodel. They were [47][48]:

- Create the meta-model from scratch using MOF, the meta-modeling language defined by OMG.
- Create the meta-model from scratch using UML profiles[49], a general extension approach to customize UML models for specific purposes.
- Extend the UML meta-model to accommodate the requirements for the new meta-model.
These available alternatives to construct the metamodel were analyzed qualitatively by comparing their advantages and disadvantages.

According to OMG specification for UML[19], UML profiles do not extend the UML metamodel and Thus, different UML modeling tools may interpret the extension through UML profiles differently[47]. Moreover, it requires the use of complex OCL[50] extensively to implement very simple constraints, which leads to difficulties in applying validations[47]. The advantages of using UML profile are that the UML metamodel is reusable here and there are a lot of available tools that support UML profiles[47][48]. According to [48], the definition of UML profile as provided by OMG is difficult to comprehend and Thus, difficult to implement.

Extending the UML metamodel is again a very difficult task and there are no proper guidelines available. Moreover, the extended metamodel becomes incompatible with the existing UML tools. Although basic constraints and validation can be achieved easily, this approach would require the creation of a new tool to support the implementation[47].

OMG has defined MOF as a platform independent and open source metamodeling language. It reuses the UML Infrastructure Library [19] and a subset of concepts from UML class diagram modeling. Thus, any UML modeling tool can be used to create metamodel in MOF[48]. Since the metamodel is created from scratch, there are several advantages[47][48] – a) customized names and icons could be created for model elements, b) relationships can be specified in a simple and precise manner, c) high flexibility and adaptability to create metamodels as required, d) does not require to include all the unnecessary concepts and complex features of UML, e) easy to apply constraints and validation. However, building a metamodel from scratch is a complex task and time-consuming task and requires greater skills than creating UML profiles. The number of available support to build tools on top of MOF metamodels is also few [47][48].

A comparison table for summarizing the merits and demerits of all three approaches given in [47] is:

<table>
<thead>
<tr>
<th></th>
<th>UML profiles</th>
<th>UML Extension</th>
<th>MOF based metamodel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expressive power</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Flexibility in design</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Clarity of semantics</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Application of constraints</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Customized Model Notations</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Tool Support</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Ease of creation</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Table 3-2: Comparison summary of UML profiles, UML extension, MOF based metamodel

Given the comparison, using MOF to create the metamodel was the chosen alternative.

Based on the high level requirement and the chosen alternative for construction, the requirements were refined to a very detailed and specific level. These detailed requirements were also expressed in a natural language. The detailed low level requirements on the metamodel are:

- Meta-model should be based on MOF.
- Meta-model should cover all the existing e3 value concepts.
- Meta-model should represent the relationship between the e3 model concepts.
- Meta-model should represent the constraints between the e3 model concepts.
- Meta-model should be expressed in human readable format (a class diagram).
The requirements on the graphical tool were created based on the requirements of the metamodel and the common features that a tool must have. These requirements are:

- Implement the e3 value metamodel as it is.
- Implement the implicit and explicit constraints on the e3 value model concepts.
- Functionality to save the model in a serialized format following xmi standards.
- Validations of the model to help the users design a correct model.
- Error prevention mechanism to prevent common errors while creating models.
- Tree view navigation of the model to navigate large and complex models.
- Functionality to print the model.
- Easy-to-use graphical interface.

Finally, the set of low level requirements were analyzed qualitatively using a checklist from [44]:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Necessity</td>
<td>Is the requirement necessary?</td>
</tr>
<tr>
<td>Consistency</td>
<td>Is the requirement consistent with other requirements?</td>
</tr>
<tr>
<td>Completeness</td>
<td>Is the set of requirement complete?</td>
</tr>
<tr>
<td>Feasibility</td>
<td>Is the requirement feasible with the given time, resources and constraints?</td>
</tr>
<tr>
<td>Unambiguous</td>
<td>Does the requirement interpret in different ways to different readers?</td>
</tr>
<tr>
<td>Redundancy</td>
<td>Does the requirement overlap with other requirement?</td>
</tr>
</tbody>
</table>

Table 3-3: Requirement analysis criteria

There are four major types of IT artifacts produced in a design science research project – constructs, models, methods and instantiation. Constructs define the vocabulary and symbols, models use constructs to represent reality, methods define standards or processes or algorithms to solve a problem while instantiation is a real world implementation of constructs, models using the methods [28]. The main artifact to be created in this project is:

- A meta-model for e3 value model based on MOF.
- A graphical editor tool to create e3 value models based on the metamodel.

The requirement gathering process closely followed the methods given in [44]. Literature review was used for requirement elicitation. The requirements were gathered initially at high level, and then gradually low level requirements, which were clear and precise, were constructed from high level requirements. The requirements were then analyzed using the checklist (Table 3.3) from [44]. Further the requirements were validated through peer reviews and it was ensured that requirements are testable when the final artifacts are delivered. Therefore the method of requirement gathering is reliable and valid.

### 3.4 Develop Artifact:

#### 3.4.1 Choice of Method

The aim of this phase is to design and develop the artifact based on the requirements gathered in the previous phase. The design of an artifact is a creative process. The primary input to the design process would be the requirements and knowledge base created from the literature study. This step is a creative process and does not have any specific research strategy or method [30]. The outcomes of this phase were:
A well-defined metamodel for e3 value modeling based on MOF.
A simple graphical modeling tool to create e3 value models on top of the metamodel.

The design process began by exploring the solution space to search through possible design alternatives and eventually narrow down to a solution which was feasible within the given resources and timeframe. The technique used here was to select some of the most popular UML modeling tools and create simple prototype e3 meta-model to get a quick understanding of the tool, its features and limitations. The three modeling tools tried were:

- IBM Rational Software Architect [40],
- Eclipse Modeling Framework in eclipse integrated development environment[51],
- Microsoft Visualization and Modeling Software Development Kit (VMSDK) in Microsoft Visual Studio integrated development environment [52].

IBM RSA is a comprehensive and robust modeling framework to create UML models. It provides tools and facilities to support all the phases in a model driven development, from creating UML models to generating compiled code. It also supports UML profiles to create domain-specific models. For the purpose of thesis, the tool was used to create a basic version of the e3 metamodel in UML. The tool was very easy to use and also had the support to apply modeling constraints using OCL. However, the tool does not support the use of custom-made metamodels to build a modeling editor on top of it for the end user. The only way to customize UML models is through the use of UML profiles. Thus, the IBM RSA could not be used further in this research.

Eclipse modeling framework provides tools to generate java code from models described in XMI, facility to view or edit models, and a basic editor. The EMF framework includes a meta-model called Ecore which is used to design and maintain the models. Ecore metamodel is very similar to EMOF metamodel[25]. EMF also provides a plug-in tool to create UML class diagrams which can then be transformed to an Ecore based model and then a basic modeling editor can be generated from it. This tool had all the necessary components to create a metamodel based on MOF and then create a basic modeling editor on top of it to create MOF based e3 value business models. A simple prototype was created to ensure that it was possible.

Microsoft’s VMSDK is integrated with Microsoft Visual Studio IDE. Microsoft.Net is a very popular development framework, and it provides support for model driven development in a similar manner as the EMF. It can be used to implement metamodels for the domain specific language, create model-driven tools to create domain models and generate code based on them. VMSDK provides an editor to define the metamodel and create a customized graphical tool based on the metamodel. The graphical tool and code generation can be customized and extended as required[26]. Based on the flexibility of the tool and my previous programming experience in Microsoft.net and c#, this tool was selected to implement the final meta-model and its graphical tool.

Once the tool was finalized, the development was carried out in following broad steps:

1. Design the meta-model for e3 value model based on MOF. The tool used was Microsoft Visio.
2. Implement the metamodel in eclipse modeling framework and create a basic modeling editor.
3. Implement the metamodel in Microsoft VMSDK and create a graphical modeling editor.

These steps were broadly followed in the given order. Each of these steps had several sub-steps. There were several iterations among the sub-steps, going back and forth to fix the bugs, issues with the metamodel, customizing the tool and so on.
### 3.4.2 Design the meta-model

The meta-model was first created in Microsoft Visio. Only the most basic elements of UML class diagrams such as – class, binary association, aggregation, attributes and primitive data types (string, integer) were used to ensure that the meta-model is MOF based. All the major components of e3 value model such as – actor, market segment, value interface, value offering, value ports, value exchanges, value objects and value activities were included. The e3 model class diagram as provided in [11] was taken as an input. The given e3 model class diagram was a very casual representation of the relations between the various components but still helpful in understanding the key relations between the e3 elements. The first version of the e3 metamodel created is shown in the following image:
Figure 3-1: e3 value metamodel
Some of the major design decisions taken during the course of development of the metamodel were:

Elementary actor, composite actor and market segment were made sub-class of the super-class Actor. According to e3 model specification [11], composite actor is a constellation of elementary actor who have decided to collaborate and represent themselves as a single (composite) actor to the external environment. This relationship has been expressed through many-to-many[53] association between elementary actor and composite actor. The association is not composite, but a simple association to provide the flexibility that an elementary actor can exist both in the collaboration as a part of composite actor and also interact with the environment as an independent entity. The market segment as defined in [11] is a set of actors that value a certain object in a very similar manner. Since they value object similarly, the actors within the market segment are not modeled individually for the sake of simplicity. Moreover, in the business environment these groups of actors are always treated as a homogeneous set with attributes defined at the group level rather than individual level. Thus, it seemed reasonable to treat ‘market segment’ as a specialization of actor and model it accordingly.

Value activity has been modeled in a composite aggregation relationship [53] with the elementary actor. According to [11], a value activity is carried out only by an elementary actor, and the actor can have many value activities within it. Furthermore, a value activity cannot exist independently and ceases to exist without the owner actor. Thus, a composite aggregation is more suitable than a simple association. The association is navigable only from the elementary actor as the actor owns its value activities.

According to [11], an actor exchanges value objects only through value interfaces. Value interfaces hide the internal working of an actor form the external environment. A value interface can exist either as a part of the actor or a value activity, but it does not exist independently or is shared between the actor and its value activities. So it has been modeled in a separate, composite aggregation relationship with both the actor and value activity. The association is navigable only from actor to value interface and from value activity to value interface as the actor and value activity are the owners of their respective value interfaces.

According to [11], a value interface can consist of at least one value offering (either incoming or outgoing) or a maximum of two value offering – one incoming and one outgoing. To apply these constraints, ‘Value Offering’ has been modeled as a super-class inherited by many sub-classes relationship. Value Offering is the super-class inherited by ‘Incoming Offering’ and ‘Outgoing Offering’ sub-classes. ‘Incoming Offering’ denotes the incoming value offering while ‘Outgoing Offering’ denotes the outgoing offering through the value interface of an actor. Both these sub-classes have a composite aggregation relationship with the ‘Value Interface’ class. Multiplicity constraint of ‘0...1’ has been applied to both incoming and outgoing offering subclasses. Additionally, value offerings cannot exist independently without the value interface; Therefore, the composite aggregation has been used. Since value interface own the value offerings, the association is navigable only from value interface to value offerings.

According to[54], incoming and outgoing value offerings consists of one or more in-ports and out-ports respectively. Value ports do not exist outside the context of value offerings. Therefore, value ports have been designed in super-class and sub-class inheritance relationship. ‘Value Port’ is the super-class and ‘In Port’ and ‘Out Port’ are the two sub-classes that inherit from ‘Value Port’. ‘Incoming Offering’ and ‘Outgoing Offering’ have composite aggregation relationship with ‘In Port’ and ‘Out Port’ respectively.

According to [54], value exchange, which connects the value ports and represent the interactions occurring in the business environment, has been classified into four types – exchange connecting opposite ports between actors, opposite ports between value activities, same direction port between composite and elementary actor, same direction port between elementary actor and value activity. In the meta-model, value exchange has been designed as a super-class and sub-class inheritance relationship. The super-class ‘Value Exchange’ is inherited by three sub-classes ‘In In Exchange’, ‘Out Out Exchange’, and ‘In Out Exchange’. ‘In In Exchange’ and ‘Out Out Exchange’ have association relationship with ‘In Port’ and ‘Out Port’ class respectively with a
constraint of 2, which means that they have to be mandatorily associated with exactly 2 ports. Similarly, ‘In Out Exchange’ is associated with exactly one ‘In Port’ and one ‘Out Port’. On the other hand, ports can be associated with zero or more exchange simultaneously. These three sub-classes of value exchange cover all the possible exchange scenarios that can exist in an e3 value model diagram.

<table>
<thead>
<tr>
<th>Conventional E3 value Exchange Types</th>
<th>Meta-model Exchange Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>opposite ports between actors</td>
<td>In Out Exchange</td>
</tr>
<tr>
<td>opposite ports between value activities</td>
<td>In Out Exchange</td>
</tr>
<tr>
<td>same direction port between composite and elementary actor</td>
<td>‘In In Exchange’, ‘Out Out Exchange’</td>
</tr>
<tr>
<td>same direction port between elementary actor and value activity</td>
<td>‘In In Exchange’, ‘Out Out Exchange’</td>
</tr>
</tbody>
</table>

Value Objects are the objects which are of economic value for actors and different actors value the same object differently. This difference in valuation results in the exchange of value objects between different actors [54]. In [54], value objects have not been further classified, but in the metamodel the value object has been specialized into four sub-classes – product, service, money and information. This classification of value object has been taken from [12]. ‘Product’ represents physical/tangible objects; ‘Service’ represents intangible objects (e.g. consultation services provided by a software consultant), ‘Money’ represents the currency used to exchange other value objects and ‘Information’ is the valuable information. This classification is an addition to the original e3 value model inspired from the classification of value objects offered in [12]. It fits perfectly well in the metamodel and does not influence any other relationships. Thus, it adds to the knowledge contained within the e3 value models. One design alternative here could have been to replace the classification with an attribute ‘object type’ having a reference to an enumeration list consisting of ‘product’, ‘service’, ‘money’ and ‘information’, but these four types of value objects have very different set of attributes. E.g. a product may have attributes like dimension, weight, packaging cost, transportation costs etc.; a service may have attributes like hourly cost, skill set, start date, end date etc.; money may have attributes like exchange rate, currency denomination, decimal separator, thousand separator etc.

Value transaction consists of value exchanges that must take place as a group to complete a business transaction. Further, a value transaction must not violate the rule that all the ports or none within an interface must exchange values. Therefore, it is more convenient to include value interfaces instead of value exchanges in a value transaction; including a value interface will implicitly ensure the rule is never violated, but e3 value model being flexible allows a value port to participate in more than one value exchange (to allow for either-or choice between two or more actors), although only one value exchange can take place at a time. To include value interface in value transaction would then mean that a value port can be a part of only one exchange. In the end, flexibility was chosen over strict validation. This would mean that the validation could not be included in the model but must be implemented dynamically in the modeling tool. Value transaction was implemented as a class that has association relationship with one or more value exchanges.

3.4.3 Implementation in EMF

The subsequent phase in the development process was the transformation of the e3 value meta-model from a UML class diagram to the Eclipse Modeling Framework’s (EMF) meta-meta model Ecore based meta-model. At first, the e3 meta-model designed in Microsoft Visio was implemented in EMF as a UML class diagram. The UML class diagram implemented in EMF was then converted to Ecore based metamodel using the conversion facility provided in the EMF’s UML editor. The Ecore based metamodel could have been created independently without the use of conversion, but it was done to ensure that Ecore based metamodel is a replica of the UML based e3 value metamodel created in the first step. A screenshot of the UML implementation of e3 value meta-model in EMF is as follows:
The Ecore based e3 value meta-model that was obtained after the conversion is as follows:
Figure 3-3: e3 value metamodel in.ecore (first version).
The Ecore based metamodel is little different from the original e3 value metamodel in UML class diagram. An additional class ‘E3BusinessModel’ has been added which has composite aggregation relationships with the class ‘Actor’, ‘ValueExchange’ and ‘ValueObject’. This class was created to bring all the elements of e3 value model under one class and have ‘E3BusinessModel’ as the starting point for any e3 value model creation. ‘E3BusinessModel’ is composed of at least one or more actor, value object and value exchange. This constraint ensures that any e3 value model created has at least one actor, one exchange and one value object which are the minimum requirement for a valid e3 value model.

The ecore model is now used to create the EMF generator model. The ecore model and the generator model will now drive the generation of the entire application. The EMF generator provides a tree view of the generator model. In our case, the model is call E3 generator model. The root level element ‘E3’ represents the whole model. We now use the EMF.Edit framework code generator to create the eclipse plug in. The EMF.Edit generates plug-in in two parts. One part is the UI-independent part which contains support classes for editing that is UI independent while the other part of the plug created needs Eclipse UI to function. The UI independent plug-in generated here is e3value.edit while the other one is e3value.editor.

The e3value.editor is a fully functional eclipse plug-in that can now be used to create, view and edit model instances. It has an icon folder which can contain customized icons for the model. ‘plugin.properties’ is the property file and ‘plugin.xml’ is the manifest file that contains information about the plug-in dependencies and extensions. A screenshot of the EMF generator model is shown below:

![Figure 3-4: EMF generator model.](image)

The plug-in e3value.editor is tested to get an understanding of the features and the model editor. The plugin is launched in a separate eclipse environment for testing.

A sample business model as created in the EMF editor is displayed in the following diagram. It is a simple e3 value model consisting of four actors- mobile phone company (MaxMobile), mobile phone subscription provider (Airtel), reseller (MobileStore) and consumer. MobileStore buys the mobile phone and subscription from MaxMobile and Airtel respectively. The reseller creates a combined package and sells it to consumers.
The model is created in a tree structure with child elements being shown under the parent element. The editor was helpful in following the rules specified in the metamodel. E.g. a value interface can have only one incoming and one outgoing offering; a value port can have only value object; value exchange connected to proper ports depending on the type of exchange. The created model was easily exported as an xml file.

The test involved creating both simple and complex business models to test the editor and the metamodel. It was helpful in bringing about some important design changes to make the metamodel more flexible and robust.

The ‘value-offering’ class and its sub-class – incoming and outgoing offering were removed from the metamodel. It was realized in testing that value-offering did not add any useful information to the model; rather it increased the level of nesting unnecessarily. Incoming offering and outgoing offering are the names given to collection of in-ports and out-ports respectively in a value interface and do not have any more significance. Therefore, it was removed from the meta-model. It reduced a level of nesting without losing any information and made the metamodel simpler. It might be for the similar reason that no shape exists for ‘value offering’ in the original e3 value model tool [55].
Figure 3-6: e3 value metamodel without value offering in ecore.
Another important design change was adding two composite aggregation relationships between the value exchange and elementary actor and composite actor respectively. This was changed because both the actors can have value activities, whose ports will participate in value exchange with other value activities within the actor, or with the value interface within the actor that participate in value exchange with external actors. These value exchanges will be inside the actor and will cease to exist if the actor is deleted.
Figure 3-7: e3 value model in.ecore (final version)
Due to the absence of a full-fledged graphical tool, creation of complex models would be cumbersome and time-consuming in the eclipse tool, but it was useful to design a better meta-model as it provided a quick feedback on the scalability and the ability of the metamodel to handle complex business scenarios. Moreover, ecore meta-metamodel is very similar to MOF meta-metamodel; Therefore, the metamodel created using ecore will follow MOF principles and the researcher does not have worry about breaking any MOF principles. Another use of this tool would be to test the exchange of serialized models between two different tools having the same MOF based e3 value metamodel.

3.4.4 Implementation in Microsoft VMSDK

The next step was the implementation of the same e3 value metamodel in Microsoft.Net Framework using Microsoft Visual Studio 2010 [56]. Implementation of the proposed MOF based metamodel for e3 value in Microsoft.net framework would give us many benefits. It will prove that the MOF based e3 value metamodel is easy to use, implement and independent of technology platforms. It will provide us a way to compare the implementation on the two most popular development environments.

The meta-meta model in VMSDK is not an exact implementation of MOF[26]. It is proprietary meta-metamodel of Microsoft and no description or design could be found. During the implementation of the metamodel, it was evident that the meta-metamodel incorporates all the features of MOF and has some additional extensions. One such extension is that the meta-meta model in VMSDK allows us to define attributes in relationships, which is a deviation from MOF. However, the e3 value meta-model was implemented in VMSDK without violating any MOF principles.

The screenshot from the MS Visual Studio IDE displays the implementation of e3 metamodel.
Figure 3-8: e3 value metamodel implementation in VMSDK.

The metamodel in VMSDK is arranged in many small parts in a way to ease designing. From the screenshot, it may seem that some classes are created twice or more, but in reality all the classes have been created only once, and the metamodel is exactly similar to the one created in eclipse EMF. Similar to implementation in EMF, the metamodel here also has a root class E3BusinessModel which has composite aggregation relationship with the Actor, ValueObject and ValueExchange class.

Each class must be associated with a shape or image. The classes on the right side in the screenshot are the shape classes that define the attributes of the shapes/images associated with the domain class. Actors (Elementary Actor, Composite Actor and Market), Value Interface and Value Activity are assigned rectangle shape with different colors. Value Exchange (InInExchange, InOutExchange, OutOutExchange), Value Objects (Product, Service, Money and Information) are assigned images and Value Ports (InPort, OutPort) are assigned circular port shapes. Ports are different form the shape in a way that they are always placed on the edge of the parent shape with a portion of the port protruding out of the parent shape. Shape classes are also
defined for the relationship connectors. Initially five connector shapes were created, one each for connectors between – value ports and value objects; elementary actor and composite actor; in-in-exchange between two imports; out-out-exchange between two outports; in-out-exchange between one import and one outport. Afterwards, the value exchange connectors were merged together to form just one connector. This change was prompted during the testing of the graphical tool where it was realized that having too many connectors could confuse the user.

The next sub-step was to create the toolbox and associate toolbox items with an icon and a domain class or connector. Icons were downloaded from the internet, and a few of them were created in Microsoft paint [57]. Icons were carefully chosen to resemble the domain class and their unique attributes. The icons downloaded from the internet were freeware or freely available for non-commercial use. The list of icons, their creator, and the website from which it has been downloaded is available in the appendix.

C# code was written to add validations to the model. The validation was designed to execute whenever the model was being saved. User friendly error messages were also written for validation errors. One difficult problem encountered was the nesting of child shapes within the parent shapes. E3 model has a maximum of four levels of nesting (elementary actor – value activity – value interface – value ports). It was difficult graphically to contain the child shape within the boundaries of the parent shape; programmatically increase the parent size when a child shape was dropped within it.

Finally, the code for xml serialization of the e3value models was written. Microsoft has its proprietary format and default settings for model serialization. At first attempts were made to serialize the model following XMI standards but it was very challenging to complete within the limited time. However, customization of the serialization process was done to bring the serialized model close to XMI format.

Developing the graphical tool from an end-users perspective took a lot of time and there were several iterations to improve the look and feel of the tool itself and the models created form the tool.

During the development phase, only those software programs for which I had valid licenses were used. These software programs were either available as open source or downloadable from the university website. Trial versions of certain software such as IBM Rational Software architect were also used. The tutorials used to learn the software programs have been clearly cited. All the code written during the development of the tool has been created during the development phase.

In terms of reliability and validity of the development phase, all the software used during development are reliable, supported and maintained by reputable organizations and are widely used commercially. The official development guide, process and tutorials[27][40], [51], [52] related with the software have been closely followed during development to ensure that the development process is valid. However, the e3 value metamodel and its tool have not been developed for commercial purpose. Thus the reliability and validity of the metamodel and its tool is limited to academic domain only and they can be seen as a proof for concept for commercial development.

3.5 Demonstrate Artifact:

3.5.1 Choice of Method:

The demonstration of the MOF based e3 metamodel and its graphical tool requires a way in which all the features of the metamodel and the graphical tool can be displayed in a clear and consistent manner. The suitable and most common choices available for such a demonstration are experiment[58], surveys[58] and
case study[58]. Each method has its advantages and disadvantages and the scenarios where they are most useful. The important parameters that influence the choice of the method are the objective of the demonstration, the amount of control to exercise over the environment and the closeness to real life scenario.

Surveys are appropriate when the objective is to discover frequency or general phenomenon related to an event, or to collect statistical data for questions of the form similar to ‘how many’ or ‘how much’[58]. Experiments are useful for in-depth analysis of an exploratory nature but in a precisely controlled environment [58]. Case studies are relevant when a detailed analysis of a real life event occurring in its natural form without any artificial control of the environment is required. It is one of the preferred methods when the focus is to answer questions of the nature ‘How’ and ‘why’ for events occurring in real life [58].

The objective here is to replicate the business model of a typical business entity with all its complexities and details to draw attention to all the essential features of the e3 value metamodel and its tool. The case study method would be the most advantageous given that the intention here is to understand the business of an organization in a natural business environment. Another benefit of using a case study is that it allows the use of multiple research methods such as observation, interviews, document review and questionnaires to collect data[58].

The demonstration is made of two parts. At first, a simple e3 value model diagram is created using the new e3 value model tool to demonstrate the key features of the tool. Then, the same e3 model is recreated but using the eclipse editor. The objective here is to demonstrate the eclipse editor and compare the serialized xml file generated from both the tools. The second part consists of a detailed case study of an existing business organization in chapter 4 and the implementation of the organization’s business model in chapter 5, section 2.

### 3.5.2 E3 value model tool:

A simple e3 value model created from the tool using all the elements along with the navigation tree is displayed in the figure 3-10. The toolbox is placed on the left, the design canvas is in the middle and on the right we have the tree view navigation of the model.

The actors are rectangle boxes of light yellow, light orange and light green color depicting elementary actor, composite actor and market respectively. A small icon of the corresponding actor is also positioned on the top right corner for distinction. Value activity is a khaki color rectangular box with round edges. Value interface is a rectangular grey colored bar with round edges. In-ports and out-ports are green and red circles respectively that fit on the edge of the value interfaces. A detailed description of each toolbox item is present in the result section 5.2.
Figure 3-9: e3 value model of Airtel Teleservices.

The tool serializes the e3 value model as an xml. This is another important feature of the tool. It generates two xml files – one file containing the details of the model and another file containing the visual diagram information. The serialized xml is based on the propriety xml format as provided by the Microsoft, but it is very similar to XMI standards proposed by the OMG. Since the underlying metamodel is MOF based, the xml further maintains the object oriented structure of the model. The serialized xml file of the above model is as follows:
Figure 3-10: Serialized e3 value model of Airtel Teleservices.

The serialized xml model is simple and closely follows the metamodel structure. ‘e3BusinessModel’ is the root node that includes all the actors, value exchanges and value objects. The business model has a unique id. Value exchanges between the actors are saved as child nodes of the root node. They contain a reference to the value ports involved in the exchange. Value objects do not belong to any particular actors, and they are also saved as child nodes of the root node. Elementary actor ‘Airtel Teleservices’ has value activity ‘packaging’ as one of its child nodes; it also embeds the value exchanges and value interfaces within its boundary. Value interfaces have value ports as child nodes. There is a reference from each port to its corresponding value objects. The references follow the hierarchical structure from the root node to the child node and are human readable. E.g. the ‘inOutExchange’ marked in red has a child ‘outPorts’ that references the outport named ‘Subscription Fee Outport’ in the value interface ‘ValueInterface1’ of actor ‘Mobile Users’ in the business model with id ‘418a68f1-5f97-4d1b-801e-b87ef61ae8c3’.

The constraints of e3 value modeling have been implemented in the tool. These constraints make it difficult for users to violate the principles of e3 value modeling while creating models.

Constraints such as nesting of elements and connection between the elements have been implemented within the toolbox items. These constraints are:

46
— Actors and value objects can only be created on the canvas and cannot be embedded in other elements.
— Value activity can only exist as embedded element within elementary actor.
— Value interface can only exist as embedded element within actors.
— Value ports can only exist as embedded element within value interface.
— Value Port-Value Object connector connects ports with only one value object.
— Value exchange connector can only connect two ports
— Each element has a default unique name assigned on creation.

Other validations, which are difficult to conduct at runtime, are done when the user saves the model. Some of these validations are:
— port must be connected to one value object.
— value exchanges must connect two ports depending on the type of value exchange.
— all the elements should have a name.
— value exchange cannot connect ports of the same value interface.
— business model must have at least one actor, one value object and one exchange.

The validation happens by default when the model is saved. It can also be initiated from the context menu in the model tree on the right. The model can also be created using the tree view, child elements can be created from the context menu of the parent elements. In the screenshot below, the context menu on the root element of the tree contains the option to validate the model and also to add a new child element such as – composite actor, elementary actor and so on. The context menu on the actor and value interface is also shown in figure 3-12.

Figure 3-11: e3 value modeling tool with validate option.
The screenshot of the e3 model shows the list of errors when the user tries to save an invalid model.

Figure 3-12: context menu on value interface.

Figure 3-13: validation error in e3 modeling tool.
Another important characteristic of the tool is that it allows the user to print, see a print preview, zoom in and out of the e3 value model diagram.

![Image of print option in e3 value modeling tool]

**Figure 3-14: Print option in e3 value modeling tool.**

### 3.5.3 E3 value model editor in Eclipse

Eclipse editor is used to create the ‘Airtel Teleservices’ business model. The editor generates the model in a tree view structure which is not a very user friendly way but it has all the essential features.
Figure 3-15: Eclipse based e3 value modeling editor.

The ‘Business Model’ is the root node which has a context menu to create child nodes. The child elements also have contextual menu to create child and sibling elements from it.
Figure 3-16: Context menu on elements in eclipse e3 value editor.

The editor also has some validations for the e3 value model.
Figure 3-17: Validations in eclipse e3 value editor.

The serialized model in xml generated from the eclipse editor follows the XMI standards and is as follows:
Figure 3-18: Serialized e3 value model in eclipse editor.

The serialized model in eclipse has the same structure as the serialized model in Microsoft VMSDK, but references to other domain objects are saved in a different way. E.g. the reference to import in the value exchange ‘Money Exchange’ in VMSDK and Eclipse is as follows:

```xml
<e3:ValueInterfaces id="2" name="ValueInterface1">
    <Ports id="3" name="Money Port" ValueObject="/e3:ValueObject4"/>
    <Ports id="4" name="Mobile Phone Port" ValueObject="/e3:ValueObject2"/>
    <Ports id="5" name="Mobile Connection Port" ValueObject="/e3:ValueObject0"/>
</e3:ValueInterfaces>

<e3:ValueActivities id="6" name="Packaging">
    <ValueInterface id="7" name="ValueInterface1">
        <Ports id="8" name="InPort1" ValueObject="/e3:ValueObject4"/>
        <Ports id="9" name="OutPort1" ValueObject="/e3:ValueObject2"/>
        <Ports id="10" name="OutPort2" ValueObject="/e3:ValueObject0"/>
    </ValueInterface>
</e3:ValueActivities>
```

```
<InOutExchange name="Money Exchange">
    <InPorts name="/418a68f1-5f97-4d1b-b01e-b67ef61e83c7/ValueObject1/Money Port"/>
    <OutPorts name="/418a68f1-5f97-4d1b-b01e-b67ef61e83c7/Mobile Users/ValueObject1/Subscription Fee Port"/>
</InOutExchange>
```

```xml
<e3:ValueExchange id="11" name="OntOutExchange9">
    <InPorts @actor="actors.0."/>
    <OutPorts @actor="actors.0."/>
</e3:ValueExchange>
```

```xml
<actors xsi:type="e3:CompositeActor" id="32" name="Mobile Expert Websites">
    <ValueExchanges xsi:type="e3:InOutExchange" id="25" name="Money Exchange">
        <InPorts @actor="actors.0."/>
        <OutPorts @actor="actors.0."/>
    </ValueExchange>
```

54
In VMSDK references are saved as child elements; in eclipse the references are saved as attributes of the elements. Both the references contain the full path to the referred element. In VMSDK, the reference contains the unique identity of the business model followed by the unique name attribute of the parent nodes; in Eclipse the reference contains some internally generated identities of the parent nodes.

Another difference between the two serializations is that all the elements in VMSDK are defined using the child class name, but in Eclipse it is defined using the parent class name and with a ‘type’ attribute defining the child class.

These differences are minor, but it makes exchange of models between the two tools infeasible. There can be ways to manipulate the serialized xml in a way to make it compatible. A console application can be created to take a serialized model from one tool as input and generate serialized model for another tool as output, but creating such an application requires an in-depth analysis of the XMI standards, its implementation in Eclipse EMF and analysis of proprietary serialization format from Microsoft (which is currently unavailable). Therefore, it is left as a future work.
4 Case Study: EU Supply Holding AB

The first step in a case study is to design the case study – a framework that would guide the entire case study and ensure that the focus is on the core objective, and the end results achieve that objective[59]. There are five important steps in the case study design[59]:

Formulate the case study question: The objective here is to explore all the major business exchanges that a business entity is involved in with its partners and customers and also its important value activities.

Create the study proposition: Case study proposition is the theoretical framework on which the research is based. It helps to focus in the direction where important information can be found. The study proposition here is that organizations collaborate in a business venture or engage in business when it is profitable to all the involved entities; the organization engages its most important resources in activities that generate objects of value and increases the profitability of the company; the customers pay in terms of money for the services/products that are of some business value, and the value derived from them is more than the value of money paid.

Decide upon the unit of analysis: Defining the unit of analysis means defining the ‘case’ and its boundaries. The major case here is the organization whose business would be analyzed. There will be two embedded units of analysis within the organization – a) inter organization business relation and b) intra organization value activities.

Choose the appropriate data collection methods: Out of the many methods of data collection, the ones which are suitable in accordance with the initial proposition and which will offer a strong foundation for analysis will be chosen[59]. The main principles in data collection are: to collect data from multiple sources, to maintain all the collected data in raw format in a database, and to maintain a chain of evidence[60]. There are six major sources of data[60] – documents, historical records, interviews, direct observation, physical artifacts and observation as a participant. I would use documents such as public documents, administrative documents, personal notes and minutes of meeting. Documents provide very specific and stable details[60]. The second method will be interviews. Interviews will be focused open ended interview with the key personals in the organization. Interviews are insightful and provide information in the current context of research[60]. They may be affected by poor recall or biased opinions of the interviewees. Direct observation will be the third form of data collection. The form of observation would be casual so as not to interrupt the normal activities. Observations are advantageous as they help to cover the events in real time and in the real context [60]. The advantage of collecting data from multiple sources would help in data triangulation – validate the same information coming from multiple sources and thus, improve the validity of the case study[60]. During the data collection phase, all the data collected in raw format would be maintained on the university computer, and a backup on the cloud data storage website for future reference.

Choose adequate analysis method to analyze the data: The general strategy to analyze data would be to rely on theoretical propositions decided earlier[61]. The propositions would form the basis to analyze the data and
discover how the business is actually carried out and what are important business transactions and value activities and the important actors. Since the data generated would be qualitative in nature, the mode of analysis will be careful reading of all the collected data and concluding the important value activities and business transactions.

After the case study design, the next step was deciding upon the type of case study. There are four different types of case study[62] - single case study or multiple case studies each with either holistic or embedded design. I have chosen a single case study design with embedded units since there are two units of analysis – inter-organizational business relation and intra-organization value activities. The rationale for choosing a single case study is that the organization will be a typical representation of a small scale software company providing unique services to its clients through its various branches and partners. The business environment is similar to other small scale software companies operating in Scandinavian countries. Therefore a detailed study of one such business organization can be generalized to other similar business organizations.

The next step was to select a single ‘case’ from the available options[59]. There are many software companies in Stockholm, but the selection of the case was done based on the requirements of the case study. The requirements were that the software company must be a small scale company (not more than 50 employees) with medium to complex business model having business in multiple countries and multiple partners. Moreover, I should have easy access to the documents and office site to collect information. Based on these criteria, ‘EU-Supply Holding AB, Stockholm’ was selected.

The next step was to define the case study protocol – a general set of rules and procedures to follow during the course of case study[59]. Procedures were decided on how to schedule and conduct interviews; prepare interview questions; defining the time line of data collection activity; observation rules; a set of checklists were also prepared to keep the case study on track and to focus on the core issues. The timeline for the data collection activity was set to two weeks and another 1 week for follow-up and review.

Ethical considerations related to case study were carefully followed during the period of case study. A permission to conduct the case study was taken from the manager of the organization. The objective and procedure details were clearly explained to the manager. No sensitive information related to the organization was revealed in the thesis. The result of the case study was discussed with the manager and then used in the thesis.

4.1 Observation

Observation is a way in which the researcher collects data by being a direct eye-witness to all the relevant events happening in the natural setting without relying on second-hand information from other people [31]. There are two kinds of observation:

a) Systematic observation: It mainly produces statistical data. The observer makes a systematic observation of the key observable events at a pre-determined time slots and may also observe specific people in the natural setting[31].

b) Participant observation: The observer makes the observation by being a part of the group and the place of study. Holistic observation is made either being undercover or openly. The researcher
collects information by listening to and questioning people over the period of time. This produces mainly qualitative data, and results in a holistic study in relation to the context [31].

For the purpose of thesis, the method of participant observation was chosen. Being an intern, direct access to the environment and people were possible. There was sufficient time to get accustomed with the overall environment. The researcher role was kept secret to preserve the naturalness of the environment. Notes on observation were made at the end of the day. Making observation without the informed consent was ethical from the point of view that the observed participants did not suffer in any way due to this observation, the identities of the participants were never revealed, and no confidential material was used or reported in the thesis. My role as a software developer intern helped me to make observations on the important value activities happening in the organization. Precaution was also taken to maintain the dual role of both as a software developer and observer. A checklist from [31] that was used to guide the observation has been attached in appendix (5.1). For observation, I attended the daily developers meetings and management – developers planning meeting and made personal notes about the on-going activities. Being an intern at the company, I could easily participate in these meetings and make careful observations. The observation notes have been attached in appendix 7.5.

4.2 Documents

Documents are also a source of information in the research [63]. Though documents can be available in many forms such as written documents, audio, video documents, in this research only written documents were relevant. Again written documents can be of different types - 'Government Publications' [63], newspapers and magazines[63], minutes of meetings[63], letters or emails [63], personal diaries [63] and website pages [63]. The relevant source of documentary information in this research were organization's official documents categorized as public, restricted and confidential, stored electronically in organization's online document repository. Emails exchanged between employees and minutes of meetings were other important sources.

Access to documentary resources was not very difficult. Organization's public documents and public website were easily accessible without any prior approval. Accesses to restricted documents - documents meant for consumption by the employees only were also accessible, as i had the required credentials and permission to view them. Confidential documents were out of the reach as it contained sensitive information and only the higher management had access to it. Being an intern software developer, I was part of the email exchanges and meetings within the development team and also received emails from the management for important announcements and general information. These emails were a good source of information on the important activities that were going on within the organization.

Validity of the documents is evaluated based on four important criteria - authenticity[63], credibility[63], representativeness[63] and unambiguous content [63]. It can be safely said that the documents were authentic since they were taken from the organization's document repository. The primary purpose of documents was to educate and to train employees on the important processes in the organization. They were regularly updated by the authorized people. Security procedure related documents were controlled by Security Manager, training documents by a training manager, infrastructure documents by the infrastructure team, customer support
documents by the support manager, software development process documents by the project leader and system architect and management documents by the company manager. Therefore, the documents were a credible source of information and were complete in their representation for the intended purpose. Since the documents had been created to give clear directions to the employees, the documents were relatively straightforward in their meaning. However, its contents were also validated during interviews as and when required. The organization's public website [64] was not frequently updated and so information from the website was not considered for data collection. Minutes of meetings were used as a source of information with a caution since they do not record everything and represented the perspective of the person taking the notes. However, being a participant observer in these meetings, I had my own personal notes which I used to validate the meeting minutes.

During the study of documents, a checklist was prepared and used based on the guidance given in[63]. As per the organization’s security policy, the documents, email exchanges, and meeting notes have restricted access only to the company employees on a need-to-know basis. Therefore, they cannot be attached or linked to the thesis.

4.3 Interviews:

According to [65], there are seven stages in an interview study: identifying the purpose of investigation, designing the interview, conducting the actual interview, transcribing from oral text to written documents, analyzing the contents of interview, verifying the collected data, and reporting.

The purpose of investigation was to validate data gathered from observation and documents and to search for any additional information relevant to business activities of the organization. Designing the interview involves deciding on the location, the candidates, and preparing the interview questions. The location was the organization’s head office and the candidates chosen were the branch manager and development team leader. Interview was designed to be an open-ended, semi structured and one-to-one interview [32]. Interview questions are given in appendix(5.3), they were designed keeping in mind the important guidelines from [65]:

- Interview questions have the dual purpose to extract knowledge and to maintain the interpersonal relationship during the interview.
- Questions of type ‘what’ generate theoretical answers while questions of type ‘how’ produces spontaneous and descriptive responses from the interviewee and keeps the interaction level high. Therefore, important questions must be formulated as ‘how’.
- Questions must be brief and simple and open-ended.

An email was sent to the interviewees to schedule interviews. The email also explained the purpose of the research. During and after the interview, notes were made. Moreover, interview is only used as data validation procedure to observation and documents. During the interview, some of the useful tips given in [66] were followed:

- Analysis, validation and reporting of the interview must be kept in mind while asking questions. This helps in clarifying the doubts, ambiguities during the interview itself.
• Important questions must be followed up by follow up questions to motivate users to think deeper and also to show genuine interest in what they say.

The analysis of the interview was done as a theoretical reading [67] in which no specific or systematic methods were followed, but the interview notes were read multiple times to understand and gather knowledge. This method was helpful because the interviewer had prior knowledge of the business modeling domain. Moreover, documents and observation had provided a suitable amount of information about the business model of the company to enable informed questioning.

4.4 Qualitative Analysis of Case Study Data

The data generated from interviews, observation and documents which are mostly in the form of words is qualitative data[68]. There are four important principles for qualitative analysis of data[68].

a) The conclusions reached must be strongly based on the collected data and evidence.

b) Careful interpretation of data through multiple and attentive reading.

c) Avoid personal biases and preconceptions developed from previous knowledge or experience.

d) The data analysis should be an iterative process.

Qualitative analysis of data is primarily based on inductive logic[68] where the logic is to discover knowledge from the data itself and to move from particular to generalized conclusions on the basis of collected data.

The process of qualitative analysis involved:

a) Preparation of data[68]: A back-up copy of the collected data was saved and then all the materials were organized in a compatible format. A word document was created to collate data from all the different sources and comments were added alongside as and where required.

b) Familiarity with data [68]: The collated data in the word document were read multiple times carefully to gain a thorough understanding of the material and comments were added alongside.

c) Interpretation of data [68]: In the phase, attempts were made to draw key concepts about the business model of the organizations, the important actors and the nature of exchanges taking place between the stakeholders. These key concepts were reviewed again to derive generalized conclusions. Any alternative explanation or theories were also carefully considered to improve the generalized model.

d) Verification of the data and analysis result[68]: The verification phase has four important parameters

   i. Validity of data,

   ii. Reliability of the data collection,

   iii. Objectivity: the extent to which the analysis is free from personal bias and preconceptions,

   iv. Generalization of the results.

The validity issue was addressed by the 'data triangulation' [68] method since there were three sources of information - interview, observation and documents. Reliability is demonstrated by a detailed explanation of the data collection and analysis procedure followed. Though it may be difficult to conclude same results using the same procedure by a different researcher, care has been taken to follow standard scientific procedures wherever applicable. Generalization of the result is difficult to achieve in a single case study. The objective here was to study a real life business organization's working business model so that it can be implemented on the newly developed e3 value model meta-
model and tool. The working model of each organization is unique in its own ways, and it is difficult to conclude how relevant this business model would be to other organizations. Objectivity again is difficult to prove in any absolute way since I was deeply involved in data collection and its analysis. However, objectivity was ensured by keeping an open mind while analyzing the data.

4.5 Results of Analysis

The result of the case study is the outline of the most important business transactions and value activities of the case i.e. EU Supply Holding AB. These activities have been presented in a tabular format with a brief explanation of each activity, involved actors and value objects that are exchanged.

Business transaction table:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Actors</th>
<th>Incoming Offering</th>
<th>Outgoing Offering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Third Party Services</td>
<td>There are a few service providers with which the organization has tie-ups. These organizations provide service at the cost of a monthly fee. E.g. Buypass- authentication service through smart cards Feide- another third party authentication services Time Stamp Authority: a third party timestamp provider.</td>
<td>Third Party Software Companies, EU-Supply</td>
<td>Software services</td>
<td>Fee</td>
</tr>
<tr>
<td>Graphical User Interface customization</td>
<td>EU Supply outsources some of the user interface customization to independent web consultants</td>
<td>Web Consultants</td>
<td>Customization Service</td>
<td>Money</td>
</tr>
<tr>
<td>Software Procurement</td>
<td>Software vendors such as Microsoft, JetBrains provide the necessary software for development and support activities</td>
<td>Microsoft EU Supply</td>
<td>Software</td>
<td>Money</td>
</tr>
<tr>
<td>Hardware Procurement</td>
<td>Computer hardware vendors such as Dell provide most of the hardware required in development and infrastructure management</td>
<td>Dell EU Supply</td>
<td>Computer Hardware</td>
<td>Money</td>
</tr>
<tr>
<td>Office space procurement</td>
<td>EU Supply rents office space, server storage space from landlords</td>
<td>Landlord</td>
<td>Office Space</td>
<td>Monthly Rent</td>
</tr>
</tbody>
</table>
Recruitment of office staff

EU Supply maintains a team of engineers for its various activities

Law and legal consultants

EU Supply receives the services of a legal firm for its various proposal preparation and to take care of legal issues arising out of contracts

Finance and accounting consultants

EU Supply receives accounting and tax services from an accounting firm. This firm takes care of salary disbursement and other bill payments.

Product reseller in international market

EU Supply has a number of redistributors in different geographic locations who earn the rights to resell the product on the payment of redistributors’ license fee and a commission of each client

Direct Sales to Customers

EU Supply also sells its product directly to prospective clients.

Additional Feature Sales to Customers

The product has a lot of optional features which clients can buy for additional fee. Customers can also request for implementation of customized features specifically for their use.

Training to customer end users

End users are provided training when they purchase the license or whenever new features are released.

Security standards certification

EU Supply has to maintain international security and environmental standards. These standards are important for the customers. EU Supply regularly calls for security audits from regulatory bodies to maintain these certificates.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Incoming Offering</th>
<th>Outgoing Offering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product development and enhancement</td>
<td>A team of software developers work on adding new features and enhancements to the software product. It is the most important</td>
<td>Software, Hardware, Software Developers</td>
<td>New features in the product</td>
</tr>
</tbody>
</table>
activity in the organization.

| Customer support and bug fixing | A support team is the first point of contact for customers with technical issues. They address the customer problems and fix bug in the online system | Customer Issues, Support personals | Customer support |
| End user training | The training dept. Provides training to end user when new features are released or when new clients are added. | Money, Professional trainers | Education |
| Infrastructure Management | The infrastructure team maintains and upgrades the software infrastructure to provide 24X7 accesses to the online system. | Software, Hardware, Infrastructure personals | Infrastructure service |
| Sales and marketing of the product | The sales team is responsible for submitting proposals to tenders and bringing in new clients to the organization. | Tenders, Sales Manager | Proposal |
| Recruitment of office staff | The recruitment team is frequently looking for technical and managerial staff for the organization. | Candidate Resume, Human resources | Salary |
| Internal Security Maintenance | The ISF () maintains the security and environment standards within the company as per international rules. | Standard Certificates, Audit Report | Certificate Fee |

Table 4-2: EU Supply Holding value activities

4.6 Case Study Report

Finally, the case study report was written. Since it was a simple single case study, the overall format was single narrative [69], with subtopics arranged in a sequence. The subtopics will consist of the case background, the issue being investigated followed by the data collection method and then their analysis and conclusion. This structure is suitable for exploratory case studies[69]. Report writing for the case study was a continuous process during the case study period. The case background, the theoretical proposition, and the data collection methods were prepared before the actual field work. During the field work, collected data was recorded. After data collection analysis, the conclusion was reported. Since the purpose of the case study is to portray a typical e-business model of a small scale software company with multiple partners and customers spread across multiple geographical locations, the identities of the case and its informants do not matter. Therefore, there is no purpose or benefit in revealing the true identities of the case or its informants[69]. However, the company website[64], a small introduction to the company and screenshot of the email from the company manager granting permission for case study have been given in the appendix. Finally, the case study
draft was reviewed by the case informants as a validating procedure to improve the accuracy of the study. An important addition to this report could have been the study of the other partner organizations and important clients to get an overall perspective of the entire business environment, but due to limited time, resources and accessibility issues it could not be done.
5 Result

5.1 E3 value Metamodel

The final e3 value metamodel that was accomplished after multiple design iterations and end-user testing of the tool is displayed in Figure 5-1.

The key features of the metamodel are:

The ‘e3 value meta-model’ covers all the important components of e3 value model. It is based on MOF meta-meta-model; therefore it is compatible with UML.

The actor class has three sub-classes: elementary actor, composite actor and market segment. One or more elementary actor can be associated to zero or more composite actor. This relationship between the elementary and composite actor has been defined through an association. Market segment is a sub-class of actor instead of being an independent entity as in [11] because it has similar attributes, relationships and functions as that of an actor. This helps to make the e3 meta-model simple and compact.

Value interface can only exist either as a part of the actor or a value activity. Actor or value activity consists of at least one or more value interfaces. A value interface can be a part of only one actor or value activity.

Value activity can only exist as a part of an elementary actor. A value activity can only be a part of one elementary actor. An elementary actor can have zero or more value activities.

Value port has two sub-classes: in port and out port. The two sub-classes have been created to simplify the relationships that exist between the ports and other e3 value model components. Value interface consists of zero or more in ports, and zero or more out ports. The in port and out port can only exist as a part of value interface.

A value port is always associated with only one value object. One value object can be associated with multiple value ports. This relationship has been established through association relationship between value port and value object. Value object has four sub-classes: product, service, information and money. This classification is missing in [11] and therefore an addition to the e3 value metamodel.

Value exchange has three sub-classes: in-in-exchange, in-out-exchange and out-out-exchange. These three value exchange sub-classes define the three different types of value exchanges that can exist in an e3 value model. In-in-exchange consists of exactly two in-ports. Out-out-exchange consists of exactly two out-ports and in-out-exchange consists of exactly one in-port and one out-port. These relationships between the value ports and value exchanges have been defined using association relationship and cardinality constraints.
Value transaction consists of at least one or more value exchanges. A value exchange can be a part of zero or more value transaction. This relationship has been defined using an association relationship and cardinality constraints between value exchange and value transaction.

This new e3 value metamodel has overcome some of the major drawbacks and ambiguities present in the e3 value class diagram as defined in [11]. The complex multiple relationships between value exchange and value port as defined in [11] has been replaced with simple association relationships between the value exchange and value port sub-classes. The relation between actor and market segment is simpler and clearer than in [11]. In [11] market segment is defined as a composed of zero or more actors and actors consisting of zero or more market segments, which is ambiguous to some extent. Moreover, cardinality constraints have been defined on the generalization relationship between elementary actor and actor, and between composite actor and actor. These constraints are unnecessary and they can be misinterpreted as actor consists of zero or more composite actor and one or more elementary actor. In [11], an additional relationship is defined between composite actor and value port, which is again ambiguous given that a relationship between the actor and value port has already been defined. In [11], the relationship between the value transaction and value exchange is missing.

The MOF based e3 value metamodel defines all the components and the relationships between them in a clear, simple and precise manner. It can be used to build design tools and create valid e3 value models.
Figure 5-1: e3 value metamodel final version
5.2 **E3 value modeling tool:**

Based on the e3 value metamodel, an entirely new e3 value modeling tool was developed using Microsoft Visual Software Development Kit. The new tool was necessary to enable end users such as business analysts, software architects, requirement engineers etc. to create e3 value models based on the new metamodel. The tool has an easy-to-use and intuitive drag and drop graphical interface.

The picture above is the screenshot of the new e3 modeling tool. On the left side is the toolbox containing tools to create e3model elements such as actor, market, value interfaces etc. In the middle is the design canvas to create the model. On the right side, the tool gives a tree-view of the model being created. The tree view is particularly useful to navigate a complex model, and to identify the relationships that exist among the different e3model elements. This tool needs Microsoft Visual Studio IDE to run, and it inherits most of the features and user interface from the IDE. An elaborate description of each of the feature of the tool has been presented in section 3.5.

A detailed description of all the items in the toolbox is given in the following table:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon.png" alt="Actor" /></td>
<td>Actor: Elementary</td>
<td>Drag and drop on the canvas to create an elementary actor,</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Actor: Composite</td>
<td>composite actor and market respectively.</td>
<td></td>
</tr>
<tr>
<td>Market</td>
<td>Cannot be dropped on any other elements such as – value activities, value objects etc.</td>
<td></td>
</tr>
<tr>
<td>Value Activity</td>
<td>Drag and drop on elementary actor to create a value activity. It can only be created inside an elementary actor.</td>
<td></td>
</tr>
<tr>
<td>Value Interface</td>
<td>Drag and drop to create a value interface. It can only be dropped on elementary actor, composite actor, market, and value activity.</td>
<td></td>
</tr>
<tr>
<td>Value Exchange: Exchange</td>
<td>Drag and drop on the canvas to create value exchanges between interfaces. It can be dropped only on the elementary actor or on the canvas.</td>
<td></td>
</tr>
<tr>
<td>Value Exchange: Out-Out-Exchange</td>
<td>Drag and drop on the canvas to create value exchanges between interfaces. It can be dropped only on the elementary actor or on the canvas.</td>
<td></td>
</tr>
<tr>
<td>Value Exchange: In-In-Exchange</td>
<td>Drag and drop on the canvas to create value exchanges between interfaces. It can be dropped only on the elementary actor or on the canvas.</td>
<td></td>
</tr>
<tr>
<td>Value Object: Product</td>
<td>Drag and drop on the canvas to create a value object of type ‘product’, service, information and money respectively. It can only be dropped on the canvas and cannot be embedded inside any other element.</td>
<td></td>
</tr>
<tr>
<td>Value Object: Service</td>
<td>Drag and drop on the canvas to create a value object of type ‘product’, service, information and money respectively. It can only be dropped on the canvas and cannot be embedded inside any other element.</td>
<td></td>
</tr>
<tr>
<td>Value Object: Information</td>
<td>Drag and drop on the canvas to create a value object of type ‘product’, service, information and money respectively. It can only be dropped on the canvas and cannot be embedded inside any other element.</td>
<td></td>
</tr>
<tr>
<td>Value Object: Money</td>
<td>Drag and drop on the canvas to create a value object of type ‘product’, service, information and money respectively. It can only be dropped on the canvas and cannot be embedded inside any other element.</td>
<td></td>
</tr>
<tr>
<td>Value Port: In Port</td>
<td>Drag and drop on the value interfaces to create value ports on its edge. It cannot be dropped on any other elements.</td>
<td></td>
</tr>
<tr>
<td>Value Port: Out Port</td>
<td>Drag and drop on the value interfaces to create value ports on its edge. It cannot be dropped on any other elements.</td>
<td></td>
</tr>
<tr>
<td>Value Transaction</td>
<td>Drag and drop on the canvas to create a value transaction. Value exchanges can be attached to the value transaction to denote transactions.</td>
<td></td>
</tr>
<tr>
<td>Connector: Value Object to Ports</td>
<td>It connects value objects to value ports. It ensures that a one port is connected to only one value object.</td>
<td></td>
</tr>
<tr>
<td>Connector: Value ports to Value exchanges</td>
<td>It connects value ports to value exchanges. It ensures that : Exchange is connected to only one out port and one in port. Out-Out Exchange is connected to only two out ports belonging to different value interfaces. In-In Exchange is connected to only two in ports belonging to different value interfaces.</td>
<td></td>
</tr>
<tr>
<td>Connector: Elementary Actor to Composite Actor</td>
<td>It connects an elementary actor to composite actor. Elementary actors cannot be embedded within a composite actor. Only a reference connection exists between the two types of actors.</td>
<td></td>
</tr>
</tbody>
</table>

Table 5-1: E3 value model editor toolbox items
The actual business model of ‘EU Supply Holding AB’ is implemented to demonstrate that the metamodel and the new tool are capable to handle complex and large business models. The case study gave a detailed view of all the major business exchanges and value activities that occur in a typical software service providing organization. The business model was implemented using the new e3 value metamodel and its tool to demonstrate all the key features. The e3 model is quite complex and large but the tool handles the model efficiently. Some readability is lost as there are too many connections. The serialized xml containing the model information for the EU Supply Holding AB e3 value model is attached in appendix.

The code for e3 modeling tool on VMSDK and the installation package for the tool have been uploaded at https://www.dropbox.com/sh/cn7vlhp7wmu7l5g/gcXD4jNgPb.

The e3 modeling editor in EMF and the eclipse plugin has been uploaded at https://www.dropbox.com/sh/cn7vlhp7wmu7l5g/gcXD4jNgPb.
Figure 5-3: EU Supply Holding AB e3 value model
6 Analysis:

6.1 Choice of Method

The final activity in the design science project is to evaluate the developed artifact for its effectiveness in addressing the identified problem and its completeness in meeting the requirements set out in the second phase. In general evaluation is a descriptive and explanatory in nature, trying to explain why the artifact is useful[30]. Before evaluation, an evaluation strategy and method must be chosen. There are two main types of evaluation ex–ante[30] and ex-post[30]. Ex-ante evaluation is done without actually using the artifact. It involves conducting interviews to gather expert opinion on the artifact. Another way is through informed arguments[30] where the researcher does self-evaluation of the artifact and provides arguments supporting the designed artifact. Informed arguments are weak in nature but inexpensive and easily available while expert opinions rely heavily on the subjective judgments of experts[30]. Ex post evaluation is done after the artifact has been put to actual use. It is stronger than an ex-ante evaluation. It is useful to evaluate the usability and effectiveness of the artifact[30].

For ex-ante evaluation, one of the several research strategies could have been used- conduct a survey to get feedback on the artifact from a large audience; theoretical analysis of the artifact based on informed arguments; interviews with experts for expert opinion. Out of these methods, survey would not produce the right result as evaluating a meta-model requires in-depth analysis; interviews with experts would be the most desirable method here, but time constraints and low availability of experts to give opinion make it infeasible. So it was decided that the researcher would present his informed arguments on the artifact as a part of the evaluation.

For ex-post evaluation, case study and experiments are the two most popular research strategies. Ex-post evaluation would be useful to evaluate the usability[82] and learnability [83] of the developed tool.

Case studies produce an in-depth analysis of the artifact, but it needs a lot of effort and competence from the researcher. To achieve generalization of the results, case study must be conducted at multiple sites [30]. A possible evaluation method would be conducting multiple case studies at organizations of varying size and complexity to collect the important information for creating their business models using the new tool. This would give an insight on the usability and efficiency of the tool.

Experiments[84], which means conducting investigation under controlled environments by varying the causal factors and making empirical observations. Another approach within experiments is ‘quasi-experimental’ method in which the variables are not controlled or difficult to control and the researcher makes observations by carefully looking at the naturally occurring experiments[84]. Though experiments are precise and have high credibility, it is always a question whether the artificial settings resemble real world situation and if the researcher can practically and ethically control the variables[84]. A possible experiment to evaluate the artifact could be to allow the business users use the tool to create e3 value models of their businesses in the organizational setting without any artificial control. The usability and learnability of the tool could be then evaluated by observing the users experience with the tool in modeling complex and real world business.

Due to lack of access to multiple organizations to conduct case studies and lack of availability of business users willing to spend time and effort in learning e3 value model concepts and then using the tools, it was very
difficult to conduct multiple case studies or a full scale experiment within the time constraints. A detailed case study at a single site was conducted as a part of the demonstration. The outcome of this case study can be used to evaluate the tool, though it is difficult to generalize the results from it.

6.2 **Application of Method**

The developed metamodel would be evaluated with respect to each requirement identified in requirement phase. There were two sets of requirements – requirements on the metamodel and requirements on the graphical tool.

6.2.1 **Requirements on the metamodel:**

Requirements on the metamodel could have been best evaluated through expert opinions, but due to unavailability of experts only informed arguments from the researcher are presented.

*Meta-model should be based on MOF:* The e3 value metamodel is entirely based on MOF. The metamodel has used only the basic concepts of object-oriented principles included in MOF such as Class [85], Generalization [53], Aggregation [53], binary relationship between the classes, primitive types such as integer and string for class attributes. The ecore metamodel is the closest implementation of MOF available for commercial use. Since the e3 value metamodel was built on ecore metamodel, it explicitly proves that the e3 value metamodel is MOF based.

*Meta-model should cover all the existing e3 value concepts:* The metamodel has covered all the e3 value concepts namely actor, market, value activity, value interface, value ports, value exchanges, value objects. However, there were two important deviations; Firstly, the concept of value offering was removed from the metamodel. During the testing of the meta-model, it was realized that value offering adds an extra layer of nesting in the value interfaces, it does not have any unique properties of its own, all the information is either on the value ports or on the value interface, it does not participate in any relationship with other elements. Therefore, practically it does not add any value to the e3 value models. It is a name given to the group of value ports inside the value interface. Its removal did not cause any loss of information or any difficulty in projecting the business models as e3 value models. It made the model simpler, and so it was rightly removed. Secondly, the value object was specialized into four child classes – product, service, money, information. This was an addition to the e3 value concept. It helped in adding specific information about the value objects. Value object is a much generalized concept. It is difficult to capture the attributes of all the possible type of things that can exist as value objects. Specialization of value objects in four broad classes helps to classify value objects based on its attributes and adds more information to the e3 value model. Addition of these child classes did not create any additional relationships with other elements or affect the existing relationships in any way. Therefore, it was safe to make this classification.

*Meta-model should represent the relationship between the e3 model concepts:* Some of the relationships among e3 value concepts are straight-forward and implemented in the metamodel as they are - Actors are specialized into two - elementary and composite actor, elementary actor contains value activities, actors contain value interfaces, value interfaces contain value ports, value objects exists as independent objects, value exchanges connect only value ports. ‘Market’ is defined as a child class of ‘Actor’ in the metamodel although it exists as a separate concept in the e3 value model definition. Market has been defined as a special case of an actor because its attributes and relationship with other e3 value concepts is very similar to that of an actor. It has value interfaces; it participates in the business as a single entity (although it represents a collection of a larger number of business entities that value the value objects in the same manner). Thus, defining market as a type of actor saves us from creating unnecessary relationship with value interface class.
Meta-model should represent the constraints between the e3 model concepts: All the constraints of e3 value model have been implemented in the metamodel. Constraints such as – value interface must belong exclusively only to an elementary actor, composite actor, market or value activity; value ports belong exactly and only to one value interface; value activity belongs exclusively to only an elementary actor; are implemented through aggregation composite relationship between actor and value interface, value interface and value ports, elementary actor and value activity respectively. Other constraints such as – value ports must be associated with exactly one value object, value exchanges must connect exactly two ports have been implemented through cardinality constraints on the relationships between value object and value ports, and between value object and value exchange respectively. Another important constraint between value exchange and value ports is that the value ports connected through value exchange will depend on the type of value exchange. These contain was implemented in the metamodel by specializing value exchange class in three sub-classes and connecting them to value port sub-classes accordingly. Thus, all the major constraints were implemented in the meta-model itself. Constraints that could not implemented in the metamodel such as – value exchange must not connect value ports of the same value interface, were implemented programmatically in the graphical tool.

Meta-model should be expressed in human readable format (a class diagram): The meta-model has been presented as a class diagram with proper names for the classes, attributes and relationships.

Meta-model should be practically useable: The use of graphical tool to create the business model of a real-world organization provides a fair evidence of its usability in practical scenarios.

The requirements set out at the beginning of the e3 value metamodel design seems to have been fulfilled based on the given explanation.

6.2.2 Requirements on the graphical tool:

Implement the e3 value metamodel as it is: The metamodel was implemented as it is with only one change. The tool required a single parent root node as a starting point. Therefore, a root node ‘e3BusinessModel’ was added to the metamodel that embedded actor, value exchange and value object class.

Implement the implicit and explicit constraints on the e3 value model concepts: Most of the constraints were a part of the metamodel and were implemented implicitly with the implementation of the metamodel. Some constraints such as value exchange cannot connect ports of the same interface were implemented in the code. The validation errors printed out user friendly messages on violation of any constraints as shown in the demonstration.

Validations to help the users design a correct model: Functionality has been implemented to validate the entire model or only part of it, either when the model is saved or when the user desires. Validation option is present in the context menu of the e3 model tree view as shown in the demonstration.

Error prevention mechanism to prevent common errors: To ease the user in creating models, some of the common errors were taken care of by the tool itself. E.g. unique names were assigned to each element by default when they were created; only valid child elements can be embedded in the parent elements – value activity cannot be dropped on the composite actor or market, value ports can be dropped only inside a value interface, value objects cannot be embedded in any other element, value exchange could connect only two ports depending on the type of value exchange and so on. These mechanisms guided the user to create a proper e3 value model.

Tree view navigation of the model: Tree view of the model is necessary to navigate large and complex diagrams. Sometimes it is difficult to follow the relationships in a large diagram which may be over-crowded
by the many connectors. So along with the graphical model, the tree view of the same model was also available to users. Users could also create model using the tree view if desired. The tree view has been shown in the demonstration.

**Functionality to print the model:** The e3 value model created has the option in the ‘File’ menu to have a print preview or print the model directly.

**Functionality to save the model in a serialized format following xmi standards:** This requirement was only partially fulfilled by the graphical tool. The graphical tool developed on Microsoft VMSDK serialized model as an xml but in the Microsoft proprietary format and not the XMI standards. This requirement was difficult to fulfill exactly as manipulation of the xml format was not allowed in VMSDK. However, the e3 value model editor created in EMF serialized the model as xml following XMI standards.

**Easy-to-use graphical interface:** It is difficult to conclude whether the graphical tool is easy-to-use without having a group of users actually using the tool and providing feedback. However, some features were implemented keeping in mind the end user. The icons were carefully chosen to represent the e3 model concepts in a way that the icons have some relationship with the concept that they represent. Different colors were assigned elementary, composite actors, market and value activity shapes for easy distinction. Different types of value exchange connectors were merged into one connector, but internally it was ensured that the ports were connected depending on the type of value exchange. E.g. connecting two in-ports would automatically draw green dashed line between the ports representing an in-in-exchange. Items could be easily dragged from the toolbox and dropped on the canvas, dropping a child element would increase the size of the parent object accordingly, child elements could not be dragged out of parent elements and so on. Some user friendly features came along with the Microsoft VMSDK. The toolbox, the canvas and the tree view window could be arranged in different orientation as the user desired; print preview was available before printing, the model could be zoomed in and out. Thus, it is evident that sincere efforts were made to make the graphical tool user friendly, but it cannot be claimed with certainty that the tool is user friendly as no user feedback was collected.

### 6.2.3 Reflection on analysis

The analysis is a descriptive informed argument from the researcher critically examining each requirement with the results. Informed argument has its limitations. Analysis could be made stronger through expert opinion on the created artifacts. The extent of user-friendliness of the graphical tool could not be concluded as there was no user evaluation of the tool. Demonstration of the tool in a case study was effective, but ideally multiple case studies would have led to a generalized conclusion in a very convincing manner. Despite these limitations, the analysis is sufficient, objective to an acceptable level, and fulfills the necessary requirements in the context of limited resources and time.

### 7 Conclusion

Based on the evaluation of the metamodel and its graphical tool, it can be said that the MOF based e3 value metamodel was successful in implementing the concepts of e3 value modeling in a way such that the relationship between e3 model elements and their constraints are well-defined, without any loss of flexibility.
and scalability that e3 value models provide. This answers the first sub-part of the research question – ‘Relationships and constraints between the e3 value model elements are well-defined’.

The e3 value metamodel is based on MOF; the UML metamodel is also based on MOF. Though no attempts were made to generate UML models from e3 value models, e3 value models will be certainly compatible with UML as they are based on the same meta-meta model. This answers the second sub-part of the research question: ‘E3 value business models are compatible with UML’.

The implementation of the e3 value metamodel in a tool proved that the metamodel can be put to practical use. The implementation of a real-world business scenario demonstrated that complex and large models can be designed based on the metamodel. Serialization of the model in standard xml format makes the model available for consumption in other software programs or tools. The tool was designed to have an intuitive, end-user interface to drag, drop and connect e3 model elements. This answers the third sub-part of the research question – ‘Graphical editor tool for end users based on metamodel can be implemented’.

Thus, it can be concluded that the research question set at the beginning of the thesis has been answered.

There have been a few research work done in the past to transfer knowledge from one perspective to another in a consistent manner. [43] suggest relating two modeling notation from the different perspective by taking the common concepts and creating a reduced model and then transforming the models from the different perspective to the reduced model. Thus, it defines an equivalence relationship between the models. They demonstrate their strategy by using e3 model for value perspective and UML activity diagram for business process perspective. Another approach ‘syntactic translation’ by [86] suggests relating modeling notation of one approach to another and then transforming one model to another. [87] suggests an approach ‘meta-representation’ where relationship is defined between the meta-models and concepts of two modeling notation and this relationship must be consistent across all instances of the two models. In [88], a reference ontology is created based on REA, BMO and e3 value model to establish the relationship between different ontologies. It is then extended to remove redundancy caused by overlapping concepts and additional sub-concepts to bridge the gap between the ontologies.

A few of the past researches have focused on combining two modeling ontologies to complement each other. Combining value modeling with goal modeling to design and analyze e-business at a strategic level is demonstrated in [89], where e3 value model complements the i* goal modeling. Another similar approach has been implemented in [12] where in an iterative process a value model is created to capture high level economic resources exchanged and then a goal model is created based on it. Goal model is further refined to derive e-services. This will increase the scope and application of business model ontologies.

In one of the past research, E3 model has also been extended as ‘e3 controls’ to include control mechanism for opportunistic scenarios in the business environment. The traditional e3 model assumes an idealistic environment for business where every transaction is upheld by the involved actors. Some control mechanisms are introduced to handle scenarios where transaction are not proper or upheld by the participating actors. [90].
The thesis has drawn inspiration from previous researches but has also contributed in a unique way. It has led a foundation step to combine the usefulness of e3 value modeling with model driven software development. E3 value model which was until now a stand-alone concept has been brought closer to software engineering without any loss of flexibility in the model design.

The thesis work has certain limitations. E3 value model concepts underwent minor changes during the creation of metamodel. A few of them was added intentionally while others were necessary to make it compatible with MOF standards. Though demonstration of metamodel and tool had no signs of drawbacks due to these changes, yet it requires confirmation through expert opinions. Evaluation of the metamodel from experts could have made the analysis robust and more conclusive. The developed tool has undergone limited testing during its development. It requires extensive testing both at the hands of professional testers and end users to check its robustness, reliability and user-friendliness before it can be made available for general use.

Due to these limitations, the result of the thesis cannot be used directly to design e3 value business models for commercial purposes. There may be undesirable social consequences on using this tool for real purposes, until the tool is thoroughly tested. The usage of the metamodels and the tool is thus limited to academic domain only and it does not replaces the existing old e3 value modeling tool.

As a possible ethical and social consequence of the conclusion, the results of the research work could be used for further research in academic domain. The developed tool does not contain any hidden or malicious code that would violate the data privacy or security of the user. The code is freely available as open source and interested students and researcher could use or modify the tool as required.

As an immediate future work, a detailed analysis of the metamodel and the tool through expert opinion and experiment with a user group respectively can be done. This would validate the results of the thesis and could generate useful insights for further improvement in the metamodel and the graphical tool. Along with it, several new features can be added to the tool - ability to minimize and maximize actors or certain section of the model to handle large models, user tips and user guide can be added to assist the user while creating model; more flexibility can be given to the user in choosing color, orientation, shape and size of the e3 value model elements.

E3 value model and UML share the same meta-metamodel, so it is possible to generate UML class diagrams for the given e3 value model. It would be interesting to investigate what type of UML models can be generated from e3 value models and whether e3 value models can be created from a UML model. This would be a significant achievement and would enable business users and software developers to communicate using models. A further step would be generating code from the given e3 value models. In model-driven development, it is possible to generate code based on the given domain model. E3 value models can be similarly used to generate code and database structures.

Another interesting area of future work can be a research on the possibility of exchange of business models between different tools. OMG has set serialization standards (XMI) of models in xml format, but most of the tools have not implemented the standard or have adopted a variation of XMI standards to maintain compatibility with their previous versions. These variations in serialization of models are an obstacle to exchange models between tools.
References


[22] M. Facility, T. Mof, X. M. L. M. Interchange, and D. We, “18 The MOF and XMI,”.


[57] Microsoft, “Getting started with Paint.”


[64] eu-supply holding ab, “eu-supply - tender mgmt, procurement and supply chain services.”


## Appendix

### 7.1 Design Science Canvas

<table>
<thead>
<tr>
<th><strong>Problem:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>High level problems:</strong></td>
</tr>
<tr>
<td>• Lack of well-defined modeling principles to construct a BM.</td>
</tr>
<tr>
<td>• Lack of well-established relationship between the components of a business model presents difficulty in extraction of knowledge from BM.</td>
</tr>
<tr>
<td>• Difficult to integrate BM with other aspects of business design such as a business strategy, business process modeling and architecture of IS to support the business.</td>
</tr>
<tr>
<td><strong>Problem in the context of e3 value model:</strong></td>
</tr>
<tr>
<td>• The relationships between the e3 model components and constraints are not well-defined enough to use e3 value models in model-driven development.</td>
</tr>
<tr>
<td>• E3 model is not compatible with UML – the most prominent modeling standard in software engineering.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Scope:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• The artifacts developed will be used by business analysts, requirement engineers and software architects to create a common line of understanding between the business stakeholders and software developers.</td>
</tr>
<tr>
<td>• To create a business value models compatible with UML.</td>
</tr>
<tr>
<td>• To integrate business model with software architecture.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Artifact:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• A meta-model for e3 value model based on MOF.</td>
</tr>
<tr>
<td>• A graphical editor tool to create e3 value models based on the metamodel.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Requirements:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>The requirements on the metamodel are:</td>
</tr>
<tr>
<td>• Meta-model should be based on MOF.</td>
</tr>
<tr>
<td>• Meta-model should cover all the existing e3 value concepts.</td>
</tr>
<tr>
<td>• Meta-model should represent the relationship between the e3 model concepts.</td>
</tr>
</tbody>
</table>
- Meta-model should represent the constraints between the e3 model concepts.
- Meta-model should be expressed in human readable format (a class diagram).

The requirements on the graphical tool are:
- Implement the e3 value metamodel as it is.
- Implement the implicit and explicit constraints on the e3 value model concepts.
- Functionality to save the model in a serialized format following xmi standards.
- Validations of the model to help the users design a correct model.
- Error prevention mechanism to prevent common errors while creating models.
- Tree view navigation of the model to navigate large and complex models.
- Functionality to print the model.
- Easy-to-use graphical interface.

Knowledge Base:

### Business Model

### E3 model
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.168.599&amp;rep=rep1&amp;type=pdf
- Website: http://e3value.few.vu.nl/

Design Science

MDA
- Website: http://www.omg.org/mdl/

UML
- Website: http://www.omg.org/spec/UML/

Constructs:

Business Model: A conceptual tool to express the business logic of a firm from a holistic perspective, focusing on the value proposition that the firm offers and the architecture of its network of partners with which it collaborates to create, distribute and market the values in an economically sustainable way. It also acts as the conceptual tool of alignment between the business strategy and business process leading to business information system.

E3 model: E3 value modeling is a light-weight, semi-formal, conceptual and multi-viewpoint approach drawing inspiration from business science, requirements engineering and conceptual modeling techniques from software engineering to model e-business ideas using graphical notations.
**XMI:** A model defines entities and the relationships among them, but it does not provide any specifications on how to represent physically those concepts. As a result, different modeling tools using the same modeling language may store the data and represent the models in a different format. It is an OMG standard to serialize model data in XML for models which has its metamodel based on MOF schema. Using XMI, the model instances between different modeling tools can be exchanged.

**MDA:** Model Driven Architecture is an architectural framework proposed by OMG which positions models at the center of information systems development. It aims to use modeling artifacts to manage the abstract information about the business aspects, functional aspects and implementation requirements of the information systems and then generate executable code using the models.

**MOF:** MOF is a meta-metamodel defined and maintained by OMG. It is used to define metamodels for models based on object oriented principles. The metamodel of UML and the XMI are defined using MOF. A model is a schema that represents the general knowledge about a domain and a metamodel is a schema that represents the general knowledge about the model in the domain.

**UML:** UML is an industry-wide standard visual modeling language for software architects, engineers or developers to design and document the software systems using both text and graphical notations. It can be used across different application domains as well as technology platforms as an all-purpose modeling language.

**Explicate Problem:**
The broad outline of the problem given by the supervisor was taken as the starting point.
Research Method used: Literature Review.
Research articles were searched on ‘Google Scholar’, KTH online library, IEEE, ACME, Springer, and Elsevier with the different combination of keywords like e3 value model, business modeling, meta-model, value modeling, ebusiness, integration of models, REA, BMO, UML, MDA, OMG, business process model.
Approximately 40 research articles were selected based on title, number of citations and the prestige of the journal.
The abstract of each of the article was read carefully to further filter around 20 articles.
These selected articles were studied in detail. The reference list of these articles led to other similar articles which were again analyzed in the similar manner.
This process continued until I could not find anything new or substantially different from what I had discovered on the chosen topic.
A few business modeling tools like ‘IBM Rational Software’, ‘Adocus metamodel agent’, ‘Protege’, ‘E3 modeling tool’ were also reviewed.
Websites of important organizations related to modeling and design such as W3C, OMG, and E3 value model were also searched for relevant information.
A set of high level problems were drawn.
Based on these high level problems, problem in the context of e3 value model was derived.

Define Requirements:
The literature was reviewed with the objective of eliciting requirements.
To derive high level requirements, the problem statement was taken as the starting point.
The high level requirements were written in natural language.
Based on the high level requirements, it was evident that a UML compatible meta-model for e3 value modeling must be constructed.
The three available alternatives to construct the metamodel were analyzed qualitatively.
A set of clear and concise requirements for e3 value metamodel was derived.
A set of clear and concise requirements for e3 value modeling tool was derived.
Finally, the set of low level requirements were analyzed qualitatively using a checklist.

Develop Artifact:
This step is a creative process and does not have any specific research strategy or method.
Explore through possible design alternatives and narrow down to a feasible solution.
Select a few of the most widely used UML modeling tools.
Create simple prototype e3 meta-model to get a quick understanding of the tool.
Finalize the tool and start the actual development.
The development was carried out in following broad steps:
1. Design the meta-model for e3 value model based on MOF. The tool used was Microsoft Visio.
2. Implement the metamodel in eclipse modeling framework and create a basic modeling editor.
3. Implement the metamodel in Microsoft VMSDK and create a graphical modeling editor.
Each of these steps had several sub-steps.
There were several iterations among the sub-steps, going back and forth to fix the bugs, issues with the metamodel, customizing the tool and so on.

Demonstrate Artifact:
The demonstration is done in two steps:
Business model of a fictitious mobile phone reseller is implemented to demonstrate all the key features of the tool.
Business model of a real software company is implemented to demonstrate the ability of the metamodel and the tool to handle complex scenarios.

Evaluate Artifact:
There are two types of evaluation ex-ante and ex-post.
Ex-ante evaluation is done without actually using the artifact but through expert reviews and informed arguments. Ex-ante evaluation here is done using informed arguments.

Ex-post evaluation is done by testing the artifact in a real world scenario. Ex-post evaluation here is done using a case study of the business model of a software company and its implementation using the new tool.

**Construction:**

The important design rationale considered during the development were:

- Elementary actor, composite actor and market segment were made sub-class of the super-class Actor.
- Value activity has been modeled in a composite aggregation relationship with the elementary actor.
- A value interface can exist either as a part of the actor or a value activity. It cannot exist independently. It cannot be shared between the actor and its value activities. So it has been modeled in a separate, composite aggregation relationship with both the actor and the value activity.
- In the meta-model, value exchange has been designed as a super-class and sub-class inheritance relationship. The super-class ‘Value Exchange’ is inherited by three sub-classes ‘In In Exchange’, ‘Out Out Exchange’, and ‘In Out Exchange’. ‘In In Exchange’ and ‘Out Out Exchange’ have association relationship with ‘In Port’ and ‘Out Port’ class respectively with a constraint of 2, which means that they have to be mandatorily associated with exactly 2 ports. Similarly, ‘In Out Exchange’ is associated with exactly one ‘In Port’ and one ‘Out Port’. On the other hand, ports can be associated with zero or more exchange simultaneously.
- In the metamodel, the value object has been specialized into four sub-classes – product, service, money and information.
- The ‘value-offering’ class and its sub-class – incoming and outgoing offering were removed from the metamodel. It was realized in testing that value-offering did not add any useful information to the model; rather it increased the level of nesting unnecessarily. Incoming offering and outgoing offering are the names given to collection of in-ports and out-ports respectively in a value interface and do not have any more significance.
- Created two composite aggregation relationships between the value exchange and elementary actor and composite actor respectively. The reason for this change was that both the actors can have value activities, whose ports will participate in value exchange with other value activities within the actor, or with the value interface within the actor that participate in value exchange with external actors. These value exchanges will be inside the actor and will cease to exist if the actor is deleted.

**Function:**

Create e3 value business model using MOF based meta-model.

Validate the model, for e3 value model rules and constraints.

Print the model as image.

The model is serialized as a human readable xml file.
| Generate UML class diagrams from e3 value model. (Future work) |
| Generate code using e3 value model. (Future work) |
| **Usability:** |
| Easy to use graphical editor. Intuitive drag, drop and connect mechanism to draw models. |
| Inherits the user friendly interface as provide by Microsoft Visual Studio. |
| User friendly validation messages. |
| Tree navigation for large and complex model. |
| E3 value model can be saved as an image or as an xml file. |
| **Effects:** |
| Design e3 value models that are compatible with UML. |
| Bring business models a step closer to software modeling. |
| Maintains the flexibility and light-weight nature of e3 value modeling. |
| One of the limitations is that the e3 value modeling tool is dependent on Microsoft Visual Studio development environment. |
7.2 EU Supply Holding AB

The company provides a web platform called CTM (Complete Tender Management System) for internet enabled procurement. CTM can be used for tender management, framework agreements between suppliers and buyers, online auction, document management, auditing and for generating reports. The company is primarily based in Stockholm with branch offices at London and Copenhagen. The Stockholm office is managed by Dr. Thomas Beergrehn. Some of its major clients are Trafikverket (http://www.trafikverket.se/) and British Nuclear Group Sellafield Ltd (http://sellafieldsites.com/profile/). More details could be found at company’s website (http://www.eu-supply.com/index.asp).

7.3 Observation Checklist

1. Have you defined clearly which form of observation and participation is being used?
2. Are you sure that total participation observation did not disturb the natural environment?
3. Have ethical issues been taken care of?
4. Is the observer role suitable in the context of fieldwork situation and access?
5. Was the time sufficient to get accustomed to the environment and make an in-depth observation?
6. Were notes made on a frequent basis?

7.4 Document Checklist

1. Is the document authentic?
2. Is the document credible in terms of:
   a. Who created the document?
   b. When it was last updated?
   c. What is the access level on the document?
   d. What was the purpose of creation of document?
   e. What is the intended use of document?
3. Is the meaning unambiguous?
4. Is any copyright law being violated?

7.5 Interview Questions

1) Could you please give a brief description of your role in the organization?
2) What are the important activities that come under your supervision?
   a) How are these activities conducted?
   b) Who are involved in these activities?
   c) What are the prerequisites for the activity to start?
   d) When does this activity end?
   e) How critical is this activity to the organization as a whole?
3) How does a customer purchase the licenses to use the software?
4) How are potential customers identified and how is the initial communication established?
5) How is support and maintenance provided to customer during the license period?
6) Could you please describe the role of partners in the business?
7.6  Observation Notes

13-Aug-2012
Daily Scrum Meeting: Meeting takes place around 10 AM and lasted for 15 minutes. Each developer briefly explains the task that was done yesterday, today's plan and highlights issues if any.
Sprint Planning Meeting: The meeting goes on for entire day with a lunch break in between. The product owner put forwards a list of prioritized task. This list contains the product features that must be developed in the next one month. Some of the features are customer specific and they are being paid for by the respective clients. Some of the features are for the overall improvement of the service itself. Some features involve bug correction and performance improvements. The cost for general feature development and product enhancement will be borne by the organization itself. The product owner explains each feature request, its business value, clients interested in it and possible release date to production environment.
After the lunch the meeting resumes. This time only the developers are present. Each of the feature requests is discussed at a technical level; time and resources required is estimated.

14-Aug-2012
Daily 10 AM meeting: meeting as usual.
Development Activity: All the developers are busy today working on the new feature request. There are some design discussions on how to implement some features among some of the developers. I had to establish contact with an external organization Buypass which provides smart card authentication services in Norway. We have to integrate their authentication services with our product as the users in Norway want to use it.
There was also discussion to provide reporting services to one of our client 'Trafikverket'. They will pay for this new feature. We provide translated and customized web page to clients to use our product online. Today some of this web page design was outsources to independent web designers by the support team.

15-Aug-2012
Meeting 10 AM as usual.
Development Activity: as usual. Nothing special to report on.
Support Activity: The support team offers first line, second line and third line support to customers. The customers pay a monthly fee for this support service. Only technical support is provided. First line support
receives customer's call or emails and tries to help them on the spot. If the issue is complicated, the second line support tries to provide solution or workaround. At last if the second line fails to provide support, a bug is logged and the third line support tries to resolve that bug. Support team also provides support to branch office in London and Estonia if they have complex cases.

**16-Aug-2012**

10 AM meeting: The product owner announces that we have a new client in Ireland and some priorities will change.

Development Activities: as usual.

Support Activities: There is discussion to provide training to the new support personals in Ireland. The new support personal will travel to Stockholm for a few days training. Training would include security training, customer interaction training, and introduction to the product, its key features, support protocol and procedures.

Hiring Activity: The manager and the system architect and one of the senior developer interviewed a few developers.

**17-Aug-2012**

10 AM Meeting: as usual

Development Activities: The system architect informed that the new version of visual studio is available for download from Microsoft. We have a yearly commercial license form Microsoft which is our primary supplier of software used in the development activity. Other vendors are Jet Brains that provide developers software to assist in development.

Infrastructure Activity: Infrastructure team maintains the infrastructure to provide software services to clients. They maintain and upgrade the hardware on regular basis. Dell is the vendor that provides most of the computer hardware and accessories. They also maintain backup of the data at different physical locations. The company has rented certain secure locations to maintain data backup in case of emergencies.

Support Activity: as usual.

Sales Team Activity: The sales team consisting of the branch manager in Stockholm, London, Copenhagen and Oslo were discussing on a proposal today. They get tender request from possible future clients or get news from government websites. The sales team prepares the proposal based on the requirements of the client, decides on the pricing and submits the quotation. This activity involves lengthy discussions among the sales team members and one senior developer was also involved.

**20-Aug-2012**

10 AM meeting: The development team leader shared the released dates for the new features of the product.

Development: The development team discussed for a short while the release plan for the features. The resources required and the time of release was decided.

Support: as usual

Training: The training manager has shared her training schedule. The end users of our product are provided training whenever new features are released or when a new client is added to organization. The client pays for the training sessions and the training manager travels to the client's location for training.

Hiring: Today a few more candidates were interviewed. There are two recruitment companies with which our company has tie up. They provide candidates matching the profile as and when required and charge fees for that.
Sales: Today the sales team continued their discussion. There is a law firm that provides legal consultancy for a monthly fee on the tenders and to verify that the proposal meets all the legal requirements.

21-Aug-2012
10 AM: meeting as usual
Development: Today we had a scrum retrospective meeting. The last sprint that was completed and delivered was discussed to find points of improvement and the good things that should be followed again. It was decided to have a test team in place. The requirement for a test team would be raised to the management.
Finance Handling: There is an accounting company called 'Sundit'. They handle all the financial matters of the company like disbursing salaries, maintaining taxes, claim settlement of the employees. A person from Sundit comes to collect the entire payment bill, claims and other such documents on every Tuesday.

Sales Activity: as usual.

7.7 Interview Notes

Interview with Product Manager at EU Supply Holding AB:

• Could you please give a brief description of your role in the organization?
  I am the CEO of this company. My primary responsibilities are managing the overall business activities of the company, maintaining the relationship with our business partners and suppliers and focusing on the sales and marketing of our product.

• What are the most important business activities in the organization?
  Sales, Product development, human resource management, finance, customer support, infrastructure management..

• How are potential customers identified? How is the entire sales process conducted?
  EU Supply also sells its services directly to prospective clients in prioritized markets (other than in those markets where the partners are better positioned and own sales is not expected to give the biggest bang for the buck of own investment). The country managers are responsible for qualifying and developing client opportunities, perform any “shaping activities” (developing clarity on latent needs, helping clients seeing opportunities to meet such needs uniquely enabled by its services), negotiating direct deals where possible, and for submitting proposals to tenders. Own country managers are also responsible for delivery of implementation services and for retention of the clients.

• Are there any additional sources of revenue?
  The product has a lot of optional features which are either exploited in order to meet contracted feature requirements, or which may be agreed out of scope, but which the clients can later buy for an additional fee. Customers can also request for implementation of customized features specifically for their use, while all features are implemented as configurable options in the single main code line.
End users are always provided training directly or indirectly (train the trainer) upfront in any new implementation when they purchase the license, or later on whenever new features are released or new staff has started.

- Could you please describe the role of partners in the business?
EU Supply has sought partnership with best positioned distributors and business operators in prioritized geographic markets where the company lacks local credibility, sales force or “inside track” to win significant business without significant own investment. Operators earn the rights to resell the services from the hosted on the basis of revenues sharing. End user revenues are reported on a quarterly basis by the operator and they self invoice based on agreed percentage royalties/commission.

- Are there any activities that have been out-sourced to other organization? How do these support organizations contribute to the organization? What services do they provide?
EU Supply receives accounting and payment services from an accounting firm. This firm takes care of salary disbursement and other bill payments. An interim CFO is consulting for the company, and a full time CFO is being recruited at the moment. Tax consultants are involved only ad-hoc on demand.
EU Supply procures legal advice services from suitable legal firms in the relevant jurisdictions for its various proposal preparations and to take care of legal issues arising out of contracts. Different legal advisors are used in different markets.

- How do you manage the human resources in the organization?
EU Supply maintains a team of software developers, sales managers, support and infrastructure personal for various activities. The employees get a monthly salary in return for their services, and other benefits (free gym, mobile). Sales managers also receives bonus based on contribution (operating profit) from own markets/accounts. Line managers (and occasional recruiters) are responsible for recruitment of technical and managerial staff for the organization. Specialist recruiters are consulting for recruiting of developers, while other specialist recruiters are involved for recruiting of any manager in sales.

- Do you have any branch offices? What are their responsibilities?
Yes, we have sales office at Copenhagen and London. We have a customer support office at Tallinn, Estonia. The sales offices are responsible for the sales, submitting proposals to prospective clients and hunting for new customers and market. The office in Estonia provides customer support in English to all our customers.

- Could you please describe the software development activity in the organization?
A team of software developers work on adding new features and enhancements to the software product. It is together with sales the most important activities in the organization. Services of independent consultants are only contracted occasionally to meet significant peaks in product development required under new contracts. All the development activity (including that of any in sourced staff) is in-house development. There is no outsourcing of development activities.

- How are software and hardware procured for use in the organization? Who are the suppliers? How do they provide upgrade, maintenance?
Software vendors such as Microsoft provide the necessary software for development and support activities. Monthly license fee is paid by EU-Supply. Computer hardware vendors such as Dell provide most of the hardware required in development and infrastructure management. Hardware is purchased as and when required via distributors providing lowest rates. Occasional large scale purchases have been done cross border
in EU. The infrastructure team maintains and upgrades the software infrastructure to provide 24X7 accesses to the online systems.

- How is support and maintenance provided to customer during the license period?
A support team is the first point of contact for customers with technical and user issues. The company contracts to provide either only 2nd line support to super users of the client or client consortium, or 1st line support to different defined sets of users. EUS 1st line support provides the necessary guidance to customers and their bidders (third parties), or report system bugs if such are verified. EUS 1st line assigns any issues to 3rd line in case it is anticipated that the issue is a product issue. Third line support consisting of one of senior developers and 1-2 new staff in 3rd line, analyzes the issues, tries to replicate any issue, tries to narrow down the conditions for when the issue arises.

- How do you maintain security standards in the organization?
EU Supply has to maintain international security and environmental standards. These standards are important for the customers. EU Supply maintains an information security management system based on ISO 27001 and regularly calls for security audits from accredited bodies to maintain these certificates. The Information Security Forum (ISF) meets monthly and has standing terms of reference for its meetings, including review of any incidents, load projections, changes planned, ISMS maintenance plans and activities.

7.8 XML Serialization of the e3 value model
<InOutExchange name="InOutExchange3"/>
<InOutExchange name="InOutExchange4"/>
<InOutExchange name="InOutExchange5"/>
<InOutExchange name="InOutExchange6"/>
<InOutExchange name="InOutExchange7"/>
<InOutExchange name="InOutExchange8"/>
<InOutExchange name="InOutExchange9"/>
<InOutExchange name="InOutExchange10"/>
<InOutExchange name="InOutExchange11"/>
<InOutExchange name="InOutExchange12"/>
<InOutExchange name="InOutExchange13"/>
<InOutExchange name="InOutExchange14"/>
<InOutExchange name="InOutExchange15"/>
<InOutExchange name="InOutExchange16"/>
<InOutExchange name="InOutExchange17"/>
<InOutExchange name="InOutExchange18"/>
<InOutExchange name="InOutExchange19"/>
<InOutExchange name="InOutExchange20"/>
<InOutExchange name="InOutExchange21"/>
<InOutExchange name="InOutExchange22"/>
<InOutExchange name="InOutExchange23"/>
<InOutExchange name="InOutExchange24"/>
<InOutExchange name="InOutExchange25"/>
<InOutExchange name="InOutExchange26"/>
<InOutExchange name="InOutExchange27"/>
<InOutExchange name="InOutExchange28"/>
<InOutExchange name="InOutExchange29"/>
<InOutExchange name="InOutExchange30"/>
<InOutExchange name="InOutExchange31"/>
<InOutExchange name="InOutExchange32"/>
</E3BusinessModel>