Master Thesis

Documenting release changes in the functional architecture: An analysis of subsequent software releases

---

Author:

Mario Narwan

Student Number: 3802132

Course: Business Informatics

Mail: M.narwan@students.uu.nl

1st Supervisor:

prof. dr. Sjaak Brinkkemper

Utrecht University

2nd Supervisor:

dr. Jan Martijn van der Werf

Utrecht University

External Supervisor:

Wilmer van Alten

HSO

Faculty of Science

Department of Information and Computing Sciences

Abstract

Releasing a subsequent version of a software product imposes a challenge onto the delivering software company to successfully communicate their subsequent software release towards the users. In order to successfully communicate the software company has to use proper documentation for the product. This project proposes to use an architecture description technique to produce functional architecture models, which would describe every subsequent software product release; its functions and features. After the release settings of the case software will be determined, a proper modelling technique will be chosen. The modelling technique will use the internal structure of the case software to produce the functional models. The final functional models will be evaluate by the case company. The main aim of the evaluation will be to determine whether the functional modelling can be used as one of the methods of communicating a subsequent software release.

Table of Contents

Table of Figures

Supervision

Author

Mario Narwan
Joop Geesinkweg 901-999
1096 AZ Amsterdam
Student Number: 3802132
Email: M.narwan@students.uu.nl
Phone: 06 465 419 19

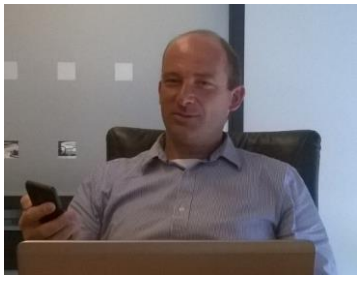
First Supervisor (UU)

prof. dr. Sjaak Brinkkemper
Universiteit Utrecht
Department of Information and Computing Sciences
Buys Ballot Laboratory, office 582
Princetonplein 5, De Uithof
3584 CC Utrecht

Second Supervisor (UU)

dr. Jan Martijn van der Werf
Universiteit Utrecht
Department of Information and Computing Sciences
Buys Ballot Laboratory, office 584
Princetonplein 5, De Uithof
3584 CC Utrecht

External Supervisor (HSO)



Wilmer van Alten
HSO, the results company
Newtonstraat 27
3902 HP Veenendaal

Chapter 1

Theoretical Background

1.1 Introduction

As the focus of this project is the new release of a software product, it is important to understand the process of a new software product release as well as its terminology. Xu & Brinkkemper (2005) define software product as follows: A software product is defined as a packaged configuration of software components or a software-based service, with auxiliary materials, which is released for and traded in a specific market." Based on the work of Ruhe (2005) and van den Akker et al. (2005) we define the software release as follows: "A software release is a collection of new and/or changed features that form a complete system that will be of value to the customer." It is important to distinguish between a release and a product version. Munch (1993) provides a simple definition of a software product version: "A version is a potential concrete instance of a software product." Therefore, it can be seen that the version is a subset of the release, and more often used within the company then outside, for example when communicating with the customers.

Any software product must evolve in order to stay competitive and productive. Siy & Perry (1998) argue that every successful software product must often evolve in several directions at once to maintain the increasingly diverse needs of the customer base; multiple versions of the same product each tuned to specific customers are evolved.

At the heart of this project is the Microsoft Dynamics AX, which is an ERP software solution from the Microsoft Corporation. Dynamics AX is a multi-language, multi-currency enterprise resource planning (ERP) business solution with comprehensive global business management features for financial, human resources, and operations management as well as additional industry capabilities (MS Dynamics, 2014). By combining comprehensive ERP with purpose-built industry capabilities, this single powerful solution delivers value across your business quickly, the agility to maximize opportunities in a changing market, and the simplicity to drive innovation—today and tomorrow (Microsoft Dynamics Resources Library, 2011). The Dynamics AX solution is usually meant for small to mid-size businesses, whose operations cannot be handled by a simple accounting system. AX enables the client to have his/hers system completely customized according to the specific requirements. On top of the complete customization of the system, AX offers specific capabilities within the following fields: Manufacturing, Distribution, Retail, Services and Public Sector.

1.2 Problem statement

Evolution of every software product is a complex process, which involves several challenges the company needs to address in order to be successful with the new release. Siy & Perry (1998) argue that the challenges of a new product release are to be able to fit the new features and functionality in the maze of already existing features. Furthermore, at any given time, multiple features are being implemented by multiple teams of developers. Not only is a developer concerned about unforeseen feature interactions between his team's feature and the existing customer base, he is concerned about undesirable interactions between his feature and those that are being made in parallel by his colleagues. As it was mentioned, software products are often customized to meet specific customer needs. These customizations further complicate the job of a developer working on a new feature. Not only does he have to contend with his colleagues working on other features in parallel, he has to make sure his team's feature fits into one or more existing customer “wishes”. At this point it is important to apply these principles into this research project.

Xu & Brinkkemper (2005) further argue that mapping market trends to software product design is becoming the central issue in new software product development. It is also important to use customer feedback during the product life cycle to enhance customer satisfaction. Usually most large scale software systems such as ERP systems, are developed

using existing software products. The software company can then decide whether it wants to do the implementation on the customer side themselves or use a business consulting company.

For the purposes of this single case study we will assume that the process of parallel development was handled properly by the development team as their exact development processes and techniques are inaccessible to someone outside of the company and even if it would be it is out of scope of this research. Furthermore, customizing of the software towards meeting a specific customer needs is being handled only after the customer decides to use the system and it is being installed on his/hers company's side. Dynamics AX as a system incorporates all the functionality of the ERP solution, which can be further customized for the customer during the introduction/installation process. Therefore, this is also out of scope of this research, which leaves us with what we are truly interested in and that is the documentation of the changes introduced in the new release and their impact on the current features and functionality as well as on the functional architecture.

This research constitutes of several main problems or issues, which need to be addressed:

- **Size –** the new release incorporates a large number of extensions and modifications and therefore there will be a vast number of documents and other information for examination.
- **Structure –** of the release documents (the ones being communicated to the partners) has to be organized in a logical way in order to effectively communicate the new implementations.
- **Clarity –** the release document has to be as clear as possible, so it is understandable even by a non-technical person.
- **Information access –** most of the documentation for this release is confidential and therefore accessing this information might prove difficult.
- **No analysis yet –** no analysis has been done yet on this new release, which could help or guide this research.

The diagram below illustrates the settings of this project and what the project is trying to achieve.

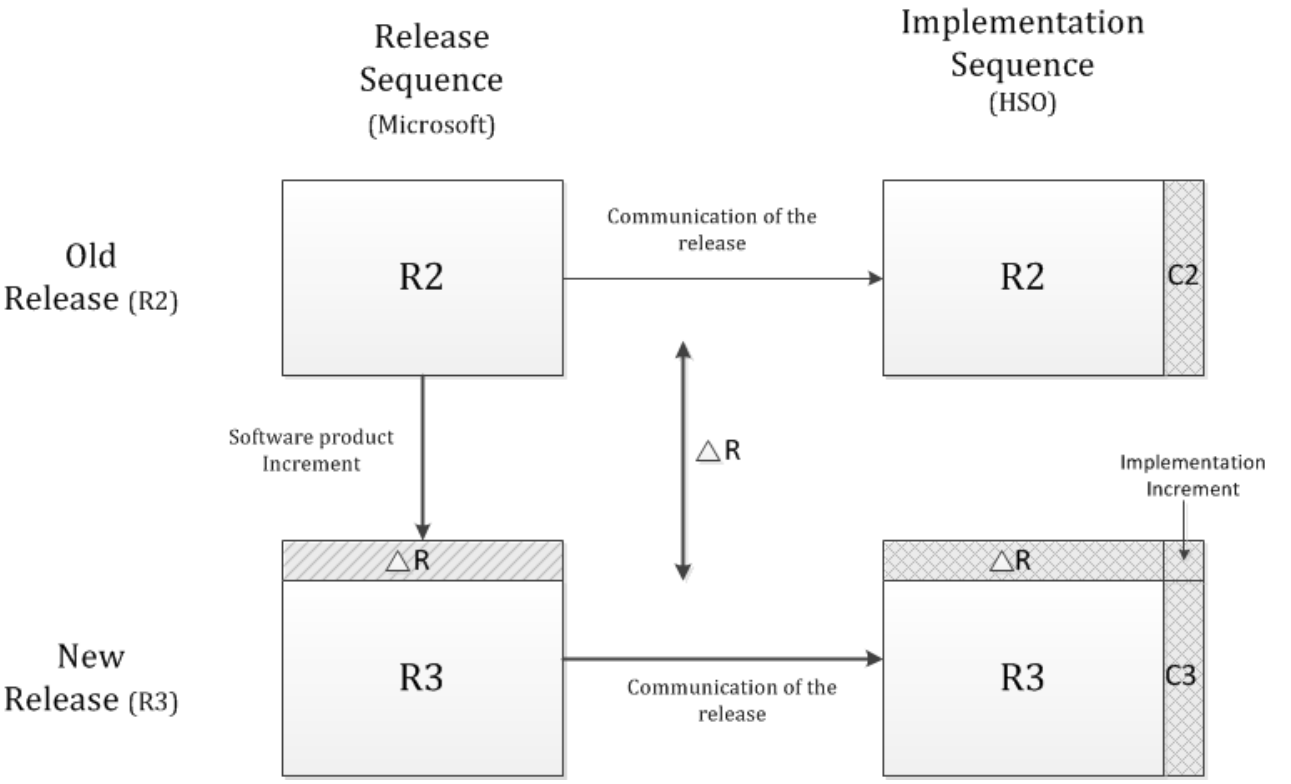


Figure 1 – Diagram of the research settings

The diagram is split into several parts: Release Sequence, Implementation Sequence and the Old & New product release. Release sequence illustrates the development process on the software development side (in this case Microsoft), whereas Implementation sequence focuses on the implementation partners, which use this developed software (E.g. HSO). Furthermore, the diagram is split into the old release (R2) and the new introduced release (R3). The transition from top to bottom means a change from old software release into a new one (update). The "shadow" of the R3 box illustrates the newly added or deleted functionality introduced in the R3 release. On the right side the released software is being implemented for customers by the implementation partners, with all the requested customizations (C2, C3). The delta (Δ R) depicts what is the focus on this research. First we want to address the changes in functionality from the old release (R2) into the new release (R3). But at the same time we want to focus on the "Communication of the release". How is the release being communicated? What kind of documents/diagrams or other information is received by the implementation partners?

Chapter 2

Research Approach

The following chapter will focus on the research approach of this project. First of all the main research question will be stated, supported by several sub questions, that will create the overall direction of this project. After a short explanation of these questions I will present the company that enables us to do this research on their system (HSO). Furthermore, a research model will be presented, which will illustrate the different processes this project is going to undertake as well as their order and interaction between one another. The most important literature and the literature review approach will be described, together will some already identified potential limitations and challenges of this upcoming project.

2.1 Research Questions

Based on the problem statement several questions need to be created in order to guide the entire research project. Each of the research questions will focus on a specific part of the project, which will split the project into logical sections. Afterwards each of these sections is going to be addressed in the proposed order of the research questions. Answering of these questions will provide an insight into the changes proposed by the new Dynamics AX R3 release, but more importantly it will show what kind of impact these changes have on the system as a whole. As Siy & Perry (1998) argue, the challenge of a new product release is to be able to fit the new features and functionality in the maze of already existing features, while not breaking or damaging the already existing functionality. Due to the fact that this release (AX R3) is only a few months old (officially released on the 1st of May), this will be one of the first analysis to provide an inside look on the changes and their incorporation into the existing system and its architecture. As it was mentioned on the beginning this whole project and all of the research questions will be focused only on the newly introduced Warehousing

functionality. Other newly introduced parts of the R3 release will be omitted (or considered to be constant) for the purposes of this study.

These are the research questions designed to guide and provide answers for this project:

> ***Q1: How are functional architectures of software products modelled?***
> ***Q2: What categories of changes are introduced to the functional architecture by the new Dynamics AX R3 release?***
> ***Q3: How are new releases being communicated to the implementation partners?***
> ***Q4: What notation should be added for modelling of release changes in functional architectures?***
> ***Q5: What are the changes in the functionality and process flows of the architecture?***

The first question (Q1) will serve as a background for this project. It will provide the theoretical knowledge behind the different software architecture modelling techniques. These techniques are going to be used to model the different functional architectures of the Dynamics AX system. These techniques are standardized and therefore a need for a new notation might occur while modelling the different parts of Dynamics AX. To answer the second question (Q2) all of the new functionality has to be documented and categorized in a logical fashion (e.g. tables) in order to gain an overview of the changes introduced with the new Dynamics AX R3 release. This data will serve as a base for answering of the following research questions. Q3 focuses on the way the new releases are being communicated to the implementation partners. This will show us whether the new releases are being communicated in an efficient and clear manner or if they could be improved using the functional models. The fourth question (Q4) is concerned with any notation, which has to be introduced for the modeling part and is not yet present in the standardized modelling languages. For example when a module is deleted (taken out of the system by the new release); How is this change going to be displayed by the model? (Colors, text or other notation) The fifth question (Q5) focuses on the changes in the functionality of the system, for example the introduction of the handheld scanning devices will force a major content change for the Dynamics AX in order to enable the usage of the new scanning devices. Moreover, the changes in the functional architecture will force the system to work with

different modules and contents and therefore the process flows of the system will change as well. The process flows are going to be modelled using the Software Architecture modelling techniques in order to illustrate the functioning and behavior of the system. Using the data and models created for Q1, Q2, Q3 and Q4 the sixth question (Q6) will focus on the impact of the new release changes onto the existing functional architecture and their interaction with the current features and functionality. This part is considered to be the main analysis part of the project, from which the main conclusions are going to be drawn. Using the Software architecture techniques for modelling, the answers to these research questions will provide us with an insight into the new R3 release and its effects on the entire Dynamics AX, furthermore this will be one of the few analysis made on this topic, which might prove invaluable for the partner companies (which are using Dynamics AX) as well as the clients themselves.

2.2 Case company - HSO International

The entire project is supported by the HSO Company (more specifically the HSO International), which is a Microsoft partner focusing on ERP solutions by using the Microsoft Dynamics AX.

"HSO is a leading provider of innovative enterprise business solutions that improve the results of our clients. Founded in 1989, HSO specializes in implementing, integrating, optimizing and maintaining enterprise solutions based on Microsoft Dynamics AX, CRM and Office365. We are working with multinational enterprises in industries such as Distribution, Retail, Manufacturing and Service." (HSO, 2014)

HSO uses the Dynamics AX to satisfy the different customer needs with a customized ERP solution designed individually for every one customer, which makes this into a very flexible solution.

HSO will play a major role in achieving the correct literature from the field of the new R3 release and any other models, figures, tables or other data that will help with the documentation of the new features and functionality, and modelling of the functional architecture. The literature review for this project is different than in an average research project as most of the materials are internal and only accessible from within the intranet of HSO or in some cases it can even be confidential, which renders the usage of this information difficult. Using the HSO partner network access it will also be possible to access the internal documents, created by Microsoft itself, which hold valuable information about the new R3 release. Furthermore, as the thesis working environment will mostly be within the HSO

Company, there will be a constant communication with the employees/colleagues, which will in itself serve as a great source of literature, information as well as quality control over the thesis project. Finally, HSO will play a major role when it comes to the analysis part of the project. The Dynamics AX experts from within HSO will take part in evaluation interviews, where the new changes and their impact on the existing system will be judged, leading to the final analysis conclusions.

2.3 Case Software product – Microsoft Dynamics AX

The new R3 release of the Dynamics AX incorporates several major upgrades of the system, however for this project we will only focus on the new warehousing management functionality (WHS). This module was adopted from a 3rd party, company which is a partner of Microsoft and as a module it is now completely integrated into the Dynamics AX ERP system. The new Warehousing Management module provides the company with great flexibility and automation of the warehouse processes as well as it provides customers with more choices on item selection, which leads to reduced operational costs that eventually translate into reduced prices for the customers. Warehouse managers gain visibility and the ability to effectively control and run their put away, quality control and picking warehouse operations. Together with the flexible warehouse workflow engine, warehouse managers will be able to rapidly respond to any changes in the business demands (MS Dynamics AX 2012 R3, 2014). The new WHS features include: Workflow creation and management, Cluster picking, Wave setup and processing, Cycle counting (quality assurance), Containerization, Reservation hierarchies, Inventory blocking and most importantly it incorporates new functionality for the usage of mobile handheld devices used for scanning and manipulating of the warehouse items. All of these functions are going to be described in detail in the future chapters.

As it was mentioned on the beginning the main focus of this project is the Microsoft Dynamics AX, more specifically the newest release of this product called MS Dynamics AX 2012 R3. This new release of software was first introduced on the annual Microsoft Dynamics AX Technical Conference on the 3rd – 5th of February, which took place in Bellevue, Washington (Dynamics AX Technical Conference, 2014). After this introduction the software product had a Virtual launch on the 10th of April 2014 and was later officially released on the 1st of May 2014 (Microsoft Dynamics AX Blog, 2014). The newly introduced release (R3) incorporates several large updates from Retail functionality through Master Planning to Transport & Warehousing Management. For this project we will be focusing on the new Warehousing

functionality, as it is a major change in this release with a lot of new functionality, which can and will have a large impact on the system as a whole.

As the warehousing module is brand new (officially released on 1st of May 2014) the amount of literature available on this topic is very limited. Most of the literature will come from within the HSO Company, which however means some of the documents might be confidential and the usage of that information might prove difficult. Furthermore, as this topic is very fresh there is almost no data or information when it comes to any kind of analysis on the topic of functionality, features and functional architecture; this will therefore be one of the big challenges of this project.

2.4 Research Design

The first step of this project will be to establish a theoretical background, by reviewing the current architectural modelling techniques (architecture description languages - ADLs). This literature review will be based on papers such as Kruchten (1995), which incorporate the current officially established modelling languages, with standardized notations. Several methods are available in this area and therefore they will be described in short, together with their advantages and disadvantages, in order to judge which of these methods would be the most suitable for this project's modelling. The background information gathered with this literature review will be used to model the different functional architectures of the Dynamics AX (with different levels of abstraction), using the standardized notation where possible; however there might be a situation where new notations will have to be introduced.

The above scientific literature review is only one half of the information and documentation gathering, the second one is aimed at documenting the changes introduced in the new Dynamics AX R3 release. The new R3 release is a major update for the Dynamics AX software, and therefore it is important to document the changes in an intelligent way, so it can be used later on for modelling purposes. To gather the release information a few specialized databases will be consulted: Microsoft Dynamics Resources Library, Dynamics AX Developer Center and Microsoft Technet. After extracting of this information from the databases, they need to be organized or categorized in a logical fashion, so they are easy to manipulate (e.g. descriptive tables). The topic of this new Dynamics AX R3 release is brand new and therefore any available literature is very limited, which means accessing many information sources may prove difficult, due to confidentiality, other security reasons or simply due to the fact that they do not exist yet. Due to this fact, this documents review will be supported by the snowball literature review technique. The snowball technique as described by Ang (2014) is

based on taking a small number of articles, journals or other scientific literature and then to gather more literature based on the references and citations of the original small number of articles. The process then repeats itself, but now the "original set" becomes larger by the amount of reference/citation articles gathered by the first round. It is important to note that the number of articles can get quite high if not handled properly with a good screening process. Another part of the snowball review will be to do a forward tracking of the citations of the original articles as described by Webster & Watson (2002). A citation database (e.g. Web of Science) will be used to track the citations of the original set of articles. Furthermore, the articles which use the "original set" as a reference (the original set is cited) are reviewed for additional information gathering.

After a new Dynamics AX release is created it is then communicated to the partner companies, which are using this software. Therefore, it is important to examine the way these releases being communicated to the partners and if they are getting all the appropriate information. Not only do they need to receive all the information about the new release, the information has to be organized in a logical fashion, so it is easily understandable and no problems or misunderstandings are introduced. An interview with a few consultants will be conducted to determine the way they receive information about new releases, as well as the structure and content of these information.

The scientific literature review will serve as a basis for modelling of the functional architecture of the different parts of Dynamics AX (with a focus on warehousing modules). The models will try to use the standardized notations described by the literature, however if it will become necessary, new notations will be created in order to describe the changes in the functional models. Any newly introduced notation will be documented together with a description. After it has been agreed on which notations are going to describe what, the functional models will be created. From the beginning it will start will a high level model of Dynamics AX with the most important modules (Sales, Orders, Innovation, Planning and so on), in order to create an environmental setting for the warehousing module. Afterwards, this model will be followed by several lower level models, which will focus on specific parts of the warehousing modules in order to display and describe the newly introduced changes to the functionality. The number of models is one of the challenges of this project, too many models can result in a confusing display, but on the other hand too few models can lead to incorrect or bias information; the correct balance of the number of models will be decided based on the information gathering.

The functional models will display all the newly introduced changes to the system (additions, deletions or modifications) by using either the standardized or newly created notations. These models will serve as a basis for the impact analysis. At the analysis stage of the project I will try to judge the changes made to the functional architecture and the functions in general, using the descriptions from the functional models created earlier. Every new addition of deletion of a module will be closely examined and I will try to determine the effect of this one new addition or deletion has on the system. The analysis will try to address each single change individually to avoid missing any important effects they might have on the system. The aforementioned analysis will present the different changes and the effects that these changes have on the system as a whole; however it is crucial to evaluate this analysis in order to judge how accurate or reliable it is. Due to the lack of literature or any kind of evaluation for the newly introduced Dynamics AX R3 release the evaluation method will reside in expert interviews, which will be conducted with experts from the HSO company as well as an outside company (which still uses Dynamics AX), in order to provide more expert opinions from different settings. The expert interviews will consist of a presentation of the analysis (to create the environmental settings), followed by an evaluation from the experts' point of view combined with a discussion, based on which the appropriate conclusions will be made.

2.5 Research Model

Below is a figure of the research model, which is going to be used for this project in order to answer the research questions. All of the main processes are depicted on this model together with the flow of information throughout these processes.

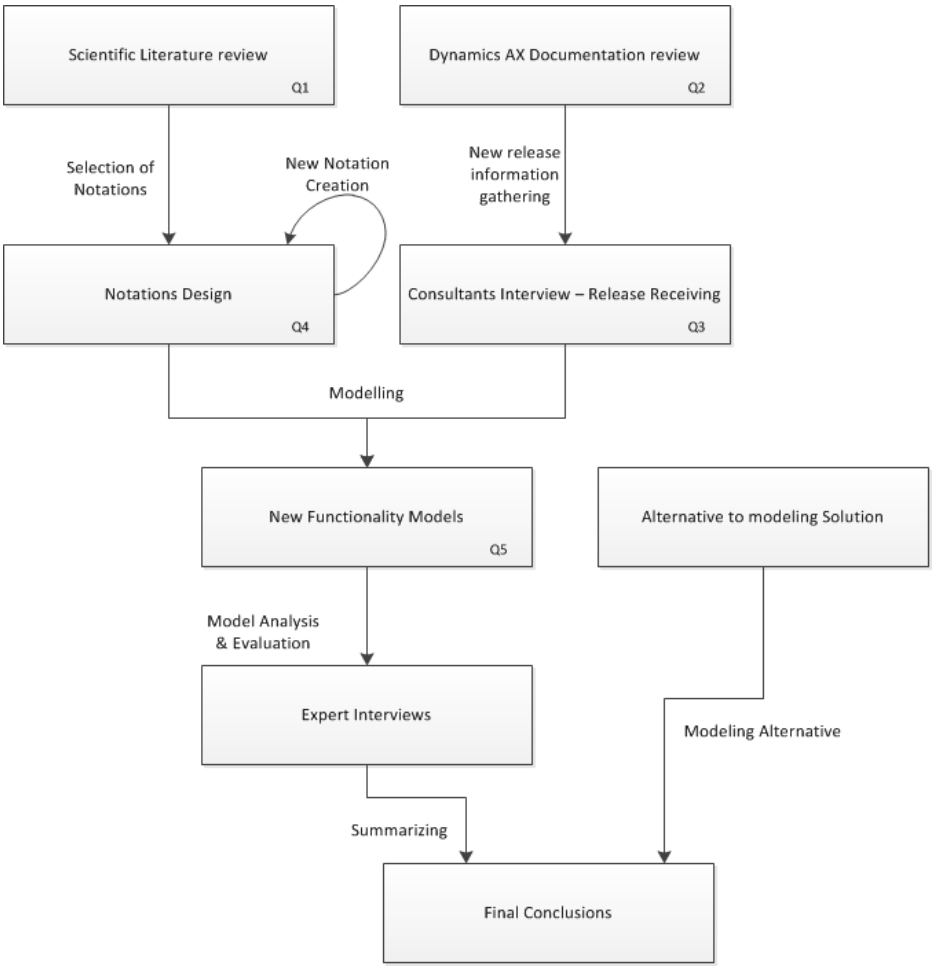


Figure 2 – Simplified Research Model

The research model begins with two processes: Scientific Literature review and Dynamics AX Documentation review, which can be executed in parallel as they do not depend on one another. These two processes will create the theoretical background for the whole project. To be able to model the newly introduced functionality it is necessary to choose the correct notations. These notations are either chosen based on the scientific literature or new notations have to be created in order to appropriately describe the new functionality (Notations Design). The information gathered from the Dynamics AX documents will be used

to setup an interview with a few consultants about the way the new releases are being communicated to the consultants. By combining the chosen notations together with the information gathered using the Dynamics AX documents as well as the consultants' interview, we create a knowledge base for the functional models. Using this knowledge base several functional models of different levels of abstraction are created. These models focus on the newly introduced warehousing module, both from the external and internal point of view. The most important part of these models is displaying the changes (additions, deletions) being introduced to the functional architecture. The models will be presented to several expert consultants, who will provide their expert opinions on the new changes and their effects and implications for the system. The expert consultants will mainly be from within the HSO Company; however several outside consultants (either from Microsoft itself or from other consultancy companies, which use Dynamics AX) will also be invited for the evaluation process to provide a different perspective towards this topic. Furthermore, during the initial work on the project I received an inspiration to explore an alternative solution to the modelling techniques; and this will therefore be another part of the project's solution, which will be presented during the expert evaluation as well. The expert evaluation will be the final step of this research project based on which the main conclusions and answers to the research questions will be provided.

2.6 Relevance

The following chapter focuses on the relevance of this research project; how useful are the findings going to be and their applications and implications for the future research, the Dynamics AX system itself and the companies that use this ERP solution.

Scientific Relevance

This project is mainly based on modelling of the functional architecture of the Dynamics AX, more specifically the Warehousing functionality. Therefore, there will be several models depicting the old Warehousing architecture as well as the new addition to the Warehousing functionality, introduced in the new R3 release. The scientific relevance of this project will be based mainly on the changes that occur within the functional architecture. How is the old functionality improved (software increment) by the new additions? How will they interact with the already existing modules? Are there any problems or issues introduced with the new release?

Using this information we will gain valuable insights into the process of a software product increment and its effect on the current system. Moreover, this research is based on a brand new product increment (Dynamics AX R3), which makes the project very current and therefore the knowledge gained will be up-to-date and of a high relevance.

Industrial Relevance

The focus of this study is the new release of Dynamics AX, the R3 release, which has only been released on the 1st of May 2014. Meaning the literature on the topic is very limited especially when it comes to any kind of analysis. Therefore, this analysis project will have a high social and business relevance as it will provide one of the first insights into the new Warehousing functionality and its behavior within the current Dynamics AX environment. The results of this research can help the company implementing Dynamics AX solutions (e.g. HSO) for clients, as it will provide important information about the internal functioning of the new modules (Warehousing) and their "position" or interaction with all the current modules. Moreover, it can provide, guide and help the customers with the understanding of the new functionality, which will eventually translate into easier use and higher customer satisfaction.

2.7 Challenges and Limitations

As it was mentioned in the introduction chapters the topic of the new Dynamics AX R3 release was only official released on the 1st of May 2014 and therefore one of the major challenges for this project is to gather any kind of literature related to this topic. The main sources of information in this case will be specialized databases of Microsoft and its related partners. However, the accessibility of this data will prove difficult as many of the official documents are either confidential or do not exist from the start.

The second part of the literature review (the scientific one) also poses a challenge for this project, which is to choose the correct modelling method and notations. As this field is quite mature, the literature review itself does not pose a challenge. However, there are many different modelling methods, which are all being regularly used in practice and therefore the challenge lies in choosing the correct method with the appropriate notations. It is possible that no one method will be sufficient enough for modelling of this project, in which case new notations will be created (and documented) to describe the models. After the correct method and notations are chosen another challenge arises, which is how many models to model and what is the level of abstraction that should be used. Too many models of different parts of

Dynamics AX might prove too confusing for the reader and render the analysis and evaluation part much more difficult. The correct number of models and their respective abstraction has to be chosen in order to cover all the new functionalities, but at the same time not to include any redundant or repetitive data. Finally each of the models should be logically linked to each other (no gaps or black boxes).

The evaluation method chosen for this project is the use of expert interviews to judge the new changes and their effects. The limitation is that due to the fact that this topic is brand new there are not many possibilities for an evaluation method. There are no standardized tests or analysis tools or any other kind of framework for evaluating of this new release. The expert interviews will provide the appropriate data for evaluating of the functional models, however they are only limited to the extent of the knowledge of the experts being interviewed. A part of a future work should include a more sophisticated method of evaluation.

2.8 PDD

The following section focuses on the process deliverable diagram. The research model created earlier serves as a basis for this diagram, which depicts the different processes of this research, but in a more detailed way than the earlier research model. The PDD follows the principles of Method Engineering and displays all the important processes as well as their corresponding deliverables.

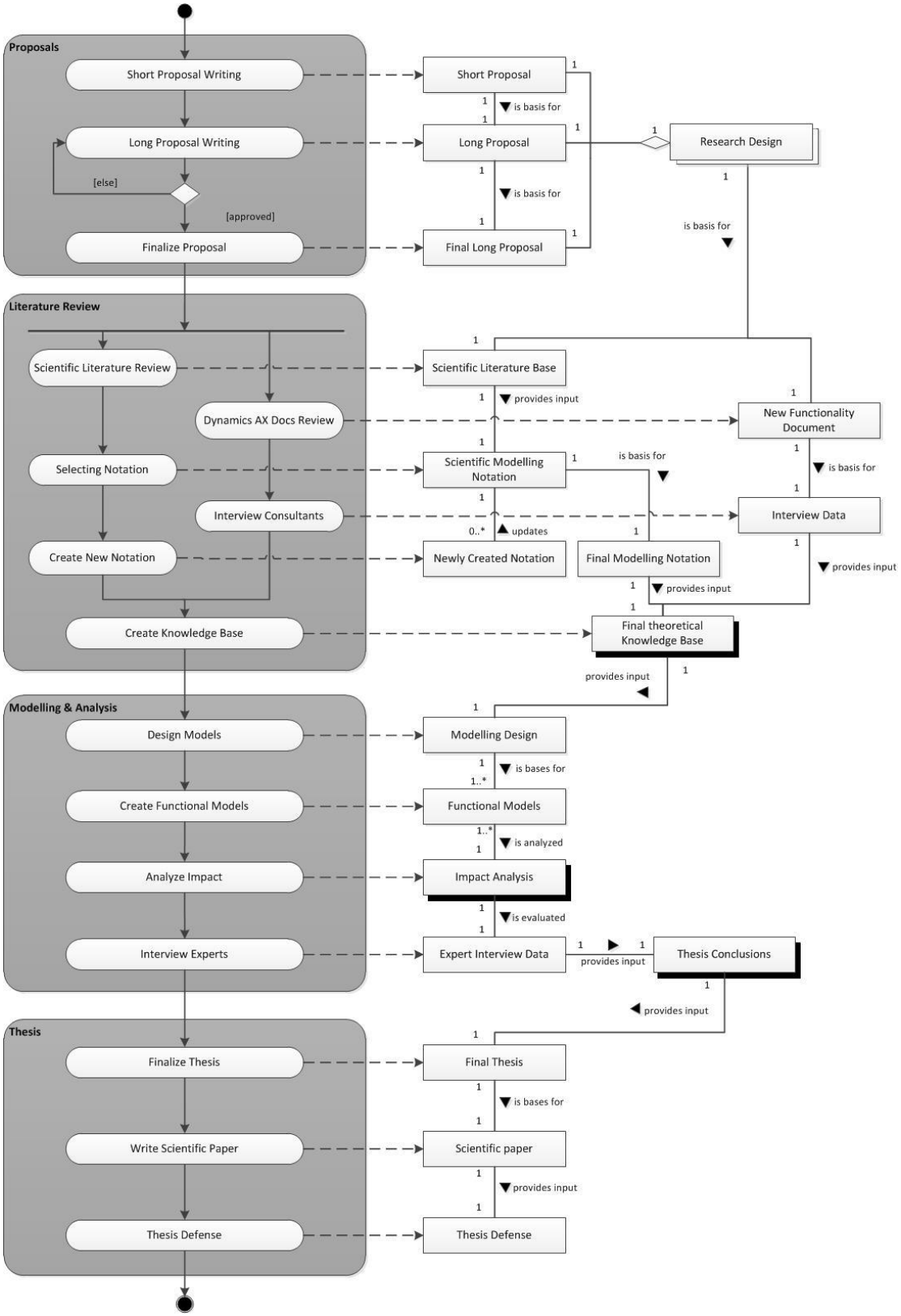


Figure 3 – Process Deliverable Diagram for this research project

2.9 Project Plan

The following table describes all the important deliverables, which are going to be a part of this project together with their respective dates of completion.

Week	Date	Deliverable
1	07.07.2014	Start of graduation project
2	14.07.2014	First Project definitions
5	04.08.2014	Short proposal
6	11.08.2014	Redefining of the project
7	18.08.2014	New Short proposal
8	25.08.2014	Introductory literature review
11	15.09.2014	Long Proposal
13	29.09.2014	Long Proposal Feedback
17	13.10.2014	Final Long Proposal
19	27.10.2014	Literature reviews
23	24.11.2014	Notation Design
25	08.12.2014	Consultant Interview
31	12.01.2015	Application architecture review
35	09.02.2015	First Colloquium presentation
37	23.02.2015	MS engineer interview
39	09.03.2015	Functional modelling
46	04.05.2015	Impact Analysis
49	25.05.2015	Expert Interviews
51	15.06.2015	Preliminary results
55	13.07.2015	Second Colloquium presentation
56	20.07.2015	Final Version of the Thesis
56	20.07.2015	Register graduation date

Table 1 – Thesis Project Plan

Chapter 3

Architecture Description Languages

The following chapter will focus on the different architecture modelling techniques or languages, which are going to be used later on in modelling of the functional architecture. This chapter is mainly based on literature review focused on Architecture Description Languages (ADLs). Clementes (1996) defines Architecture description languages (ADLs) as formal languages that can be used to represent the architecture of a software-intensive system

3.1 Modelling Language principles

According to Kruchten (1995) it is important to know which model or formal representation focuses on which part of the architecture. Kruchten (1995) defines four types of architecture views, which split up the entire software architecture into more manageable parts: Logical, Development, Process, Physical view and finally they are supported by the scenarios (use cases). In this project we are only interested in the functional architecture of a software product, which is in this case the Logical view. Kruchten (1995) describes the logical architecture as primarily supporting the functional requirements—what the system should provide in terms of services to its users. The system is decomposed into a set of key abstractions, taken (mostly) from the problem domain, in the form of objects or object classes. The logical view will enable us to see all the functions and features of the software release as well as their connections, which will help us in identifying the changes and possible issues in the new software release.

There are many techniques that address the issue of architecture modelling. However, only one can be used to model the functional architecture for this project. Therefore the next part of this project will address the evaluation and choosing of the most appropriate technique for the task at hand. The modelling techniques have to be evaluated according to several criteria, which also address the specifics of this project. First of all, the most common criteria have been chosen from the scientific literature; followed by an expert interview, where we

looked at these literature criteria from the point of view of the project itself. This way we are using the scientifically valid principles in the context (scope) of our own project.

As cited by Cheng (2004): According to Ross and Schoman (1977) any modelling technique is characterized by the following four principles:

- **The purpose of the model:** Modelling must have a final goal in mind. This creates a direction and a framework for the process of modelling.

- **The scope of the model:** This is the environment in which the model is situated, defining the covered range or domain.

- **The viewpoint of the model:** The viewpoint defines which parts of the system (or organization) will be included in the model and which parts are going to be omitted as well as the viewpoint from which the modelling is drawn (see Kruchten 1995).

- **The detailing level of the model:** This defines the level of precision or granularity of the model regarding the reality being modelled. It depicts the level of detail that will be displayed in the models. Different degrees of detailing provide different levels of model flexibility.

These basic principles are extended by Vernadat (1996) as cited by Whitman (1999) and Cheng (2004), using the company (enterprise) modelling concepts:

- **Principle of separation of concerns:** Separating the model into logical parts, each part focused on a certain department or logical area. Decreasing the models' complexity.

- **Principle of functional decomposition:** A principle which revolves around splitting up functions into more specific and detailed parts, which put together make up the parent function. Closely related to the Model Detailing levels.

- **Principle of modularity:** Defines the building blocks of a model, the components which put together define the entire model (functions & features of the model). This is one way of dealing with enterprise complexity, and it makes the models easier to manipulate and maintain.

- **Principle of model genericity:** There are many functions and activities which support many different functionalities; however they usually have a certain bases in common. It is therefore advantages to create general classes on functions, which may later be specialized in the desired fashion. Further reduces the model complexity.

- **Principle of reusability: "**To reduce modelling efforts and increase model modularity, predefined building blocks or partial models must be reused and customized to specific needs as much as possible when modelling new parts of the system. This is referred to as ease of customization.**"** (Cheng, 2004)

- **Principle of separation of behavior and functionality:** The model needs to distinguish between the functionalities and the behavior of a system. An alteration in either of these views must not trigger a change in the other.

- **Principle of process and resource decoupling:** According to this principle the model has to separate the actions being performed (business functions) and the agents performing these actions (resources).

- **Principle of conformity:** This principle deals with the accuracy of a model. How well does the model represent the given system or situation using semantics and syntax?

Furthermore, Vernadat mentions the following principles gathered from Ward and Mellor (1985) as cited by Whitman (1999):

- **Principle of model visualization:** This principle addressed the ability of the model being easily communicated towards the stakeholders.

- **Principle of simplicity versus adequacy:** The modelling technique should be easy enough to for the users to learn, but at the same time complex enough to express everything that needs to be expressed.

- **Principle of management of complexity:** This principle describes the ability of a model to display entities of great complexity.

- **Principle of rigor of representation:** The modelling technique should not incorporate ambiguity or redundancies in its models.

- **Principle of separation of data and control:** The technique should have the ability to separate the data used by a process and the events that control that process.

All the above principles were gathered via literature search and review. However, they all need to be further evaluated in order to find you if there are suitable for our project. To achieve this we have scheduled an interview with an expert from the field (a consultant), with whom we have examined the above principles and chosen the most appropriate for our project. The final chosen principles will be used to evaluate and pick the most appropriate modelling technique.

At the beginning we have the four basic principles of scope, purpose, viewpoint and detailing. The scope and the purpose of a model describe the focus and field of the modelling technique and will therefore be used as the first evaluation criteria for choosing of the appropriate technique. The scope and purpose of each technique will be examined and based on how suitable a technique is for this given project it will be included or excluded from the final evaluation. These criteria are not going to be used in the final evaluation. The last two basic concepts are the detailing and the viewing point of the model. The viewpoint itself has already been chosen (the warehousing module) and therefore will not be used for further evaluation. The level of details of the models will be decided during the modelling phase (therefore not included in the evaluation) and will be based on the desired level of complexity for the models.

In the next part of the interview with an expert consultant we have examined the concepts proposed by Vernadat. First and foremost concept which had to be included is the Model visualization, which describes the understandability of a model. As it was mentioned several times, one of the main aims of this project is to create models, which are easy to understand even for non-technical staff. Therefore model visualization will be one of the most important evaluation criteria for our modelling techniques. Closely related to this concept is the principle of Conformity, which addresses the issue of how well does a model, represent the desired information; in other worlds: “How precise a model is?” The conformity will serve as another evaluation criterion. Another aim of this project is to make the models easy to use, in order for them to be effective in communicating a new software release. The usability is closely related to the simplicity versus adequacy principle as described by Ward and Mellor (1985), cited by Whitman (1999). Using the functional models we not only want to represent the current software structure of the system, but the newly introduced changes as well, which brings us to the concept of Extensibility. Each of the architecture description languages is useful in representing of the current structure; however none of them deal with

representing of the changes in the architecture. Therefore, the methods themselves need to be extended by new notations, which will handle the representation of the newly introduced changes. The technique which can accommodate the new notations the easiest is the most desired technique. Last but not least is the evaluation of the models. The final chosen technique should provide models which are going to be easily analyzable by the standard architecture evaluation techniques (e.g. ATAM or SAAM). The analyzability of the models will be the final evaluation criterion for choosing of the appropriate modelling technique.

Architecture Description Languages evaluation criteria:

- **Usability:** *The modelling technique should be easy enough to for the users to learn, but at the same time complex enough to express everything that needs to be expressed.*
- **Conformity:** *The ability of a model to accurately display the given system architecture.*
- **Model visualization:** *The ability of a model being easily communicated towards the stakeholder.*
- **Extensibility:** *The ability to easily accommodate new modelling notations.*
- **Analyzability:** *The modelling technique should be analyzable by the standard architecture evaluation techniques.*

3.2 Chosen modelling techniques

As it was mentioned in the evaluation criteria section the first step towards the technique evaluation is to examine the scope and purpose of the techniques. This is going to be a pre-evaluation step where we chose the most suitable technique from a number of possible modelling techniques found during the literature review phase. The main reason for this pre-step is to match the purpose and scope of a technique to the ones of this project.

These are all the technique found using the literature review and included in the pre-evaluation phase:

Name of a technique	Authors	Year of first publication
Aesop	Garlan, Allen, & Ockerbloom	1994
Architecture description interchange language	Garlan, Monroe, & Wile	2010
ArTek	Terry, London, Papanagopoulos, & Devito	1995
Functional Architectural Models	Brinkkemper & Pachidi	2010
Functional Flow Block Diagram	TRW Inc.	1950s
IDEFx	Menzel & Mayer	1998
Rapide	Luckham, Kenney, Augustin, Vera, Bryan, & Mann	1995
Systems Modeling Language	Friedenthal, Moore, & Steiner	2014
Unified Modelling Language	Rumbaugh, Jacobson, & Booch	1997
Wright	Allen & Garlan	1994

Table 2 – Architecture description languages summary

First we need to examine the scope and purpose of this project and match it to the most suitable technique. Architecture modelling can be viewed from several perspectives as described by Rozanski & Woods (2011) in their architecture viewpoint model (Figure 9).

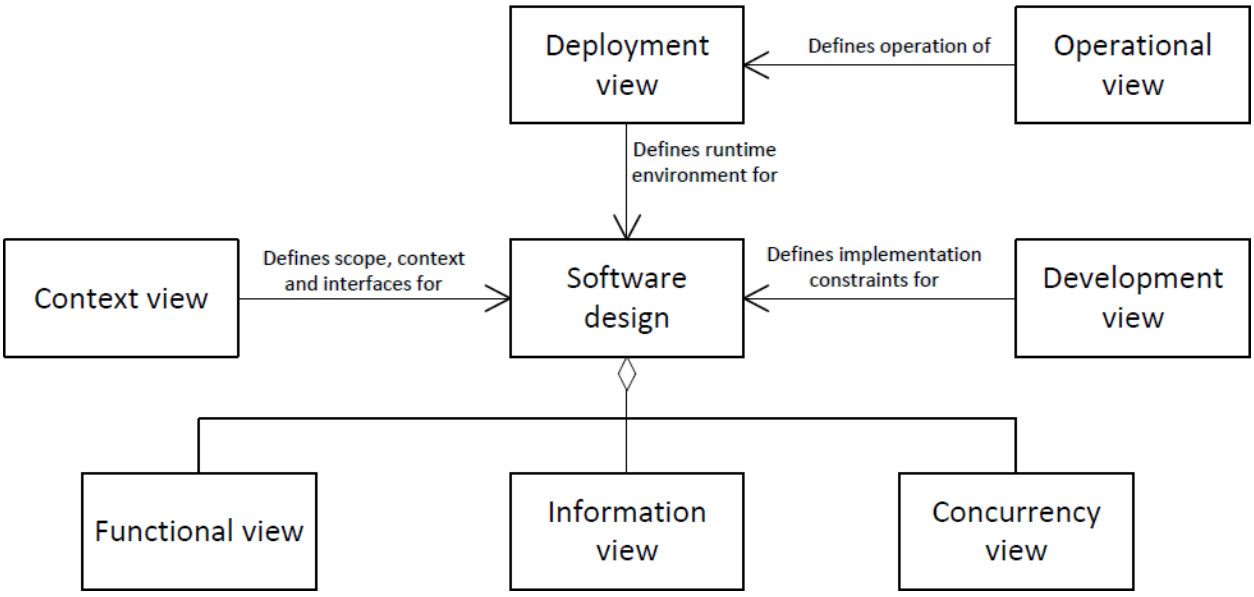


Figure 4 - Rozanski & Woods (2011) architecture viewpoint model

When considering the above model and the fact that this project is aiming to model the functions and features of a software system, we consider the "Functional view" as our scope. The following techniques were chosen as the most suitable in modelling functional architecture views:

- **Functional Architectural Models (FAM)**
- **Functional Flow Block Diagram (FFBD)**
- **Systems Modelling Language (SysML)**
- **Unified Modelling Language (UML)**

Below is a short description of all the evaluated techniques.

Functional Architectural Models (FAM)

According to Brinkkemper & Pachidi (2010) the Functional Architecture Model (FAM) includes all the necessary modules and structures for the visualization of the Functional Architecture of a software product and its relevant applications in the business domain. The models resulting from this technique are mainly focuses on the stakeholders like customers, marketing, sales employees or end-users. Overall they should be easy to understand by all the stakeholders groups. The functional models reflect a software product from the usage point of view; such models should resemble the functions performed in the individual user context. Most of the notations are borrowed from the Enterprise Function Diagrams.

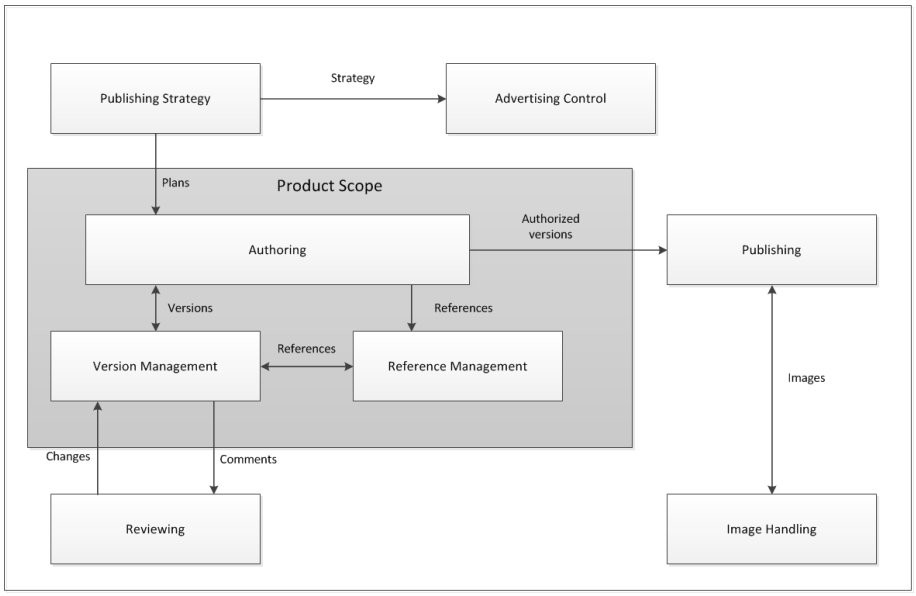


Figure 5 – Example of a Functional Architecture Model (Brinkkemper & Pachidi, 2010)

Functional Flow Block Diagram (FFBD)

"Functional Flow Block Diagram portrays the sequential relationships among functions at each given level, and provides a framework for deriving performance requirements for the system and/or all subordinate system elements." (Davis, 2005). The first Functional flow block diagrams were favored by the systems engineers, but they continue to be widely used by others even today. The main trait of this technique is that each function is represented by a block (module) which has certain inputs, outputs, support and control flows. Furthermore, the technique provides a multi-layer system representation, which enables the vertical traceability through the levels. Blanchard & Fabrycky (1990) state that the FFBD does not contain any information relating to the flow of data between functions, and therefore does not represent any data triggering of functions.

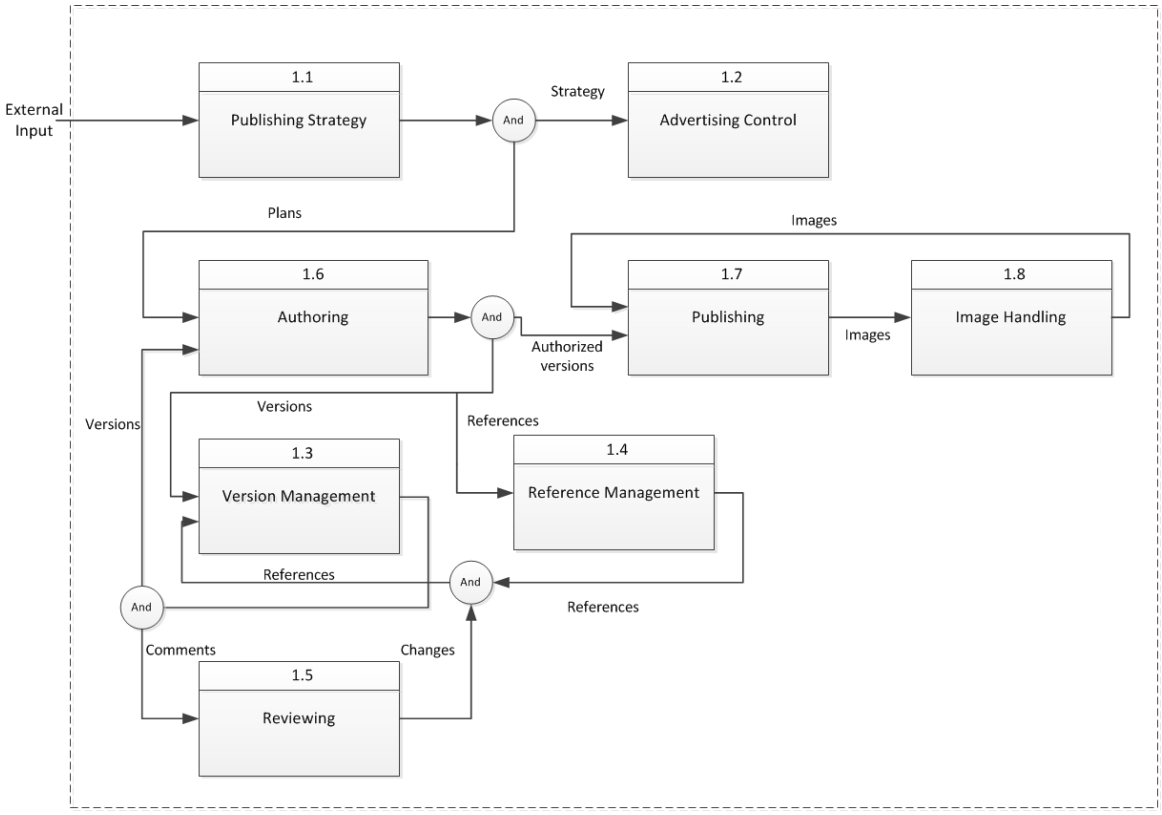


Figure 6 – Functional Flow Block Diagram (FFBD)

Systems Modeling Language – Block definition diagram (SysML)

Friedenthal, Moore & Steiner (2014) argue that SysML is a general-purpose graphical modeling language that supports the analysis, specification, design, verification, and validation of complex systems. SysML is defines as an extension of the Unified Modelling Language (UML), using the UML mechanism. Weilkiens (2011) argues that while UML is a strong modelling language, it has some shortcomings when it comes to systems engineering, such as requirements modelling. Moreover, UML is rather software-oriented, whereas SysML is interdisciplinary. This technique incorporates several types of diagrams from behavior towards the structural diagrams; from which only the Block definition diagram is of interested for this project.

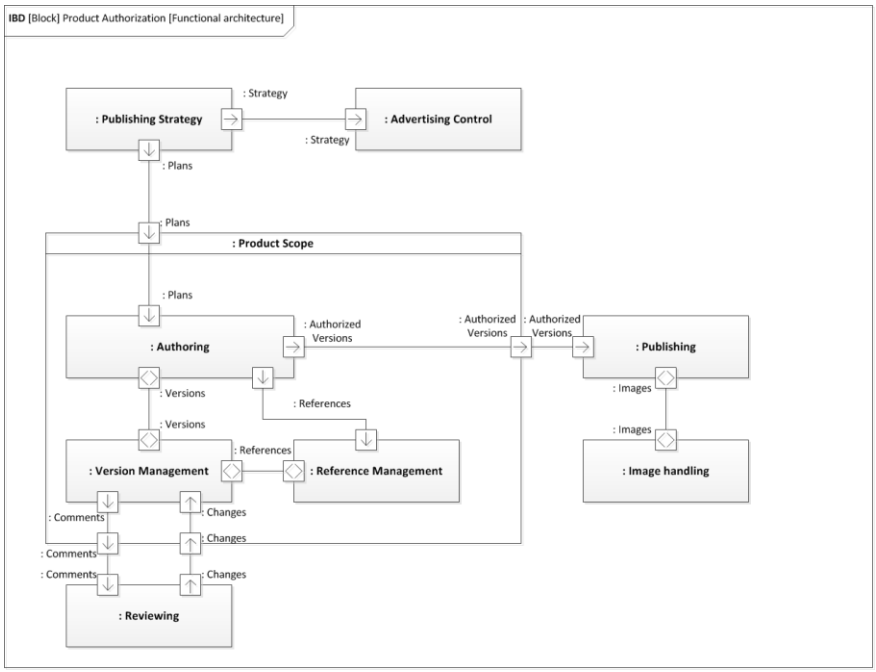


Figure 7 – an Example of a Block definition diagram SysML model

Unified Modelling Language - Component Diagram (UML)

UML is a well know modelling language with a semiformal syntax and semantics. Weilkiens (2011) states that UML was made a standard for system engineering by the INCOSE in 2001. The technique revolves around four diagram levels with at least one meta-model. Pooley & King (1999) state that the UML contains several different types of diagram, which allow different aspects and properties of a system design to be expressed. In this project we focus on the functional view of the architecture, which in case of UML is covered by the Component Diagram. Medvidovic et al. (2002) argue that the UML address a number of issues: classes and their declared attributes, operations, and relationships; the possible states and behavior of individual classes; example scenarios of system usage and the system behavior in those scenarios and several others. Overall the technique is quite complex and open to extensions; however the user has to have prior knowledge about the language in order to use it effectively.

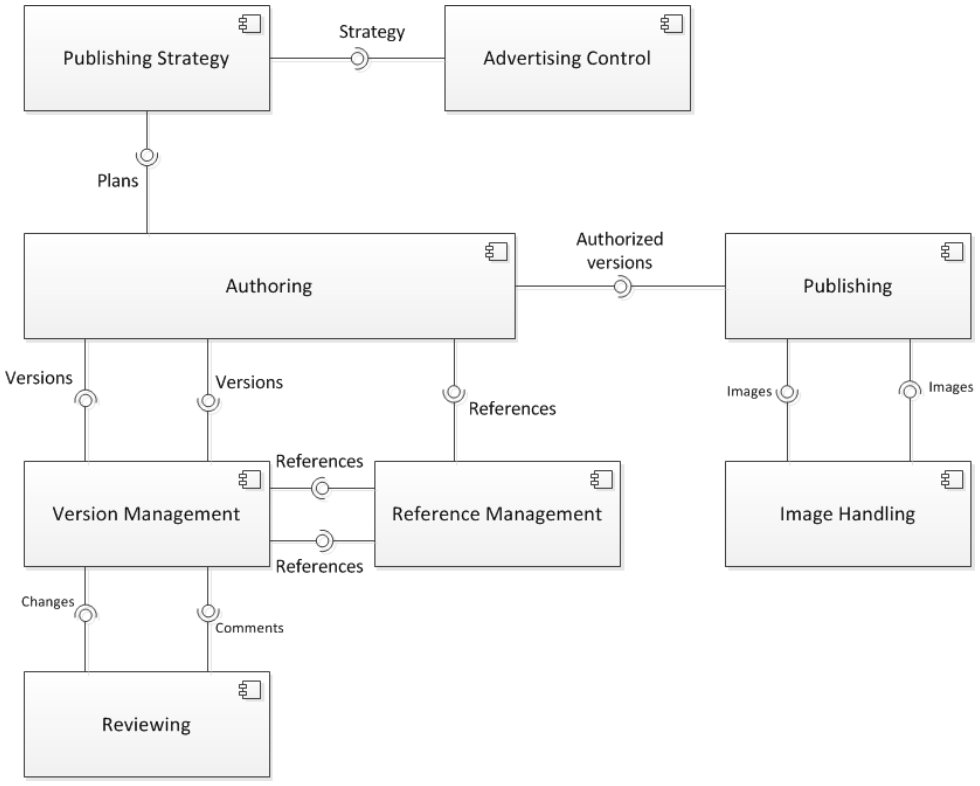


Figure 8 – an UML Component diagram

3.3 Evaluation

The last stage of this chapter is to evaluate each of the four chosen techniques according to the five criteria we have chosen in the 3.1 section. Each technique will be discussed separately with regards to the chosen criteria.

First however we need to define how are the chosen criteria going to be measured.

Usability: The usability of the system will be judged according to how difficult it is to use the method. Whether the notations used are straight forward and understandable even by someone who does not know the exact working of the method. Methods with the most intuitive way of displaying information will be the most desirable.

Conformity: The models need to display as much information as possible, but at the same time the display should not be too complex. The modelling technique which provides most suitable information while still keeping the complexity in order, is the most desired one.

Model Visualization: At one side the models need to display the appropriate information, but at the same time the models need to be readable. The technique with most intuitive displaying of information (using intuitive nouns and verbs to label the modules and information flows) are going to be chosen.

Extensibility: As the final chosen technique has to be extended by new modelling notations, its structure should be designed in a way that best accommodates the new notations. Techniques with simpler structure and flexible rules are desired.

Analyzability: The chosen method should have a structure that is easily analyzable by the standard architecture evaluation techniques, such as ATAM or SAAM.

Score: Each of this criteria will be scored from 1 – 5. 1 - Being the most desired value; 3 – Being neutral value; 5 – Being the most undesired value.

Functional Architectural Models (FAM)

Usability: The cornerstones of the FAM technique are the modules representing the functions & features of the system, combined with the directional arrows, which represent the flow of information. Therefore using and manipulating the resulting models is quite simple and straightforward. Any additions or other changes are easily updated in the models themselves.
Score: 1

Conformity: The semantics and syntax of the FAM models provide the user will lot of the necessary information about the architecture, mainly by using the descriptive names for every module as well as its connections to other parts of the system. The only downside of this method is that it does not provide complex semantic structures. For example modelling a splitting information flow, which is affected by a Boolean variable might prove challenging.
Score: 3

Model Visualization: The FAM models are composed of modules (functions) with verb noun labels complemented by the connections between these modules which are labeled using nouns. The connections clearly display the flow of information using directional arrows. Together they provide a very clear and straight forward overview of the system. Overall the technique is easily understandable even by someone without any prior experience.
Score: 1

Extensibility: Due to the simple nature of the notations in FAM, extending it with new notations should not prove to be a difficult task. The models can be easily supplemented by colors or simple markings to describe the changes in the architecture.
Score: 1

Analyzability: FAM is one of the standard modelling techniques used in the field of functional architecture, combined together with the simple nature of the notations of the models, makes the FAM technique very suitable for all the standard analysis methods (e.g. ATAM, SAAM).
Score: 1

Functional Flow Block Diagram (FFBD)

Usability: The bases for the Flow Block technique is that each module has its inputs, outputs, support and control, which results in a more rigid design. The user needs to have extensive prior knowledge about the technique in order to use it.
Score: 3

Conformity: Due to the nature of the flow diagram notation the diagram does provide more detailed information as compared to the FAM technique. However, the same constraints can limit simplification of the models, resulting in models which are more complex and harder to understand.
Score: 2

Model Visualization: The FFBD technique uses similar modules and directional arrows as the FAM, however it also incorporates logical operators (AND, OR, XOR), which increase the complexity of the design. Furthermore, the fact that each flow has its own place (e.g. input always from the left) makes the diagrams harder to read.
Score: 4

Extensibility: While extending this method is possible (in a similar fashion as the FAM), the aforementioned constraints (rules) of the method make the extending into a difficult task. The new notation would have to be useful while not damaging the existing rules.
Score: 3

Analyzability: FFBD is a technique which has been around for many years and therefore should be easily analyzable by any of the standard evaluation techniques. Analysis might prove more difficult than with FAM due to the nature of the FFBD technique.
Score: 2

Systems Modeling Language (SysML)

Usability: The SysML technique consists of many different diagrams each suitable for different situation. Therefore it is necessary to have some prior knowledge in order to use this technique. However, the overall model notation is on a similar level than the FAM, making it quite simple to use once the user has a bit of prior knowledge.
Score: 2

Conformity: The method displays information using modules and directed arrows similar to the previous techniques, as well as the interfaces each of the modules has. Therefore the amount of display information is high, but might be lacking in the really complex instances.
Score: 3

Model Visualization: This method is based on modules and arrows with descriptive names, which make it easy to read and understand, however the user does need some beforehand knowledge to understand the interfaces and coupling of the modules.
Score: 2

Extensibility: As it was mentioned the notations of this method are quite straight forward, which makes extending this method very possible. The user does need a bit more knowledge for extending as he/she does for the FAM technique.
Score: 1

Analyzability: Analyzing of a technique usually depends heavily on the style of notations, which in this case are similar to the FAM method.
Score: 2

Unified Modelling Language (UML)

Usability: The UML method contains several types of diagrams, each with its own style of notation (in this case the component diagram). Therefore the user has to have extensive knowledge on UML in order to use this method effectively.
Score: 3

Conformity: The tradeoff for the method complexity and required knowledge is the amount of displayed information, which is the highest among these methods. There are special notations for each function or action the system does, making the method very descriptive.
Score: 1

Model Visualization: The method does indeed display a great amount of useful information, but on the other hand this makes the models harder to read; nearly impossible without any prior knowledge.
Score: 4

Extensibility: As the method is so complex with rigid rules, extending it is a difficult task. The UML notations are so specific that adding onto them requires one to be an expert from the UML field.
Score: 5

Analyzability: As UML is not a standard technique used to model functional architecture, combined together with its complex notations, it might be more difficult to use the standardized evaluation techniques to analyze the models.
Score: 3

All the above information has been summarized in the following table. The chosen criteria were scored from 1 – 5; 1 being the most desired score while 5 being the most undesired one (color coded as well).

Technique	**FAM**	**FFBD**	**SysML**	**UML**
Usability	1	3	2	3
Conformity	3	2	3	1
Model Visualization	1	4	2	4
Extensibility	1	3	1	5
Analyzability	1	2	2	3
Average Score	1.4	2.8	2	3.2

Table 3 – Summary of the ADLs evaluation

The scores displayed in the above table are all based on scientific journals, papers or books describing the different modelling techniques. If the scientific literature was lacking the necessary information that aspect of a technique was compared to the other ones reviewed. As it can be seen the technique that faired the best according to the chosen criteria is the FAM technique, due to its simple design and easy readability. This is the technique that is going to be used to model the functional architecture in the following chapters. This does not make the FAM technique superior to the others mentioned, but in the light of the chosen criteria the FAM is the most suitable option.

3.4 Change modelling

The above presented modelling techniques address the issue of architecture modelling, however they do not provide any notations or information about the change in between two versions of an architecture. Therefore, a small part of the literature review was focused on finding appropriate modelling notations to depict architecture change.

The following is a figure of new notations Hendrickson & van der Hoek (2007) use to depict the architecture change.

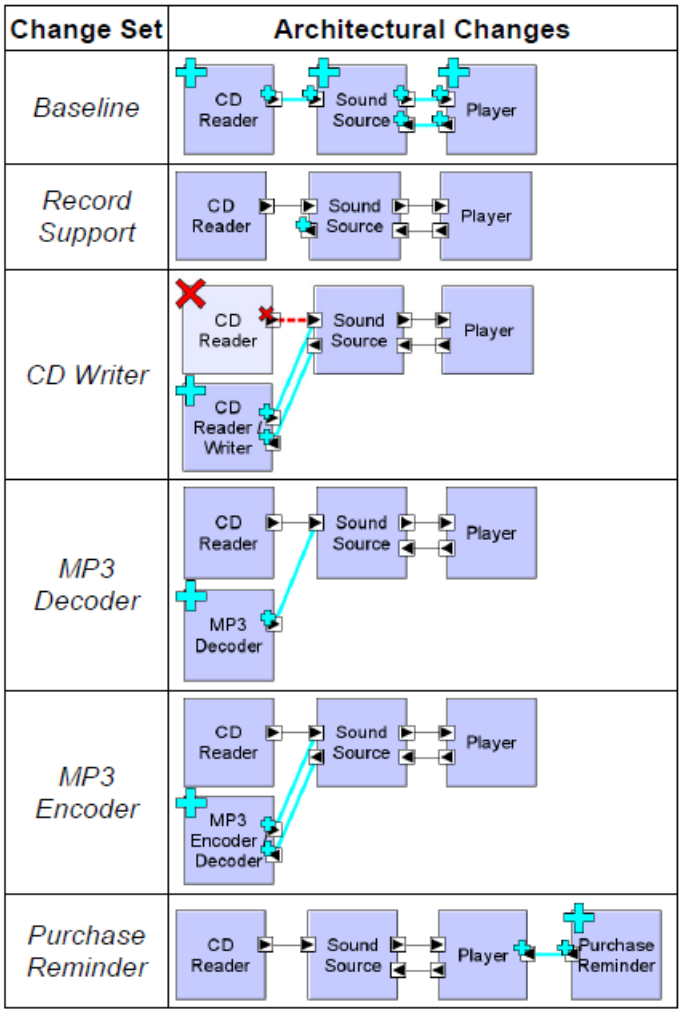


Figure 9 – Modell depicting changes in product line architecture (Hendrickson & van der Hoek, 2007)

In the Figure 3 Hendrickson & van der Hoek use a “plus” and “cross” sign in the top corner of a module to depicted whether it was deleted of added into the system. They use the same logic for modelling the change in the information flows; they label the deleted ones with a red “cross” sign and the new additions with a blue “plus” sign. Last but not least the modules which are deleted are filled with a slightly more transparent color that the ones added or remaining unchanged. This way the users can immediately know, what composed the actual working system = All the modules with the non-transparent module fill.

Chapter 4

Communication of a software release

4.1 Interview Setup

The main deliverables of this project are the functional models, which will display the overall structure of the different modules (functions & features) as well as their inter-connections and communications. Therefore, these models can be used to communicate the functional structure of software product releases, in our case company it is the new Dynamics AX R3 release. With this in mind, the project wants to address the way the new releases are being communicated from the development company (in this case Microsoft) towards its implementation partners (HSO). We want to find out what are the mechanisms Microsoft uses in order to communicate a new software release, which is directly related to the research question Q3.

As it can be seen in the research design (Q3) a consultant interview has been proposed to collect data about the way the new releases are being communicated. At this point it was important to decide which type of interview will be most suitable for information gathering. A choice between structured, semi-structured and unstructured interviews; structured interviews often provide quantitative data due to their exact questions and answers, however this consultant interview has to be more exploratory and therefore semi or unstructured interviews are more suitable. Finally the semi-structured interview was chosen as the best option for the following reasons. Louise Barriball & While (1994) describe the advantages of a semi-structured interview:

1. It has the potential to overcome the poor response rate of a questionnaire survey;
2. It is well suited for exploration of attitudes, values, beliefs and motives;
3. It enables probing for more information or clarification of answers;

4. It provides the opportunity to evaluate the validity of the respondent's answers by observing non-verbal indicators;
5. It ensures that the respondent is unable to receive assistance from others while formulating a response.

Due to these factors a semi-structured interview was chosen as for example opposed to using a questionnaire. DiCicco-Bloom & Crabtree (2006) describe these interviews as generally organized around a set of predetermined open-ended questions, with other questions emerging from the dialogue between interviewer and interviewee. The basic research question may well serve as the first interview question, but between 5 and 10 more specific questions are usually developed to delve more deeply into different aspects of the research issue. Furthermore, the interviewer should be prepared to depart from the planned topic during the interview, because digressions can be very productive as they follow the interviewee's interest and knowledge.

4.2 Consultant Interview

Based on the description above, the interview was scheduled with one of the experienced consultants at the case company. The consultant has already had experience with several new software releases and therefore is perfectly suitable to answer the communication questions. The interview itself did not have a given time frame to ensure that all the crucial information will be gathered.

These are the questions formed in order to guide this consultant interview:

- ***How are the new Dynamics AX releases being communicated to the implementation partners?***
- ***What documents or materials are being communicated by the development company (Microsoft)?***
- ***What is the structure of these documents?***
- ***Are these documents or materials clear and easy to understand even by the non-technical staff?***
- ***Do these documents provide enough information about the new release? Or do you think you could receive better quality information?***

The following section is going to address and answer each of the interview questions according to the interviewee responses.

How are the new Dynamics AX releases being communicated to the implementation partners?

The first concept to find out was: How is a new release being communicated towards the implementation partners.

Press releases: According to the interviewee the very first pieces of information are gathered from the press releases, which only contain the basic information on what is new or changed.

Microsoft Conferences: The real communication of information takes place at the Microsoft (technical) Conferences also called Convergences, where all the implementation partners get their first hand on information about the new software release. However, these conferences only provide as much information as you would expect from a commercial company communicating their new product.

Much of the specific information is only being communicated with the chosen partners, whereas the others usually need to seek the information themselves via:
Technical websites (technet)
Approaching other partners
Reading white paper documentations

What documents or materials are being communicated by the development company?
&
What is the structure of these documents?

It is important to know what the structure of the communicated documents is, and whether there is enough information for the consultants to be able to understand the new changes. Furthermore, the information should be easy to understand even by non-technical staff. The developers themselves will probably always understand even the most technical documents, however there are many how do not have such an in depth knowledge about the technical

side and therefore need differently structured information. The development company (Microsoft) has several sources of information or documents, however almost none of them are being actively communicated. The information is usually made available to all the partners; however every consultant has to search for them him/herself.

The following are the different documentation types used to communicate a new release:

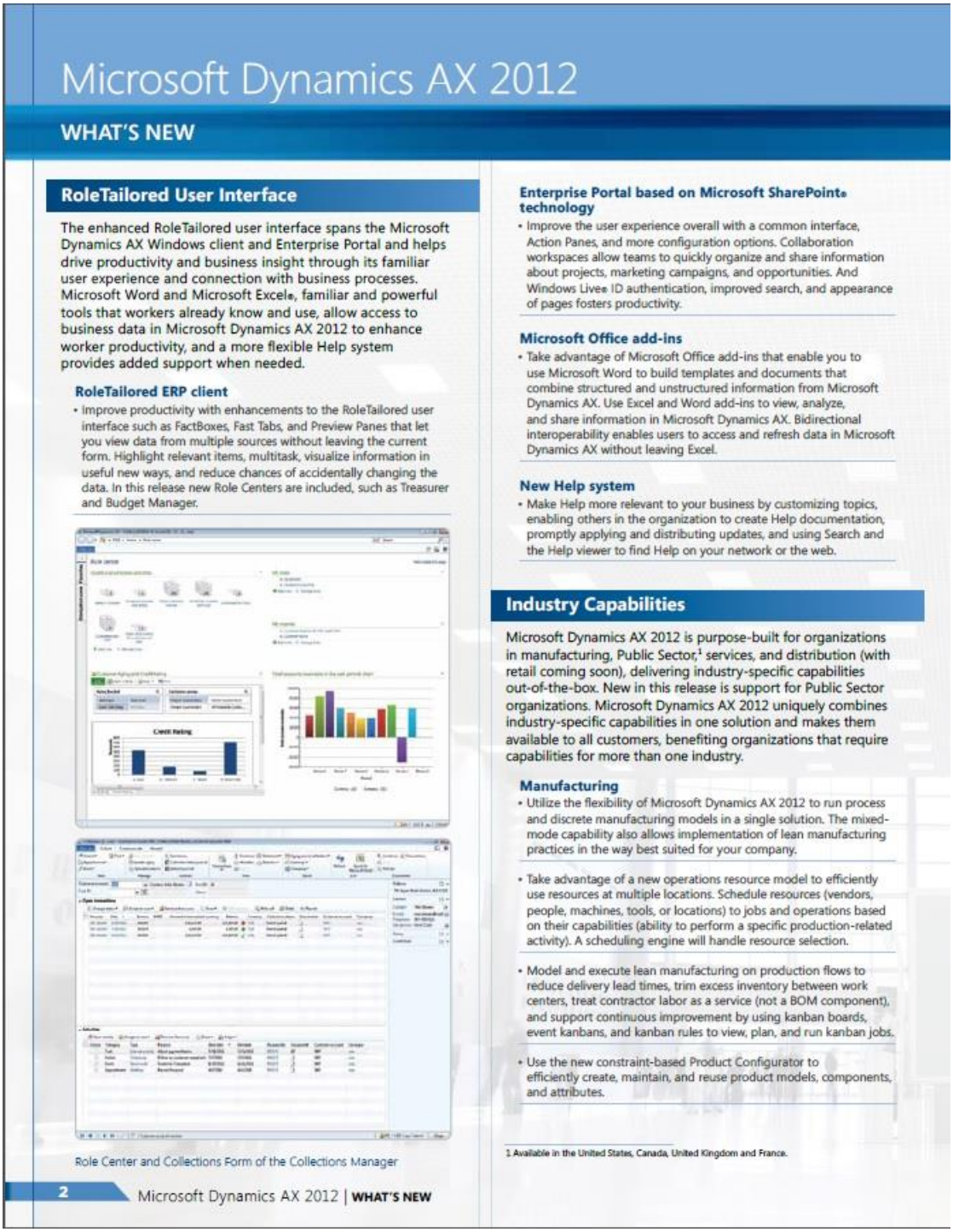
1. **White paper documents:** Written documents, usually in the form of a pdf, which contain detailed information about every aspect of the new release. Depending on the size of the release ranging from 50 – 600+ pages.

2. **Technet database:** Technical database website, created and maintained by Microsoft. It contains detailed information about the whole dynamics AX system, both older and newer releases. It contains several hundred links and pages describing the systems.

3. **Presentations:** Usually PowerPoint presentations created by the Microsoft staff. Featured at the Microsoft conferences. One presentation usually covers one topic of the new release.

4. **Screenshot tutorials:** Provided by Microsoft in a similar fashion as the white paper documentations. The tutorials are written documents supplemented by screenshots from within the AX environment. Only the very simple examples are usually described.

The main source is the Microsoft Information Source website, which incorporates all the documents released by the development company. An example of such an introductory white paper document can be seen in figure 1 below. Other tools for communicating are the example tutorials, which incorporate screenshots from within the software itself, as well as guiding explanations, to help the user understand the new changes (see figure 2). Technet is a website database with more detailed description of all the new functionalities and features; an example of the technet website can be seen in Figure 3.



Microsoft Dynamics AX 2012

WHAT'S NEW

RoleTailored User Interface

The enhanced RoleTailored user interface spans the Microsoft Dynamics AX Windows client and Enterprise Portal and helps drive productivity and business insight through its familiar user experience and connection with business processes. Microsoft Word and Microsoft Excel®, familiar and powerful tools that workers already know and use, allow access to business data in Microsoft Dynamics AX 2012 to enhance worker productivity, and a more flexible Help system provides added support when needed.

RoleTailored ERP client

- Improve productivity with enhancements to the RoleTailored user interface such as FactBoxes, Fast Tabs, and Preview Panes that let you view data from multiple sources without leaving the current form. Highlight relevant items, multitask, visualize information in useful new ways, and reduce chances of accidentally changing the data. In this release new Role Centers are included, such as Treasurer and Budget Manager.

Role Center and Collections Form of the Collections Manager

Enterprise Portal based on Microsoft SharePoint® technology

- Improve the user experience overall with a common interface, Action Panes, and more configuration options. Collaboration workspaces allow teams to quickly organize and share information about projects, marketing campaigns, and opportunities. And Windows Live® ID authentication, improved search, and appearance of pages fosters productivity.

Microsoft Office add-ins

- Take advantage of Microsoft Office add-ins that enable you to use Microsoft Word to build templates and documents that combine structured and unstructured information from Microsoft Dynamics AX. Use Excel and Word add-ins to view, analyze, and share information in Microsoft Dynamics AX. Bidirectional interoperability enables users to access and refresh data in Microsoft Dynamics AX without leaving Excel.

New Help system

- Make Help more relevant to your business by customizing topics, enabling others in the organization to create Help documentation, promptly applying and distributing updates, and using Search and the Help viewer to find Help on your network or the web.

Industry Capabilities

Microsoft Dynamics AX 2012 is purpose-built for organizations in manufacturing, Public Sector,[1] services, and distribution (with retail coming soon), delivering industry-specific capabilities out-of-the-box. New in this release is support for Public Sector organizations. Microsoft Dynamics AX 2012 uniquely combines industry-specific capabilities in one solution and makes them available to all customers, benefiting organizations that require capabilities for more than one industry.

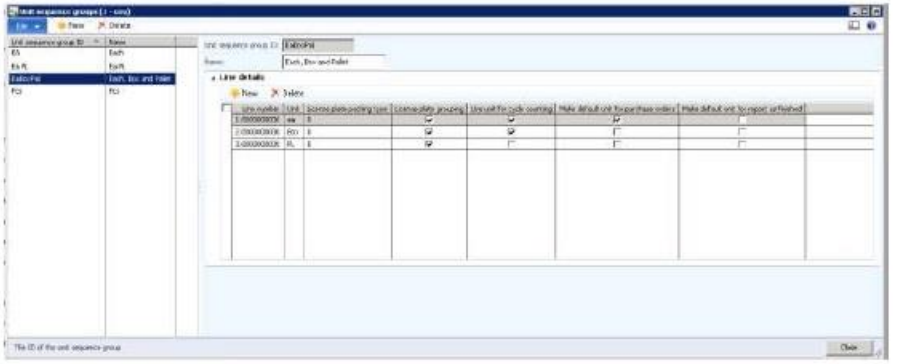
Manufacturing

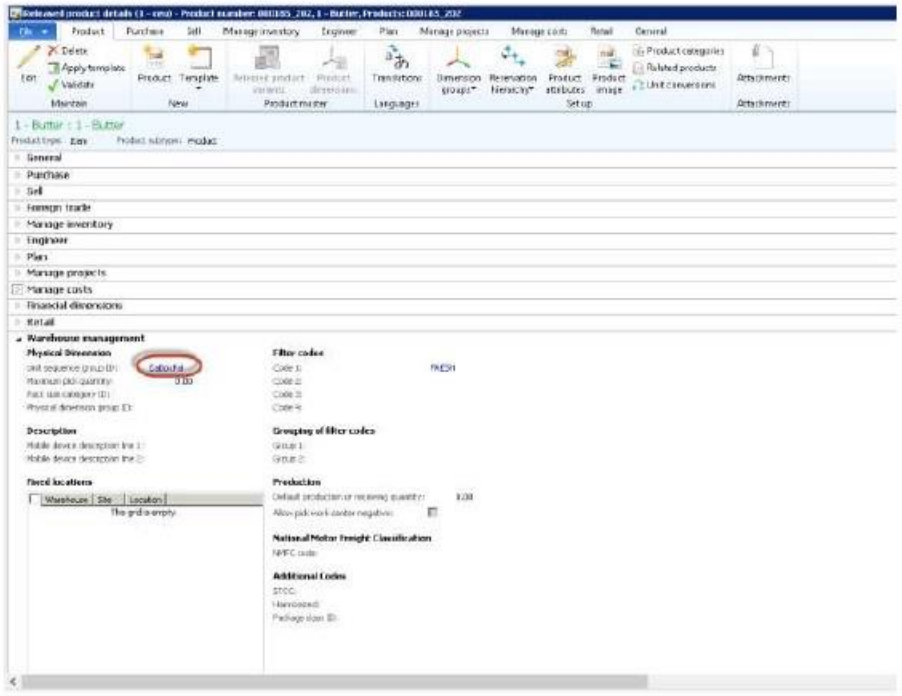
- Utilize the flexibility of Microsoft Dynamics AX 2012 to run process and discrete manufacturing models in a single solution. The mixed-mode capability also allows implementation of lean manufacturing practices in the way best suited for your company.
- Take advantage of a new operations resource model to efficiently use resources at multiple locations. Schedule resources (vendors, people, machines, tools, or locations) to jobs and operations based on their capabilities (ability to perform a specific production-related activity). A scheduling engine will handle resource selection.
- Model and execute lean manufacturing on production flows to reduce delivery lead times, trim excess inventory between work centers, treat contractor labor as a service (not a BOM component), and support continuous improvement by using kanban boards, event kanbans, and kanban rules to view, plan, and run kanban jobs.
- Use the new constraint-based Product Configurator to efficiently create, maintain, and reuse product models, components, and attributes.

1 Available in the United States, Canada, United Kingdom and France.

2 Microsoft Dynamics AX 2012 | WHAT'S NEW

Figure 10 – An Introductory white paper document for new software release communication





After you define unit sequence groups, ensure that you set the **Unit Sequence Group ID** on the **Warehouse management** FastTab.

Note that for LBs you need to define a Unit sequence group of Lbs, box and pallet, and assign that to the fresh items created, that primary stocking is not eaches.

Microsoft Dynamics AX Warehouse Management System User Guide 29

Figure 11 – Example functional tutorial

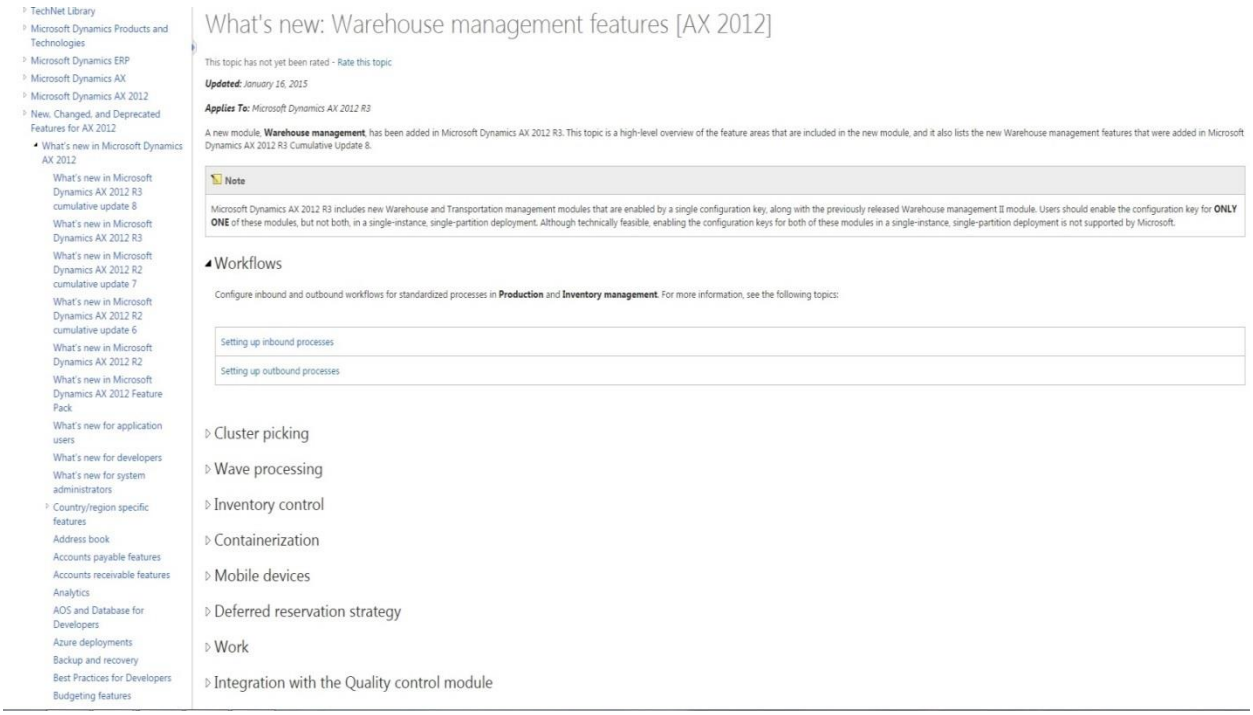


Figure 12 – The TechNet information source website

Are these documents or materials clear and easy to understand even by the non-technical staff?

At this point it seems like the information is available and easy to reach, however the problem with this many information sources is that the data is usually scattered all around and the consultants have to do an extra amount of work just to gather all the information they need at the moment. According to the interviewed consultant it is also difficult to see the connections between the different functions, as they are usually described each on their own, but never in the context of the others.

One of the focuses of this project is to be able to communicate the new release changes even to a non-technical staff (consultants), because consultants usually do not share the same extend of technical knowledge as the developers do. As for the documents themselves, there are always several types; some suitable only for developers with in depth technical knowledge, some suitable for non-technical staff. However, as it was already mentioned with such an amount of documentation much of the information is scattered and gathering it means extra work for the staff. The interviewee mentioned that he has to revisit several websites until he finds the information he needs.

Do these documents provide enough information about the new release? Or do you think you could receive better quality information?

The last question of the interview was whether the documents themselves provide enough information for the consultants. The interviewed consultant said that from the technical point of view you can usually find all the information you need, however that demos or presentations lack real life examples: "There is usually only one simple example and that is it. You have to figure out the complex ones on your own." Furthermore, the features and functions are being discussed in detail, but only on their own; there is no global overview where a person can see the different connections between the functions. The interviewee said himself that he would like to see an overview of all the new functionality in order to how are they interconnected.

4.3 Interview Conclusions

Based on this interview it is obvious there are many information sources when it comes to communicating a new release, however they have their limitations. Much of the information is scattered and requires extra work in order to become useful. When considering the technical information it is usually covered very well; however there might be a problem with understanding for the non-technical staff. To sum it up we have three issues:

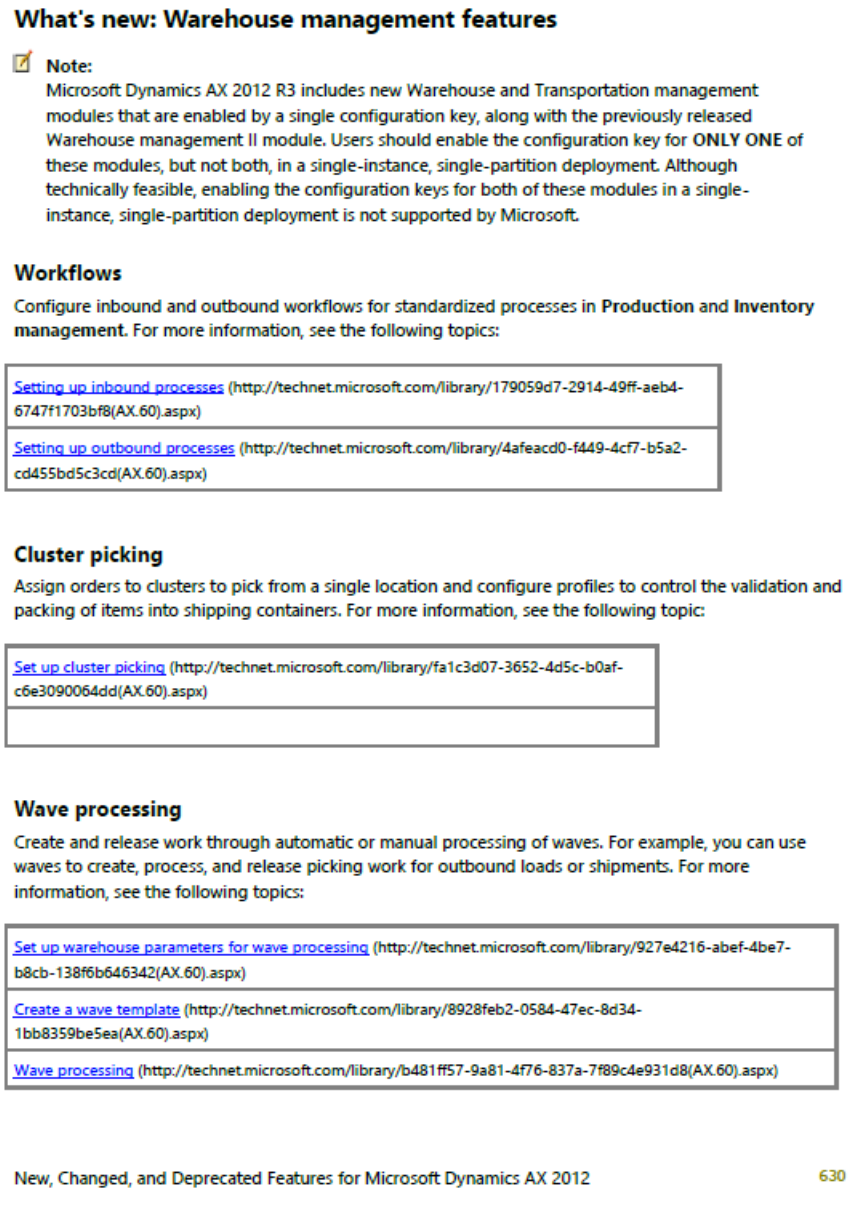
1. **No global overview of the functions and their connections**
2. **Lack of real life examples**
3. **Documents not being suitable for non-technical staff**

The lack of real life examples in the introductory tutorials is beyond the scope of this project, which is focused on the global functional overview (together with their connections) as well as on creating a document (information source) suitable for non-technical staff. However, when focusing on the two "in scope" concepts it is clear that the functional models will help to create an overview of the new and changed functions as well as their connections and communications. Moreover, the models are going to be structured in a way that even a non-technical staff can understand. Based on this interview it is clear that the functional models produced during this project are going to meet an important need for communication of a new software release.

4.4 Types of changes in the new release

The new notations need to be based on the types of changes introduced with the new software release. The new release of our case software product contains three major types of changes: new additions, deprecations and modifications.

Additions mean a completely new part of the software has been introduced, which usually means an addition of a new functionality or a feature. Below is an example of these additions extracted from the official “new release” document.



What's new: Warehouse management features

Note:
Microsoft Dynamics AX 2012 R3 includes new Warehouse and Transportation management modules that are enabled by a single configuration key, along with the previously released Warehouse management II module. Users should enable the configuration key for **ONLY ONE** of these modules, but not both, in a single-instance, single-partition deployment. Although technically feasible, enabling the configuration keys for both of these modules in a single-instance, single-partition deployment is not supported by Microsoft.

Workflows

Configure inbound and outbound workflows for standardized processes in **Production** and **Inventory management**. For more information, see the following topics:

Setting up inbound processes (http://technet.microsoft.com/library/179059d7-2914-49ff-aeb4-6747f1703bf8(AX.60).aspx)
Setting up outbound processes (http://technet.microsoft.com/library/4afeacd0-f449-4cf7-b5a2-cd455bd5c3cd(AX.60).aspx)

Cluster picking

Assign orders to clusters to pick from a single location and configure profiles to control the validation and packing of items into shipping containers. For more information, see the following topic:

Set up cluster picking (http://technet.microsoft.com/library/fa1c3d07-3652-4d5c-b0af-c6e3090064dd(AX.60).aspx)

Wave processing

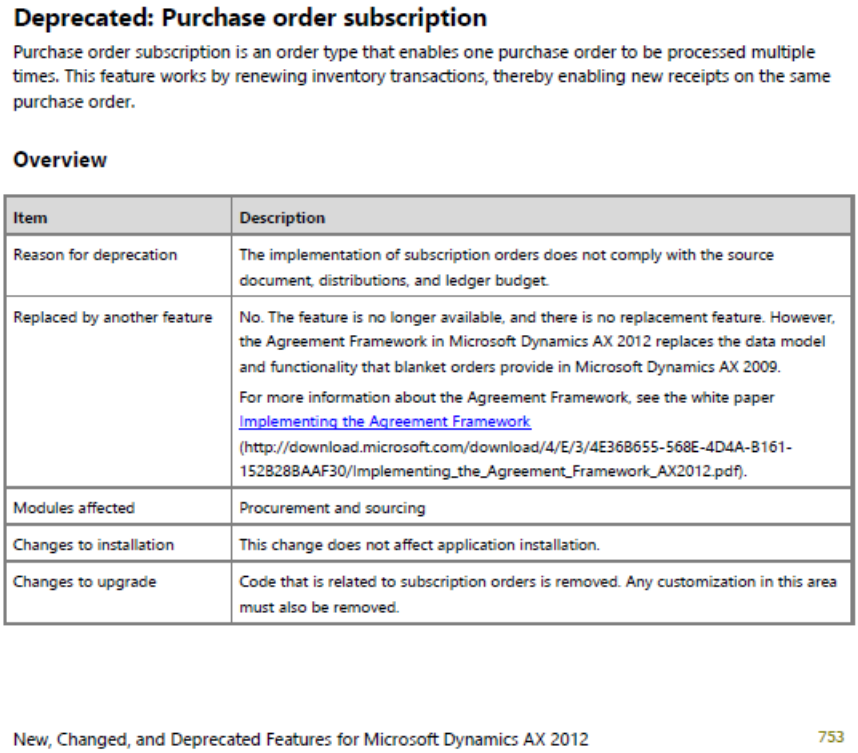
Create and release work through automatic or manual processing of waves. For example, you can use waves to create, process, and release picking work for outbound loads or shipments. For more information, see the following topics:

Set up warehouse parameters for wave processing (http://technet.microsoft.com/library/927e4216-abef-4be7-b8cb-138f6b646342(AX.60).aspx)
Create a wave template (http://technet.microsoft.com/library/8928feb2-0584-47ec-8d34-1bb8359be5ea(AX.60).aspx)
Wave processing (http://technet.microsoft.com/library/b481ff57-9a81-4f76-837a-7f89c4e931d8(AX.60).aspx)

New, Changed, and Deprecated Features for Microsoft Dynamics AX 2012 630

Figure 13 – an Example of new software additions (Microsoft Dynamics AX, 2012)

On the other hand there are several features and functions which are a bit out dated or rendered unimportant by the introduction of new additions. These parts of the software need to be deprecated from the system, or in other worlds a deletion of parts of the software needs to take place. There are several deletions of features mentioned in the new release documents.



Deprecated: Purchase order subscription

Purchase order subscription is an order type that enables one purchase order to be processed multiple times. This feature works by renewing inventory transactions, thereby enabling new receipts on the same purchase order.

Overview

Item	Description
Reason for deprecation	The implementation of subscription orders does not comply with the source document, distributions, and ledger budget.
Replaced by another feature	No. The feature is no longer available, and there is no replacement feature. However, the Agreement Framework in Microsoft Dynamics AX 2012 replaces the data model and functionality that blanket orders provide in Microsoft Dynamics AX 2009. For more information about the Agreement Framework, see the white paper Implementing the Agreement Framework (http://download.microsoft.com/download/4/E/3/4E36B655-568E-4D4A-B161-152B28BAAF30/Implementing_the_Agreement_Framework_AX2012.pdf).
Modules affected	Procurement and sourcing
Changes to installation	This change does not affect application installation.
Changes to upgrade	Code that is related to subscription orders is removed. Any customization in this area must also be removed.

New, Changed, and Deprecated Features for Microsoft Dynamics AX 2012 753

Figure 14 – an Example of a software feature deletion (Microsoft Dynamics AX, 2012).

Both the new additions as well as the deletions are not always straight forward. Sometimes an entire feature is added into the system, without any further modification of any other modules. However, there are many cases of deletions or additions, where the process of adding or deleting triggers a modification elsewhere in the system. Therefore, the last type of modelling notations has to address the issues of system modification: A change in the modules that can be neither labeled as an addition nor deletion, but something in between, a modification.

Chapter 5

Architecture modelling notations for release extensions

As stated in chapter 4 the chosen technique for the final functional modelling will be the FAM technique. The technique itself fulfils all the necessary criteria: ease of use, easy to understand, expressiveness, but most importantly it is easily extensible. This becomes an essential part of this project as there is no way to model the changes in the architecture with the FAM language as it is. Therefore, several new notations must be introduced and used to extend the FAM language in order to model the changed (additions, deletions or modifications) in the functional architecture.

5.1 Architecture change modelling

As it was mentioned in the previous chapters, there is no standardized notation for modelling a change in a functional architecture. Therefore, notations that address the architecture change will be one of the main contributions of this research. In the previous section 4.4 it was stated that there are three distinct types of changes introduced in the new release:

- **Additions**
- **Deletions**
- **Modifications**

Each of these types will be met with the appropriate new notation depicting the nature of the change. These notations will be incorporated into the FAM framework, which will later result in functional models, which can depict the changes in the architecture.

The only inspiration for these new notations was a paper by Hendrickson & van der Hoek (2007) which was focused on modelling the changes in product line architecture. The field of study does differ from the one of this project; however they are similar enough to make their paper relevant. Other small inspiration came from the field of Method Engineering, where the difference between two PDDs is being noted by the use of different colors for the changes modules and their respective information flows. The use of colors is one of the easiest and most commonly understood methods of depicting changes.

5.2 New modelling notations

Based on the paper by Hendrickson & van der Hoek (2007), which can be seen in chapter 3.4, and the change modelling between two PDDs from Method Engineering, each of type of the architecture change is depicted by our new notation.

New additions

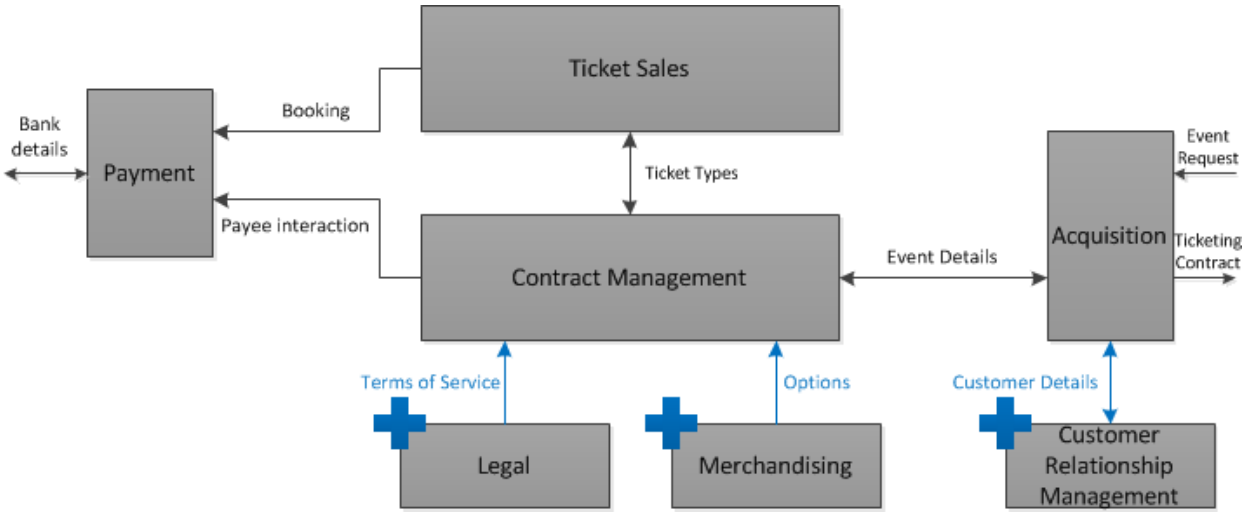


Figure 15 – New modelling notation: New module addition

Above we can see a part of a FAM diagram modified using the newly introduced notation. The newly added modules "Legal", "Merchandising" and "Customer Relationship Management" are marked with a blue plus sign in the upper left corner, depicting that these modules are one of the newly introduced ones. Newly added information flows that supplement and connect the new module to the system are also colored blue as well as their descriptive text. Furthermore, the new module has the same transparency of color as do the already existing modules, which depicts the active modules of the system.

Deletions

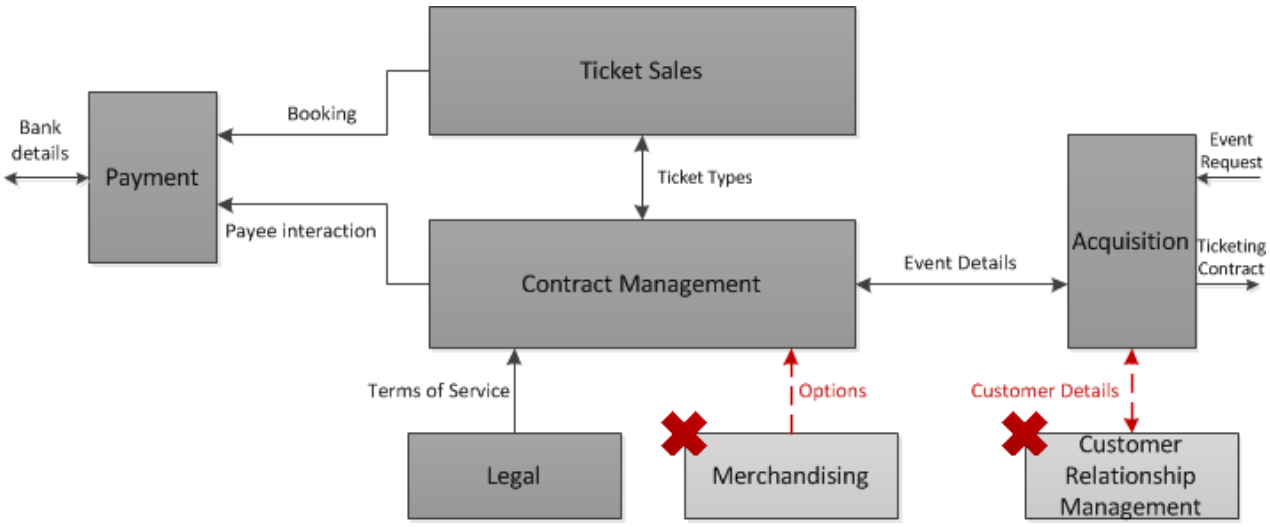


Figure 16 – New modelling notation: Module deletion

The deleted modules "Merchandising" and "Customer Relationship Management" are marked using a red cross sign, representing the deletion of the given modules in the new release. Moreover, the transparency of the modules is increased to depict that the "Merchandising" and "Customer Relationship Management" modules are no longer a part of the active system. The deleted information flows that the modules used are described by a dotted red line with a supplementary red text to label the flow.

Modifications

Modifications are a bit more complicated that the simple additions and deletions. Some of the modules will not be deleted nor added, but some of their internal structure will be changed (e.g. the use of a new algorithm). Any modified modules will be marked with an orange circle in the upper left corner. Modifications can incorporate the use of deletions and additions and therefore are split into three categories.

Modification only

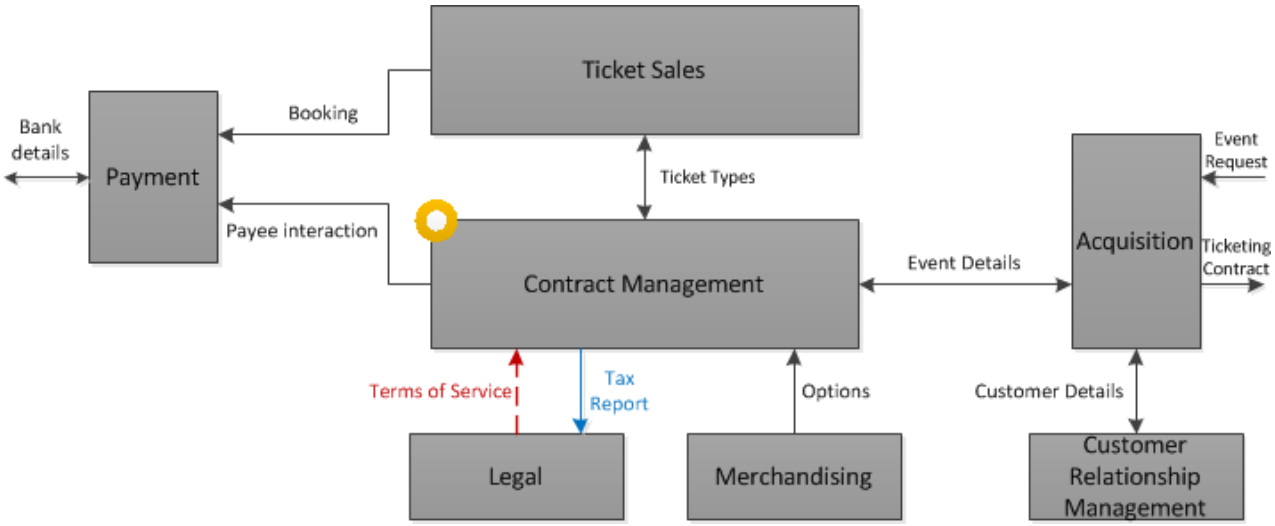


Figure 17 – New modelling notation: Modification only

The last figure 6 shows a situation where a module is modified, but no new additions or deletions are present. In this case the module is marked with an orange circle, but no additional patterns occur. This means the module itself was changed. However the size of the module remains the same. For example if an algorithm is changed from a waterfall approach to a quick search approach.

Modification with additions

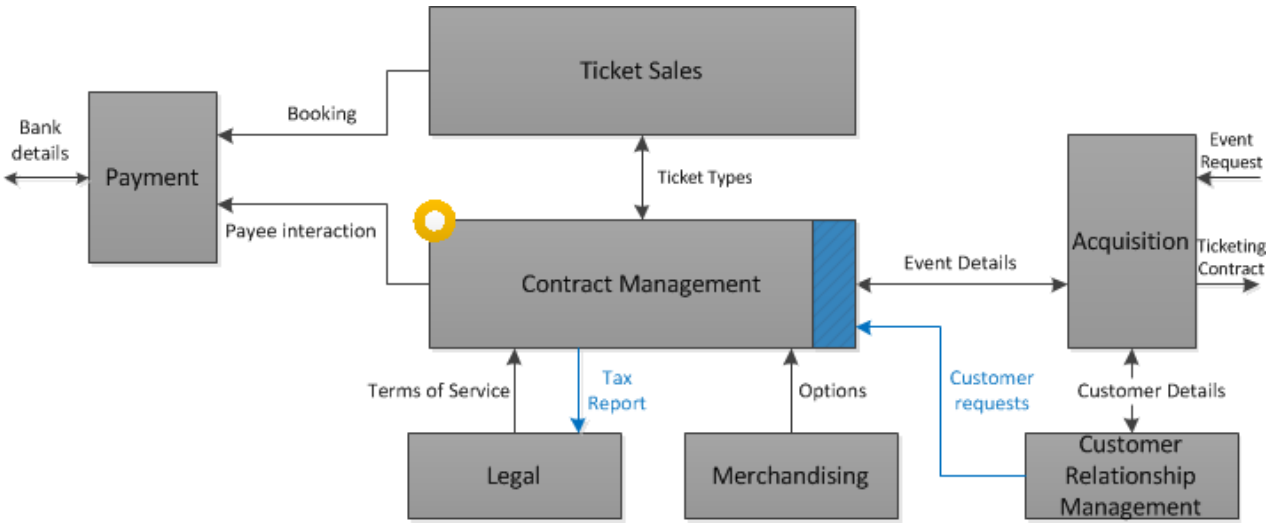


Figure 18 – New modelling notation: Modification with addition

In figure 7 we can see a changed module "Contract Management" marked with an orange circle, which means a part of the module has been changed. In the above case the "Contract Management" module has received new additional functionality, which is displayed using a blue pattern on the right part of the module. The blue pattern part depicts that something has been added to the current existing module.

Modification with deletions

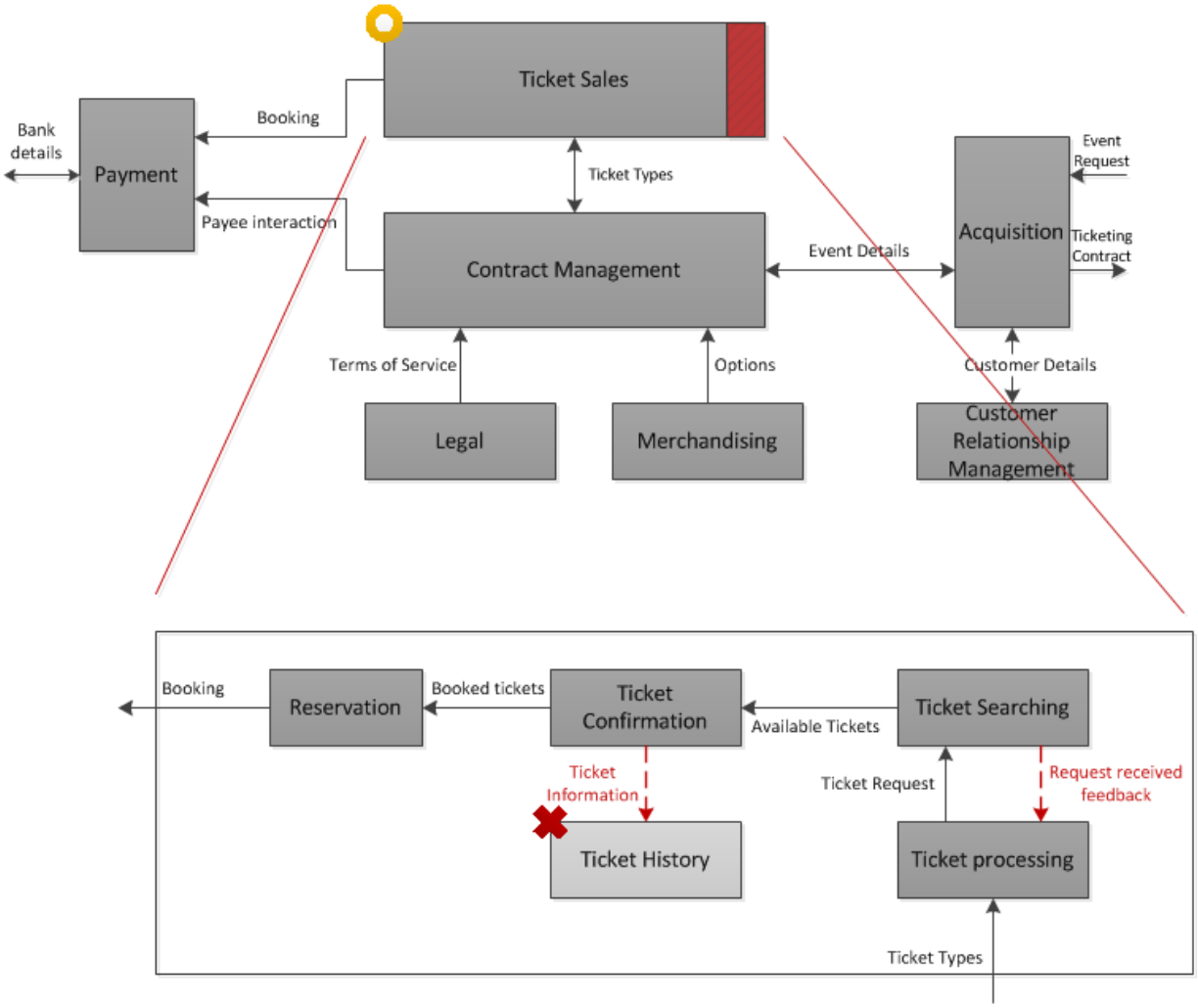


Figure 19 – New modelling notation: Modification with deletion and second level of abstraction

Figure 8 displays the module "Ticket Sales" as modified, combined with a deletion. Meaning a part of the module has been deleted, but the rest remains the same. The deletion is depicted using a red pattern on the right side of the module. Furthermore, another level of abstraction can be created in order to model the changes happening in the higher level "Ticket Sales" module. The bottom part of figure 8 shows a model with a lower level abstraction, which presents the inside structure of the higher level module (in this case "Ticket Sales"). In figure 8, from the higher level model we can see that the "Ticket Sales" module was modified with a deletion, and the said deletion is depicted in the lower level model in the bottom part. In this case the 'Ticket History" module was deleted from within the "Ticket Sales" module.

All of the notations from Figure 4 to Figure 8 will be used to extend the current FAM framework. The notations will tackle the task of modelling the architecture change from one software release to another. The notations themselves have the potential to be incorporated into a standardized modelling technique (in this case the FAM).

Chapter 6

Functional Architecture models

This chapter will present and describe the functional models of the internal dynamics AX architecture. This is the point where the theoretical literature research and the dynamics architecture review combine together to describe the main warehousing functionalities. The prior setup for this models will be discussed on the beginning of this chapter, followed by presentation and descriptions of the models. Finally a few preliminary results will be made.

6.1 Modelling setup

This project combines on one side the theoretical knowledge about architecture modelling and on the other the technical parameters of a new software release. The theoretical part, which was discussed in previous chapters, was quite accessible due to the vast number of scientific sources. However, the technical part focused on the case study product (Dynamics AX) was much harder to manage. The main reason was that the architecture of the case product was not publicly available. It was only accessible to special Microsoft accounts. Due to some help from the case company staff, we were able to contact a Microsoft Software Engineer who could help with accessing of the Dynamics AX architecture.

Two online interviews/meetings were setup with the Microsoft Software Engineer, which aim was to gather information about the internal dynamics architecture. The interview had

to be online, because the interviewee is located in Denmark. It was setup around a fix collection of questions, but the style of interview was semi-structured to inspire conversation. Each of the interviews took between 1 – 2 hours.
Two main results were achieved from these interviews: One was the access to a private Dynamics AX environment, where the entire structure of the system could be examined. But more importantly the interviewee provided a reverse engineering method, which can be applied within the Dynamics environment in order to extract the functional architecture. The method in itself is simple and well embedded within the Dynamics system, and therefore proved to be easy to use and quite useful.

Using the information from the interviews with the MS software engineer, the interview with an expert consultant, plus all the prior data from literature research, the functional models can be created.

6.2 Functional Models explanation

As Dynamics AX is a software with a vast technical architecture it was decided that this project will only focus on a certain part of the system. During the introduction interviews at the case company we established that the most interesting part for us to examine is the warehousing module, mainly because warehousing functionality was a major focus in the recent Dynamics release (R3). However, after the first interview with the Microsoft software engineer we agreed to reduce the modelling to only those parts of warehousing, where we could show the most changes in the architecture.

Based on the latest modifications the creation of the functional models began. The reverse engineering method was the bases of these models. First, the entire warehousing architecture was extracted from Dynamics AX using the reverse engineering method. The functionality was exported in the form of ERD-like diagrams. After both the older R2 warehousing functionality (WMS) and the new R3 one (WHS) were exported it became apparent how different they are. The two architectures almost did not share any module or information flows together. The new R3 warehousing functionality was bought by Microsoft from an outside company and therefore its programming and architecture were completely different from the older R2 functionality. Due to this fact it was necessary to reduce the diagrams to their bare essentials in order to find at least some common ground.

At first is the model of the older R2 warehousing functionality, which will be considered as a base model for comparison. The R2 model can be seem bellow:

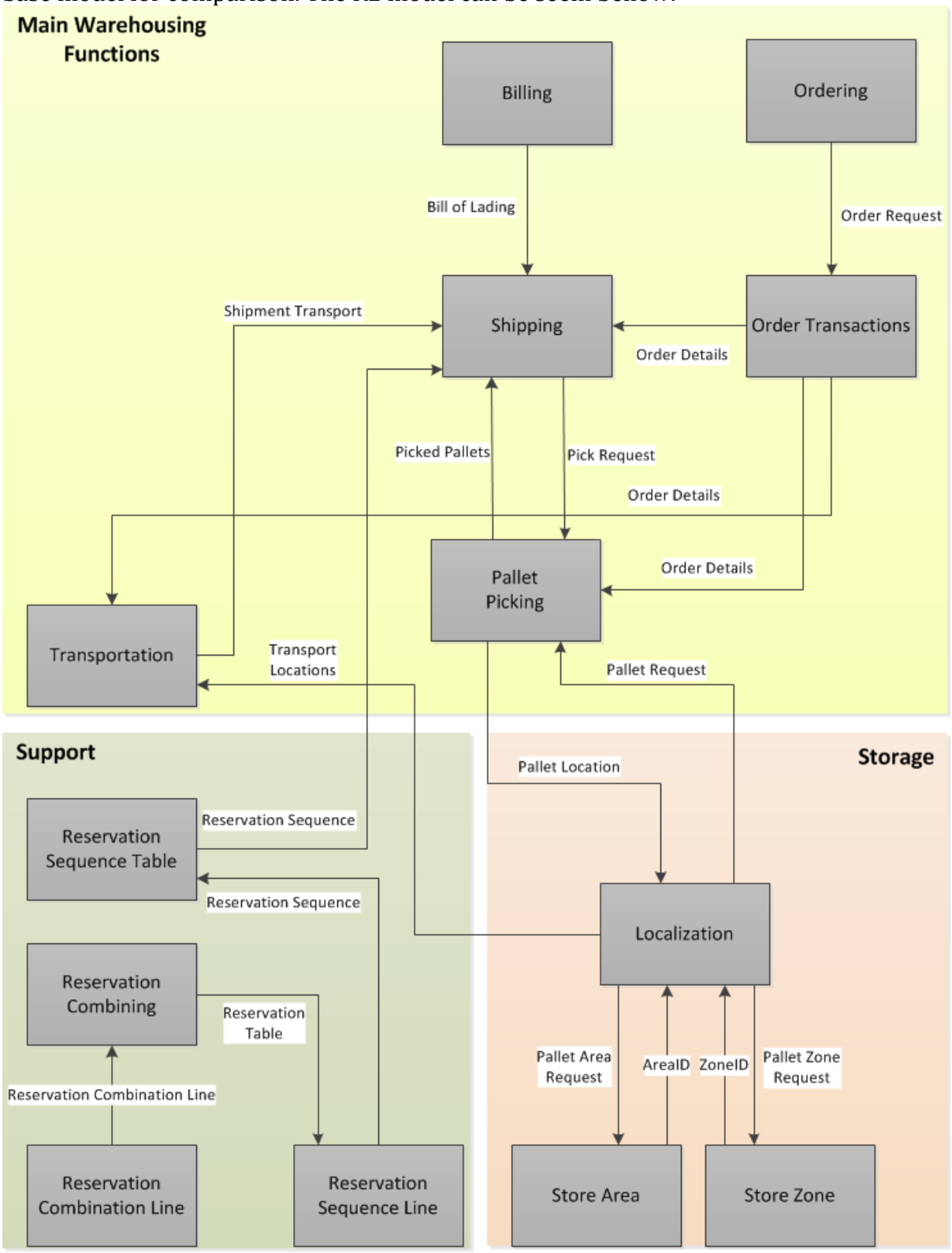


Figure 20 – Functional architecture model of the warehousing module – Dynamics AX R2

The model is split into three logical parts. On the bottom left is the Support, which incorporates the reservation hierarchies and tactics of the system. The hierarchies are directly connected to the main Shipping function; their task is to control which inventory items get released at which time to which location. On the bottom right, the Storage section covers the physical management of space and items. The Localization function takes the input from Pallet Picking, based on which it decides which items to pick and whether or not they can be picked. On the top are the main warehousing functions. The process usually starts from Order and Order Request, which further communicates with the Order transactions. All the necessary order information gets distributed into Shipping and Picking functions. The Picking function picks all the needed items with the help of Localization and sends it all to Shipping, which is the main function for releasing orders. Transportation function provides Shipping with the appropriate locations and addresses, which makes the Shipment ready for final expedition.

The above diagram serves as a basis for the new R3 release diagram. The R3 diagram can be seen on Figure 2. Unlike the R2 diagram the R3 uses the new notations introduced in chapter 5. The explanations of this model will focus on the changes made in the architecture.

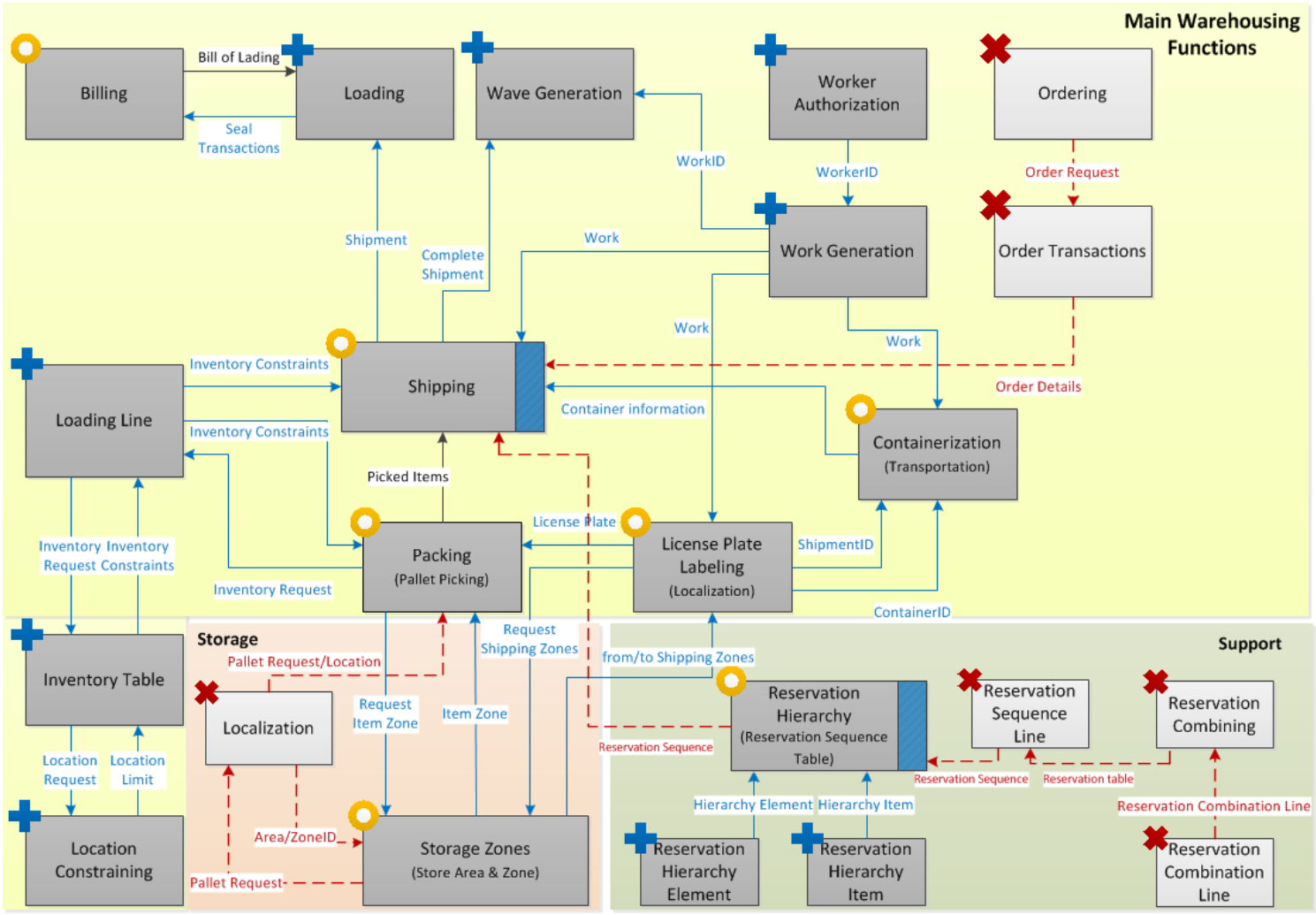


Figure 21 – Functional architecture model of the warehousing module – Dynamics AX R3

One of the major changes was with reservation hierarchies. The older R2 functionality was almost entirely replaced by a new one. The modules Reservation Sequence and Combining were deleted, whereas the main Reservation Sequence Table was reworked into Reservation Hierarchy. It does share some of the older functionality, hence the "modified with addition" label. Furthermore, the Reservation hierarchies no longer provide direct input into the Shipping function.

Storage zone and area were merged into one function Storage Zones, however it holds similar information and therefore was marked only as "modified". The Localization function was deleted entirely and replaced by License Plate labelling. They do however function differently, because the system no longer uses pallets as a unit, but license plates; this is one of the main reasons these two releases do not have so much in common. The Pallet picking was replaced by Packing, but it works in a similar fashion.

The Order and Order Transaction functions were both deleted from the new release and replaced by the concept of work. Each action in the warehouse is considered to be "work". Picking an item from a zone, sending it to shipping, labelling shipment, these are all instances of work. Therefore, the entire process is usually started by a certain worker trying to execute a certain type of work: Work Authorization -> Work Generation. From Work Generation the information gets distributed into all the functions.

Last but not least it is important to note that there are many additions that do not correspond to any part of the older R2 functionality. They are marked with blue color flows and blue crosses (e.g. Wave generation). Only a few were chosen to be displayed here, due to the R3 functionality being too large, containing mostly parts unrelated to the old R2 functions.

Model statistics

The following is a summary of the changes depicted in figure 2. These statistics reflect only the seen model, not the entire real life architecture, as the model only depicts a certain part of the complete system.

	Modules	**Flows**
Additions	9	25
Deletions	6	10
Modifications (only)	5	N/A
Modifications (addition)	2	N/A
Modifications (deletion)	0	N/A

Table 4 – R3 Functional model statistics

6.3 Preliminary results

Some preliminary conclusions can be made based on these models. The main research question states: ***How are functional architectures of software products modelled?***
Using these FAM models combined with the newly introduced notations we can see that the functional architecture can be modelled and the way how can be seen in the models themselves. The models provide an overview of the functionality from a higher than the programming level, which makes them more suitable for less technical oriented staff.

The complexity of the models can be adjusted according to the one creating them. However, one of the aims is to make the models easy to understand and therefore the higher the complexity the harder it is to understand them. It is more sensible to keep them at a lower complexity as there is plenty of information when it comes to the deep technical specifics of the system (e.g. developer technet).

The most important conclusion which can be drawn here is that this FAM enhanced technique is more suitable for an incremental software release, meaning that the old functionality gets improved and enhanced, but not entirely replaced by a new one. Because as in this instance, the whole warehousing module was implemented from an external company, with a completely new programming. This limited the outcomes of these models, as they do not share many functions.

The final evaluation will provide even more specific conclusions, as the models will be analyzed by experts from the field.

6.4 Alternative solution to the architecture description languages method

During the first presentation of this project it was suggested that there might be an alternative solution to the architecture description languages, which provides similar results. The alternative solution is based on using automated tools to model the architecture of the system and at the same time compare the differences between two models. The resulting models might be similar to the ones from FAM, but the major difference is that the tools work automatically whereas the FAM models have to be created by hand.

There are several modelling tools that address this issue. For this section we have chosen only one technique as an example. The name of the technique is BP-Diff, which is a tool for behavior comparison of business process models. The reason why we have chosen this tool is because it focuses on communicating of the different between two models, which is exactly what this project is aiming to do. The BP-diff was developed by: Abel Armas-Cervantes, Paolo Baldan, Marlon Dumas, and Luciano Garc´ıa-Ba˜nuelos (2014). "BP-Diff is a tool for identifying and diagnosing behavioral differences between pairs of business process models. BP-Diff identifies behavioral discrepancies involving pairs of tasks and provides both verbal and visual feedback to help users to understand each discrepancy. The verbal feedback explains how a given pair of tasks is related in one model in contrast to the other model. Meanwhile, the visual feedback allows users to pinpoint the exact state where the discrepancy occurs. "(BPdiff, 2014). The technique currently works based on BPMN models, which get mapped to Petri nets in order to make the comparisons.

On Figure 3 we can see the BP Diff tool at the beginning of the comparison process. The tool displays both the models next to each other to compare. Using the Petri nets the tool determines the differences between the given models. The differences are displayed on the left hand side in form of notes. The user can then click on any of the notes on the left side to see the details. Figure 4 shows what happens when a user clicks on a note (difference) on the left side; all the appropriate parts of the structure are highlighted in a logical fashion. If there is a problem the tool marks the problematic parts will red color, whereas the working parts are marked green. The tool also displays a number for each module to illustrate how many times a given module needs to be executed for the system to work properly.

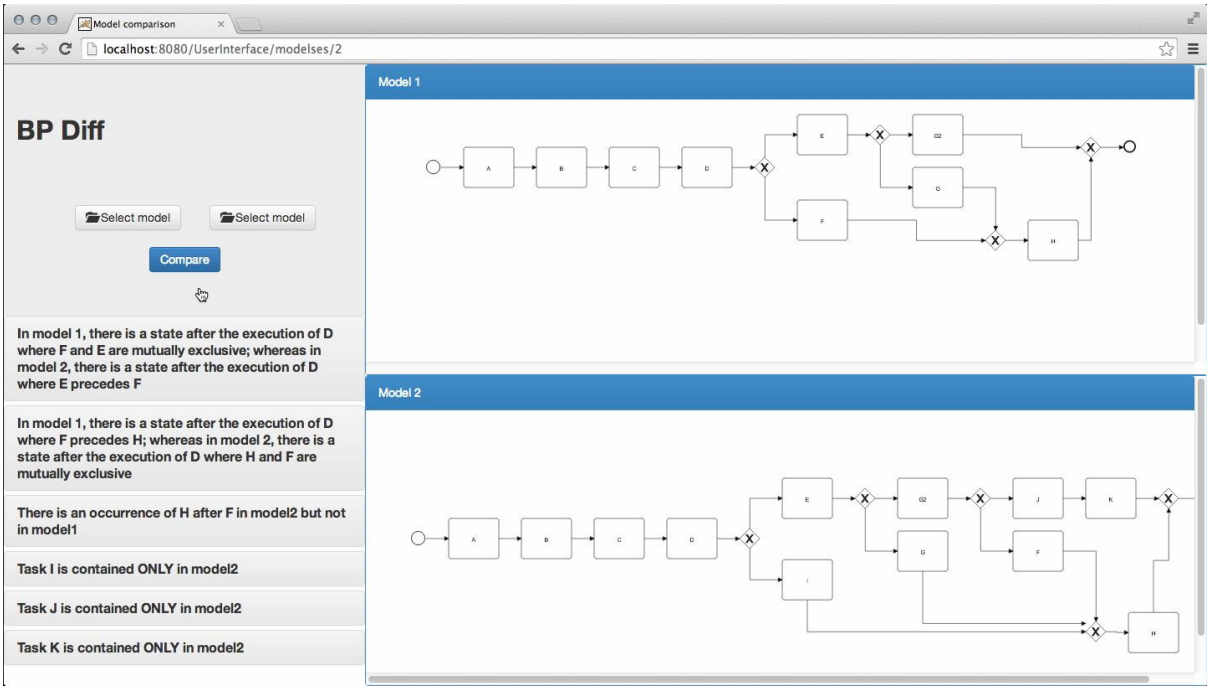


Figure 22 – BP-diff basic models

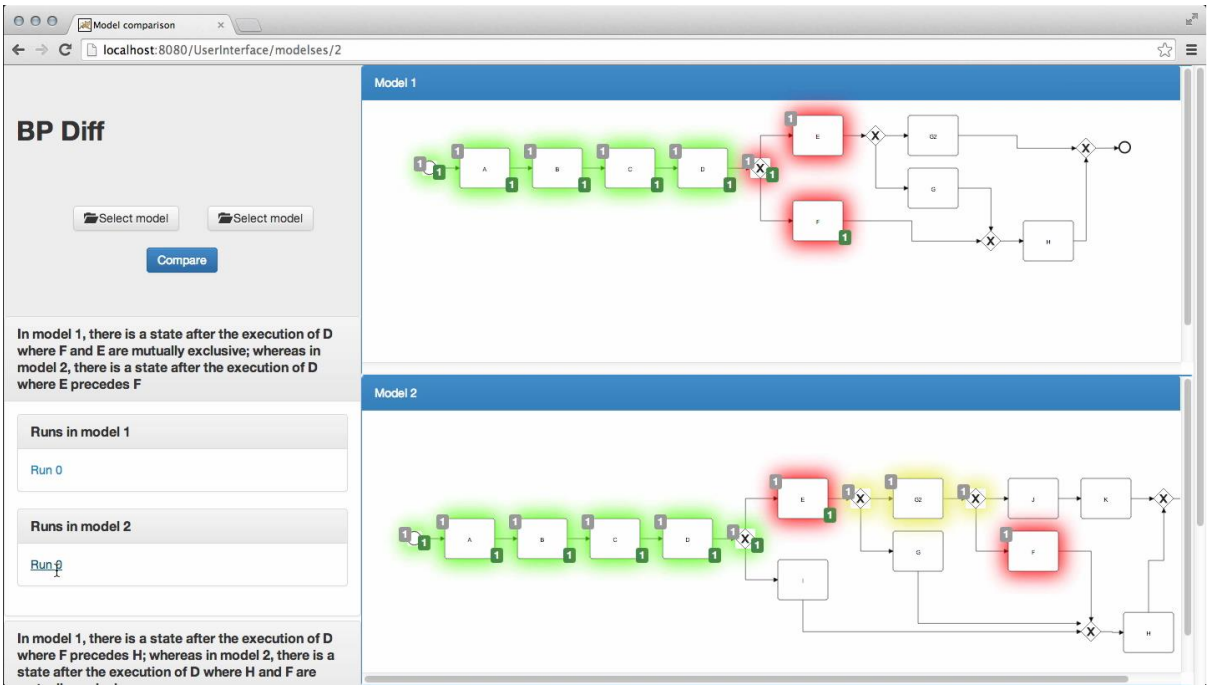


Figure 23 – BP-diff comparing models

The main difference between the BP-diff and the FAM technique is that the BP-diff is an automated tool, whereas the FAM models need to be created by hand. This gives the BP-diff an edge and considering large number of models and comparisons. Furthermore, the BP-diff provides very specific and detailed way of communicated the different in the system structure. This provides a detailed overview of the architecture of the system and its internal working.

The BP-diff tool uses Petri-nets to map all the tasks, events, and gateways in order to make comparisons between two models. Dijkman, Dumas & Ouyang (2008) describe the mapping process as follows: A task or an intermediate event is mapped onto a transition with one input place and one output place. The transition, being labelled with the name of that task or event, models the execution of the task or event. A start or end event is mapped onto a similar module except that a silent transition is used to signal when the process starts or ends.

The process of mapping and the later comparisons are done in an exact way, the two compared models need to share some common ground, such as task or event names, specific information flows. In a case where the two models share a common ground the BP-diff method can be very effective, however when the two models are different (e.g. create but two separate programming teams) the comparative results from the BP-diff method will be limited.

When comparing with the FAM technique the BP-diff provides more details and specific information, but that also means the BP-diff models are more complex and detailed than the FAM ones. However, for this project we wanted to provide a technique which would create model that are easy to understand even for a non-technical staff.

To sum it up, there is no best technique; the choosing of a technique depends on the level of complexity and details the user wants for their models. The FAM provides a higher level overview of the system functions, whereas the BP-diff provides more in depth information.

Chapter 7

Expert Evaluation

The functional architecture models were successfully created in the previous chapter. In order to validate and evaluate the models, they need to be examined by experts from the field. Therefore, the focus of the next chapter is on presenting and evaluating of the functional models to the experts from the field of Enterprise Resource Planning.

7.1 Evaluation Setup

The expert evaluation was setup at the case company HSO in Amsterdam. The evaluation session was attended by three experts from the field of Dynamics AX. The session took around two hours and was split into the following parts:

Presentation

A PowerPoint presentation was presented, which described the entire thesis project and presented all the preliminary results. At the end of the presentation were questions intended to open a discussion.

Discussion

The questions at the end of the presentation were meant to guide the discussion that followed. The purpose of the discussion was to evaluate the functional models in a semi-structured way. All the experts gave their opinions on the questions as well as the functional models presented. The discussion was recorded for future references.

Questionnaire

The discussion session was further supplemented by a questionnaire, which held questions addressing the issues of the presented materials. Some of the information might be redundant with the discussion session, however this way we assure that we gather all the necessary information.

HSO Evaluation Questionnaire

Master thesis project evaluation questionnaire - Mario Narwan

How suitable do you think the FAM functional models are for communication of a new software release?

- Very Suitable
- Suitable
- Neutral
- Limited
- Not Suitable

How hard is it to understand the functional models?

- Very easy
- Easy
- Neutral
- Difficult
- Very difficult

Do you think a consultant with little technical in depth knowledge can use this models?

Choose "Other" if you want to mention a particular reason.

- Yes
- No
- Other:

Do you think the information presented by the functional models and the way they are presented is useful?

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

Is there a particular issue these models have, that would limit their potential as a method of new release communication?

How suitable do you think the alternative (BP-diff) method is for communication of a new release?

- Very Suitable
- Suitable
- Neutral
- Limited
- Not Suitable

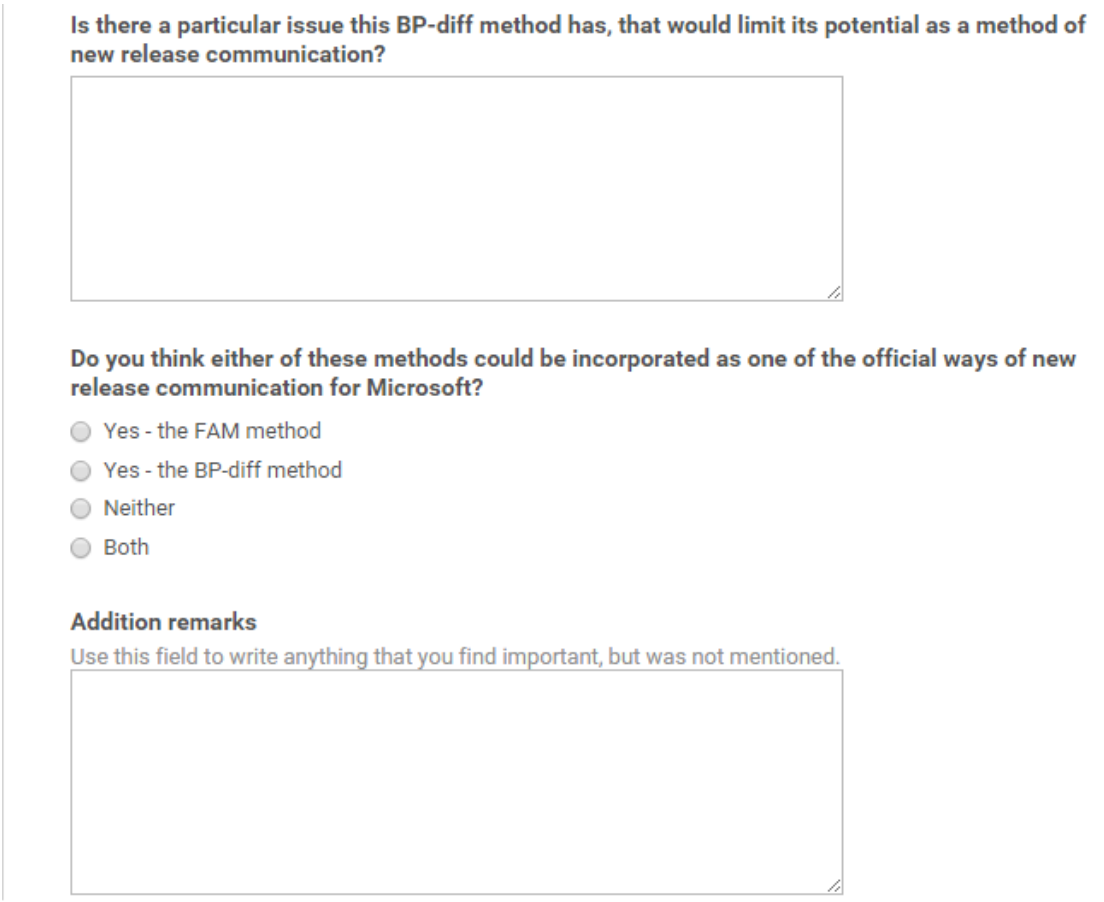


Figure 24 – The questionnaire used for the expert evaluation session

7.2 Evaluation results

The evaluation session was a success and it provided many different insights. To begin with the experts found the method to be understandable, even when considering the case where the user does not have too much in depth technical knowledge about the system. Therefore, this fulfilled one of the project requirements, which is producing functional models, which are easily understandable even for non-technical staff.

Furthermore, the experts valued the degree of clarity the models brought. They argued that an advantage of the models is that they provide an overview of the functions of the system, which cannot be found anywhere else.

One of the key topics of the evaluation session was the degree of complexity of the functional models and the impact of the complexity on the readability of the models. The experts argued

that when the models describe a large portion of the system, they get too complicated to understand and read. This becomes quite an issue, because Dynamics AX in itself is a large software product and trying to model the entire system will prove to be difficult.

One of the experts provided a possible solution to this problem. He suggested to use several levels of abstraction to split the models into logical parts. The final functional model, seen in chapter 6, is already split into three logical parts: Main Warehousing functions, Support and Storage. The idea is to keep one higher level model, which would incorporate all the logical parts and supplement that will several lower level models, each representing one logical part. Illustration of this idea can be seen in figure below. It is important to note that all the information flows from one logical part to the other need to be consistent with what is displayed in the higher level model.

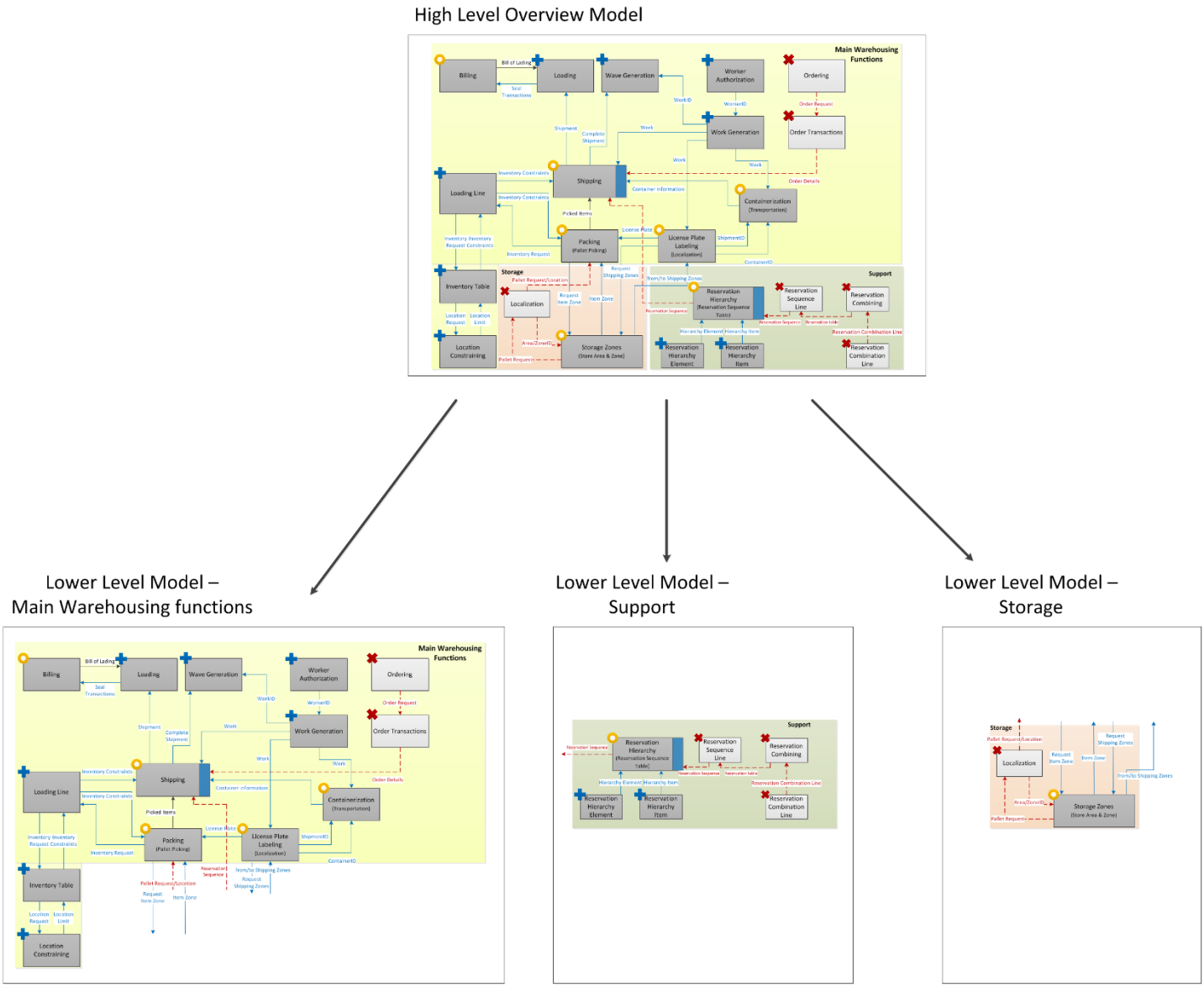


Figure 25 – Higher level functional model split into logical parts

Another suggestion for improvement was to use the size of the newly introduced notations in order to judge the size of the change which has happened in the given module. E.g. In figure 25 the yellow circles on top of the module would be larger or smaller depending on the amount of changes present within the module. This is in itself a good idea, however the size changes would introduce even more complexity into an already complex modelling environment, which would eventually cause the models to be less useful.

Finally the BP-diff method was originally a part of the evaluation session, however due to time constraints the BP-diff method could not have been examined as closely as the FAM method, which limited the evaluation of the BP-diff method heavily.

Questionnaire results

Below is a summary of the questionnaire results. Only questions with exact answers are displayed. Questions will open ended answers are discussed in the above section.

Questions	Expert #1	Expert #2	Expert #3
How suitable do you think the FAM functional models are for communication of a new software release?	Suitable	Suitable	Limited
How hard is it to understand the functional models?	Neutral	Easy	Neutral
Do you think a consultant with little technical in depth knowledge can use this models?	Yes	Yes	Yes
Do you think the information presented by the functional models and the way they are presented is useful?	Agree	Neutral	Neutral
How suitable do you think the alternative (BP-diff) method is for communication of a new release?	Not Suitable	Neutral	Neutral
Do you think either of these methods could be incorporated as one of the official ways of new release communication for Microsoft?	Both	Yes - the FAM method	Yes - the FAM method

Table 5 – Questionnaire Summary table

Conclusion

The aim of this project was to find a way to model the functionality of a software system. The proposal was to use an architecture description language to model the functional architecture of the system, which would provide us with an overview of the features and functions of the software. At the begging of the project we explored the possible architecture description languages, which could be used for this task. We evaluated the techniques according to several criteria found in the literature and chosen the FAM as the most suitable one. This was followed by extensive literature review on both the scientific side as well as on the side of the case company.

In order for the FAM technique to tackle the main problem we need to extend it with new notations, which would depict the changes in an architecture. With the combination of the FAM technique and the newly introduced notations we were able to produce the needed functional models. The functional models did provide a higher level overview of the system's functions, and provided additional clarity toward communicating a new software release.

The model themselves however do have certain limitations as was explored during the expert evaluation session. The FAM models tend to be quite complex, containing many modules and information flows, which can be harder to read or understand. Thanks to the evaluation session we managed to address this issue to some extend; and that is by creating several smaller models of a lower level abstraction, which would eventually get combined into a high level model (see Figure 25 above).

The alternative solution explored does provide some advantages over the FAM solution. It is an automated tool, which means the comparison of the model gets done quickly, and the information provided is exact. However, on the other hand when the two compared models are too different the result of this tool comparison is limited.

To sum it up, the experts do think that the FAM method together with the new notations can be used as a way of communicating a subsequent software release. However, the model complexity should be addressed and handled properly to avoid producing overly complex models.

References

Ang, S.H. (2014). Research Design for Business & Management. Victoria University of Wellington, *SAGE Publications Ltd*.

Armas-Cervantes, A., Baldan, P., Dumas, M., & Garcıa-Banuelos, L. (2014). BP-Diff: A Tool for Behavioral Comparison of Business Process Models.*Proceedings of the BPM Demo Sessions*, 1-6.

Blanchard, B. S., Fabrycky, W. J., & Fabrycky, W. J. (1990). *Systems engineering and analysis* (Vol. 4). Englewood Cliffs, New Jersey: Prentice Hall.

BPdiff (2014). Behavioural Process Model Comparator. Retrieved from: http://diffbp-bpdiff.rhcloud.com/

Brinkkemper, S., & Pachidi, S. (2010). Functional architecture modeling for the software product industry. In *Software Architecture* (pp. 198-213). Springer Berlin Heidelberg.

Cheng, E. (2004). An Organization Model to Support Dynamic Cooperative Work and Workflow. Birkbeck College, Department of Computer Science, London.

Clements, P. C. (1996, March). A survey of architecture description languages. In *Proceedings of the 8th international workshop on software specification and design* (p. 16). IEEE Computer Society.

Cusumano, M. A., & Smith S. (1995). Beyond the Waterfall: Software Development at Microsoft. (Working paper No. 3844-BPS-95). Retrieved from http://dspace.mit.edu/

Cusumano, M. A. (1997). How Microsoft makes large teams work like small teams. *Sloan Management Review*, *39*, 9-20.

Cusumano, M. A., & Selby, R. W. (1997). How Microsoft builds software. *Communications of the ACM*, *40*(6), 53-61.

Davis, D. (2005). SMC Systems Engineering Primer & Handbook. *United States Air Force Space & Missile Systems Center*.

Dijkman, R. M., Dumas, M., & Ouyang, C. (2008). Semantics and analysis of business process models in BPMN. *Information and Software Technology*,*50*(12), 1281-1294.

DiCicco-Bloom, B., & Crabtree, B. F. (2006). The qualitative research interview. *Medical education*, *40*(4), 314-321.

Dynamics AX Technical Conference. (2014). Microsoft Dynamics AX technical Conference. Retrieved from http://www.microsoft.com/dynamics/axtechconference/About.aspx?p=topnav

Feiler, P. H., Gluch, D. P., & Hudak, J. J. (2006). *The architecture analysis & design language (AADL): An introduction* (No. CMU/SEI-2006-TN-011). Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.

Friedenthal, S., Moore, A., & Steiner, R. (2014). *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann.

Garlan, D., Allen, R., & Ockerbloom, J. (1994, December). Exploiting style in architectural design environments. In *ACM SIGSOFT Software Engineering Notes* (Vol. 19, No. 5, pp. 175-188). ACM.

Greer, D., & Ruhe, G. (2004). Software release planning: an evolutionary and iterative approach. *Information and Software Technology*, *46*(4), 243-253.

Hendrickson, S. A., & van der Hoek, A. (2007). Modeling product line architectures through change sets and relationships. In *Software Engineering, 2007. ICSE 2007. 29th International Conference on* (pp. 189-198). IEEE.

HSO the Results Company. (2014). *HSO: About Us*. Retrieved from http://www.hso.com/content/us/

Kruchten, P. B. (1995). The 4+ 1 view model of architecture. *Software, IEEE*,*12*(6), 42-50.

Leonard, J. (1999). *Systems engineering fundamentals*. DEFENSE SYSTEMS MANAGEMENT COLL FORT BELVOIR VA.

Louise Barriball, K., & While, A. (1994). Collecting Data using a semi-structured interview: a discussion paper. *Journal of advanced nursing*, *19*(2), 328-335.

Medvidovic, N., Rosenblum, D. S., Redmiles, D. F., & Robbins, J. E. (2002). Modeling software architectures in the Unified Modeling Language. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, *11*(1), 2-57.

Microsoft Dynamics AX 2012 R3. (2014). Microsoft Dynamics AX 2012 R3 Preview. Retrieved from http://download.microsoft.com/download/5/A/E/5AEC22B0-3DBF-4E57-8E9C-7D63E2204234/Microsoft%20Dynamics%20AX%202012%20R3%20preview%20final%20English%20040914.pdf

Microsoft Dynamics AX Blog. (2014). Microsoft Dynamics AX R3 released. Retrieved from https://community.dynamics.com/ax/b/kayaconsulting/archive/2014/05/01/microsoft-dynamics-ax-2012-r3-released.aspx

Microsoft Dynamics AX. (2014). What is Microsoft Dynamics AX? Retrieved from http://www.microsoft.com/en-us/dynamics/erp-ax-overview.aspx

Microsoft Dynamics Resources Library. (2011). Microsoft Dynamics AX 2012 A New Generation in ERP. Retrieved from http://www.microsoft.com/en-us/dynamics/resource-library.aspx

Microsoft Dynamics Resources Library. (2011). Microsoft Dynamics AX 2012. Powerful. Agile. Simple. Retrieved from http://www.microsoft.com/en-us/dynamics/resource-library.aspx

Microsoft Dynamics AX (2012). New, Changed, and Deprecated Features for Microsoft Dynamics AX 2012. Retrieved from http://www.microsoft.com/en-us/dynamics/default.aspx

Munch, B. P. (1993). Versioning in a software engineering database—the change oriented way. *Division of Computer Systems and Telematics, The Norwegian Institute of Technology*, 1-284.

Pooley, R., & King, P. (1999, February). The unified modelling language and performance engineering. In *Software, IEE Proceedings*- (Vol. 146, No. 1, pp. 2-10). IET.

Rozanski, N., & Woods, E. (2011). *Software systems architecture: working with stakeholders using viewpoints and perspectives*. Addison-Wesley.

Ruhe, G. (2005). Software release planning. *Handbook of software engineering and knowledge engineering*, *3*, 365-394.

Ruhe, G., & Saliu, M. O. (2005). The Science and Practice of Software Release Planning. *University of Calgary*.

Salfischberger, T., van de Weerd, I., & Brinkkemper, S. (2011, August). The functional architecture framework for organizing high volume requirements management. In *Software Product Management (IWSPM), 2011 Fifth International Workshop on* (pp. 17-25). IEEE.

Siy, H. P., & Perry, D. E. (1998). Challenges in evolving a large scale software product. In *Principles of Software Evolution Workshop. 1998 International Software Engineering Conference (ICSE98), Kyoto Japan*.

van den Akker, M., Brinkkemper, S., Diepen, G., & Versendaal, J. (2005). Determination of the Next Release of a Software Product: an Approach using Integer Linear Programming. In *CAiSE Short Paper Proceedings*.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *Management Information Systems Quarterly*, *26*(2), 3.

Weilkiens, T. (2011). *Systems engineering with SysML/UML: modeling, analysis, design*. Morgan Kaufmann.

Whitman, L. (1999). A Methodology for the Classification of a Living Model of the Enterprise. University of Texas, Arlington.

Xu, L., & Brinkkemper, S. (2005). Concepts of Product Software: Paving the Road for Urgently Needed Research. In *CAiSE Workshops (2)* (pp. 523-528).