

# Enterprise Architecture Modeling of Core Administrative Systems at KTH

A Modifiability Analysis

PETER ROSELL



**KTH Electrical Engineering**

Degree project in  
Ind. Info. & Control Systems  
Master Thesis,  
Stockholm, Sweden 2012

XR-EE-ICS 2012:014

**ABSTRACT**

This project presents a case study of modifiability analysis on the Information Systems which are central to the core business processes of Royal Institution of Technology in Stockholm, Sweden by creating, updating and using models. The case study was limited to modifiability regarding only specified Information Systems. The method selected was Enterprise Architecture together with Enterprise Architecture Analysis research results and tools from the Industrial Information and Control Systems department of the same University. Jointly used with the ArchiMate modelling language, to create the models and perform the analysis. The results demonstrated to be very varied in regards to system models and modifiability. The Alumni Community system seemed to have very high modifiability whereas the Ladok på Webben system seemed to have the low modifiability, and other systems ranging differently or in between. The case study results found three slightly more critical systems of all the systems analysed: Ladok på Webben, Nya Antagningen & Ladok Nouveau. The first two showed to have either very low or low modifiability while being highly coupled to the other systems. Therefore any modification to these two systems would most likely cause effects that would require change in interconnected systems. Whereas Ladok Nouveau, while having average modifiability, has a critical position to process activities, is nearly isolated from all other systems, making them indirectly dependent on the system through the interconnected LADOK database. The study showed that the systems developed at KTH are comparable with systems developed by commercial enterprises in terms of modifiability. The study also provided insight into an Enterprise Architecture where the systems have different development origins and how this could affect modifiability and analysis.

**KEYWORDS**

Enterprise Architecture, Enterprise Architecture Analysis, Case Study, Modifiability, Maintainability, EAAT, KTH, Public Sector, Information System, Modelling

## 1 ABBREVIATIONS

EA – Enterprise **A**rchitecture

EAA – Enterprise **A**rchitecture **A**nalysis

EAAT – Enterprise **A**rchitecture **A**nalysis **T**ool

IT – Information **T**echnology

ICT – Information and **C**ommunication **T**echnology

KTH – **K**ungliga **T**ekniska **H**ögskolan (Royal Institute of Technology)

VoS - **V**erksamhetsuppföljning **o**ch **S**tudiedokumentation (Student records and informations systems)

LADOK – **L**okalt **A**DB-baserat **D**OKumentationssystem (Local ADB-based Documentation system)

ITS – Enheten för IT-stöd och systemutveckling (IT department of Umeå university)

ITA – IT Avdelningen (IT department of KTH)

## 2 TABLE OF CONTENTS

1	ABBREVIATIONS.....	3
2	TABLE OF CONTENTS .....	4
3	INTRODUCTION .....	7
3.1	BACKGROUND .....	7
3.2	PURPOSE.....	8
3.3	RESEARCH QUESTIONS .....	8
3.4	DELIMITATIONS .....	9
4	RELATED WORK .....	10
4.1	ENTERPRISE ARCHITECTURE FRAMEWORKS .....	10
4.2	ENTERPRISE ARCHITECTURE ANALYSIS.....	13
4.3	MODIFIABILITY.....	13
5	METHOD .....	15
5.1	CASE STUDY .....	15
5.1.1	SELECTION .....	16
5.1.2	VALIDATION.....	17
5.1.3	ADAPTED METHOD.....	18
5.2	ENTERPRISE ARCHITECTURE .....	22
5.3	ENTERPRISE ARCHITECTURE ANALYSIS.....	25
5.4	ENTERPRISE ARCHITECTURE ANALYSIS TOOL.....	27
6	MODIFIABILITY.....	29
6.1	ARCHIMATE .....	32
6.2	P-OCL .....	33
6.3	META MODEL .....	35
6.3.1	META MODEL DETAILS .....	36
7	EMPIRICAL DATA .....	40
7.1	INITIAL.....	42
7.2	MODELLING .....	42

7.3	METRICS AND VALIDATION .....	43
8	ANALYSIS.....	46
9	RESULTS .....	50
9.1	KOPPS .....	56
10	DISCUSSION .....	61
10.1	THEORY AND METHOD .....	61
10.1.1	ENTERPRISE ARCHITECTURE .....	61
10.1.2	EAAT.....	62
10.1.3	MODIFIABILITY.....	63
10.1.4	META MODEL .....	65
10.2	EMPIRICAL DATA .....	66
10.3	ANALYSIS.....	67
10.4	RESULTS .....	69
11	CONCLUSION .....	77
12	FUTURE WORK.....	78
13	ACKNOWLEDGEMENTS.....	81
14	REFERENCES.....	82
15	APPENDIX - SYSTEMS.....	91
15.1	OVERVIEW .....	92
15.2	LADOK.....	93
15.2.1	NOUVEAU .....	94
15.2.2	MINA SIDOR.....	98
15.2.3	LADOKPING.....	102
15.3	NYA .....	104
15.4	ALUMNI COMMUNITY .....	111
15.5	TIMEEDIT .....	115
15.6	DIAK .....	118
16	APPENDIX – MODELS.....	120

17	APPENDIX – UPDATED PROCESSES .....	129
17.1	STUDENT PROCESS .....	129
17.2	PROGRAM PROCESS .....	130
17.3	COURSE PROCESS .....	132
17.4	SCHEDULE PROCESS.....	133
18	APPENDIX – DATA.....	134
18.1	EMAILS.....	134
18.2	WORKSHOP.....	144
18.3	INTERVIEW NOTES.....	146
18.4	TELEPHONE INTERVIEW NOTES.....	154
18.5	ESTIMATIONS.....	155
19	APPENDIX – p-OCL STATEMENTS .....	159
20	APPENDIX - EAAT CALCULATION HISTOGRAMS .....	161

### 3 INTRODUCTION

Sometimes claimed to be the most important part of any presentation; the introduction presents the compelling background regarding a problem, the reason why it has to be solved and what is not considered part of the problem. This introduction first introduces the background, followed by the purpose and research questions, followed by the limitations. Rounding up is the outline of the project report.

#### 3.1 BACKGROUND

Companies in the world today cannot compete in the modern market unless they have a business strategy model, preferably aligned with their IT strategy. All larger companies and enterprises need information systems in order to maintain their communication, information and not to mention coordination, among other things. Information technology can be represented in a company through several systems, processes, services and applications, such as customer support systems, accounting systems, etc. (1) These systems communicate and integrate with each other for the enterprise to function. However as the enterprise grows larger new systems, processes and applications are added, modified and removed, which might lead to confusion and disorder in the enterprise if these actions are not properly planned and executed (2). Information technology is everywhere and enterprises need to know how to plan for change.

When constructing a building you need design notes and blueprints to know which wall goes where, this is also true when it comes to companies and enterprises. A sort of plan of the organization, the methods, the systems and the processes (3) is needed to define objects and how they all can interact. A plan for where the figurative walls are, where you can build your roof i.e. a blueprint of your enterprise, an Enterprise Architecture (4). Enterprise Architecture can provide a modelling approach for presenting enterprise information systems, processes and architecture through models (5). The models provide abstract to detailed views over the systems, which gives a better understanding of the enterprise information technology, a blueprint of the enterprise.

Using these blueprints for analysis would be to perform Enterprise Architecture Analysis, which brings out better understanding of an enterprise, system objects and processes; this depends of course on the form of the analysis. Analysis, as implied, specified in certain areas offer insight into development or modification as to how the IT of the enterprise could be improved, altered or simply better understood in certain aspects. Such analysis can prove to be very valuable to a company or organisation in many ways (6), enterprise decision makers can use the analysis results for decision, while employees can get a better understand of the enterprises different functions.

As mentioned earlier the Enterprise Architecture provides an understanding of the enterprise through models and figures. These models need to be created using frameworks and tools. There are several frameworks available for Enterprise Architecture, such as DoDAF (7), TOGAF (8) and Zachman (9), and several modelling tools for creating models such as Metis and Aris etc. There is also the Enterprise Architecture Analysis Tool (10), developed at KTH, which is a modelling tool, and is especially useful for Enterprise Architecture analysis. EAAT can model relations between classes and attributes, attributes are different variables or values of systems or processes, and also has functions available for evaluation of attributes and relations. This of course is only a few parts of EAAT functionality and more details can be found in section 5.4. The EAAT calculations from the attributes and their relations make it possible to analyse different aspects of an Enterprise Architecture, such as how modifiable a system or parts of it could be.

Enterprise Architecture Modifiability; defines as the ability to change the components of your architecture (11). Depending on whether it is attractive to your enterprise or not, analysing modifiability in your architecture will tell you which parts of your IT is easier to modify. Certain systems or components might be harder to modify depending on, for instance, whether these components are highly coupled to other systems or other components, they might even be coupled in such a way that modifying them might involve the other systems or components. (11)

### 3.2 PURPOSE

KTH - The Royal Institute of Technology in Stockholm, Sweden is a University and as most universities, KTH has many organisations, groups and handles a lot of information. This makes KTH comparable with an Enterprise, and as well as an enterprise KTH needs IT to maintain their communication, information and coordination. This of course means that like companies and enterprises, KTH suffer similar problems as Enterprises, which means solutions available to enterprises should also be possible to apply to KTH. This degree project will be investigating a part of the current administrative systems of KTH as a case study.

### 3.3 RESEARCH QUESTIONS

When managing a large amount of students, such as the ones at KTH, it is important that the IT that support them and the systems that they use and depend on are working. But to manage these systems you must know, among other things, which and what the systems are and how they relate to each other. Some of the current IT and business process models, including other documentation, at KTH have proven to be out of date, inefficient and confusing. If they would want to improve or replace a system, the effects this could have on the rest of the IT is unknown.

This raises a question that needs to be answered:

1. What does the current information systems of VoS look like? And how are they connected to each other?

The need for change can come with introduction of new technology, information and other events can change the needs of the systems, in order to adapt to these needs one must modify the system, but in order to find what systems you can modify one needs to know what systems you are using, and what kind of consequences modifying them will cause. And even if you know which systems that you use, to find out the consequences of modifying them you need to know how the systems relate to each other.

This raises another question that needs to be answered:

2. What kind of results can be found regarding the modifiability in the VoS systems?

This information can then provide a foundation for answering more practical questions such as:

- If we remove system x, how could this affect system y?
- Which of the systems x, y or z should we expect to be hardest to modify, i.e. which would probably take more of the resources?



### 3.4 DELIMITATIONS

This degree project focuses only on the VoS department, and not the entire administration of KTH. The excluded parts, which are also part of the administration, are salaries, human resources, etc. The VoS department is responsible for support of statistics, processes and management of the central systems at KTH, these systems are used for supporting the administration of courses and programs at basic, advanced and research levels.

The VoS department of KTH were primarily interested in how the application systems that they use are connected, especially since these systems are critical for the entire institution of KTH. This meant that the focus of the study was to be committed to the systems and less relevant was the processes and technology. Since Enterprise Architectures are usually divided into different architectural layers this meant the study was to focus on the application layer with only small insights into the other layers. This was also because there was also a limited amount of time for the project which meant that focusing on more layers would mean less time spent on each layer. With the Enterprise Architecture focus of the study thus limited to modelling only the application layer the modifiability analysis thus had to be adapted to support this. This meant focusing on modifiable information related to systems and components. Another reason that the technology was considered less important was because of LADOK, the database to which many of the systems are connected, and is provided to KTH from an external developer for the main student information storage. Further the systems to be investigated were limited only to those which VoS are responsible for at KTH:

- NyA – Antagningssystem (admission system)
- KOPPS – Kurs och programkatalogen (Course and programme catalogue), information about available courses at KTH, including research courses and programmes.
- Mina Sidor (My Pages)
- Mina sidor för studenter (My Pages for students), students can view their results, print certificates, change address etc.
- Mina sidor för forskarstuderande (My Pages for research-students), research courses, certificates, change address etc.
- Mina sidor för anställda (My Pages for employees), employees can view student information, expected course attendees, administer exams etc.
- Diak, databas för internationella aktiviteter på KTH, används även av KTH Bostad (Database for international activities at KTH, also used by KTH residence)
- TimeEdit, schedule planner.
- Alumninät - Nätverk för KTHs alumner (Network for KTH alumni).
- LadokPing – A system retrieving information from LADOK.
- Nouveau – Administrationsverktyget för LADOK(Administration tool for LADOK)

Externally developed parts of the systems were not considered to be part of the actual modifiability analysis other than the coupling to them, i.e. metrics such as size and complexity were not pursued for those components since they were not considered to be within the limitations of the project.

## 4 RELATED WORK

The related work is divided into three different categories. Firstly, Enterprise Architecture Frameworks and Enterprise Architecture in general, the section also mentions work in Enterprise Architecture related to the public sector such as hospitals and governments, furthermore some different tool alternatives for modeling are presented. Secondly, different Enterprise Architecture analysis methods and papers are then presented. Thirdly, related work on modifiability and maintainability, and specifically ones dedicated to modifiability analysis are presented.

### 4.1 ENTERPRISE ARCHITECTURE FRAMEWORKS

Enterprise Architectures frameworks are different ways of communicating the resources of your enterprise; defining how the views and structure of the organization should be presented in models, standards, principles etc. in order to achieve Enterprise Architecture (12). As Enterprise Architecture has its presentation of enterprises concentrated in models, it would be almost impossible to grasp the architecture without knowing what the different elements in the various models mean. To understand what the model is all about, you need a standard or description of the elements represented. A weakness in Enterprise Architecture is its immaturity, one can loosely say that Enterprise Architecture is a rather new concept or rather it has many different sources (13). As such there is no common standard established on how to develop and construct Enterprise Architecture, and therefore different organizations have their own way of investigating and creating the models.

Ever since the 1980's there have been several different frameworks developed in order to present Enterprise Architecture (13). Some of these frameworks are more popular than others and have their different advantages and disadvantages. The first one created, according to Johnson & Ekstedt (13), was the Zachman Framework (9), which is still used today (14). This framework presents an enterprise through a matrix model and was developed based on commercial enterprises. But the frameworks are there to help develop an Enterprise Architecture, which does not necessarily have to be a commercial corporation, so choosing the correct framework to use is important (5). The Zachman framework focuses on models in forms of matrixes, where there are thirty-six cells which answers questions such as whom, what, when etc. together with other aspects such as data, function, network etc. This can be useful for determining different aspects of an enterprise and describing an Enterprise Architecture, but it is not really suited for presenting graphical system models.

The American Department of Defence also developed in the early 1990's another Architecture Framework called the Department of Defence Architecture Framework (DoDAF) (7). The main focus of the DoDAF framework is to primarily support the Department of Defence Chief Intelligence Officer (CIO). The framework differs from other frameworks by focusing on architectural data, rather than products. Though framework was developed for the American department of defence, this framework can be modified for commercial use. (15)

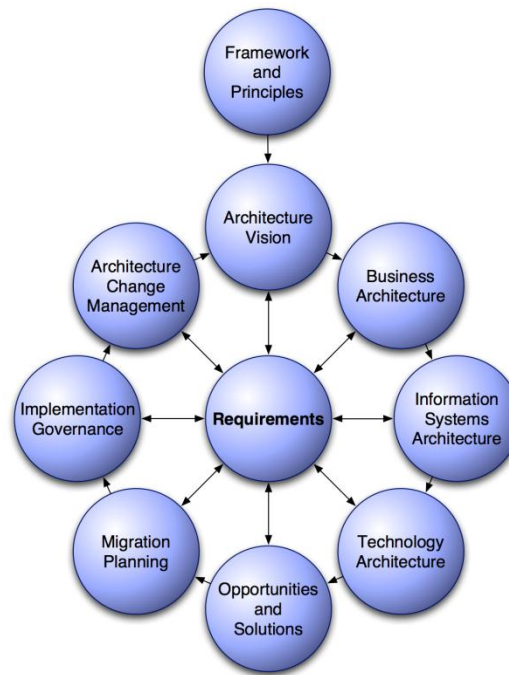


Figure 1: The TOGAF, a widespread framework used for establishing and maintaining Enterprise Architectures, figure from (16).

The Open Group Enterprise Architecture Framework (TOGAF) was first presented in 1995 (8). TOGAF presents an enterprise in four different architectures, business, application, data and technical architectures. Although TOGAF is rather abstract to these different architectures and what they should look like, TOGAF does present a more detailed approach in the process of enterprise architecting (13). Figure 1 presents the TOGAF model and the process to which the framework adheres to in order to establish an Enterprise Architecture.

The Enterprise Architecture frameworks presented previously are different processes, tools, standards etc. more for creating and maintaining an Enterprise Architecture. Enterprise Architectures concerns relations such as those between processes and technology to increase the productivity of the enterprise (12). They are relevant to this study as they present different alternatives for mapping your enterprise's architectures such as systems. Selecting the right framework depends on what kind of views, standards etc. that the enterprise intends to focus on.

E-Government or electronic-government is referred to the use of technology, Internet and web applications to allow information that is relevant to the public, business, employees and other agencies to be accessed more easily and efficient; using electronic commerce to cut costs of information delivery and improving government efficiency through IT use. (17) As the information systems at KTH are focused at providing and managing information to and regarding the students and its employees, and the universities of Sweden are part of the public sector and associated with the government. To share this information and allow faster delivery of public services requires that different systems and business processes share information between each other. In order to achieve this (18) presents different ways to achieve such transparent IT solutions using an Enterprise Architecture approach. However when modification in e-government and the public sector is not without its disadvantages and (19) shows several reasons of what can go wrong when employing IT solutions and investments.

There have been many different papers published on Enterprise Architecture which include implementation and analysis based on case studies in the public sector: (20; 21; 22; 23; 24; 25), some of which mention

the differences between the private and the public sector when using Enterprise Architecture. As such this is something that KTH needs to be concerned of in its Enterprise Architecture development.

A successful modelling language called ArchiMate, provides an extensive meta model for Enterprise Architecture, and will be used in this degree project (5; 26). Additionally this modelling language is possible to integrate with the TOGAF framework, which however is not used in this degree project since it entails a lot more and different tools, views and processes than what is used in this project. This however does not limit KTH from using the TOGAF framework afterwards if they choose to do so.

Different tools for creating models for Enterprise Architecture are available; some of the more advanced tools are focused on specific architecture. Tools such as Aris (27), Metis (28) and Metastorm BPA (29) can be mentioned. These aforementioned tools are all acceptable choices when modelling Enterprise Architecture but they lack analysis capabilities for system quality attributes and are not used in such context in Enterprise Architecture analysis.

## 4.2 ENTERPRISE ARCHITECTURE ANALYSIS

Different analytical methods for analysis on an architectural level are available, methods such as: ATAM (30), Architecture Trade-off Analysis Method, SAAM, Software Architecture Analysis Method (31), and formalism based on Bayesian Network statistics (32), among others. While these are all valid analysis methods they are not applicable to the Enterprise Architecture domain.

Explained in (33) are many different kinds of analysis applicable to Enterprise Architecture, these are analysis performed in different areas using different techniques. Several papers have been published regarding Enterprise Architecture analysis in different areas. Some of the areas are Information Security (34; 35), Interoperability (36; 20) and Modifiability (37; 38) the last of which is part of this project.

Different techniques have been adapted for Enterprise Architecture Analysis some of which are presented here:

- In (37) a method is presented for performing Enterprise Architecture Analysis using Architecture Theory Diagrams (ATD) with different properties to measure different information from Enterprise Architecture models.
- In (36) it is described as a step-by-step where combinations of elements are evaluated pair-wise and according to a certain scale, deciding which element of the pair that is more important, and by how much.
- Enterprise Architecture Analysis has been performed using XML. (39) The paper shows how static and dynamic analysis of architectures can be performed using XML, together with Rule Markup Language (RML).
- Enterprise Architecture Analysis has been performed using extended influence diagrams, which are influence diagrams extended with Bayesian networks to encapsulate probability. (40)
- Several papers have been published using PRM and formalism based on Bayesian Network statistics (32), among others. Probabilistic Relational Models specifies a template for probability distribution over an architecture model (1); also here using several different properties for what kind of analysis you wish to perform in the architecture.

Another technique used in several studies is the Probabilistic Object Constrain Language (p-OCL), studies such as (41) and (42). The p-OCL technique was also used in this degree project, the reason for this is due to the querying of information which can be made to the models, and is further described in its own section 6.2. It is here used to aid in the Enterprise Architecture Analysis which can be, as mentioned earlier, directed towards several different areas, one such area is modifiability.

## 4.3 MODIFIABILITY

Modifiability is the cornerstone of this degree project. Thus a segment regarding modifiability, its definition and use is presented in section 6.

There have been several case studies performed in software modifiability analysis which have been used in research papers such as (43; 44). In the field of Enterprise Architecture the interest of modifiability analysis is a hot topic of research. Studies regarding modifiability in Enterprise Architecture can be found in the several papers which have been published regarding software modifiability e.g. (45; 11). These are based on different studies and surveys done at different organisations.

Modifiability is also sometimes presented as maintainability and the costs of developing and maintaining a system (46). Several methods have been used for analysing software change and maintainability costs, but they do not specifically consider the impact of modification has on the surrounding software environment and rather within the system itself. Some of the methods also only focus on software development rather than modifying already existing systems (45):

- Constructive Cost Model COCOMO 2 (47), an updated version of the original COCOMO from the 1980's. For instance the new COCOMO 2 considers the flexible nature of the modern day's software development process and not only the now outdated waterfall model. However it is still one of the models that are only considered with development. The COCOMO model is used for determining the cost of change, i.e. modification, to a system.
- Architecture-Level Modifiability Analysis (ALMA) (48) is as the name implies a modifiability analysis technique specified for software modifiability at an architecture level. This method only considers software however and modifiability only when the case is: *"The modifiability of a software system is the ease with which it can be modified to changes in the environment, requirements or functional specification."* (48)
- Performance Assessment of Software Architectures (PASA) (49) is a method for performance analysis in software architectures, which is also used for analysis of other quality attributes such as maintainability (50). It uses software architectures for analysis basis and is mostly directed towards the developers of a system. PASA can be used for analysis in pre, during and post production of a system.
- Definition and Taxonomy for Software Maintainability by Oman (51; 52) defines modifiability as taxonomy. The taxonomies are used for identifying systems or architectures which needs to be modified and finding maintainability risks in the systems. The taxonomy is presented as a hierarchic tree for software modifiability, where every branch and leaf is an attribute related to maintainability.
- ISO/IEC 9126, an international standard for software quality in software engineering. (53; 54; 55; 56) This standard defines different qualities and what characteristics influence these attributes. Using these definitions the quality models can validate software's different qualities from a set of criteria and requirements. The standard provides definitions for measuring maintainability quality. (57)

Earlier mentioned analysis methods such as SAAM and ATAM can be adapted to be used for modifiability analysis but this is not necessarily the primary intention of the method (41). One thing in common with all these methods and techniques that none of them focus on modifiability in an Enterprise Architecture domain (57).

Considering all this the uniqueness of this degree project is presented in the project itself; i.e. a case study of Enterprise Architecture used to model some of the KTH administrative systems with analysis related to modifiability, with KTH being an institution in the public sector of Sweden. Also by focusing on Enterprise Architecture for system modelling by adapting the ArchiMate modelling language, and extending the ArchiMate language with modifiability analysis using system quality analysing techniques designed and performed in Enterprise Architecture context. This using a modelling tool which employs these analysis methods together with only software system metrics from Enterprise Architecture modifiability research results.

## 5 METHOD

This segment explains the method used in this degree project. Explained first is the general case study method to which this degree project adheres to. The following segment explains the Enterprise Architecture method used in modern enterprise planning and modelling. The third segment is regarding the Enterprise Architecture Analysis used for the analysis part of a case study. Lastly follows the tools used in order to achieve the results and to aid in the analysis.

### 5.1 CASE STUDY

Yin defines a case study in two technical definitions, firstly: *“A case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context, when the boundaries between phenomenon and context are not clearly evident.”* in (58) . That is, use the case study method where you have intentionally chosen to investigate related circumstances, because they are thought to be relevant to the study which is being performed. Yin then also distinguishes the case study method from other kinds of research methods such as surveys and histories, by comparing the requirements of those methods with case studies (58). A further explanation to selecting case study as the research method is explained in the selection section.

The second technical definition from Yin is concerned with data analysis and collecting strategies: *“The case study inquiry copes with the technically distinctive situation in which there will be in any more variables of interest than data points, and as one result especially relies on multiple sources of evidence, with data needing to converge in a triangulating fashion, and as another result benefits from the prior development of theoretical propositions to guide data collection and analysis.”* (58) This shows that a case study is not merely a method of gathering data and analysing, but actually a complete strategy for doing research.

A case study is a flexible design strategy, as the processes described can be performed iteratively, i.e. if you need more information or evidence, as the one you have has proven insufficient, you can plan for more data collection. However a point of inflexibility of a case study is the goals, if you change the goals, the case study become a new case study and not a change to the current one (59).

Yin in (58) and Runeson & Höst in (59) provides five guidelines for conducting a case study:

1. Design: This phase is the planning phase, where the research-goals are carved out as well as the delimitations are considered. The planning of the case study is also considered in this phase; how the study will be performed in terms of time and what limitation that needs to be set. A preliminary theory is established about the subject as well, in order to achieve the necessary knowledge to form a method to reach the goals. This phase concludes with a plan containing the aforementioned items. (58)
2. Preparation: Once the goals and a preliminary method have been created the method needs to be more specified. This is the phase where the theory is built, i.e. gathering information about the method and how to best carry it out. This phase is very important as the theory establishes how the analysis and data collection will be performed, and what is to be collected. This phase establishes the foundation for the validity of the case study, without the correct theory the evidence collected might not be the evidence required to perform the analysis. (58)
3. Collecting evidence: Data collection for any study should be very thorough. This is especially true for qualitative case studies. The data collecting was as mentioned, performed in a qualitative manner; by reading and evaluating documents, and by performing informal interviews, workshops and



- investigations. The quality of these methods, e.g. the questions for the interviews, depends much on the theory of the degree project. In order to validate the data the information should be collected from different sources if possible, or at least confirmed by different sources (58).
4. Analysis: The method for analysis is established in the theory part. This phase is to carry out the information from the evidence collected. The analysis has to be adapted to the information available, thus if no information was collected only poor analysis was possible. Should the situation that the theory gathered about the different attributes not seem to pass the bar for the analysis, there must be a way to backtrack in the information gathered; a chain of evidence (58).
  5. Report: No matter what form the report takes it is important that the report targets the right audience, maintains a comprehensive structure, and that the information is reviewed by informed persons such as the ones who have been part of the study. The general advice is to start the reporting early, already at a methodology section and not at the end during the analysis. (58)



Figure 2: The case study as a process.

But of course, they are only guidelines for a general case study, and the specific case study that is performed does not need to heed the order in Figure 2 to the letter and Yin explains that it is allowed, sometimes even necessary to collect more data even though the analysis has started. This study uses similar steps but executed differently in an adapted method which will be presented later.

Runeson & Höst further distinguishing case studies in (59) from other research methodologies by defining its characteristics as exploratory, i.e. finding what is happening, why it is happening and providing materials and conclusions for further research. There are qualitative and quantitative research methods, but that does not necessarily distinguish the two according to Yin in (58). A qualitative case study does not need to rely on quantitative evidence, and quantitative research methods sometimes use qualitative evidence in their studies.

### 5.1.1 SELECTION

Mentioned in section 3.2, there were not any specific system models that were up to date for the systems owned by the VoS department of KTH. In order to create and update these models, information had to be collected from various sources. This information had to be collected in a systematic way.

Since the degree project investigated part of the current administrative system of KTH, which has unique system architecture, a quantitative research method was out of question, such as a survey. Therefore in order to find a solution, a more in depth qualitative method was performed. According to Yin, this can be achieved through experiments, history or case study (58). The reason that a case study was chosen for this project was because it's not possible to consider the system architecture of KTH without the context. And since there will be no arbitrary control over the system as the degree project progresses, experiments cannot be conducted on the systems. Finally since the degree project will focus on contemporary events a history is not suitable (60).



Considering all this, a qualitative case study seemed like the best method. In a qualitative case study the chain of evidence needs to be very clear, following the collected data there should be no problems to understand the results and conclusions (58). This includes the decisions and steps of the study should also be presented (59). What is also common to the qualitative method is having analysis carried out in parallel with the data collection; this analysis could also bring out new insights into the study which then requires new kinds of data. In order to collect this data, data collecting methods such as interviews need to be updated and this update needs to be systematic in order to maintain the chain of evidence (59).

Using the case study research strategy is not without disadvantages however; Smith mentions in (61) two weaknesses of the qualitative case studies; first, that they might bring the researcher too close to the problem, which might cause problems when it comes to objectivity. Smith also mentions (61) the political acceptance of the case study research method; that research performed in a case study manner might have trouble to explain the results and benefits of the research, since case study is not always politically accepted.

On the other hand, Smith then explains that the strength of the case study lies in how the case study is used and the logic and validity of the analysis (61).

---

### 5.1.2 VALIDATION

The information gathered in the case study needs to be validated. Validation denotes the reliability of the results, how much of the data gathered that is true and not constructed from the researchers' point of view (59). The validity of the case study needs to be maintained, and is an important part of the case study results (61) and starting the validity at the analysis phase is too late (59).

Yin describes in (58) four different kinds of validity which Runeson & Höst in (59) extends to:

- Construct Validity: Establishing correct operational measures for the concepts beings studied. If the data collection methods however are ambiguous or interpreted incorrectly there would be a problem with the validity (59).
- Internal Validity: Establishing a causal relationship, whereby certain conditions are shown to lead to other conditions, as distinguished from spurious relationships. Only for explanatory or casual studies only, and not for descriptive and exploratory studies. A problem to this validity is that if there is an unknown factor that affects the study it would compromise the validity (59).
- External Validity: Establishing the domain to which a study's findings can be generalised (58).
- Reliability: Demonstrating that the operations of a study – such as the data collection procedures – can be repeated with the same results. What could become a problem in this kind of validity is if it was unclear how the data was collected (59).

Of those validity kinds the ones that are the most relevant here is the reliability and construct options. However as IT is an iterative process, if someone were to perform the same study at a later date some of the systems might have changed, documents remain out of date, and people who were interviewed might have quit. Thus the validity will be harder to verify as time passes, but the constructed operations and methods must be clearly defined and understood, and not due to the researchers subjective view.

As described earlier the case study is a flexible design and if there is a part missing in a step of the process one can usually iteratively return to that step and repeat it. Thus if there is data that is inconsistent, invalid or missing then the case study can return to the data collection step and collect more data in order to vali-

date the data properly (62). The validity of the information gathered affect how the information can be further used once the case study is over (59).

There are several ways to improve the validity of the case study however, such as triangulation, maintaining detailed case study protocols, peer reviews, having collected data reviewed by the case subjects and researching contradicting cases (59). Triangulation of sources, data types and researchers is a strategy used in order to research a subject from multiple angles. Several methods are usually used in order to improve validation (60).

This presented how validity can be maintained in a case study in general, however as mentioned case studies can be flexible in their execution and this project used its own adapted method.

---

### 5.1.3 ADAPTED METHOD

As presented earlier a case study can be described in five steps. This is as previously mentioned an iterative process, and one does not have to completely finish a step in order to move on to the next, e.g. you can return and collect more data if needed as your analysis is underway. As can be seen in Figure 3 the adapted method for this case study is not the same as the one presented in Figure 2; the original guidelines in separate segments. The full process is depicted later in Figure 7.

**Design:** Planning of the case study and goal determination and delimitation.

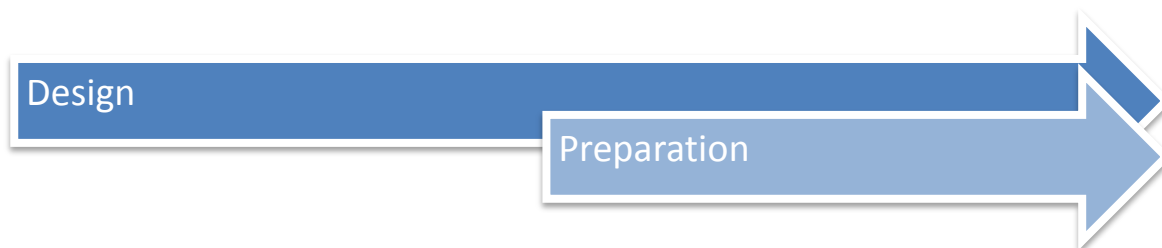


Figure 3: The adapted design part of the case study.

Through meetings and feedback from the supervisors the goals of the project was established. This is then further improved through limitations. Although the goals are established and kept static through the entire research process, the limitations are more flexible and as such priority to the system research fluctuated, making some systems more of a priority to model than others.

**Preparation:** The procedure of collecting the data is determined, through study of theory.

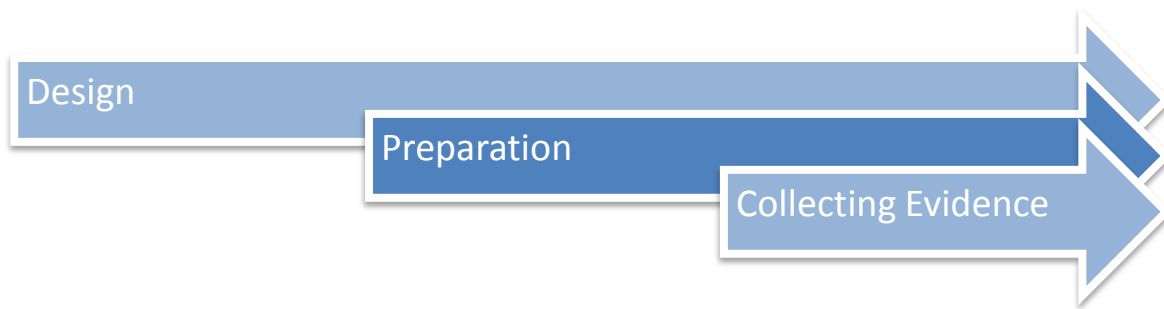


Figure 4: The adapted preparation part of the case study

The theory established in this project takes its final form in a meta model (section 6.3). A meta model consists of entities, relations connecting entities and attributes of variables related to the entities (63), these are abstract instances of that provides a blueprint for the business and system models (11; 64). This includes the relations and dependencies of the different layers of the organisation. The ArchiMate modelling language provides an extensive model that spans an entire enterprise while still keeping the layers and aspects separated, from the technology layer to the business process layer. Creating a new meta model from scratch is a project in itself, therefore this degree project adapt the ArchiMate model in order to establish the necessary theory foundation to perform the modifiability analysis. When creating a meta model in an effort to perform analysis you need to consider what kind of analysis that the meta model will be used for. This is because the attributes required in the model to analyse modifiability differ from the attributes required to measure e.g. security or interoperability (11). ArchiMate does not contain any attributes for analysis and thus the model has to be extended in order to include analysis.

Using the limitations from the design and planning phase and the theory collected. Extending the theory phase in accordance to the flexibility of the case study method, early amounts of data collection can be started. This reason for this is in order to find what kind of information is available in order to perform the analysis and thus better adapt the meta model to its purpose. This also reflects to the planning phase and was taken into consideration, and if there is limited information available in certain areas then the data collection will have to focus elsewhere. This phase is finished once the research method and knowledge of how to collect the data required has been established, this was then concluded and presented in a research plan.

**Collecting evidence:** Using the procedure for collecting data to gather the evidence needed.

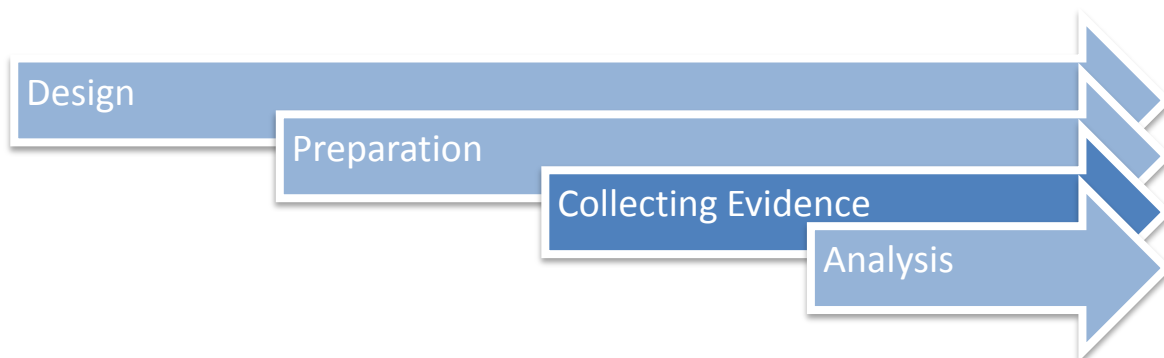


Figure 5: The adapted evidence collection part of the case study.

The data collection was divided into several minor phases, in accordance to the limitations. That is, due to the abstract nature of the project different data had to be collected in a specified manner. First the data for constructing the models had to be collected, and once the models were done the different metric needed for the analysis could be collected. This was because of the lacking current information already available at KTH. Without having any models of the systems, more specific questions could not be asked about the different components which realize the system. In the evidence collection validity was considered by referring to the validity presented in section 5.1.2, and heeding to Yin's advice on how to maintain validity in those aspects.

Yin says the strength in interviews is the targeting focusing on the topic and insightful perceived casual inference (58). The weakness in this is: Bias due to poor questions, bias due to responses, inaccuracies due to poor recall, reflexivity interviewee gives what interviewee wants to hear. These issues are addressed by creating asking non leading questions and documenting the interview, and in interviewing the right personnel. That is, people related to the systems.

Further Yin explains that validity weaknesses in the documents are irretrievability (58), selectively biased if the collection is incomplete, reporting biased based on the author, and access may be deliberately blocked. Irretrievability is addressed by relying on other sources than documentation, same with access. Selectively biased and reporting biased is addressed by verifying the accuracy by comparing with results from interviews.

In parallel one can also maintain the validity by following a strict chain of evidence, i.e. explicit links between the research questions and the conclusions drawn. This is aided by carefully documenting all the materials collected in order to maintain the evidence collected.

In addition by using triangulation, i.e. establish reliability by using triangulation, the validity of the project can also be maintained. Triangulation is, as mentioned earlier, accomplished by using different sources which points towards the same evidence. Thus in order to achieve this multiple sources for information was pursued during the data collection.

**Analysis:** Analysing the evidence collected in order establish an outcome of the goal.

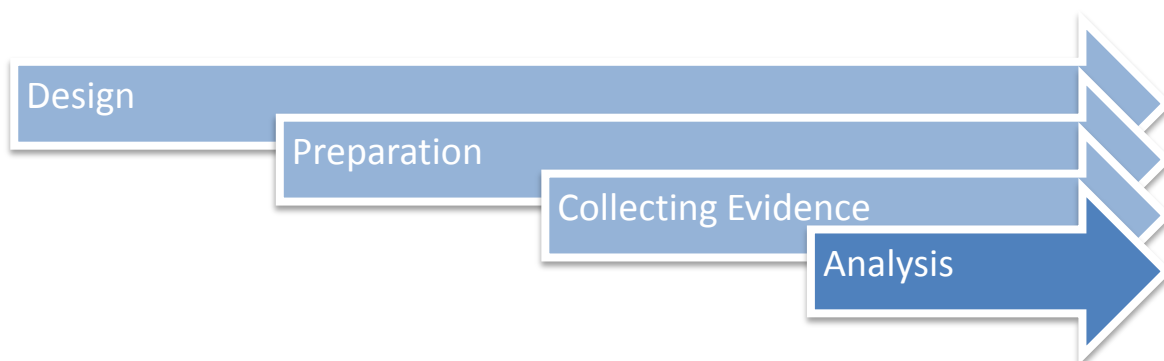


Figure 6: The adapted analysis part of the case study.

The meta model which was created during the theory or preparation part is essential in the analysis phase. Using the meta model as a basis for the analysis and the information gathered from various sources, the different system models are constructed. Using these models and software metrics, calculations are performed using a calculation tool, and conclusions about the different systems can be drawn. Relevant enti-

ties in the meta model are equipped with attributes; different properties which can be found in the analysed systems.

**Report:** Reporting the gathered findings.

Reporting was done continuously during the project. This to make sure the information was not forgotten between phases, i.e. so that once the analysis phase started, all the theory had been forgotten.

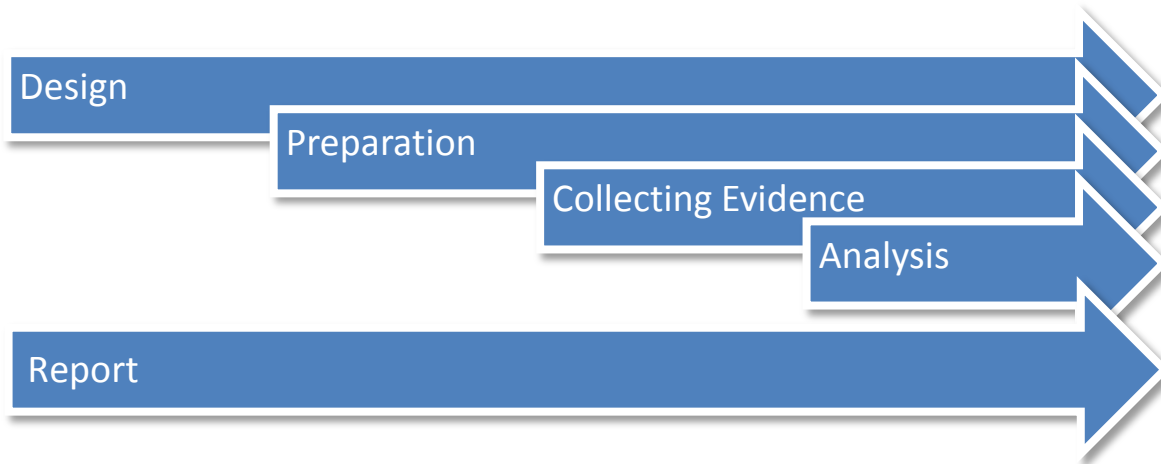


Figure 7: The adapted the case study method.

The design, preparation, data collection and analysis were all performed at different stages as according with the case study but never considered finished; however the reporting was done continuously from the start in different ways, i.e. through diaries, notes, a project and a research plan.

Different models were created containing the different entities, relations and attributes for all the systems that was part of the degree project. Aside from the models for the degree project, information from several documents was also collected into a single system description document which would complement the models. Added to these were also notes and pictures from interviews workshops and investigations. This document was, together with the continuous reporting, what was to become this report and the models which were created can be found in the APPENDIX - MODELS, models which depict part of the enterprise's application architecture.

## 5.2 ENTERPRISE ARCHITECTURE

Enterprise Architecture is generally agreed to have emerged in the late 1980's (13). This as a response to the importance of information systems increasing and developing into core parts of the enterprises. As mentioned by Johnson & Ekstedt in (13) Enterprise Architecture has a very brief history and therefore the term can be ambiguous in its meaning, i.e. what should be included and how things should be addressed. Several different definitions of Enterprise Architecture can be found in (12).

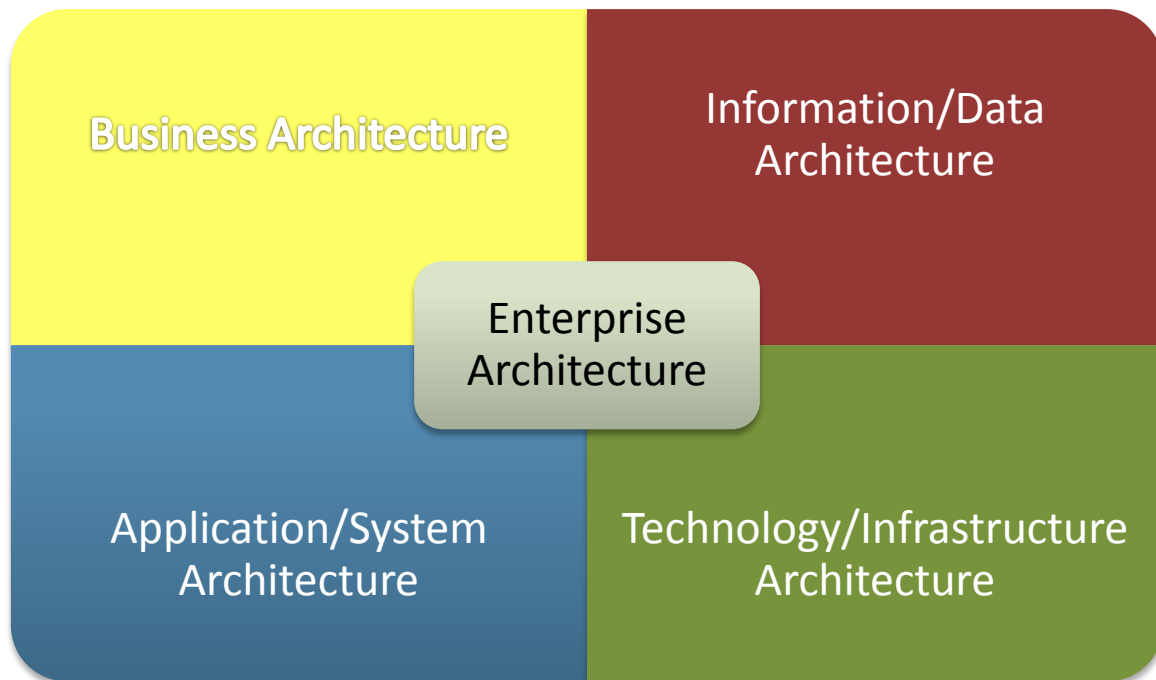


Figure 8: Enterprise Architecture works as a common language between different architectures.

In (4) Lankhorst defines Enterprise Architecture as: *“a coherent whole of principles, methods, and models that are used in the design and realisation of an enterprise's organisational structure, business processes, information systems and infrastructure.”* this states what is used to create an understanding of what it is that realizes an enterprise. What makes Enterprise Architecture unique from other enterprise information systems management techniques is the importance and focus on models and graphic representations of the enterprise to provide a holistic point of view (3). This by using different models to present itself as a communication language between the different architectures of an enterprise, as depicted in Figure 8. The view is of course very important to the stakeholders (4), the parties who are interested in the enterprise or system. Different stakeholders have different interest in an enterprise, this includes as to what the various outcomes or possibilities are available (4). As such an Enterprise Architecture should be possible to explain to all stakeholders, at least the part of the Enterprise they are interested in.

In the world today, new software and systems are created and developed all of the time, old systems are improved and extended into the new ones (13). Enterprises that expand or modify their business, for instance to meet new customer demands or business goals; need to plan for integration with their systems. When these systems use different standards and terminology a lot of confusion which could impede the enterprise progress might occur (2). To achieve further business success, enterprises today need plans and knowledge that encompass a broader area than IT and technical infrastructure, plans and knowledge of how the entire enterprise is built, a view of the entire enterprise's architecture (4). This architecture can

then be used as an underlying foundation for different enterprise decisions. A well-constructed Enterprise Architecture defines and translates the importance between the business strategy and the system functions (3). Thus having a good Enterprise Architecture that promotes the business strategy and aligns IT to it results in better profit for the corporation or organization (2) through various means, such as better return on existing investments and more efficient processes (5).

So what is included in Enterprise Architecture? The definitions can vary (e.g. (5) & (14)) however Enterprise Architecture usually includes several different layers of the enterprise, which can then be further divided into sub-layers (26), this project followed the scope of an enterprise divided into three different general layers (12), which are:

- Business Architecture. Generally where you find the effects of the business strategy, processes, functional requirements. Things represented here are usually business actors, network relations and other services/products for external customers and suppliers (14) (26) (33).
- Application Architecture. This layer supports the business processes and business functions in the enterprise with different systems, applications and services (14) (26) (33).
- Technology Architecture. This layer contains the infrastructure information to realize the services available in the higher layers. Infrastructure such as storage, hardware and communication points (14) (26) (33).

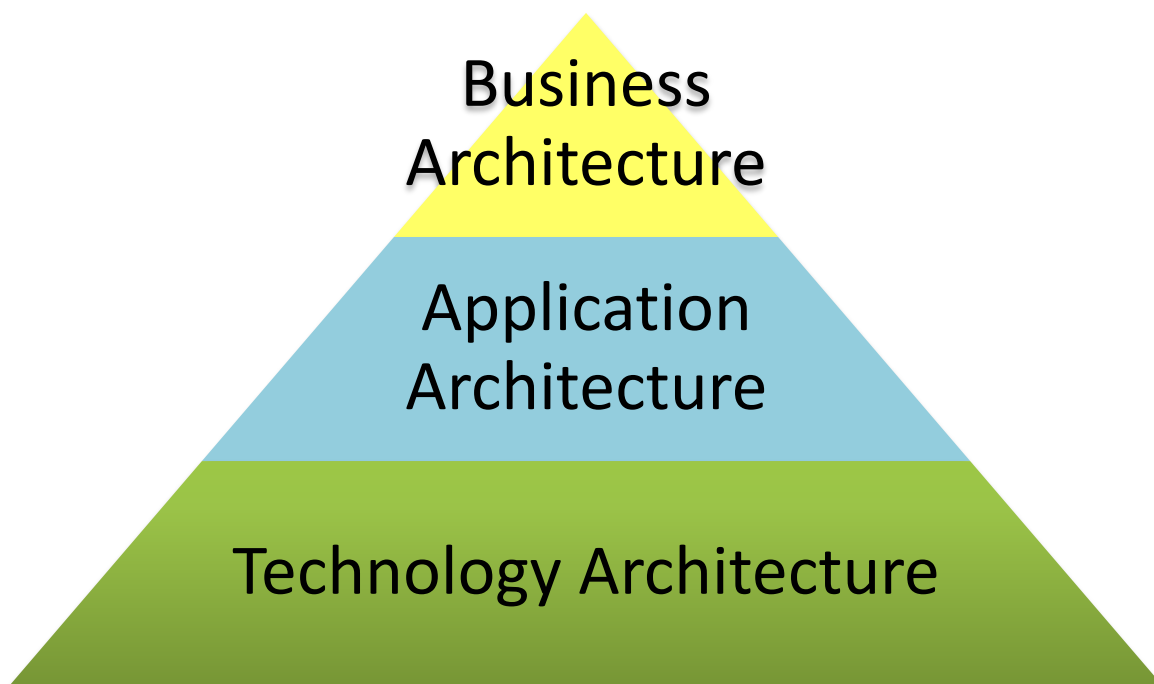


Figure 9: An Enterprise Architecture pyramid representing the layers of Enterprise Architecture.

With these layers Enterprise Architecture can model an entire enterprise on various scales and with different levels of detail and especially the relationships between these levels. As mentioned, some Enterprise Architecture frameworks further divide the different layers into sub-layers, e.g. Application architecture is sometimes divided into information layer and application layer (13), (14), (8), (33) and the business architecture can sometimes be divided into business strategy and process layers.

All the various enterprise information systems management techniques have their own advantages and disadvantages, but some of the reasons as to use Enterprise Architecture are:

- The models and methods of Enterprise Architecture provide an impact view of various stakeholder concerns (4).
- Models are usually much easier to understand, which might benefit the communication (13).
- Flexibility and adaptively can be improved in the architecture while still maintaining the enterprise business core (3).

The reason Enterprise Architecture is used in this degree project is because of the advantages which presents themselves in models (13). As explained Enterprise Architecture sets itself apart from other enterprise information system management methods in the models, models which are used in order to increase communication between the different people, departments and organisations in an enterprise. Enterprise Architecture models can also present the enterprise's structure and information, and can be mapped and often optimized in order to drive the enterprise forward. Par example the enterprise could find using these models and their relations that there are two different systems performing the same function and thus move the enterprise organization or department to use only one of them and reduce costs. The models can as mentioned also present more than applications; such as business processes, technology mapping and system models, system models which to create is one of the goals of this project. Using Enterprise Architecture models to reach one of the goals, and by extending the models with information they could also be used for quality analysis.



### 5.3 ENTERPRISE ARCHITECTURE ANALYSIS

To know how introduction, change or removal of systems and information will affect an enterprise can be challenging, since reading the future is not really available... yet, however one can make an educated estimation as to what consequences will transpire, and from this make a decision of what would be most beneficial, this is called Enterprise Architecture Analysis (13). In (13) Enterprise Architecture Analysis is exemplified as a machine (Figure 10) for decision making, where models or specifications and different scenarios are fed to the machine, and the machine analyses the different decisions and presents what outcomes the different decisions have. This is to be done before the change, to report the results after the decision would, according to (11) simply be book-keeping. Although there are a lot of different Enterprise Architecture methods available today, they do not really agree on what information and architecture model should contain (37). Even though having a good looking, easy to explain model, which captures and explains the different parts and relations of your enterprise. Having the model providing even further insight into your enterprise would be even better; to provide a view of how different scenarios, e.g. changing database provider or introducing a new process, might affect the rest of the Enterprise Architecture (13).

What parts of your enterprise require the most resources? What parts work best with others? Which provides the best quality? These are the questions that Enterprise Architecture Analysis can answer, given the right kind of criteria and information. To find this information about what is most important about the analysis, you need to find what goals you wish to achieve with your analysis (65), the specification of analysis as presented in Figure 10. Thus if your goal is to understand which parts of your enterprise has the most operability, an analysis of security might not say as much as an analysis of interoperability (37).

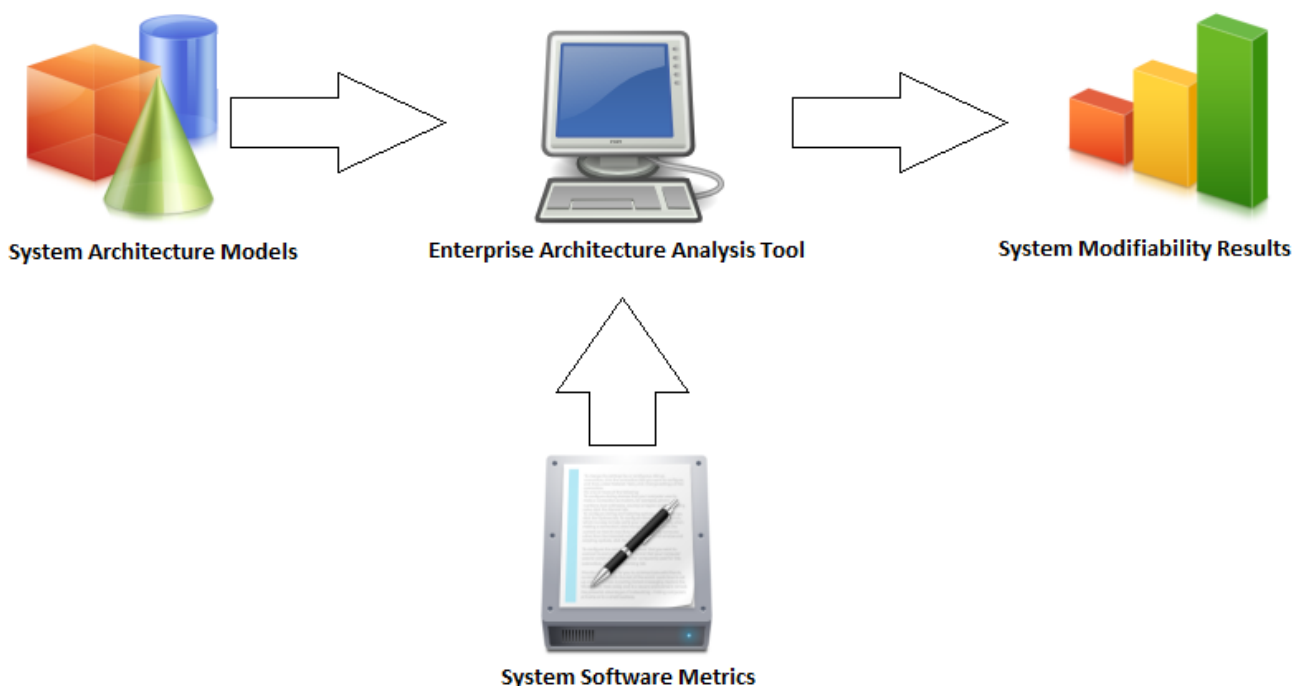


Figure 10: A picture which represents the Enterprise Architecture Analysis as used in this study, picture adapted from (40).

Uncertainty can be found in all analysis; this is because to achieve complete certainty is almost impossible. In (66) they even mention that some research and experiments are less accurate if they do not take heed to uncertainty. To ignore uncertainties, especially ample uncertainties, might not be an option when performing decision-making (66). There are several kinds of uncertainty, (40) separates them into four different kinds:

- Definitional uncertainty, some terms have different meaning to different people, which expresses uncertainty. As an example somewhat related to this degree project would be a Ph.D. student, to some administrative department of a university this is a student, whereas to the payroll department this is considered an employee because they have a salary.
- Theoretical heterogeneity. Enterprise Architecture does not have a framework standard, and different frameworks encapsulate different things into the architecture.
- Empirical uncertainty. Data collected and models that have been created might contain errors, or perhaps be out of date when used.
- Casual uncertainty. Casual events are not necessarily guaranteed, it is uncertain that a student which studies actually pass the exam.

Uncertainties need to be taken into consideration when performing an analysis, but preferably the idea is to remove uncertainty by performing the analysis, to get a better understanding of the phenomenon you analyse. Uncertainties are usually expressed as probabilities (66), thus certain analysis methods which express uncertainty express it as probability.

Enterprise information is required in order to perform an analysis, and different kinds of information, or attributes, is relevant for different kind of analysis (11). Finding values for the measurements can be done through different ways such as, expert opinions, case study or literature review, etc.

In order to determine the modifiability of some the critical systems at KTH, this degree project extends parts of the ArchiMate modelling language with probabilistic uncertainty and attributes related to modifiability which are instantiated using empirical data and calculated to perform analysis. To this data is also added a certain degree of uncertainty when necessary. This uncertainty is expressed as percentage using the *coefficient of variation* in normal distribution, e.g. when there is high uncertainty; the *variation* of the distribution is then expressed as a low *coefficient of variation* to the *mean* value.

So by taking Enterprise Architecture models, and extending them with analysis capabilities through attributes and possibility for uncertainty assessment, both the goal to create system models and the goal to analyse modifiability can be reached. So in order to lighten this modelling and calculation process this project used a tool for Enterprise Architecture modelling and analysis.

## 5.4 ENTERPRISE ARCHITECTURE ANALYSIS TOOL

The Enterprise Architecture Analysis Tool is, as the name applies, a tool for modelling and analysing Enterprise Architectures. This degree project only offer a basic explanation of the Enterprise Architecture Analysis Tool (EAAT), as the specifics of the tool could be concluded into a paper itself. Also since the tool was still being developed when this report was written, further extensions and functionality might have been created. However some more details of the tool's functionality and specifics can be found in (6) and (42).

The basic functionalities of the tool however are represented in the two aspects of the tool, the abstract modeller and the concrete modeller (Figure 11 presents a screenshot of the concrete modeller of the tool in action). These basic functionalities are among other things, modelling various objects found in UML (67) such as, *entities*, *relations* and *attributes* (10).

Additional to the basic functionalities available in EAAT are the *attribute relationships*. These relationships signify the probabilistic dependence between two attributes, which can span between several entities, thus if a child entity attribute is altered, this relationships signify that change occurs in the parent entity as well (10).

The models, which is one of the main advantages, of Enterprise Architecture, can vary from either being small focused models of a small system, or extended large scale enterprise models spanning over several processes, systems and servers. Drawing these models by hand can be quite tedious, if not impossible to some extent. Tools for drawing models however are already widely available, different tools for drawing UML models have existed for years, e.g. (68).

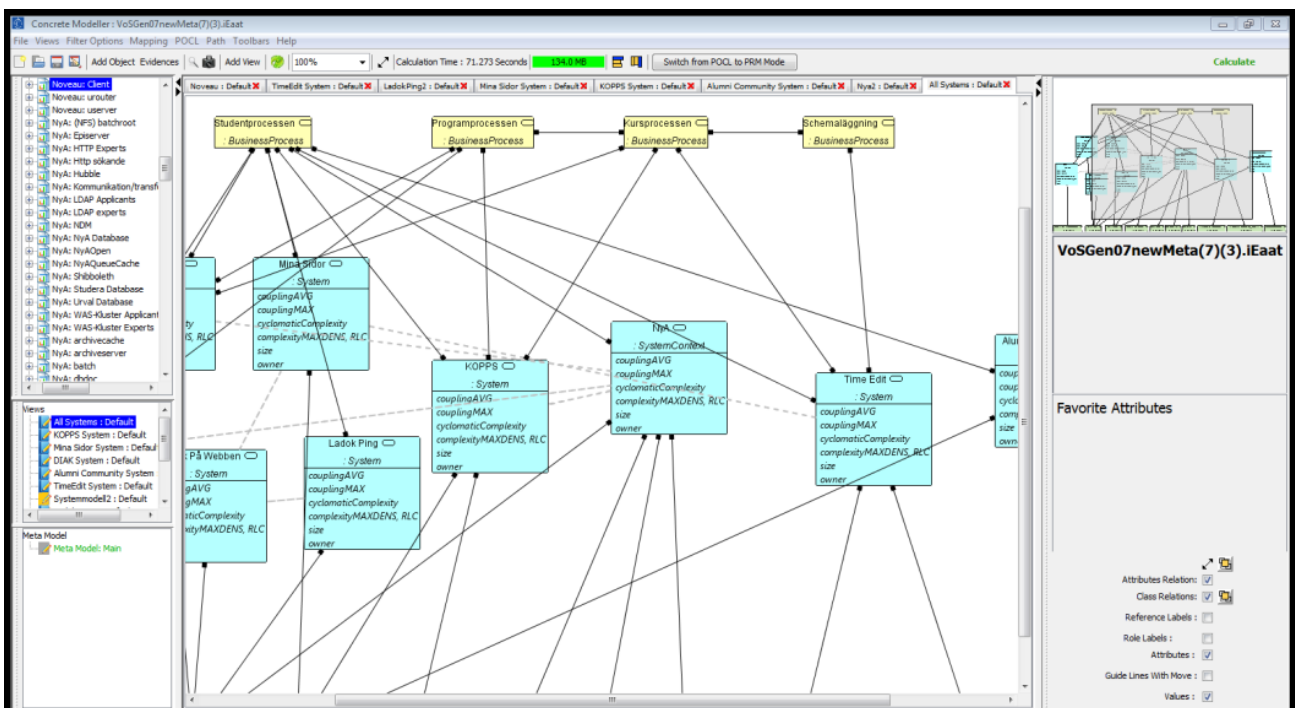


Figure 11: A screenshot of the EAAT tool drawing a model.

EAAT differs from earlier mentioned tools through its extensive calculation functions (6), using either PRM (Probabilistic Relational Models) featuring Bayesian networks, or p-OCL (Probabilistic Object Constraint Language) statements for an analysis, calculations can be performed in the tool. The empirical data for the-

se calculations are used as the evidence input which can be used as input in different format e.g. XML (10). This alleviates, among other things, the need for extended calculations performed outside of the modelling. EAAT is being developed at the Industrial information and Control Systems (ICS) department of KTH.

In order to express uncertainty the EAAT adopts several different probabilistic distributions which are commonly used to express uncertainty in statistics. These distributions are during the time of this writing: *normal* distribution, *binomial* distribution and *log-normal* distribution. The distributions can be used for various different kinds of analysis and also depending on the data that was provided. In EAAT there are also histograms which can be used to present the distributions used such as Figure 12.

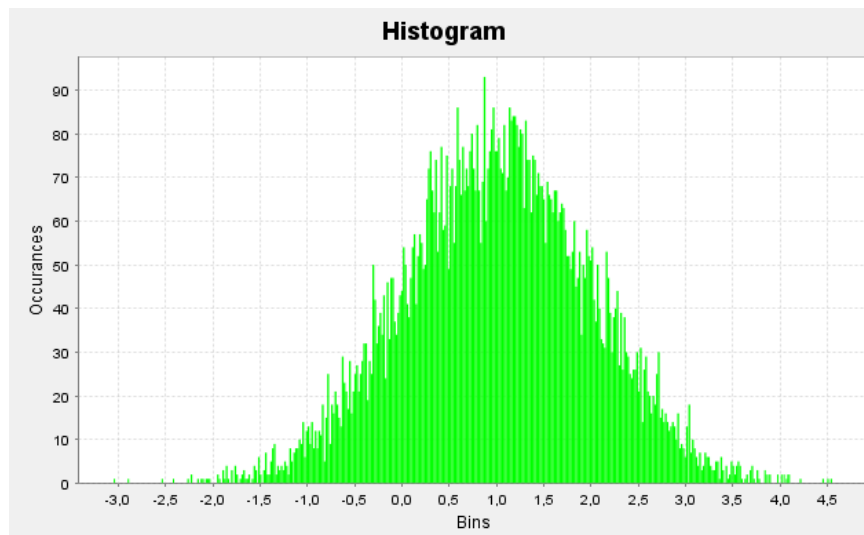


Figure 12: A histogram from EAAT depicting a normal distribution identifiable by its bell shaped form.

The calculations that can be performed with EAAT provide information regarding the unique quality analysis which was presented in the previous section. When presenting these calculations as histograms, different important values such as *variance* and *mean* are also presented with them. These histograms can be used to analyse the uncertainty and other things using already well-established mathematical algorithms, such as par example the *coefficient of variation*.

EAAT provides, in its multitude of functions, also a way to present different parts of a model while hiding the rest. This makes the drawn models possible to contain their specific calculations, attributes and relations, while still providing a more specific viewpoint of the model, and thus the models can be made more presentable. This is implemented through various views (10). EAAT was used in this project to create the models of the different systems and calculate the modifiability of the systems using a probabilistic extension to ArchiMate for analysis. With the modelling and analysis method defined to reach the goals, a tool to model and calculate, the only thing missing is finding the attributes or metrics of the systems which affect modifiability and how to calculate and interpret them.

## 6 MODIFIABILITY

This section presents the theory behind modifiability, the ArchiMate modelling language and the p-OCL language, which are all tools used to create the meta model used in this degree project.

The reason why modifiability(also called maintainability (45)) analysis is an issue is because a lot systems today are usually interconnected or integrated, which means if you try to change a single system this might correlate effect upon another system which you had no real intention to change. When there is a system or process that is in need of change or improvement, because it's becoming a bottleneck for the enterprise, or the functionality of the system is becoming irrelevant (69). However there could be several facts that could prevent from initiating a change, these facts might range from simple ignorance of system functionality to software access, i.e. you do not own the system. These are all attributes more or less relevant to the modifiability analysis. Removal, adaptation, extension or restructuring of the system software is dependent on the system modifiability (70). Important to mention is the fact that software or system modifiability is not only relevant during system development, but a prominent fact during the entire management process (45).

(69) and (11) presents extensive insight into what entities, relations and attributes that can be used in order to assess modifiability. The modifiability meta model which is presented ( Figure 13) forms a basis for what different attributes are relevant for modifiability analysis throughout all the Enterprise Architecture layers, and is not focused on a single layer such as application.

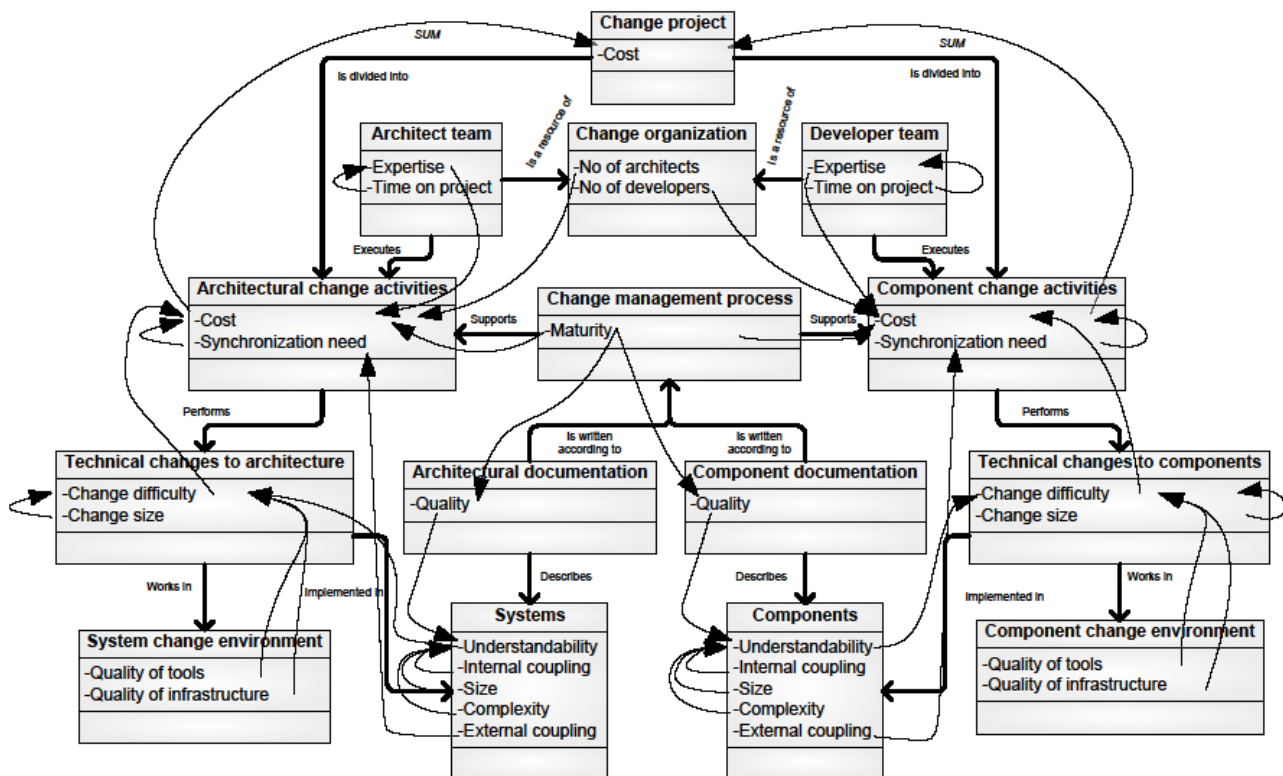


Figure 13: The modifiability meta model (11)

Using this meta model as a foundation for modifiability analysis, it was then further adapted to fit the limitations and goals of this degree project. The attributes from the modifiability meta model which became most relevant were the ones found in Systems and Components, as it is there which the project was focused. The attributes which become relevant from the meta model are:

- Size, this is usually equivalent to the lines of code that a system or component has, and a system being realised by components. (71)
- Coupling, the number of couplings between components of a system, both internal and external. (72)
- Complexity, a member of the definitional uncertainty mentioned earlier, complexity can be defined in several different ways: Halstead complexity (73), Lines of Code and McCabe's Cyclomatic complexity (74).

These attributes are facts largely established to be relevant to software maintainability and reusability which has been established in (11), (45) and (69). The papers also explain what might affect the modifiability but those attributes are outside of the limits of this degree project.

Size affects modifiability, and although there are no specific limits to size in order to assure high modifiability, the size of a component does affect the difficulty of conducting a change (75). As mentioned size is presented as Lines Of Code (LOC) or Source Lines Of Code (SLOC). SLOC is lines of code where certain code lines are not in the calculation, such as comments or empty lines.

Coupling is defined as how different components and systems communicate and are connected. Components with a tighter coupling usually results in more error prone modules (72). The tighter the coupling between components the more likely a change to one object will affect the other (76). This degree project uses the coupling measurements found in (72) to determine how tight or loosely coupled a component is:

- Content coupling – A module x branches in and changes variables and logic in another module y.
- Common coupling – A module x refers to the same global data as another module y.
- Control coupling – A module x passes a parameter to module y to control its behaviour.
- Stamp coupling – Modules x and y accept the same kind of record type, but only parts of the data are used.
- Data coupling – Module x and y communicate through relevant parameters.

Using couplings further down in this list incorporates looser couplings, i.e. data coupling is the loosest (except for no coupling) and content coupling the tightest. The looser the coupling the less chance that any modification done to a module will affect another coupled to it.

The algorithm established in (72) used in this project for measuring couplings is:

$$M(x, y) = i + \frac{n}{n + 1}$$

Where the measurement M between module x and y is equal to the tightest type of coupling i added with the number of interconnections n divided by n + 1.

Complexity as mentioned before can be measured in several different ways, however in this degree project the chosen complexity measurement is McCabe's Cyclomatic Complexity, a value calculated from the number of execution paths in the source code. The cyclomatic complexity was chosen because it is a common and widely accepted measurement for software complexity. Complexity values are recommended to be below 10, but exceptions can be made if they are reasonable, and modules with complexity up to 15 have been proven manageable (74; 77; 78).

Cyclomatic Complexity is calculated as:

$$v(G) = e - n + 2$$

Where  $v(G)$  is the cyclomatic complexity of flow graph  $G$  of a program,  $e$  is the number of edges in  $G$ , and  $n$  is the number of nodes in  $G$ .

Cyclomatic complexity density (79) and Relative Logical Complexity (80; 81) is shown to have a predicting effect upon software maintainability, and is derived from Cyclomatic complexity and size, which are both metrics measured, thus this value can be calculated as another indicator of maintainability. It expresses an amount of logic in terms of the amount of source code. Larger modules tends to have larger complexity values, this metrics defines the complexity in relevance to lines of code. The higher the cyclomatic density values the more logic per lines of code and thus the harder to understand the code. Too low density values however there is abundance of source code in comparison to logic. Cyclomatic Density is defined as:

$$C_{MAXDENS} = \frac{v(G)}{KNCSLOC}$$

Where  $C_{MAXDENS}$  is cyclomatic complexity density,  $v(G)$  is the average cyclomatic complexity as defined previously and  $KNCSLOC$  is the thousand lines of executable code, i.e. size.

Using (82) as reference for limitations regarding the cyclomatic complexity density values; suggested values should be between 0,14 and 0,42 in order to be simple and comprehensible. Anything below should reasonably be considered too low, and anything higher can reasonably be considered too high.

The advantage of analysing modifiability gives as mentioned insight into what systems or components whose modification might correlate effect upon other systems. By analysing modifiability of the systems conclusions can be drawn as to how much time and resources are needed to be invested to modify them (11). In addition, modifiability analysis can also be used to provide a sense of priority as to which systems to extend, remove or upgrade. Therefore a modifiability analysis can be useful as a basis for further system development and planning which is what was requested by KTH. And although modifiability can be found in the entire enterprise, as mentioned this degree project will focus on the modifiability of the systems and their components, i.e. the application layer.

## 6.1 ARCHIMATE

Unified Modelling Language (UML) and other successful ICT modelling languages are exceptional at defining detailed processes and systems, however they have limited capability when modelling relations between these processes and systems, including process to process. Therefore they are not optimal when it comes to modelling Enterprise Architectures (83).

ArchiMate is the product resulted from the combined effort of The Open Group, and several companies, specified in (5). ArchiMate is intentionally similar to UML (67), but more lightweight and scalable, and expressive enough to provide all layers and aspects of an organisation in an Enterprise Architecture model (5).

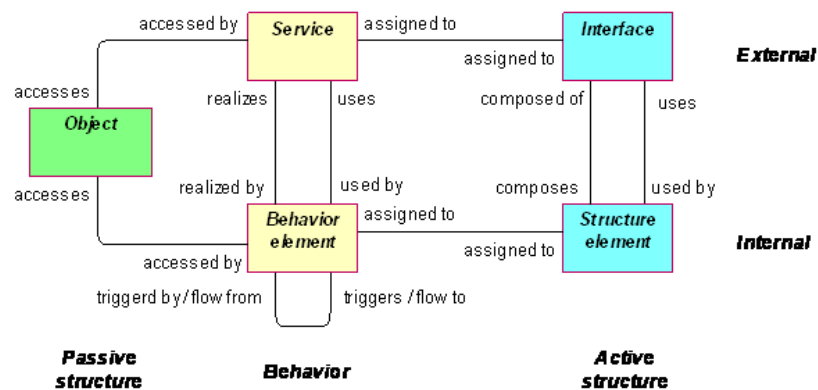


Figure 14: The internal and external view together with the different aspects of ArchiMate (5).

In addition to the three general layers or architectures of Enterprise Architecture described above, the business, application and technology architecture, the elements in these layers can further be divided into three aspects: information aspect, behaviour aspect and structure aspect (83). These aspects also represent the more passive parts, the behaviours and active parts of the enterprise, where the active structure elements (actor, application components, etc.) propose behaviour on the passive elements (5) or data objects.

Added to this is the external and internal view (see Figure 14), the external view is composed of interfaces connected to the services, these together with non-functional requirements provide the external view, where the higher layers and business stakeholders need not be concerned with the internal behaviour or view of the layer as long as the services are available (26). The internal view provides a more in-depth knowledge of the specific layer elements, which is important to internal stakeholders of the enterprise. This can be presented using the EAAT views which were described in section 5.4.

Alignment issues between business and IT arises from problems when communicating between layers and domains, the descriptions of how to communicate between them are nearly non-existent (83). Communications in ArchiMate shows the importance of the use relation of service oriented applications and models, higher layers use lower layer services. Realisation relations allows mapping of lower level artefacts or objects to higher level objects. In addition to these there are several other relations between elements that are available, such as triggers and aggregations. These are also present in UML, BPMN (84), etc. and are needed to properly express the model elements interaction and integration. (85)

The most important part of the Enterprise Architecture model is the meta model; this contains all the key artefacts of the enterprise. The modelling language ArchiMate represents a balance and coordination of the



languages that comes from different architecture domains and layers, a general set of concepts that all the systems and processes of the enterprise can relate to (83). Figure 15 shows the entire ArchiMate model.

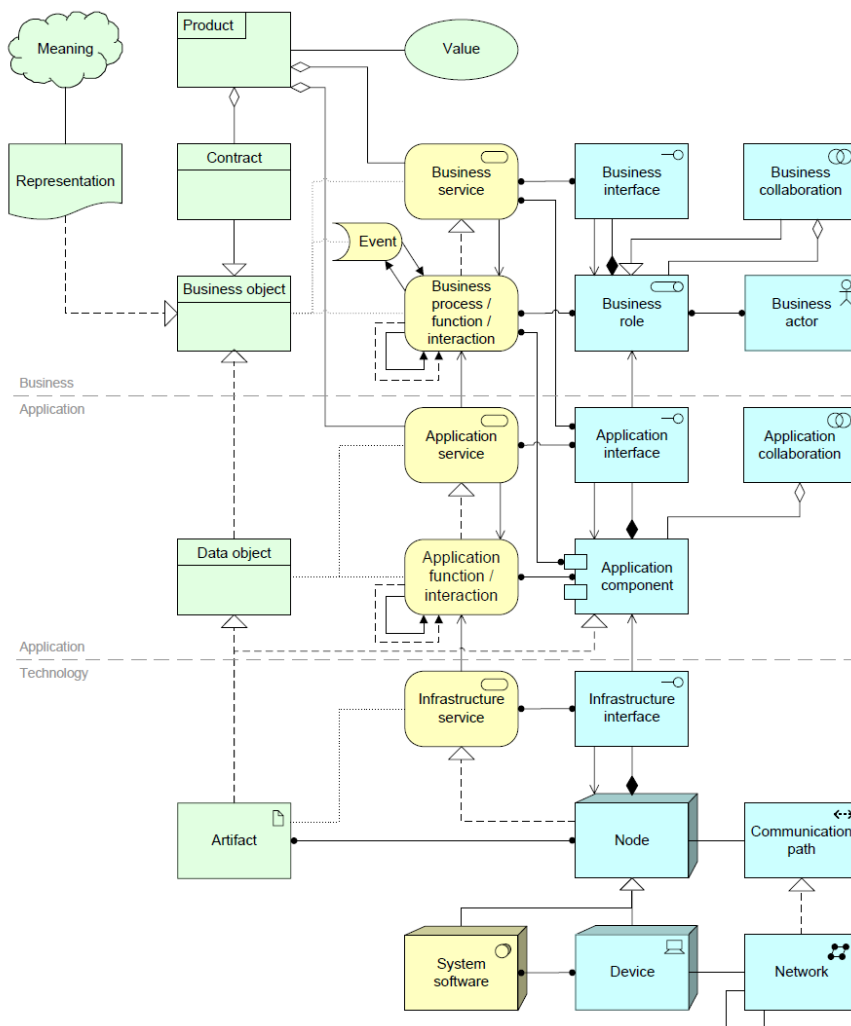


Figure 15: The complete ArchiMate model. Picture from ArchiMate Language Primer

However when developing a meta model using ArchiMate, it only creates a skeleton of the enterprise in terms of analysis (63). But by extending and formalizing the meta model to contain more information about the enterprise it is possible to perform analysis (63). Therefore in addition to providing a language to represent an entire enterprise, the reason that ArchiMate is used in this project is because of its ability to provide analytical insight and knowledge into the enterprise in question if properly extended. Further information about the different components and entities of the ArchiMate modelling language can be found in (5). This degree project focus mostly on the application layer of ArchiMate, but this does not mean that the other layers are excluded completely.

## 6.2 P-OCL

Object Constraint Language (OCL) was developed to describe expressions in UML models, to specify invariants conditions and queries (86). Traditional languages that existed before OCL required extended mathematical insight to specify the constraints. However evaluating OCL expressions do not cause any side effects on the system and since OCL is a deterministic language with pre- and post-conditions, uncertainty in the system is non-existent (41). In order to express uncertainty in the models OCL has been extended with

probabilistic assessment, resulting in *probabilistic* Object Constraint Language (p-OCL), additional information can be found in (41) there introduced as Pi-OCL. With this extension p-OCL can be used to describe uncertainties in the relations, objects and attributes of models that implement it.

A simple p-OCL example query:

```
let n : Integer = numberOfConnections() in getWorstType() + n / (n+1)
```

This is a p-OCL representation of the software metric for determining the coupling type of a number of couplings between two components as defined by (72), which is queried to the entity that implements the operation. This expression is also a valid OCL expression.

In (42) two kinds of uncertainties are explained: First, stochastic attributes, to be able to express attribute values as probabilities, e.g. the system stability is 0.9, that is it is stable 90% of the time it is running. Second, that the existence of objects and relations are uncertain, this is expressed through a variable E(Existence) which is required for all classes and relationships.

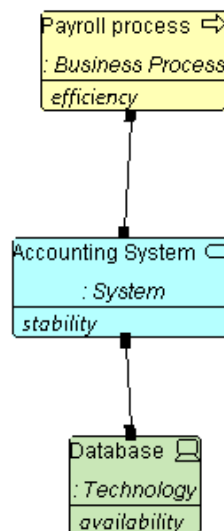


Figure 16 - A simple figure drawn in EAAT to explain uncertainty.

In Figure 16 the stochastic attributes are defined inside the different entities, in this example as *efficiency*, *stability* and *availability*. An example of a stochastic probability would be if the database has proven to be overloaded due to traffic and its *availability* is only 70% of the time, thus would get a value of 0.7. This can then affect the *stability* of the **accounting system**, which further affects the *efficiency* of the **payroll process**. Existence probability can appear when information or data is ambiguous, e.g. the documents could be out of date or two expert opinions claim different components realizes a system. Existence probability can be used in order to express this uncertainty.

(42) presents three ways to implement p-OCL:

1. Attributes specifications, how to specify different attributes.
2. Invariants constraints, defining how the model may or may not be built.
3. Operations, methods that aid the invariants and attributes.

Since p-OCL is able to specify some kinds of uncertainty, this is very useful when performing an analysis, especially when as mentioned earlier; there are different kinds of uncertainties in an Enterprise Architecture Analysis, and the presence of these needs to be defined in the model. Thus the degree project used the p-OCL extension to assess the uncertainties and provide the calculations from the collected data in order to develop an estimate of modifiability in the systems.

### 6.3 META MODEL

The meta model (Figure 17) was created from adapting the meta model from ArchiMate modelling language (Figure 15) and trimming it down to the components needed. By using the already established relations and components from ArchiMate, the modelling of the systems and the processes was alleviated.

As mentioned in the earlier sections the complete ArchiMate model spans an entire enterprise, from the abstract information entities to the dedicated server machines of the technology. And as mentioned in the scope section this degree project focuses mostly on the application layer. Therefore the ArchiMate meta model is used as a base and then trimmed to fit the need and goals of the project. There was no advantage to have the major parts of the technology and organisational layer in the meta model if they are not part of the goals. The trimmed meta model (Figure 17) was then adapted to the necessary requirements, i.e. modified with other objects or relations that was not strictly the ones available in ArchiMate, but required in the project. This was an iterative process, and the first proposed meta model did not turn out to be ideal from the start, and thus changed as the work went on, as some aspects or object might turn out to be unnecessary or outdated. More attributes were added as statements when new information was obtained.

The reason for creating this meta model was rather straightforward and as mentioned earlier, by using the trimmed ArchiMate modelling language as a standard of base components and then extending it with attributes related to modifiability and probabilistic extensions in forms of Object Constraint Language, EAAT was used to calculate different values of interest, such as the coupling average or cyclomatic complexity of components and systems, which was the core of the analytical interest.

With information about documentation, source code and API's, different attributes, relevant to modifiability, was determined in the meta model.

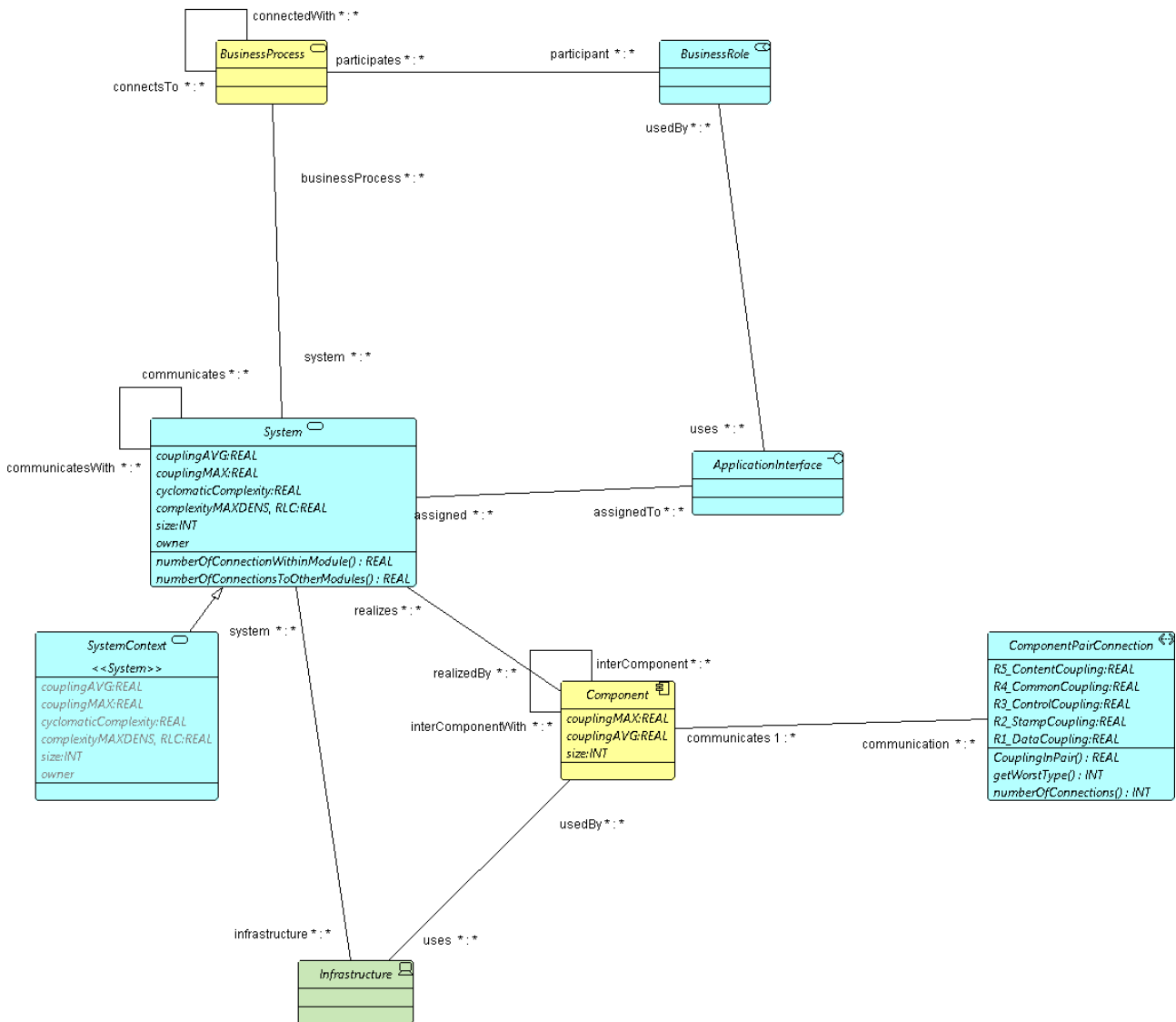


Figure 17: The meta model created and used in this degree project.

### 6.3.1 META MODEL DETAILS

#### Entities:

- **BusinessProcess** – The business organisational layer is composed of business processes, to determine the work activities.
- **BusinessRole** – The different kind of roles that a user assumes when using the system. Different users use different Application Interfaces to access the system.
- **ApplicationInterface** – The different Interfaces used to access the Systems, e.g. web browser or an expert client.
- **System** – The collective name of the Components which realises the System.
- **SystemContext** – This class inherits from the System class, thus having the same attributes, relations and operations. Unlike the system class however this class does not have derived or calculated values for the attributes but are instead used for direct input. This class was used when component level metrics were not available.

- **ComponentPairCoupling** – This class was created to represent the different couplings between components.
- **Component<sup>1</sup>** – Different parts of the system, represents a single or a group of parts, i.e. application functions or objects.
- **Infrastructure** – This class represent the different technology or infrastructure that the System and Components are run upon.

**Attributes:**

- **System.couplingAVG** - The sum of all Component.coupling realizing the System divided by the number of Components. For p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.
- **System.couplingMAX** – The highest form of coupling used in the system. To see the p-OCL code used to calculate this, see APPENDIX p-OCL STATEMENTS.
- **System.complexityMAXDENS, RLC** - Cyclomatic Density, that is cyclomatic complexity divided by the size of the system in source statements. CMAXDENS is the Cyclomatic complexity divided by LoC/1000, which is similar to the Relative Logical Complexity value. To see the p-OCL code used to calculate this, see APPENDIX p-OCL STATEMENTS.
- **System.size** – A System is constructed by a certain amount of code. The amount of code, calculated in lines represents the size. Size (LoC) of the System, with (LoC) or without (SLoC) comments.
- **System.cyclomaticComplexity** – McCabe's cyclomatic complexity, the Components are used as nodes and the ComponentPairCoupling are the edges. Relations to Components outside of the System are also considered. To see the p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.
- **Component.size** – The amount of code, calculated in lines represents the size. Size (LoC) of the Component, with (LoC) or without (SLoC) comments.
- **Component.couplingAVG** – The average coupling that the Component have, i.e. the combined value of all ComponentPairCoupling for the component. To see the p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.
- **Component.couplingMAX** – The highest or tightest coupling in a ComponentPairCoupling the Component is connected to. To see the p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.
- **ComponentPairCoupling.R5\_ContentCoupling** – An attribute to represent the number of Content Couplings between Components as defined in (72).
- **ComponentPairCoupling.R4\_CommonCoupling** – An attribute to represent the number of Common Couplings between Components as defined in (72).
- **ComponentPairCoupling.R3\_ControlCoupling** – An attribute to represent the number of Control Couplings between Components as defined in (72).
- **ComponentPairCoupling.R2\_StampCoupling** – An attribute to represent the number of Stamp Couplings between Components as defined in (72).
- **ComponentPairCoupling.R1\_DataCoupling** – An attribute to represent the number of Data Couplings between Components as defined in (72).

**Operations:**

---

<sup>1</sup> Component is also used to represent databases, however these Components do not realize a System as the database can be used by different Systems.

- `System.numberOfConnectionsWithinModule()` – A p-OCL operation used to calculate the sum of all connections between components inside the same System. To see the p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.
- `System.numberOfConnectionsToOtherModule()` – A p-OCL operation used to calculate the sum of all connections from components to outside the System To see the p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.
- `ComponentPairCoupling.CouplingInPair()` – This is a p-OCL operation to derive the global coupling between to Components as defined in (72). Where  $i$  is represented by `ComponentPairCoupling.getWorstType()` and  $n$  is represented by `ComponentPairCoupling.numberOfConnection()`. To see the p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.
- `ComponentPairCoupling.getWorstType()` – An operation to determine the worst kind of coupling in the `ComponentPairCoupling` class. To see the p-OCL operation used to calculate this see APPENDIX P-OCL STATEMENTS.
- `ComponentPairCoupling.numberOfConnections()` – An operation used to determine the total number of couplings as provided by the values in the attributes. To see the p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.

#### Relations:

- `BusinessProcess.system *.* to System.businessProcess *.*` – This connection was created in order to create a system overview view in EAAT, to show which BusinessProcesses use which systems. There are no calculations connected to this relationship. It does however provide insight into which systems have more dependencies from BusinessProcesses. Several processes can use several systems and vice versa.
- `BusinessRole.participates *.* to BusinessProcess.participant *.*` – This explains what business processes have which business roles that use the systems. There are no calculations connected to this relationship and it is only used to provide an understanding of the business to system relationship in terms of roles and application interfaces. Different roles can be assigned to different processes.
- `BusinessRole.usedBy *.* to ApplicationInterface.uses *.*` – This shows what business roles are can use what application interface to access the system. There are no calculations connected to this relationship and is only used to provide an understanding of business process. Several interfaces are used by different business roles.
- `ApplicationInterface.assigned *.* to System.assignedTo *.*` – This explains what application interfaces that are available to the system. There are no calculations connected to this relationship and is only used to provide an understanding of system. Several application interfaces are assigned to different systems.
- `System.infrastructure *.* to Infrastructure. *.*` - The relationship used to represent connections between Systems and Infrastructure in the overview model. This connection is not used for any calculations.
- `System.communicates *.* to System.communicatesWith *.*` - This is a connection used to calculate how the different Systems are connected using the Components. If a Component is connected to another Component using a `ComponentPairCoupling` then this relationship calculates and draws this direct relationship. To see the p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.
- `System.realizedBy *.* to Component.realizes *.*` – This is the relationship which defines which Systems are realized by which Components. The relationship is used in several p-OCL calculations and

also in the calculation of the System.communicates relationship. To see the p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.

- Component.interComponent \*.\* to Component.interComponentWith \*.\* – This is also a derived calculation similar to the System.communicates relationship, and it is used to calculate and draw connections between different Components in views where ComponentPairCoupling is not visible. To see the p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.
- Component.uses \*.\* to Infrastructure.usedBy \*.\* - This relationship is used to describe what technology or infrastructure that are used by the Components. This relationship is not used in any calculations.
- Component.communication \*.\* to ComponentPairCoupling.communicates 1.\* - This relationship connects the Component class and the ComponentPairCoupling class. It is used to represent the communication between Components together with the ComponentPairCoupling class. To see the p-OCL code used to calculate this see APPENDIX P-OCL STATEMENTS.

As mentioned earlier, the different views for external and internal parts of the meta model are of varying value to contrasting stakeholders. This is represented here especially in the Interface, System and Component entities, some stakeholders are only interested in the interfaces and systems, whereas others are interested in the components.

This degree project also proposes to add the attribute to the System class in this specific case:

- **Owner**, depending on whether the owner or developer of the system is KTH themselves, this could affect the modifiability since documentation and source code might not be available in the extent necessary. That is if modification is even possible.

Using this meta model and data collected through various means, instantiated models of the different systems are created, and these models reflect the actual systems. Included in this meta model are also the probabilistic extensions that will be used for the calculations called p-OCL.

## 7 EMPIRICAL DATA

The data collection of the case study (see section 5.1) was divided into three different phases. The first phase, the initial phase was conducted concurrently with the theory collection, and was mostly intensified towards the end of the theory phase as the project started its data collection phase. The second phase, the modelling phase was regarding the information which was gathered to create the models. Once this phase was starting to reach its end and the models were starting to take form, the third phase could start, the metrics and validation phase, where information used towards the analysis was gathered. Validation was done concurrently with this using the different alternatives established in the evidence collected part of the adapted case study method in section 5.1.3.

As the systems which the project enveloped were all part of the KTH, the information required for completing the project was first assumed to be found at KTH, however several system are externally developed and licenses bought which prompted the owner attribute (see section 6.3.1), and information collection had to be extended further to other sources. The initial investigation of what kind of information was available also showed that documentation on almost all of the systems was out of date or inexistent. This prompted alternative methods for collecting data to adapt the meta model to its needs.

Following this segment introduction information regarding what and how information was collected, and also how the collection was specific for each phase. However the primary forms for collecting data were:

- Emails.
- Interviews, formal and informal.
- Document investigations.
- Workshop.

Email communication was the primary form of communication with the KTH developers and non-KTH personnel in all phases. Informal interviews<sup>2</sup> were also performed during all the phases especially for validation with VoS personnel. Data collection was done rather informally due to the fact that the researcher was stationed close to the personnel responsible for the systems.

Email communication and telephone interview was performed with a system architect from MIRA for the Alumni Community system, the system which is used for maintaining relations with Alumni (graduated students) at KTH. This system is developed and maintained at MIRA networks, a commercial software development company.

In order to collect information for the NyA system, the national admission system for higher education in Sweden, email communication was performed with a system architect for the NyA system and the process leader at VHS, the Swedish agency for higher education services.

LADOK is a national system for study administration within higher education in Sweden. It consists of several sub systems and the information for was collected concerning these sub systems as individual systems. One of the sub systems is Nouveau the administrative tool for the LADOK system. Second is the Ladok på Webben (LpW) system which is a set of web services which via API can be integrated into the individual's institutions web portals. Thirdly is the LadokPing system which enables communication between the different institutions LADOK installation. For Nouveau, LpW and LadokPing emails were sent to ICT Services and

---

<sup>2</sup> Scheduled interviews with constructed questions but not documented audio tapes etc. only notes.



System development department (ITS) at Umeå University whom are responsible for the LADOK development. The respondents there were different two developers and the project leader, each one responded for each system. The developers responded to the questions regarding Nouveau and LadokPing, and the project leader for LpW.

Some might argue that LADOK is only one system with three different applications, LadokPing, LpW and Nouveau. Some might even argue that LadokPing is a part of LpW and that LADOK only has two applications LpW, Nouveau and a database. Neither of this is considered wrong, however in this project the entirety of what is considered the LADOK system is divided into three systems and a database. There are several reasons for this but the main reasons were:

- Different technology – The Nouveau system is developed and deployed in an entirely different environment from the other two.
- System functionality – The functionality and purpose of the systems differs entirely.
- Diversity of conclusion – Analysing each system to some extent provides results to each different system rather to only one.
- LADOK database – It was early established that most systems has some connection to the LADOK database, and if this was then only explained as a connection to the LADOK system the abstraction would be too high.

There are as mentioned other less relevant reasons for this such as how to best present the systems in the models and component conclusions. But in the end the alternative chosen is the one presented here in this project.

KOPPS the system for Course and Programme management and Mina Sidor the web portal at KTH, are developed at KTH by the IT department ITA. At KTH different people were consulted regarding the KOPPS and Mina Sidor system; three developers at ITA were contacted, two of which helped with the modelling, and one whom helped with the metrics acquired. Two of these developers also helped regarding the DIAK system, KTH's local system used for managing international and exchange students.

At the KTH study administrative department (VoS), five people were consulted in regards to validation of models and processes. These were the VoS department head, the system administrators for KOPPS, Nouveau and Mina Sidor at KTH. At the department for collaboration with commerce and industry of KTH a customer relationship manager was interviewed regarding the Alumni Community system. At the admission department of KTH two admission administrators were interviewed regarding the NyA system and its use at KTH.

The documents found were regarding the NyA system, the DIAK system, the Nouveau, LpW and LadokPing system, the KOPPS system and the TimeEdit system. The TimeEdit system is the central scheduling system used by KTH, the system is developed by the commercial development company Evolvera. VHS provided an out-of-date Software Architecture Document and a Product map regarding the NyA system, which were carefully investigated. The documents found regarding the DIAK and TimeEdit systems were very old and didn't provide much information. The documents regarding the Nouveau, LpW and LadokPing systems were abundant but very abstract and outdated when it came to information regarding the system architecture and software metrics. A component map regarding the KOPPS system was also provided by the VoS department; however it was out of date and had to be updated.

Triangulation was done using the few out of date documents that were found, together with interviews and the workshop to establish different sources to create the models. Information regarding the processes was easier to come by and was triangulated using other out of date documents, some up to date documents and interviews to confirm their validity and accuracy. Triangulation for metrics was not possible as different sources for source code were not available, thus it is assumed the metrics that were given were accurate unless specified (Some metrics were on a more abstract level and were not specified exactly what or how it was measured by the sources, see LpW and NyA).

## 7.1 INITIAL

The initial phase was the data collection which commenced during the theory part of the project. The reason for starting the data collection early is to be able to judge what kind of information that would be available to perform the analysis. This is to better specify what should be in the meta model in terms of; entities, relations and attributes, and also what shouldn't be in the meta model. The different kind of information available shaped the meta model into what it is, if there had been other information available then the meta model would have looked different and perhaps used different attributes.

Specific to the initial phase was emails which were conducted in order to establish interview slots with relevant personnel in order to collect information regarding the different systems use and connections to business processes. To establish to whom to ask for specific software metrics, and in those cases where interviews were not available email interviews and exchanges were performed instead.

Informal interview were performed with people responsible for the systems, i.e. systems and object owners or coordinators. These interviews were to establish process to system information, different interfaces as well as information regarding whom to ask for the technical information. The interview questions were standard questions specifically altered to match the system in question.

Informal interviews were performed with people for the Alumni Community system, NyA system (restricted to KTH personnel), Mina Sidor system, DIAK system and KOPPS system. Interviews with NyA and Alumni system were together with business related personnel, and not system developers or architects. Only in the case of Mina Sidor, DIAK and KOPPS, was one meeting performed with developers. The rest of the interviews or questions were answered over email.

The meeting with the system developers at KTH also provided information regarding what software metrics would be possible to receive for the KOPPS and Mina Sidor systems. Documents regarding different systems were provided, but as mentioned most of them were out of date. But still the documents provided gave some insight into which systems that were available to be modelled and analysed.

## 7.2 MODELLING

Before the final phase, the data metrics collection could begin the models had to be completed, in order to know what components that realizes the systems, and thus limit the data measurements to specific components rather than the entire system over all where available. However in the case when measurements could not be performed on a component view, a large system measurement was provided for some systems when possible.

In order to establish what the systems actually look like, documents regarding system composition and architecture was requested from developers and external software companies and organisations. However

documents up to date were very scarce and only available on a few of the systems. Thus interviews were requested in order to get an understanding of what components realized the system.

In order to get an overview of how the systems are connected at KTH, a system presentation was attended where notes regarding the different systems and their connections were taken. A telephone interview with MIRA networks was conducted over the telephone, where they provided information regarding the setup and architecture of the Alumni System. Another more specific workshop or meeting with the developers at KTH was given, where specific information regarding the components and connections of KOPPS, Mina Sidor, DIAK and TimeEdit API was given (pictures available in APPENDIX B).

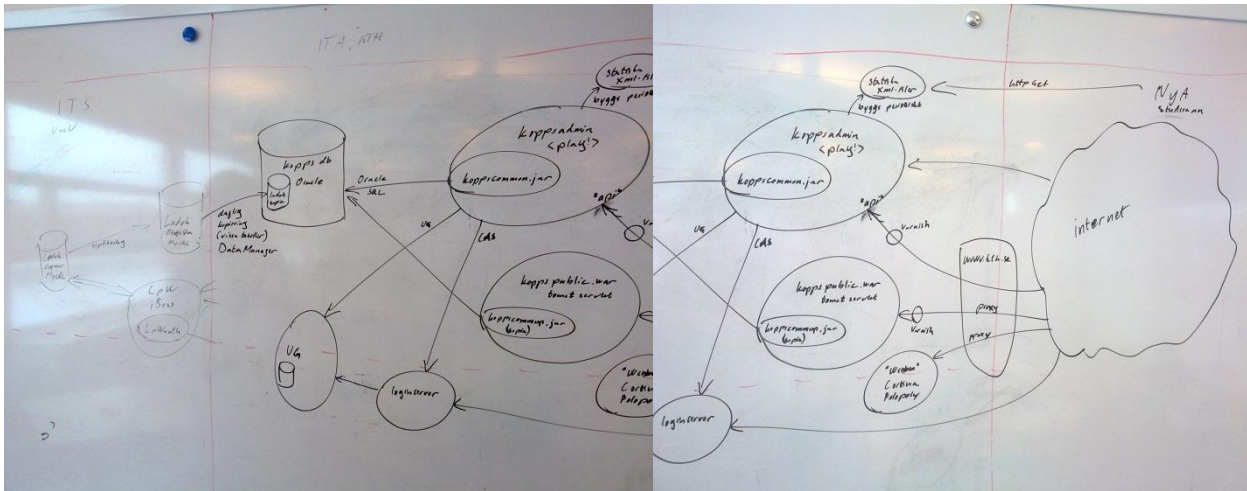


Figure 18: Pictures taken at the meeting with the developers at KTH.

When meetings and interviews with architectures or developers were not an option or not available, investigation of documents and emails were performed. This was the case for the NyA, LADOK and TimeEdit.

The documents provided for the NyA system were the one most up to date, a Software Architecture Document (SAD) and a Product Map. Documents regarding LADOK were collected and reviewed on the LADOK website (<http://www.ladok.se>), these documents varied in regards to updates; however a lot of them had version history which reassured relevancy of the contents. Documents regarding the TimeEdit system were abstract at best, explaining the product, uses and how it is generally setup.

Emails were sent to all the organisations responsible for each system, requesting for more information, asking further insight into what was not explained in the different documents or unclear during the interviews. However Evolvera did not respond on requests for information and after the telephone interview MIRA networks did not respond to any further inquiries regarding vague information. A request regarding more up to date and valid information of the Nouveau system was never answered. With the models complete the collection of metrics could commence as the borders of the components of the systems were established. This meant also that models could be validated.

### 7.3 METRICS AND VALIDATION

Once the models were finished and drawn from the data collected in the previous phase, the models were used as a base for requesting information regarding specific metrics related to software modifiability. The information that was requested in the metrics and validation phase was related to perform the analysis of each component and system. The information requested from the sources available was as established in

the theory, about the cyclomatic complexity in the components, the size i.e. source lines of code and what kind of couplings between the components which were drawn in the models, if this information was not available, then information about cyclomatic complexity and size about the entire system in itself was requested. Couplings however were still requested between components. In the cases where metrics could not be provided, educated estimations were made regarding couplings.

Information gathering for system metrics was not as successful as the previous data collection for constructing the models. The Alumni Community system metrics regarding cyclomatic complexity were hard to understand, any further attempts at contacting MIRA resulted in no response. The Nouveau system information regarding metrics was never received either. Evolvera still did not respond to the request regarding information.

However not all requests ended up fruitless, the metrics for LadokPing, NyA, and LpW was acquired, at least regarding the cyclomatic complexity and the size of the systems. These values however were not specifically measured as requested with the models, and instead was given as an abstract total system measure, i.e. a measurement of the entire systems rather than specific components.

In order to receive more accurate and up to date metrics for the LadokPing and LpW systems, the developers at ITS referred me to the developers at KTH, however the developers at KTH only had a limited amount of time to spare to help, which was put to use in order to receive metrics from KOPPS, Mina Sidor and TimeEdit API. Those metrics however were precise and easy to understand as presented in Table 1.

(APPENDIX – DATA)

Component	Metric	Value
<b>KoppsAdmin</b>	Lines of Code	48701
	Source Lines of Code	30718
	Cyclomatic Complexity Average	1.85
	Cyclomatic Complexity Max	16
<b>KoppsPublic</b>	Lines of Code	5286
	Source Lines of Code	4436
	Cyclomatic Complexity Average	1.33
	Cyclomatic Complexity Max	4

Table 1: Metrics example as found in the APPENDIX - DATA.

These are the values which were collected regarding two KOPPS components, KoppsPublic and KoppsAdmin. The definition of the values and how they are defined can be found in section 6. They are also further interpreted in the following analysis and results sections.

In order to determine the accuracy of the models emails were exchanged with ITS, VHS and ITA. And further validation regarding the processes and the models were performed with interviews with the system owners and coordinators at VoS.

Although the interviews were performed with a variety of personnel, from system owners to system developers, most of them were with the system owners rather than with the developers. Although the system owners were very good at answering questions regarding use and process work, they did mostly have limited knowledge regarding the technical system and components aspects regarding the systems. But even so they were related to the systems and could validate the models, as it was mostly the technical details to which their knowledge was limited. Most of the documents provided were usually at least 2 years out of

date and system developers and architects were usually limited on interview or emailing time. Thus most of the sources were not individually especially reliable, but together they still seemed to point in the same direction. Thus with the evidence collected and validated the case study could continue to its analysis stage.

## 8 ANALYSIS

The analysis of this study was performed using the theory previously described in the method and modifiability sections, i.e. section 5 & 6, and the Enterprise Architecture Analysis Tool calculations. By establishing the meta model, including the p-OCL statements and attributes which were derived from the documentation and the literature reviewed, the base for the analysis was completed, i.e. the theory and the developed method. With the p-OCL based on already established analytical algorithms as presented earlier.

The data collected as described above was used to create instantiated models using the meta model classes to represent the different part of the systems. The models were drawn in EAAT and as the different entities of these models then contain the attributes related to modifiability as can be seen in Figure 19. The full extent of the models can be found in APPENDIX – MODELS. The attributes were then instantiated using the software metrics which were collected from the systems as part of the data.

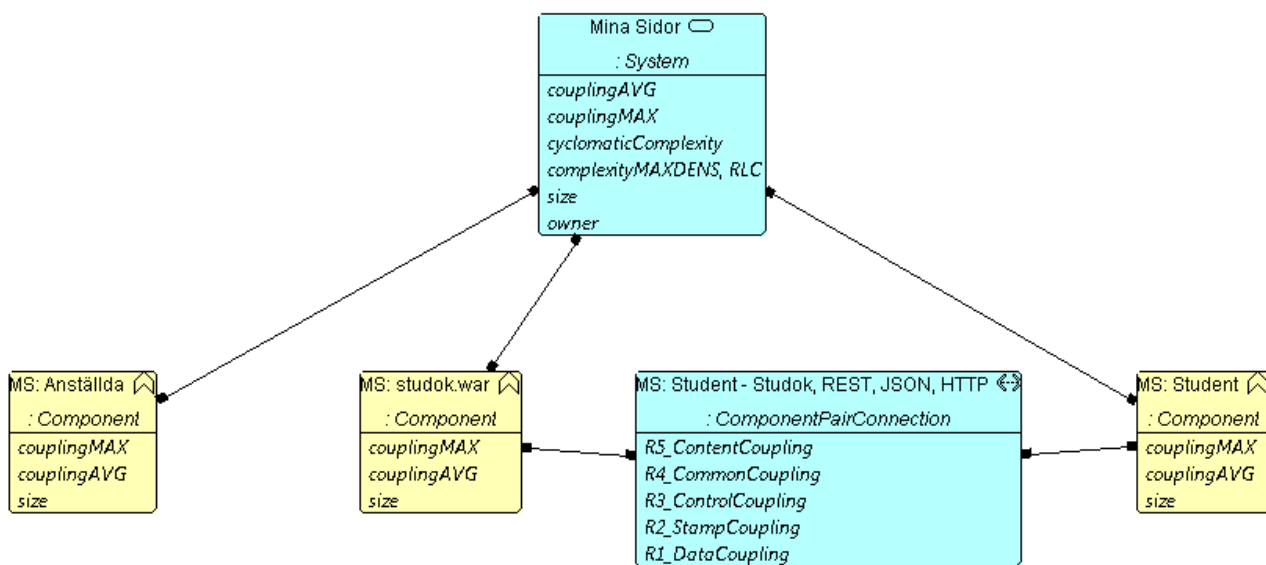


Figure 19: An extract of the models drawn from the information gathered. The figure is a small part of the Mina Sidor system but shows the attributes of the different entities.

In the cases where the collected data was not enough to determine exactly which kind of coupling between two objects were available, educated estimations were performed using the information that was available and probability distributions were added to order to express uncertainty. For those components and metrics where information was scarce or uncertain, probabilities and uncertainty of existence in EAAT was applied.

E.g. information regarding how the different external components (see the Alumni Community system in the APPENDIX – SYSTEMS) in the Alumni Community system were coupled was not obtainable. But making an educated estimation regarding what kind of coupling there could be in regards to the product, and then adding a probabilistic value in form of a normal distribution and the values coefficient of variation in order to express the uncertainty. Different coefficient of variation was used in order to express different levels of uncertainty, e.g. high uncertainty was determined using a coefficient between 75-90% and low uncertainty between 10-25%. The quality of the information obtained or the expressed uncertainty from the source determined the uncertainty level used. Another example: Where the coupling number values are uncertain, normal distribution was used, e.g. LpW to LADOK, where the amount of common couplings was found to be

around 90 from the documentation. However the documents were not up to date and no confirmation regarding the estimated couplings were given. Thus a normal distribution around 90 with a coefficient of variation of 25%, i.e. 22.5 was used to express this low level of uncertainty.

Thus with Enterprise Architecture and the ArchiMate modelling language to create the models, and the attributes related to Enterprise Architecture modifiability analysis in place within the entities of the models and instantiated using the metrics and uncertainty applied, calculations using EAAT were performed.

The calculations in EAAT were performed using a forward sampling method with 7500 samples. Forward sampling is a node based sampling method, where every following node is only evaluated based on its predecessor. Every sample is an independent probability evaluation through these nodes, thus the higher amount of samples the better probability measurement will be shown. The nodes in this case are the entities in the models and their attributes.

These calculations took about 500 to 550 seconds to complete. Three attempts were made using ten thousand samples, but EAAT crashed at different stages in the calculations, in the end 7500 samples were used instead. There were a total of three calculations conducted with 7500 samples performed in order to extract all the information from the program but these calculations were performed with the same input. These calculations then provided a variation of different values regarding software modifiability as defined in the modifiability and meta model segment. As explained earlier in the method EAAT section, these values can then expressed as histograms, as can be seen in Figure 20, for further insight. The varying values in these histograms were because of the probabilistic distributions used when values and metrics were uncertain. With these values together with the theory and instantiated models, conclusions regarding the modifiability of each system were made.

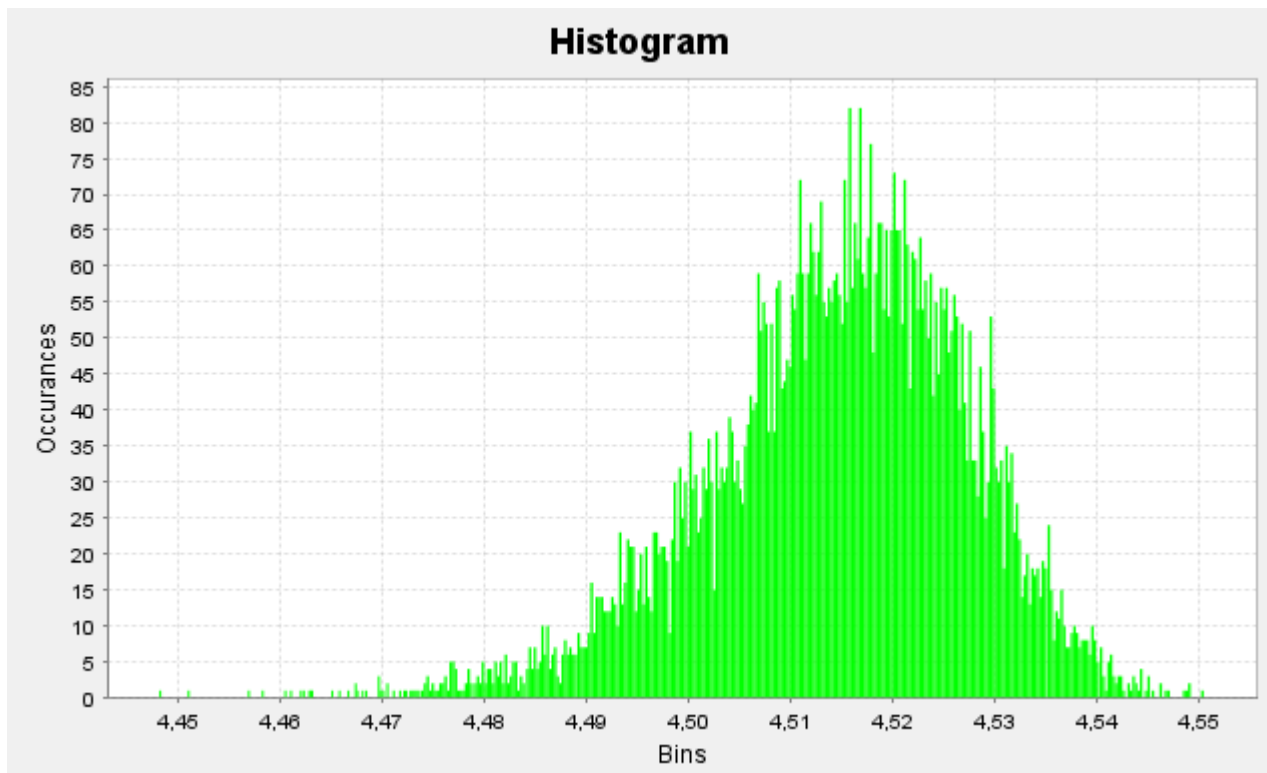


Figure 20: The histogram of the KOPPS system's coupling average measurement.



As can be seen in the histogram (Figure 20) there is no absolute value for the coupling average measurement, but instead it is distributed with a peak around the *mean* 4,514413, which represents uncertainty. The *variance* of the histogram was calculated to be 0,005682 in EAAT. The histogram shows many samples around the 4,52 value and shows a very slight distribution and most values are peaking in between 4,50 and 4,53. The entire distribution is rather gathered around the 4,51 in general, and no samples can be found lower than 4,44 or higher than 4,56. This suggests low uncertainty and that with high probability the coupling value of the KOPPS system is somewhere close to 4,51. With the *mean* and *variance* values calculated in EAAT the *coefficient of variation* can easily be calculated, which presents the uncertainty and distribution in terms of percentage.

To calculate the coefficient of variation the variance was square rooted to calculate the standard deviation, which was then divided by the mean.

$$\text{Coefficient of Variation} = \frac{\sqrt{\text{Variance}}}{\text{Mean}}$$

CouplingAVG	Mean	Variance	Deviation	Coefficient of Variation (%)
Alumni	3,04507	0,028378	0,168458	5,5%
LpW	4,09661	0,004497	0,06706	1,6%
Nouveau	2,931113	3,766976	1,94087	66,2%
Mina Sidor	3,785904	0,003822	0,061822	1,6%
KOPPS	4,514413	0,005682	0,075379	1,7%
DIAC	3,661697	0,004202	0,064823	1,8%
NyA	3,526906	0,00341	0,058395	1,6%
Ladok Ping	3,785967	0,003894	0,062402	1,6%
TimeEdit	3,914292	0,026207	0,161886	4,1%

Table 2: The different values used for calculating the coefficient of variation which expresses the uncertainty.

The coefficient was calculated for all the histograms which expressed an uncertainty in the histograms. As can be seen the coefficient of variance for KOPPS was only 1,7%, which further validates the low distribution and low uncertainty regarding the value calculated.

The uncertainty calculations were limited to mainly the couplingAVG & couplingMAX attributes for most of the systems, but also the cyclomatic complexity value for the Nouveau system, whose components did not all have confirmed existence which was another point of uncertainty as explain the p-OCL section.

CouplingMAX	Mean	Variance	Standard Deviation	Coefficient of Variation (%)
Nouveau	4,529797	3,431223	1,852356067	40,9%
DIAC	3,694901	0,00425	0,065192024	1,8%
LpW	4,994295	0,006652	0,081559794	1,6%
KOPPS	4,689835	0,006319	0,079492138	1,7%
NyA	4,722305	0,006161	0,078492038	1,7%
TimeEdit	4,662025	0,007526	0,086752522	1,9%

Table 3: The different values used for calculating the coefficient of variation for the couplingMAX attribute.



Once the conclusions regarding the code metrics were made, an observation regarding external components and owners of the systems was added. Modification to external third part components are considered limited unless they are Open Source, and even then considered quite limited, but measurements regarding such components were not obtained.

Using the models and the calculations the analysis was completed and results regarding the goals of the project could be concluded. The models provided results regarding the system models and the attributes; size, complexity, couplings and density provided insight into whether further change of either components or systems would be more or less difficult, a modifiability analysis.

## 9 RESULTS

This section presents the results regarding the research questions asked in this study. Considering the degree project spanned several systems and a lot of information, in order not to overwhelm the reader, this section answers the research questions in a general sense and more information regarding specific details can be found in the APPENDIX – SYSTEMS. Following this section is the KOPPS system which can be used as an example of how the information regarding the remaining systems is described. By following the report one can see the key elements of the method is used to present the results, Enterprise Architecture for the models, Enterprise Architecture Analysis and EAAT to determine modifiability.

Research Question 1: What does the current information systems of VoS look like? And how are they connected to each other?

This question was answered using the strength of Enterprise Architecture: models. Using the data collected from various sources and analysing it using the methods and tool described in the Enterprise Architecture Analysis and Enterprise Architecture Analysis Tool sections, the results are the models of the systems. There are two questions to answer the first of which; what the current information systems look like and following this result section is the KOPPS section of the results, which provides the answer to this question regarding the KOPPS system. There are similar sections for every system that was investigated (see APPENDIX – SYSTEMS).

In order to answer the second part regarding how the systems are connected, models were created to present several views, some dividing larger models into smaller views which provide insight depending on which system and what kind of interest to the system that the reader has (e.g. only business layer). Some models created were more general like the overview picture (Figure 21) which presents a general view over all the systems and their connections to other systems, together with the processes and databases.

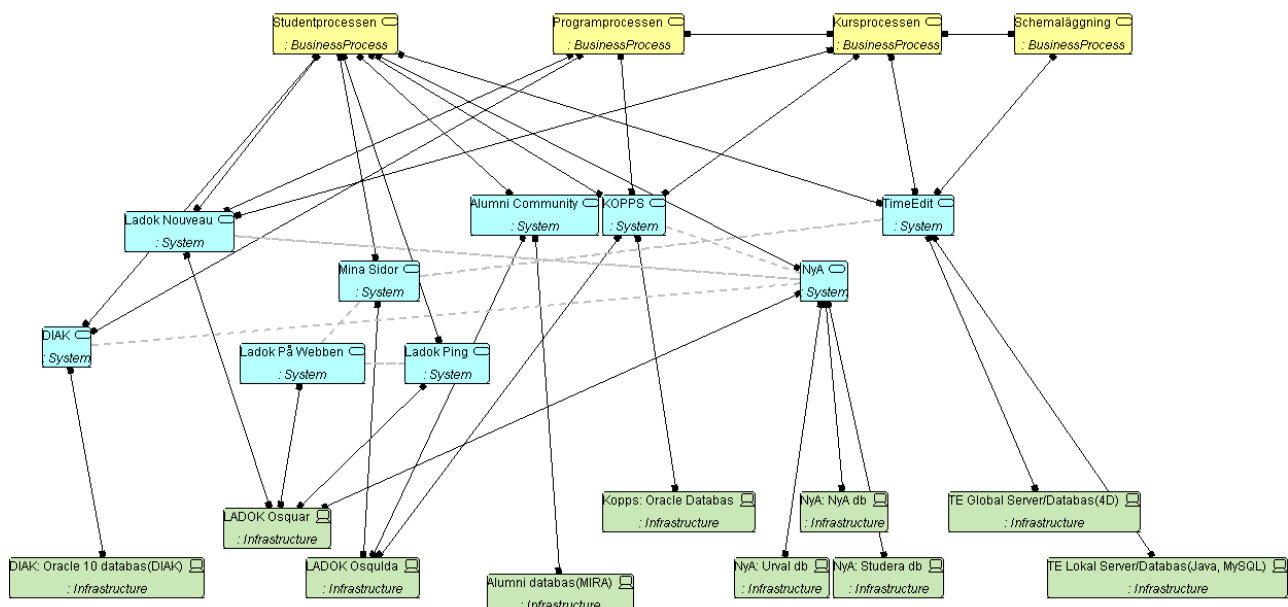


Figure 21: A model created to show the entire overview of the processes, systems and databases. A larger picture is available in APPENDIX - MODELS.

As can be seen the systems that were investigated are all represented in the figure and shows how they are connected, where the dotted lines are connections between the systems which answers the connection

question. In addition to showing the different connections between systems and their databases, Figure 21 presents the processes and what systems that the processes use. This is summarized in Table 4 which is another form of a model, to show the connections in another perspective, i.e. which systems the processes use:

	Student	Program	Course	Schedule
TimeEdit	X		X	X
NyA	X			
Mina Sidor	X			
LadokPing	X			
Nouveau	X	X	X	
KOPPS	X	X	X	
Alumni Community	X			
DIAK	X	X		

Table 4: The updated processes what systems they use.

There were in total four different business processes investigated which are currently used by the university; the student process, the program process, the course process and scheduling process. The updated process models and information regarding them can be found in APPENDIX – UPDATED PROCESSES. The results show that all the systems are used in the student process, but only specific systems are used by the other processes.

However as the application layer of Enterprise Architecture was the focus, most of the models that were created were of the individual systems. In these models the entire system is presented from business to application to technology, although mostly focused on application as explained, presenting the entire model and its components. These individual models were then further broken up into different views in order to address those who are interested in the different layers using the external and internal views as reference from ArchiMate for instance Figure 22. Most of these figures are presented in the section for each individual system in the APPENDIX - SYSTEMS section which is part of the results. Here the different components of the system are also explained and described, together with the connections to the other systems. The full pictures are available in the APPENDIX - MODELS.

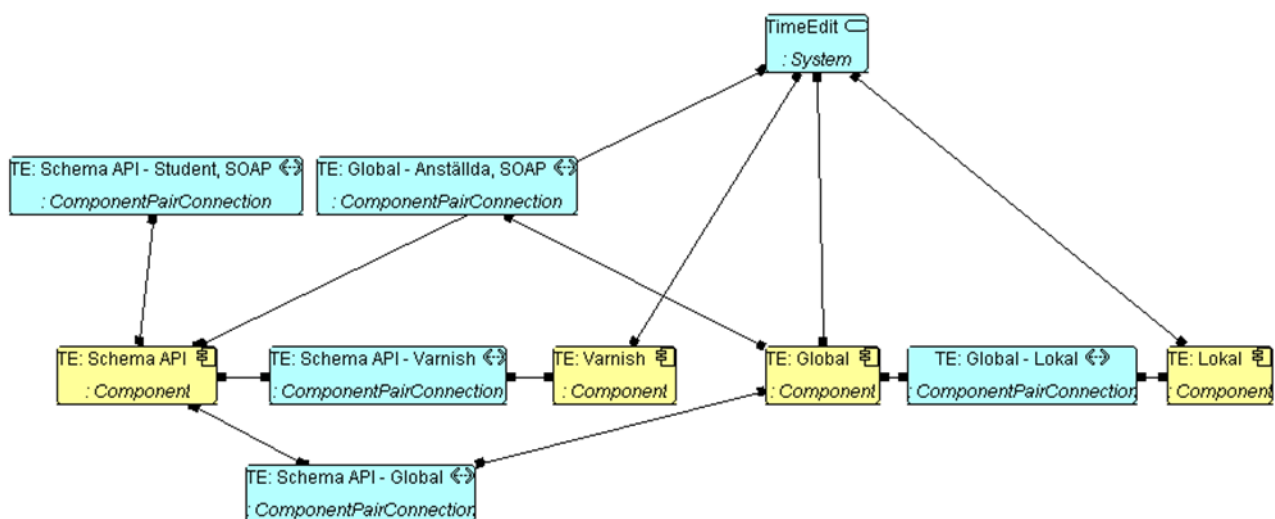


Figure 22: A scaled down model created to present a view of information to those interested in for instance the System and its components.

Other models were also created; some which highlights to some extent parts of the technology layer which can be found in the APPENDIX – SYSTEMS, one model which presents the connections to the LADOK data-bases (Figure 28). Also in order to present critical and highly connected systems other models were created where the only model entities are the systems and their connections. These figures are further presented and discussed in the discussion section, i.e. section 10.

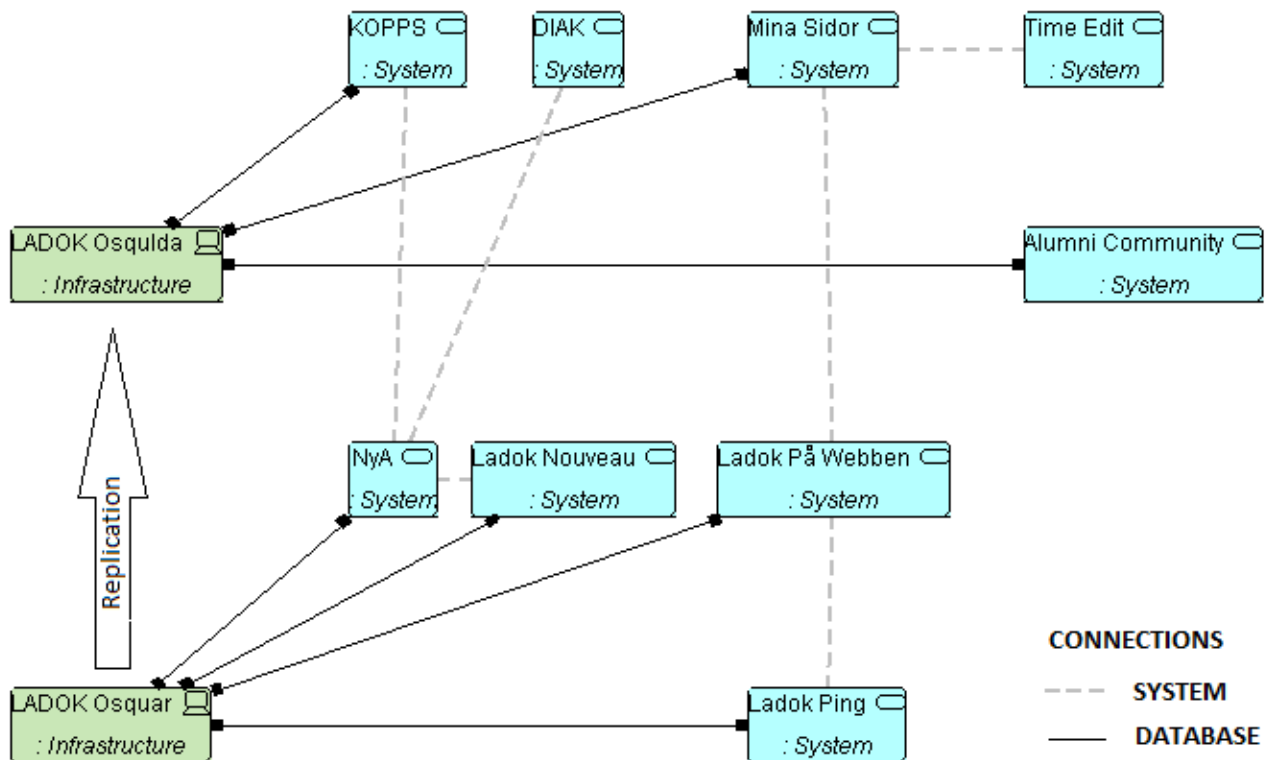


Figure 23 – The connections to LADOK is existent in almost all the systems.

As can be seen in both Figure 21 and Figure 23 several systems are connected to each other, both directly and in some aspects also indirectly through LADOK as almost all systems are connected to them. Coupled systems that are modified has as presented earlier in section 6 a chance to cause ripple effects and affect other systems to which it is connected. This leads to the other question that was asked to answer in this study:

Research Question 2: What kind of results can be found regarding the modifiability in the VoS systems?

The second question that was asked in this study was regarding the modifiability of the systems. Using the method established in previous sections together with EAAT and the collected data, the different attributes which were considered to be the most relevant to system modifiability were calculated and analysed. The values calculated for these attributes can be found in Table 5.

In the data collection the different sources had varying levels of uncertainty coupled to them, and as described in the analysis section, uncertainty was in this project represented as a coefficient of variance to the different data collected depending on how relevant, accurate, verifiable and up to date it was. The results of the analysis of the different systems were thus provided with a level of uncertainty as presented in Table 5.

Presented in this section (Table 5) are the calculations from EAAT based on the metrics received regarding all the systems. These are the results regarding the systems, for specific information regarding individual components you can find the information for each individual system in APPENDIX - SYSTEMS.

System	Metric	Value (Mean)	Uncertainty
Mina Sidor	Size	11323	0 %
	Cyclomatic Complexity	12,0	0 %
	CouplingAVG	3,79	1,6%
	CouplingMAX	4,67	0 %
	Cyclomatic Density	1,06	0 %
Ladok på Webben	Size	178758	0 %
	Cyclomatic Complexity	6,00	0 %
	CouplingAVG	4,10	1,6%
	CouplingMAX	4,99	1,6%
	Cyclomatic Density	0,03	0 %
System	Metric	Value	Uncertainty
LadokPing	Size	45266	0 %
	Cyclomatic Complexity	4,00	0 %
	CouplingAVG	3,79	1,6%
	CouplingMAX	4,50	0 %
	Cyclomatic Density	0,09	0 %
System	Metric	Value	Uncertainty
Nouveau	Cyclomatic Complexity	2,75	15,9 %
	CouplingAVG	3,16	66 %
	CouplingMAX	4,53	40,9 %
System	Metric	Value	Uncertainty
KOPPS	Size	35156	0 %
	Cyclomatic Complexity	7,00	0 %
	CouplingAVG	4,51	1,7%
	CouplingMAX	4,69	1,7 %
	Cyclomatic Density	0,20	0 %
System	Metric	Value	Uncertainty
Alumni Community	Cyclomatic Complexity	3,00	0 %
	CouplingAVG	3,05	5,5 %
	CouplingMAX	4,50	0%
System	Metric	Value	Uncertainty
TimeEdit	Cyclomatic Complexity	3,00	0 %
	CouplingAVG	3,91	4,1 %
	CouplingMAX	4,66	1,9%
System	Metric	Value	Uncertainty
DIAK	Cyclomatic Complexity	4,00	0 %
	CouplingAVG	3,66	1,8 %
	CouplingMAX	3,69	1,8%
System	Metric	Value	Uncertainty
NyA	Size	819341	0 %
	Cyclomatic Complexity	18,00	0 %
	CouplingAVG	3,53	1,6%
	CouplingMAX	4,72	1,7%
	Cyclomatic Density	0,02	0 %

Table 5: The collected results of the EAAT calculations of every system, where some values are missing such data were not available.

These different attributes are presented earlier first in the previous sections where they are first presented theoretically in their relevance to systems and systems modifiability, the attributes themselves are then further explained in the meta model presentation. To quickly reiterate their function and meaning:

- **Size** is the source lines of code without comments, in which the larger the system or component it is harder to conduct a change.
- **Cyclomatic complexity** is represented by the system through its components and their couplings to other components. The higher the value the more complex the system is in terms of components and couplings.
- **Coupling** is here represented by two values: the average coupling value representing how the tightly coupled the system is overall, and also the max coupling value which is the highest coupling found in the system. The range of the coupling is from one to five, where one is the loosest kind of coupling and five represents a very tightly coupled system.
- **Cyclomatic Density** represents the average amount of logic or complexity for every thousand lines of code. A too low value indicates a system where the logic is very low considering the size of the system. Whereas a too high a value suggests that there might be too much logic or complexity in the system.

As explained in the modifiability section above most of the attributes that were measured have a range of what is considered better in terms of modifiability, e.g. looser couplings are considered better since then modifications to one components of the system is less likely to affect another part. Systems with low complexity, i.e. under a certain recommended limit are considered easier to modify as they are thus seen as easier to understand and cyclomatic density has both an upper and a lower recommended limit.

The size attribute however does not have a clearly defined limit as to what is considered a large system, as it is a rather subjective attribute, and rather the higher the value of size the harder it is considered to modify. Therefore subjective limits chosen by the author that seemed relevant to the study was chosen in order to categorize them in terms of each other. Using a limit which would categorize them all as either small or large didn't make any sense, e.g. having a lower limit at one million.

The values in Table 5 are translated in order to express their values in relevance to the limits, how the author concludes they reflect on modifiability and in other terms than numbers. While as explained in the modifiability method part of this study some of the attributes are rather subjective and there have been several systems according to McCabe which exceeds the different recommendation limits to for instance cyclomatic complexity and are still considered highly modifiable. The recommendations and how they were considered can be found in Table 6:

Attribute	Too Low	Low/Small	Average/Medium	High/Large	Too High
Size	-	<100000	100k-500k	500k<	-
Coupling	-	1-2	3-4	4<	-
C Complexity	-	<10	10-15	15<	-
C Density	<0,14	Close to 0,14	0,23-0,32	Close to 0,42	0,42<

Table 6: The categorization used in this project summarized from the modifiability section.

Following these recommendations and categorizations the values in Table 5 the results acquired from the analysis the author used the values in Table 6 to draw these conclusions regarding the system modifiability:

System	Density	Coupling (Avg)	Size	Complexity	Modifiability
--------	---------	----------------	------	------------	---------------

<b>KOPPS</b>	Low	High	Small	Low	Average-High
<b>NyA</b>	Too Low	Average	Large	High	Average-Low
<b>Mina Sidor</b>	Too High	Average	Small	Average	Average
<b>LpW</b>	Too Low	High	Medium	Low	Low
<b>LadokPing</b>	Too Low	Average	Small	Low	Average-High
<b>Nouveau</b>	-	Average	-	Low	Average-High <sup>3</sup>
<b>Alumni Com- munity</b>	-	Average	-	Low	Average-High
<b>TimeEdit</b>	-	Average	-	Low	Average-High <sup>3</sup>
<b>DIAC</b>	-	Average	-	Low	Average-High <sup>3</sup>

Table 7: The author's modifiability conclusions for each system.

The modifiability expressed in Table 7 answers the second question, the modifiability which was found regarding the systems. As can be seen in the last column of table the modifiability is rather varied ranging from low to high, most of the systems however have an average to average-high modifiability, and there are only two systems that have low or average to low modifiability.

The different attributes that have been categorized by the limitations in Table 6 are used to conclude the modifiability, where the attribute coupling has been considered slightly to have more weight to the modifiability conclusion. This is because, as mentioned in the previous modifiability theory section, coupling can have impact on the nearby coupled systems and components. In order to see how however, one needs to look at the systems individually.

<sup>3</sup> Small amounts of data were collected for these systems and were used with higher uncertainty.

## 9.1 KOPPS

**Kurs Och Programkatalog Planerings System (KOPPS)**, developed by KTH, this system is used for administrating and maintaining the different programmes and courses that are available at KTH. This system was developed in house at KTH which means that they have full control over what kind of changes they wish to perform, as long as it is technologically possible. The system was constructed to replace the student hand-book, which contained the information regarding programmes and courses which are available at KTH. The system is realised by two components: The aptly named KoppsAdmin, which is the administrative part of the system, and the also aptly named KoppsPublic, which is the public part of the system.

Of the four processes which were considered in this degree project, three of the processes use the KOPPS system as a part of its activity, the student process, the course process and the program process. However the fourth process the scheduling process is also indirectly dependent on the system. The different processes, the connections to the business roles and the what interfaces they use can be seen in Figure 24.

- The student process: The KOPPS system is used in the student process as it is required in order to offer the student the necessary information regarding courses and programs for the students to apply for the program or course.
- The course process: The KOPPS system is the core system for the course process, as this is where the information regarding the courses is kept, and administration for the courses is performed.
- The program process: The KOPPS system is also the core system for the program process, and the system is also in this process used for administrating and keeping information regarding the programs.
- The scheduling process: In this process the KOPPS system does not play a direct role, but indirectly course abstracts are used in order to create schedules in the scheduling process.

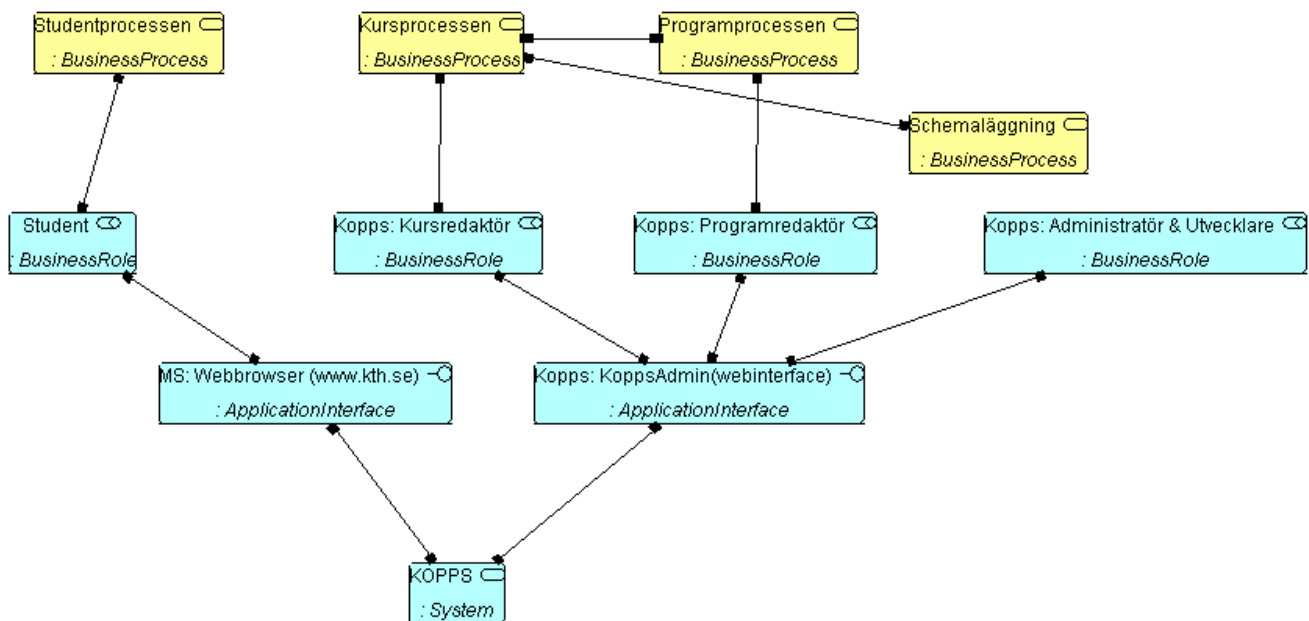


Figure 24: The processes, roles, interface and how they are related to the KOPPS system.



There are two interfaces for accessing the KOPPS systems, both are web interfaces, but one is for the public e.g. students, and one for the KOPPS administrators i.e. KTH employees responsible for the courses and programs. The first is the actual course and program information which is available at the KTH webpage, the public information. The second is the administrative part which is only available through login by personnel. The components that realize the system are:

- **KoppsPublic** – This is the component of the system which realizes the publically available part of the system, i.e. the part which is used to present the different information about courses and program.
- **KoppsAdmin** – This is the administrative component of the system. The KoppsAdmin is used by course and program administrators when creating, updating or removing different courses and programmes.
- **Varnish** – A component realized by an external component called varnish, this program is used for improving the loading time for users through caching. That is, for instance since the course websites information are static and does not get changed at the website and only presents the information that is in the database, Varnish helps caching this information for higher performance. This component is used by both KoppsPublic and KoppsAdmin to increase performance.
- **DataManager** – There is also another external component called IBM Cognos Data Manager. This component is used to copy, map and translate tables and information from LADOK Osqulda which is a MySQL database, to the KOPPS Oracle database; this is data intensive and only happens once a day.

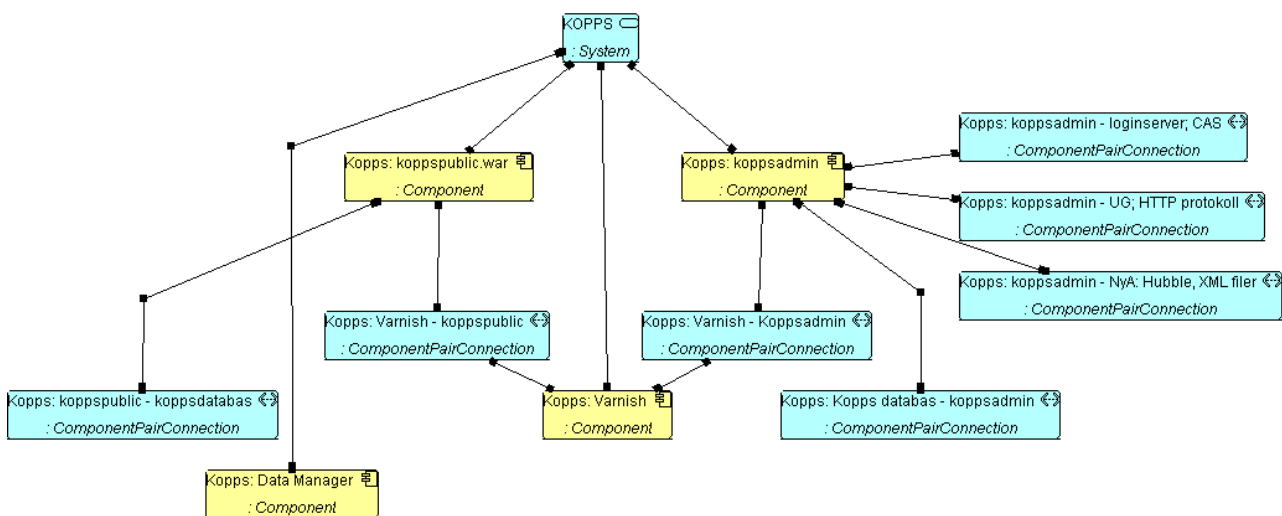


Figure 25: The different components that realize the KOPPS system and how they are connected, KoppsAdmin also has external system connections.

**KoppsAdmin** - The KoppsAdmin component is comprised of an archive named *koppsadmin.war* and is developed in the Play! Framework (87) and runs on a Play! server. The Play! Framework is a lightweight Java based framework for developing highly scalable web applications. The source code metrics for the KOPPS system were measured using the MetricsReloaded (88) tool. And for the KoppsAdmin measurements the results were 48701 LOC in size, of which 30718 lines are actual source code. The cyclomatic complexity average was measured from the methods to be a value of 1.85 in the component's source code, however of all the methods in the component, a max cyclomatic complexity in a method was found to be 16. Figure 25

shows the different components and connections between the components, as well as the external connections from KoppsAdmin to other systems.

The couplings data for the KOPPS system were given directly from the developers and the extent of them can be found in APPENDIX - DATA. Using the p-OCL statements for measuring coupling average and coupling max values for components the following values were obtained for KoppsAdmin:

Component	Metric	Value
KoppsAdmin	CouplingAVG	4,07
	CouplingMAX	4,68

These are rather high value of couplings, which suggests that changes to this component would most like affect other coupled components, or vice versa.

The login to KoppsAdmin is handled by the Loginserver used by most KTH systems. Once logged into the KoppsAdmin however there are different levels of access available dependent on what the user is authorized to in the KTH User Group system. Included in the *koppsadmin.war* is the *UgClient.jar* file which is a properties file which facilitates a connection to the UG system. KoppsAdmin also has a connection to the NyA system through XML files which are used for publishing the different courses and programs for NyA applicants.

**KoppsPublic** - The KoppsPublic component is comprised of an archive named *koppspublic.war*. This archive is deployed on several Tomcat Servlets (89) and contains the interfaces used to present the information. The metrics for this component was also obtained using the MetricsReloaded tool and the measurements were given directly from the developers. Compared to the KoppsAdmin, the KoppsPublic is a rather small component, only 5286 lines of code, of which 4436 are actual source lines of code. The cyclomatic complexity average measured from the methods in the component's source code was found to be 1.33 and the highest cyclomatic complexity value found for one of the methods was 4.

The coupling values calculated from these coupling using the same p-OCL statements as the defined in the meta model were:

Component	Metric	Value
KoppsPublic	CouplingAVG	4,66
	CouplingMAX	4,69

These coupling measurements are also rather high which can be explained by the common coupling. Thus any changes here would also likely affect the coupled components, or vice versa.

According to the developers at KTH, in order to connect to the database KoppsAdmin and KoppsPublic components have an identical properties file called *koppscommon.jar*. Thus any change in the database requires an update to both archives.

There are two databases used by KOPPS:

- KOPPS Database – The Oracle database used to contain all the information regarding the courses and programs.
- LADOK OPEN (Osqulda) – KOPPS reads and translates information from this database which is then later used in the KOPPS database.

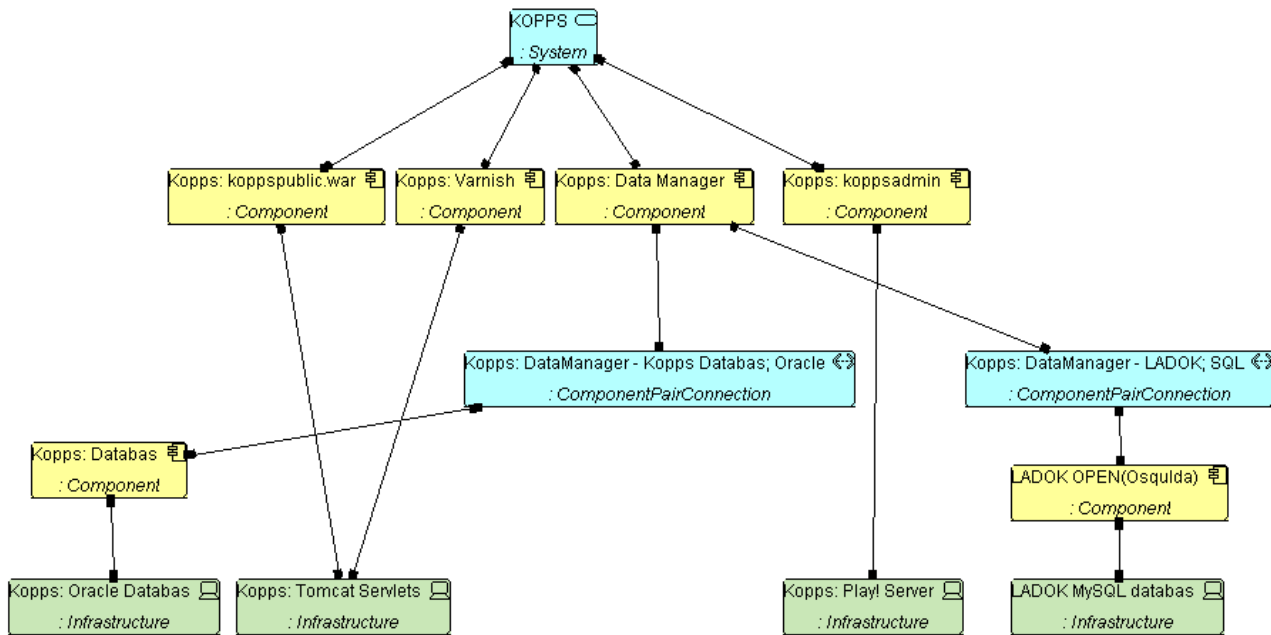


Figure 26: The databases, and the different kinds of technology that runs the components and the databases in KOPPS.

As can be seen in Figure 25 & Figure 26, there are external components used in KOPPS; Varnish and Data-Manager, and these also have couplings which are used in the calculations for the whole KOPPS system. As these are external components measurements such as size and cyclomatic complexity was not available, however a coupling measurement was obtained since they are still implemented and connected to other components.

The developers expressed uncertainty when it came to the coupling values to Varnish. In accordance to this, these values were thus used as input together with probabilistic additions; these additions were expressed as normal distributions with a variation of coefficient at 25%.

The coupling to Varnish and DataManager calculation tables presented earlier, and the calculations from EAAT are:

Component	Metric	Value
DataManager	CouplingAVG	4.67
	CouplingMAX	4.67
Varnish	CouplingAVG	4,67
	CouplingMAX	4,68

According to VoS, the System was built to persist, that is the system was built to last and not be replaced in a few short years by another system and thus care is taken in further development.

Using all the metrics previously presented together with the attributes, and the p-OCL statements found in APPENDIX P-OCL STATEMENTS, these System metrics were calculated for KOPPS:

System	Metric	Value
KOPPS	Size	35156
	Cyclomatic Complexity	7,0
	CouplingAVG	4,51
	CouplingMAX	4,69
	Cyclomatic Density	0,20

The system size is rather small compared to other systems, which suggests high modifiability within the system. The complexity of system is also within the limit which is defined in the theory, which further suggests high modifiability within the system. The cyclomatic density of the system is also according to the theory within the limits which suggests good modifiability. The coupling values however are rather high which suggests that any changes would have ripple effects to other components to which the system is connected, requiring modification to those as well. The system is owned and developed by KTH, which gives them almost full reign when it comes to what modifications are needed, at least to the non-external components.

## 10 DISCUSSION

This section presents the discussion regarding the different aspects of the project, first presented is the theory and method section and several points regarding the choices and limitations. Second is the discussion regarding the empirical data obtained and how it was used to create part of the results. Thirdly is a discussion regarding the analysis, the samplings used, the accuracy regarding the uncertainty in the analysis etc. Last is a discussion regarding the result which then leads to the conclusion in the next section.

### 10.1 THEORY AND METHOD

As presented in the method segment the case study provides information at a very insightful level into the phenomenon being studied, which in this case was the KTH core systems. Unlike a survey, experiments, history etc. Case studies are usually intense investigations of a singular or unique phenomenon with high focus on the analysis and conclusion. This leaves the case study open to much criticism in several aspects such as relevance and individual observations of the study and other subjective insights. As the subjective insights of the case study is unique to the study and the researcher any replication of the study to determine the accuracy of the results is hard if not impossible to perform. The limitations of the case study to unique phenomenon does provide critique in terms relevant sources and related work, thus the analysis, chain of evidence and reliability of a case study must be particularly resilient.

The alternatives, i.e. survey, history and experiments, are methods not quite suited for this kind of phenomenon as they have requirements which cannot be given, for instance experiments require control, which in this case could mean making software changes to systems for determining effect. The KTH systems being a unique setup, a case study seemed the most relevant research method.

There are some aspects in the adapted methods that can be questioned however, such as the study could have recorded the interviews and provided transcripts for the data collection, rather than simple interview notes. Travels and personal interviews rather than emails and telephone could have provided better data as well in some cases. The reporting could have been performed more thoroughly during the entire project. The models and meta model were refined several times during the project in order to adapt to the circumstances.

---

#### 10.1.1 ENTERPRISE ARCHITECTURE

To draw models of systems using Enterprise Architecture might not be considered the best approach if the goal is to draw detailed system models. One of the advantages of Enterprise Architecture is the abstractness to illustrate the connections between the IT and Process layers. To create system models using this method can thus be questioned as the abstractness of Enterprise Architecture in a sense prevents a certain level of detail in the systems.

Alternatives better suited for drawing system models where information regarding the systems would for instance be the Unified Modelling Language (UML), an extremely well accepted modelling language which is aspiring to become de facto standard in system development. UML is often used for drawing systems and the various aspects of a system, from user interaction with the system to technology which the system runs on. UML is in many aspects similar to the ArchiMate modelling language as mentioned in the ArchiMate section.

So why not use UML instead? One of the reasons is the audience, while UML is a modelling language directed at people mostly knowledgeable in information systems, the abstractness of Enterprise Architecture makes it possible for many different audiences to participate in the information. Another reason is the relations between the systems. As UML is focused on modelling object-oriented systems it is rather lacking when it comes to modelling in between systems and especially business processes.

Other mentioned languages and methods in related work such as ARIS, RM-ODP and BPMN are all focused rather on process mapping than systems and the relations between systems and processes. But since systems and the application layer was the focus of this study, this would be rather counterproductive in that sense. There are of course combinations of these different methods; one in particular which would be suited would be UML/BPMN, a combination of the UML and the BPMN, as they can be combined to express processes and systems. However we then return to the detailed level that is UML whose, as mentioned earlier, audience is not quite as wide compared to that of Enterprise Architecture.

As mentioned the advantage to Enterprise Architecture is the models; a picture says more than a thousand words. This phrase however suggests that all the spectators speak the same language, and this rings true to Enterprise Architecture as well. Previously established in the method segment, the Enterprise Architecture concept is still rather young and is still maturing. The modelling language used in this project is the ArchiMate modelling language; however this is not an international standard and other Enterprise Architectures use different languages or frameworks. Therefore the pictures here address those invested in the ArchiMate modelling language, those invested in Enterprise Architecture and those interested in KTH. However, if the models here could be presented in another figurative language, perhaps they could be addressed to more people? What does a colourful picture say to the colour-blind? Could the ArchiMate language in itself be a limitation to the audience? Not an entirely irrelevant metaphorical observation. What is important to mention is the fact that using Enterprise Architecture does not exclude the use of the previously mentioned frameworks, languages and methods. That is another advantage to the abstractness of Enterprise Architecture; if further details are necessary of the systems then UML is still available for drawing those details within the Enterprise Architecture method, framework or language.

TOGAF and Enterprise Architecture frameworks are mentioned in the in the theory section, however an Enterprise Architecture framework is not used in this study. As a framework entails much more than creating system models; it is an entire lifecycle to maintain Enterprise Architecture with a framework, e.g. it defines how to organize and structure the views of the Enterprise Architecture, provides different methods and tools for modelling those views and more. The modelling part which this project consists of is merely a small part of an Enterprise Architecture framework.

---

### 10.1.2 EAAT

As reiterated several times in this report, one of the most important parts in Enterprise Architecture is the models and in order to draw the models different tools are needed, depending on the models one wants to create. Tools for drawing models have been available even before the introduction of modern IT, and even though there are many tools better suited for drawing Enterprise Architecture than others, some models could still be drawn in simple drawing programs.

The main advantage of EAAT is not the modelling part of the application, and as mentioned in the related works section previously there already exists several tools for drawing and analysing Enterprise Architectures. So why use EAAT? The main reason as to why it was chosen is due to the uniqueness in the analysis.

All of the tools mentioned do not have noteworthy proficiencies to system quality analysis (6; 10; 41), that is analysis of non-functional requirements such as security, interoperability and as analysed in this study; maintainability. In addition EAAT provides functionality for providing uncertainty to capture assessment uncertainty, data quality and system configuration, which are important factors for Enterprise risk management (6).

So what would the alternatives be? Using one tool for modelling and something else for the analysis? Although this is possible especially in the modelling sense as there are as mentioned several tools for modelling Enterprise Architecture, the different analysis options for modifiability are restricted to different methods which are not suited for the Enterprise Architecture domain, e.g. ALMA, SAAM & ATAM (6; 10; 41).

A full scale comparison of the different tools and the advantages in the modelling sense is beyond this report, but the fact that EAAT combines a competent modelling Enterprise Architecture tool with a unique analysis framework for different kinds of system quality analysis such as modifiability, makes it a perfect choice for this study.

---

### 10.1.3 MODIFIABILITY

Unlike the previous mentioned modifiability methods, SAAM etc, the modifiability meta model used in this project addresses some issues that the other methods do not consider. Issues such as the diversity of people in an enterprise i.e. architects, developers & managers etc. and other aspects such as increasing number of systems and how they are coupled to each other (90). Also as mentioned above they do not consider such methods in an Enterprise Architecture domain.

The modifiability meta model for Enterprise Architecture which was used for this project is as mentioned the results of extensive research and work, however it is also very widespread. That is it considers many different aspects of the Enterprise in order to analyse the modifiability or change cost of an enterprise or project. This is of course something to appreciate when investigating an entire enterprise, but as this study was only mostly focused on systems it would have been abundant to consider all parts if only some of the information is related to the systems.

Starting out at the System and Component part of the modifiability meta model, the related parts can be roughly put into two categories, change and documentation. Change was interpreted to be changes in a hypothetical scenario to the system, components or architecture. As the project did not consider any change scenario as part of the study, since no such scenarios existed, the change category was not considered. It was learned in the initial data collection and pre-study that the documentation of architecture and components were out of date or non-existent, which as mentioned in the background was one of the reasons of this project. Thus as one of the goals was to create part of the documentation, consulting documentation for quality analysis of modifiability was not an option. As a result the only parts of the modifiability meta model considered was the systems and components.

As the modifiability analysis performed in this study was only based on a small segment of the entire meta model presented in the modifiability segment of the report, criticism towards the results must be taken into consideration. Criticism such as: How reliable are the results? If the results are accurate, does one need to consider the rest of the meta model? If not, how reliable are the results? A full analysis of the topic is beyond this study but the conclusions drawn in this study can be considered sound in regards to the

technical aspects of the systems. But it does raise one point of interest which should be mentioned, referencing Aristotle: "The whole is greater than the sum of its parts." That is, if the meta model is divided into segments and conducted as individual studies, would you reach the same conclusion if you used the whole meta model in one study? In essence both topics are related to validity, but the first topic regarding the results is related to this study, and as such should be further considered here. As the meta model is the result of extensive research of modifiability in Enterprise Architecture and has been proven to be accurate in several studies (69) it can be safe to assume that each part of the meta model is reliable to use by itself. But nevertheless the results are still considered with uncertainty for this and other reasons.

How to justify only using part of the modifiability meta model? As mentioned in the modifiability section the attributes which are used in the study are considered relevant to software maintainability, some decades before their adaptation in the modifiability meta model. The modifiability meta model however proves that they are still relevant in the Enterprise Architecture domain. The modifiability meta model however is not perfect: There are some exceptions to when the modifiability meta model has proven inaccurate. This is when the change project is considered small, i.e. below 2000 hours (57). However this study is not related to change project resource estimation as presented in the modifiability meta model but rather to system modifiability, and the system attributes adapted in the meta model of the study are as mentioned still relevant to system modifiability.

The attributes in the modifiability meta model mentions understandability, size, complexity and couplings both internal and external. Understandability was not considered since as in can be seen in the modifiability meta model (section 6) it is related to the technical documentation of the system which was not considered in this study. After all, as mentioned above, one of the goals of the study is to model the systems to aid in the documentation that does not exist or is out of date.

Size, a rather loose term can be considered in many aspects, such as number of components, number of methods or number of classes etc. In this study it was used as lines of code, firstly because it has been used before as a measurement of systems, and secondly because the alternatives would require more detailed modelling of the systems than was chosen in this study.

Both the Halstead and McCabe complexity measurements are accepted complexity measuring methods, both focused on the code and language of a software system and both have their advantages and their drawbacks. The Halstead measurement does not have a specific value which determines when a system is considered complex (73), unlike the McCabe cyclomatic complexity used in this study.

The internal and external couplings attributes were both considered as the coupling attribute since the same measurement was used for both coupling types, i.e. coupling as presented in the theory section. As the coupling attributes defines the connection between systems and also determines how tightly or loosely they are coupled to each other, one might argue the coupling attribute provides the most information regarding modifiability.

Another attribute used was the cyclomatic density as defined in (79), which is not present in the modifiability meta model but is essentially calculated using the two already established attributes: size and cyclomatic complexity, and is considered a measurement for modifiability. An alternative attribute to cyclomatic density which is also presented in the theory segment is Relative Logical Complexity (80; 81), but as can be seen in the referenced material it is essentially the same calculation. As mentioned it determines relative logic to size of the system or component, the disadvantage of the attribute however which is that the measurement relies on two other measurements which have to be accurate.



In a previous section another attribute was suggested to be investigated in this study, the *owner* of the system. This is of course a rather ambiguous term and could for instance mean the developers, the customers who bought the system, the people whom support it or perhaps a combination of all three. In this study the owners were considered the ones who maintain the system, i.e. the ones whom support the functionality and maintainability, this to reflect the analysis of the study i.e. maintainability. But how does the owner attribute affect modifiability? A naïve answer would be that if you don't own the system then you cannot modify it, however this is not the case. First of all, modifiability of a system does not vanish because you don't own it, looking only at the system it does not affect the inherent modifiability of the system, however what it does affect is the control or extent of what can be modified.

In terms of Enterprise Architecture modifiability however; not owning a system can have modifiability consequences. Information regarding the system not developed or maintained by the enterprise can be limited. As an example, in this study the data collected regarding KTH's own developed systems was superior and easier to obtain. Third party developers could be reluctant or hesitant to provide information regarding system attributes or documentation which is, as can be seen, relevant in the modifiability meta model for Enterprise Architecture modifiability analysis. As most systems are continuously developed to meet the needs of an enterprise, system modifiability is obviously an important fact to consider in any enterprise. Thus if system customization or information is limited due to third party developers this would most likely affect the modifiability of an enterprise.

---

#### 10.1.4 META MODEL

The meta model which was created for this study was as can be seen in the meta model section based on the ArchiMate model. Starting with the ArchiMate model as a base would create a strong foundation in both an Enterprise Architecture context and a modelling context. The end result of the meta model however does not contain that many ArchiMate entities, and the ones that do are mostly not part of the focus of the study. The reason for this is as mentioned that the focus of the study was the application layer of the ArchiMate model, thus removing most parts of the business and technology layer. However removing them completely could be considered counterproductive to mapping the connection between the layers, which is one of the advantages of Enterprise Architecture.

Further investigation into the different systems and the information available about them was also diverse, thus requiring a more abstract view than what ArchiMate could provide. As can be seen in the report segments for some systems the information was abundant compared to others in which no information was found or given at all. Thus in order to create a meta model which would then be able to somehow describe all the different systems with the varying amount of information, a certain level of abstraction is required, otherwise the meta model would not be able to model all the systems. This led to the *System* and *Component* entities which can be seen in the meta model. This was fortunately still applicable to the modifiability meta model and the attributes which were to be used for the modifiability analysis. ArchiMate however does not consider quality analysis which was the second goal of this study, and thus the theory behind the coupling attribute could not be applied to the relationships between entities in the model. Thus another entity called *CouplingPairConnection* was created to be able to represent the coupling attributes theory more accurately.

As the project spanned several systems of varying size and complexity, and the project was limited to a certain amount of time. The depth of the case study had to be limited to a level of every system in order to provide enough time to investigate every system. This was reflected in both the meta model and the in-

stantiated models, which can be seen in the results. In the end the ArchiMate was used as a foundation for the meta model of this study, but circumstances related to the study, as explained above, required certain adaptations in order to proceed.

However due to the level of abstraction taken, this was of course then reflected when modelling the systems and the components. Some components modelled could therefore represent several different smaller components or packets, which could have been modelled individually had the abstraction level been more detailed. Some of these smaller packets or components might then be more easy or harder to modify as those segments could be smaller, have fewer couplings and be less complex than all of them put together.

Although the theory excerpts data couplings as the best or loosest kind of coupling and in order to improve reusability and maintainability. The theory proven best case is sometimes not possible when it comes to reality implementation. Most software development projects aspire to achieve loose couplings and low complexity in the development, however this is also not always possible. First of all the attributes are rather subjective; a system with small size, low complexity and no couplings could be considered a very modifiable system as it is very easy to change due to low complexity and small size., also it will with very high probability not affect any other system. However these conditions can be achieved with a simple “Hello World!” program, a system that simply prints “Hello world!” on a screen. Thus the best one can aspire is to maintain small size, low complexity and loose couplings when developing systems but these attributes are sometimes sacrificed to achieve other goals, such as low costs, standards, security and performance. An example would be the LADOK system itself, providing a monumental standard to all its participants, which increases the communication efficiency between LADOK installations and universities; however as can be seen in the results one of the LADOK systems also has the worst modifiability of the systems investigated.

Reading through the report the following thought might occur: “Using only Enterprise Architecture analysis methods and tools which have been developed at KTH in order to analyse another department at KTH.” But reading this discussion section will have explained the reasons and advantages in choosing the methods and tools to collect and analyse the data for this study.

## 10.2 EMPIRICAL DATA

The empirical data collected was for two purposes as according to the goals of the project; creating models of the systems and analysing the modifiability. As mentioned in the empirical data segment this was divided into several phases for different purposes. The empirical data collected however was as mentioned very varying and distinct; some information was very detailed and up to current date, some information was very abstract and some information was abundant, detailed but out of date. Worst of all, some information was not given at all. Of all the information provided the ones whom provided the most amount of relevant information were the developers of KTH themselves to their systems.

With this diverse kind of data careful consideration regarding the accuracy, validity and the selection had to be done. In the cases when the information as out of date and could not be updated to current date, there was different levels of uncertainty regarding this data. These levels of uncertainty varied from high to low, i.e. from very abstract out-of-date data to certain up-to-date data. In most cases, when the information was irrelevant or abstract and out-of-date it was discarded, otherwise the data was updated where there were questions or irregularity. However at some parts there was no other data available and the out-of-

date data was the only data available, thus high levels of uncertainty was applied when modelling those models and applying the data to the attributes.

The data obtained especially in the metrics segment was somewhat lacking compared to the information regarding the models. There are several reasons for this, such as limited time available from the developers at the different departments or code analysis tools were not available, but the foremost was most likely the time invested in the models. As mentioned in the empirical data segment there were almost no already existing models for each system, these had to be made from scratch in order to determine what components realised the system, this data collection and modelling took more time than estimated to collect which then unfortunately affected the following data collection.

Validation regarding the models created was done through interviews with key personnel when possible; however validating metrics obtained was not possible as the values were simply handed to the researcher without any validation possibilities.

Despite the lack of size, coupling and complexity values of some systems; how the systems were coupled within and with other systems were still obtained or otherwise estimated. With this information conclusions about modifiability of certain parts could still be found to some extent.

### 10.3 ANALYSIS

Uncertainty can be expressed in different ways, in this project it was used as a normal distribution from mathematical probability statistics. However there are several different kinds of probability mathematical statistics methods, binominal, log-normal, etc. The normal distribution which was used in this project was chosen because of several reasons, but mostly because the *coefficient of variation* which was used to express uncertainty as a percentage. However if the data about the systems had been more accurate or detailed then perhaps there could have been a better method of expressing uncertainty in EAAT.

Some systems were analysed with very high levels of uncertainty as the attributes for these systems were estimated from inaccurate sources, e.g. Nouveau and NyA were estimated from out-of-date documentation. In accordance to analysis theory these were applied with probability in order to represent the uncertainty. However despite the probability applied, the values are still estimations and could be wrong to some degree.

In the cases where there was no uncertainty in the measured attributes then there would be no distribution and the histograms would look like the ones for Mina Sidor couplingMAX value, in which all the values were the same. This value as can be seen in Figure 27 is a single value at 4.7 for all the 7500 samples used.

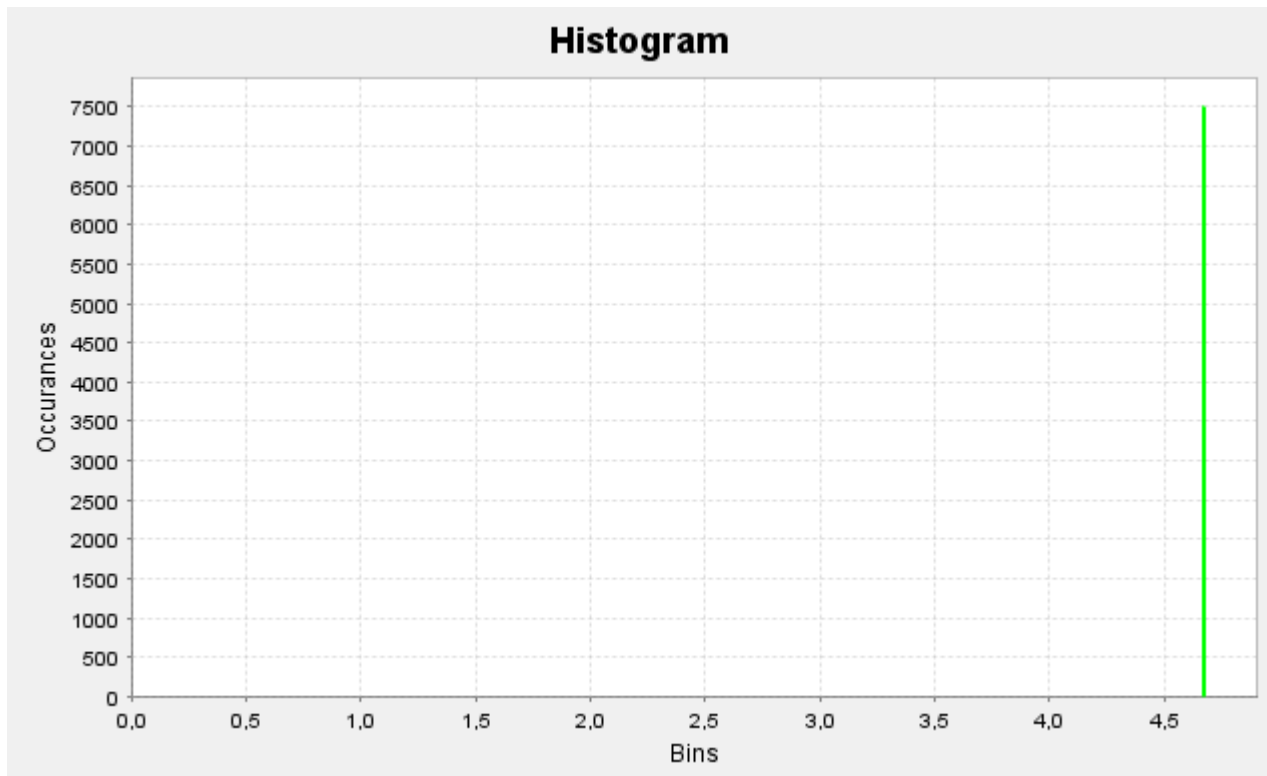


Figure 27: The histogram which depicts the samples of the couplingMAX of the Mina Sidor system, as can be seen there is no uncertainty.

The uncertainty applied was rather subjective to the information gathered, the sources which were used and how it was interpreted. In the case of Nouveau whose uncertainty was quite large compared to the other uncertainty measured of the other systems, suggests that the Existence probability seems to have a greater impact on uncertainty. However the reason could also be that the general information gathered regarding the Nouveau systems, the estimations regarding the system and the Existence probability all provided to the uncertainty. A complete comparison and investigation of the EAAT is beyond this study.

The analysis of the systems was done through the EAAT application which was developed and provided by ICS. Although it's effectiveness in modelling and its tremendous assistance in the analysis method, it is still a software application that is currently being developed. This contributed to the analysis time, as the application crashed at several occasions and had to be restarted and the calculations had to be performed again. This however was not all negative, the different calculations provided results which were all similar which further validated the calculation results. Using a system that is not considered complete and still in beta could be the target of criticism, but EAAT has been used in several different studies, and has been presented in several research papers as accurate which can be seen in section 5.4.

As mentioned in the analysis section of the report, the EAAT program was tried with higher number of samplings in order to assert a better mean value and graphs, etc. However as mentioned the application crashed when trying 10000 samplings, which urged lower sampling numbers. The sampling size was determined in respect to probability, i.e. the larger the size the better distributed and more accurate the probability will be presented. With the 7500 sampling numbers the program did not crash during the calculation and partial analysis was possible to be performed. However as mentioned, due to unidentified technical issues the application eventually became unresponsive and had to be restarted and had to perform another calculation. The results from the different calculations were compared and were very similar in the systems. This in a way provided more sampling numbers in total even though it is not visible in the pictures, but one

could argue that there were a total of 22500 samplings done over the three calculations, hence furthering the distribution of probability. That is, as the three calculations of 7500 samples results all provided almost the same results, one could enforce reliability in the calculations and data.

The modifiability analysis was done using the attributes acquired in the data collection, however in extension to this the models was able to describe different dependencies upon systems as presented in the results. These dependencies were merely composed from the models and the information regarding the systems. Other than being a goal for the project; the analysis is also highly dependent on the accuracy of the models, therefore this accuracy is critical in order to draw the conclusions regarding modifiability. This contributes to their priority in the project, therefore the priority in data collection for models before the software metrics. This could also be one of the reasons that the uncertainty results, i.e. the *coefficient of variation*, were rather low for most of the systems.

## 10.4 RESULTS

The models that were the end result of this study is presented in their own segments and one of the main uses of the models were to give an accurate current description of the systems as they are constructed. The full use of the models however is not limited to this however, for instance in this study they were used for analyzing the modifiability. If they are further extended with attributes related to other analysis, e.g. interoperability, the models could be used for other kind of quality analysis as well if the models are properly extended with the related attributes and the data required is collected.

The full extent of the models can be found in APPENDIX - MODELS, however as these models are quite large and do provide a lot of information regarding all three layers, they were further divided in order to present information less intimidating or confusing, not to mention provide models which could be useful to specific audience. However determining exactly what models are more important and how they provide information is hard to determine without any set goals regarding what models or views to create. The only goal set regarding the models was the system views and how they were connected, which was achieved with the aforementioned system models and the overview model.

With the results focused on the systems, the relations between the different enterprise layers, i.e. business to technology, are rather limited in respect to them being one of the advantages of modelling Enterprise Architectures. With the technology and business layers reduced to mostly one or two entities respectively, the relations between the different layers are in this study rather streamlined and abstract, not really depicting in full detail the entire business to system relationship. The business processes were however described further and investigated to a degree as mentioned earlier, but this did not contribute to the relationships mapping. The technology layer was represented by the single infrastructure entity, which represented all the different kinds of technology to which the systems run on. Due to these limitations of the entities, the ArchiMate external and internal views were not represented in these layers. This could have provided more detailed information regarding for instance how the different processes are related to the systems, and through what interface that the process activities are supported, and as can be seen the student process is related to all the systems.

The technology layer was represented by a single entity, here also not using the internal and external view, to present the different technological aspects which the systems run upon. From application server to database, this was all presented by a single entity. And while the systems all use several different kinds of technology and data storage, there is one database which is related to several systems the LADOK database.

What can be seen by including the business processes and the technology is what the critical systems of KTH are focused on. The student process is related to all the systems, the LADOK database is used for student information, thus the system focus on what is essentially the product of KTH, the students, and all the systems developed for and by KTH are meant to support the administrations of this.

The components that realise the systems represent rather large parts of the systems. The models created are relevant and usable by VoS, and if further investigations need to be performed at a less abstract level then the project already provide a limitation as to what the components contain.

And in that regard the models can tell us more, for instance as mentioned, almost all systems have a connection to the LADOK database, either directly to LADOK Osquar or to its read-only counterpart LADOK Osqulda. The only two systems which do not have a direct connection to this database are TimeEdit and DIAK as can be seen in Figure 28.

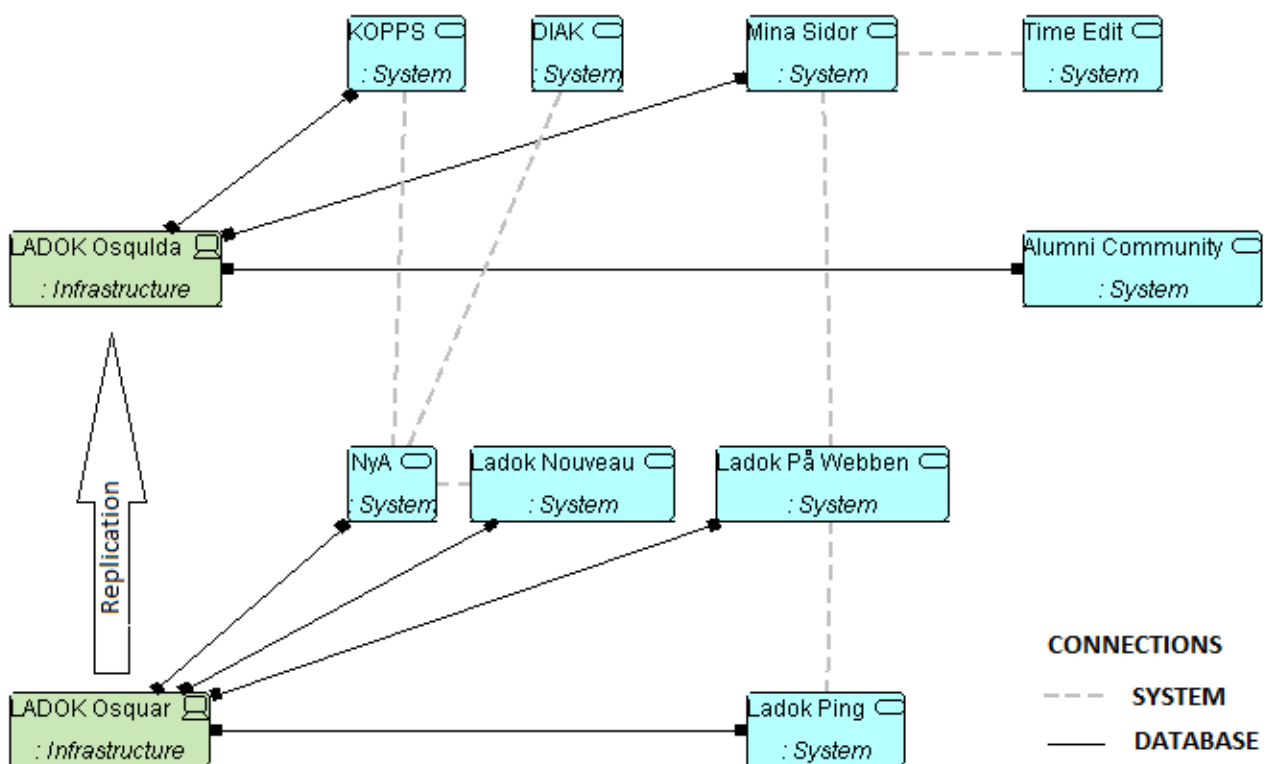


Figure 28 – The connections to LADOK is existent in almost all the systems.

The administrative tool for the LADOK database is the Nouveau system, i.e. it is the only way for LADOK administrators to directly administer input into the database. This suggests an indirect dependency on the Nouveau system for administrative purposes, as the other systems only have mostly read access and otherwise limited specific write operations. This makes the Nouveau system in a sense the most critical system to the KTH administrative processes. With that in mind, could modifications to Nouveau indirectly affect other systems? As can be seen in the models the only system to which Nouveau is connected is the NyA system, and although a change in Nouveau could affect NyA or even LADOK if developers were to make such a change, it is unlikely for this to happen. This is because LADOK is such a widely connected database and as such any change to it could affect all the connected systems, a property which is common with databases. In such a sense changes to Nouveau could affect related systems, but then this could be said about all systems connected to LADOK.

In the results it was found that most systems had average or higher coupling value which suggests that ripple effects could be quite common, the data however did contain several data couplings which however are not really represented in the calculations since the theory regarding the couplings express that couplings are to be represented by the worst type of coupling available. Thus in order to actually determine how the ripple effects occur one should look at the components and not only the systems.

The second goal of the study was the modifiability analysis, which was achieved using the models and analysis method described in the theory. EAAT provided the calculation results and histograms, but in order to interpret the calculations certain estimations and limits had to be used in order to categorize them. Most of the categorizations are based on the theory regarding the different attributes or software metrics, however the categorizations regarding for instance the size metrics is more subjective and the resulting categorization could have been different if one were to compare other systems in another company. E.g. small size could be categorized as what is in this study considered medium. While on the topic of attributes, the value max coupling was not considered in the system modifiability conclusions as it states the highest coupling found. This is an interesting observation in regards to the coupling average in a system, but the calculation does not specify where this coupling is located and as such is mostly interesting on a component level.

The resulting conclusions regarding the categorization can also be discussed, in the results the modifiability conclusions are rather subjective and considered from the authors interpretation of the results and categorization. What this means is that the modifiability conclusions which were reached from the results could be interpreted otherwise if someone else used the results and another categorization. In this study the different attributes were considered with small considerations to priority, i.e. the only attribute which was considered to have slightly more impact on modifiability was the coupling attribute, and the attribute cyclomatic density was the only attribute considered having slightly less impact on modifiability compared to the others which were otherwise all considered equal. The reasoning behind this is that the coupling attribute represents connections between systems and thus also represents the aforementioned ripple effect that modifications can have. The cyclomatic density attribute, while still an attribute for determining modifiability, has not been used in the related works to which this study builds upon and thus is harder to confirm the exact extent it has on modifiability compared to the other attributes, thus considered to have less impact. While the attributes were based on the theory the exact extent of the priority is not included and would be hard to measure as they thus considered from a subjective viewpoint. Thus the modifiability conclusions, which are still relevant and should be considered, can be the target of criticism since interpretation of the numbers and prioritizations of the attribute could differ and thus reach different conclusions.

By combining the models and the modifiability results the model presented in Figure 29 is a colour schemed picture of which systems are more critical and also harder to modify while also presenting how the systems are connected.

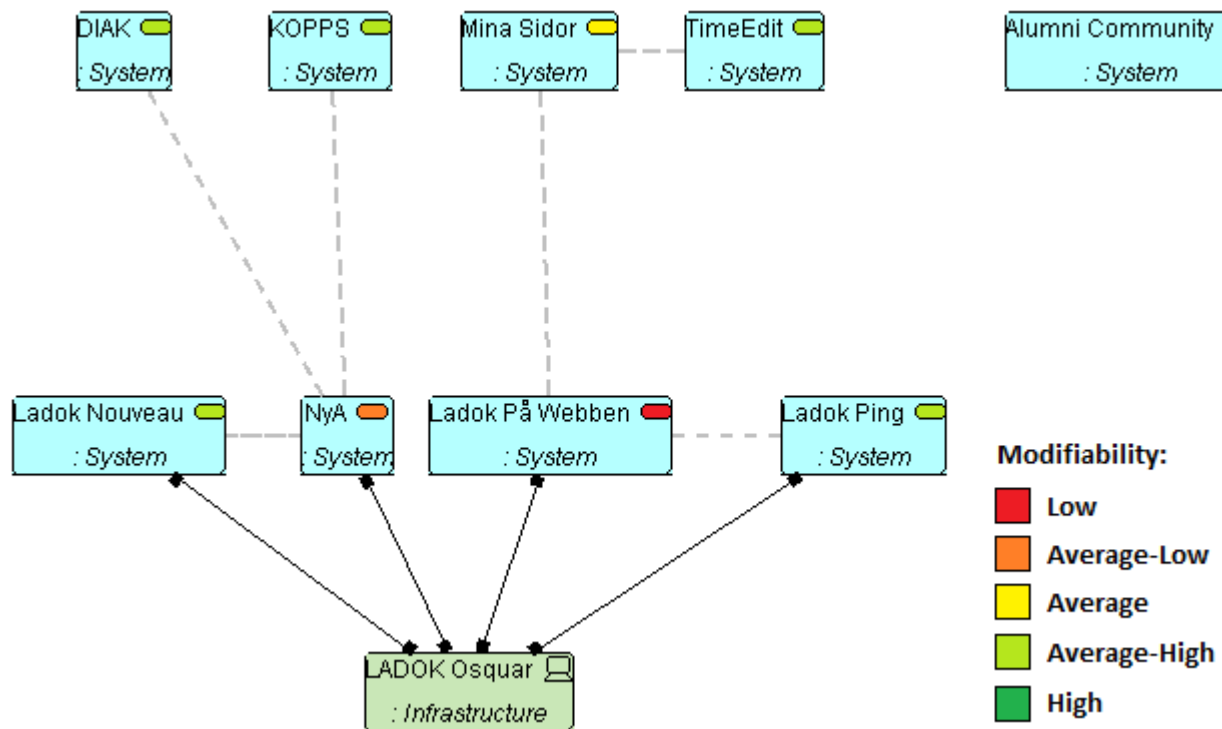


Figure 29: Modifiability, system connections and LADOK access. Nouveau, TimeEdit and DIAK are very uncertain.

In the figure there are three critical systems; two of which are coupled to several other systems: NyA & LpW. The other system is Nouveau, and KTH has no direct modifiability control over any of them other than requesting new features. What is interesting to notice which can be seen in Figure 29 is that these three systems together with LadokPing are the only systems which have direct contact with LADOK Osquar, however LadokPing is mainly used for reading operations while the others actually can perform write operations. There are 39 different LADOK installations and even more schools which use NyA for their admission process and not simply KTH alone, which could be an excuse for the low modifiability. However scalability should not interfere with modifiability. Therefore connecting any other system to LpW or NyA would most likely only decrease the modifiability if done carelessly, and thus any changes done in either of these systems could likely affect the newly connected systems as well. The models in Figure 31 & Figure 29 also provide an understanding of which related systems could be affected if modification were to be done on a specific system. As can be seen especially NyA and LpW are coupled to the other systems of this study, and thus modification to NyA or LpW would more likely cause effects on other systems. More information regarding the systems can be found in their specific segments for NyA & Mina Sidor.

Another point of discussion is the uncertainty; as can be seen in the results Table 5. Here the uncertainty is expressed as a calculation of the distribution in the histogram which was provided by EAAT. Comparing the results with previous related work (57) where uncertainty has been applied shows that the uncertainty calculated is rather small in comparison. The reason for this could be that most of the values and data while diverse still were represented by mostly the same values, i.e. coupling values in the components. Another reason could be that the systems are rather small in terms of components and as such the different couplings between them are, as the coupling theory explains, considered mostly by the worst type of coupling which could become more diverse if the components were considered smaller and more spread out. It is most likely though due to the different values of uncertainty which were applied to the data where values were highly uncertain. Especially in regards to how spread out the uncertainty was applied. This is shown



quite clear when comparing all the systems with the Nouveau system, especially the coupling attribute. Here the uncertainty is quite a lot larger compared to the other systems, this is most likely that all the data use was applied with a high level of uncertainty, whereas in other systems mostly light or medium level of uncertainty was used, and then only to some values. The Nouveau system is also the only system which has uncertainty applied to the complexity result, this is because of the Existence probability which was applied to one of its components, which none of the other systems had. It is also possible that the Existence attribute could affect the values of the coupling uncertainty, however this is not very likely as the only uncertainty which was expressed with the Existence probability in a component did not have any couplings to other components.

So how does the uncertainty in the results affect the modifiability conclusions? One answer to this question is the fact that the modifiability in the conclusions could be considered quite different. Due to the uncertainty, the modifiability could actually be better or worse than what was reached in this study. This is most obvious in the results for the Nouveau system whose uncertainty is clearly presented. However there is another aspect of uncertainty which is also depicted in the results and that is the lack of values and metrics for different systems. In the case where small amount of information is provided educated assumptions can be made, but where no information was provided, such as size, such assumptions cannot be made.

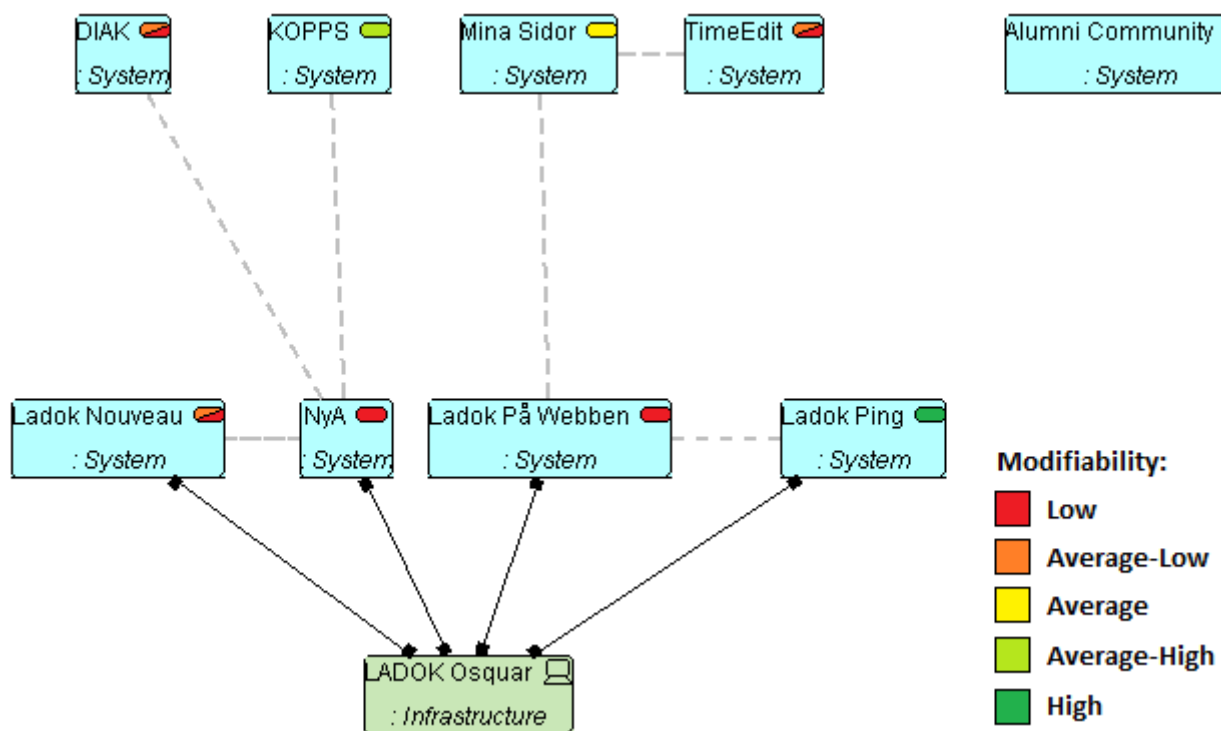


Figure 30: Theoretical modifiability analysis scenario based on the uncertainty of the data.

DIAM, Nouveau and TimeEdit systems did not provide any software metrics, thus the analysis to these systems were very probabilistic. This actually suggests that these systems could be less modifiable than presented in Figure 29 or as given by the analysis of the values obtained in Table 5 or presented in Table 7: The author's modifiability conclusions for each system. both of which can be found in the result section. Especially in the case of Nouveau and TimeEdit as these are not maintained by KTH themselves. This could also affect the systems to which they are coupled, such as Mina Sidor, as changes in TimeEdit thus can affect the solution in Mina Sidor. A theoretical modifiability analysis is presented in Figure 30. Even though some systems where data was highly estimated, e.g. Nouveau, the modifiability turned out to be average from the

data, this system could turn out to be very much harder to modify in reality based on more accurate metrics. This is of course entirely hypothetical and even though the metrics for the system was estimated, the actual model is still based on documentation. The functionality and location of the system still remains the same, thus the conclusions regarding the model is still sound. Although many attributes were used with levels of uncertainty the uncertainty to existence of the components is very low, i.e. the components existence is rarely questioned. Therefore the models are considered very accurate when describing how the systems look.

Another aspect of the project which was excluded due to the limitations was the external systems. This project only considers the connections to the systems which are being studied, with a few exceptions e.g. UG. There are for instance a large amount of systems which are connected to NyA the same way as KOPPS, at least one for each university which wants their courses information published on the admission website. In addition to those there are, among others, the Statistiska Centralbyrån (SCB), Statistics Sweden, and Posten, the national mail service in Sweden, which uses information from both NyA and LADOK. These connections could further contribute to the different modifiability results of the systems, should a more extensive study be performed. However as the systems which already are the most critical and concluded to have low modifiability, more connected systems would most likely only provide an even lower modifiability or more systems which would be affected by the modification.

What is important to notice when it comes to these systems (NyA, LpW and Nouveau) is that they are not owned or developed by KTH, but developed and maintained by ITS at Umeå university. This provides the insight that the two most interconnected systems, as well as the administrative tool for the core process database, are not KTH's own developed systems but rather national systems which they have to integrate with, as can be seen in Figure 31. This is not necessarily something entirely undesirable and the effects from this national integration could provide more benefits overall compared to other alternatives, but from a system modifiability perspective this can be seen as that there are dependencies on systems from which KTH do not have explicit development control over. Therefore any modifications requested by other universities will affect KTHs systems as well. To the NyA and LpW systems themselves this could provide a form of restriction or limitation to modifiability as the modification could unintentionally affect the related systems. An additional notice is that these two systems are as mentioned national systems, and there are a total of 39 different institutions of higher education in Sweden which use the LADOK systems, and even more institutions use the NyA system for admission. Thus any modification to NyA and LpW can therefore cause further effects on these other related systems as well and can consequently be considered more precarious to modify.

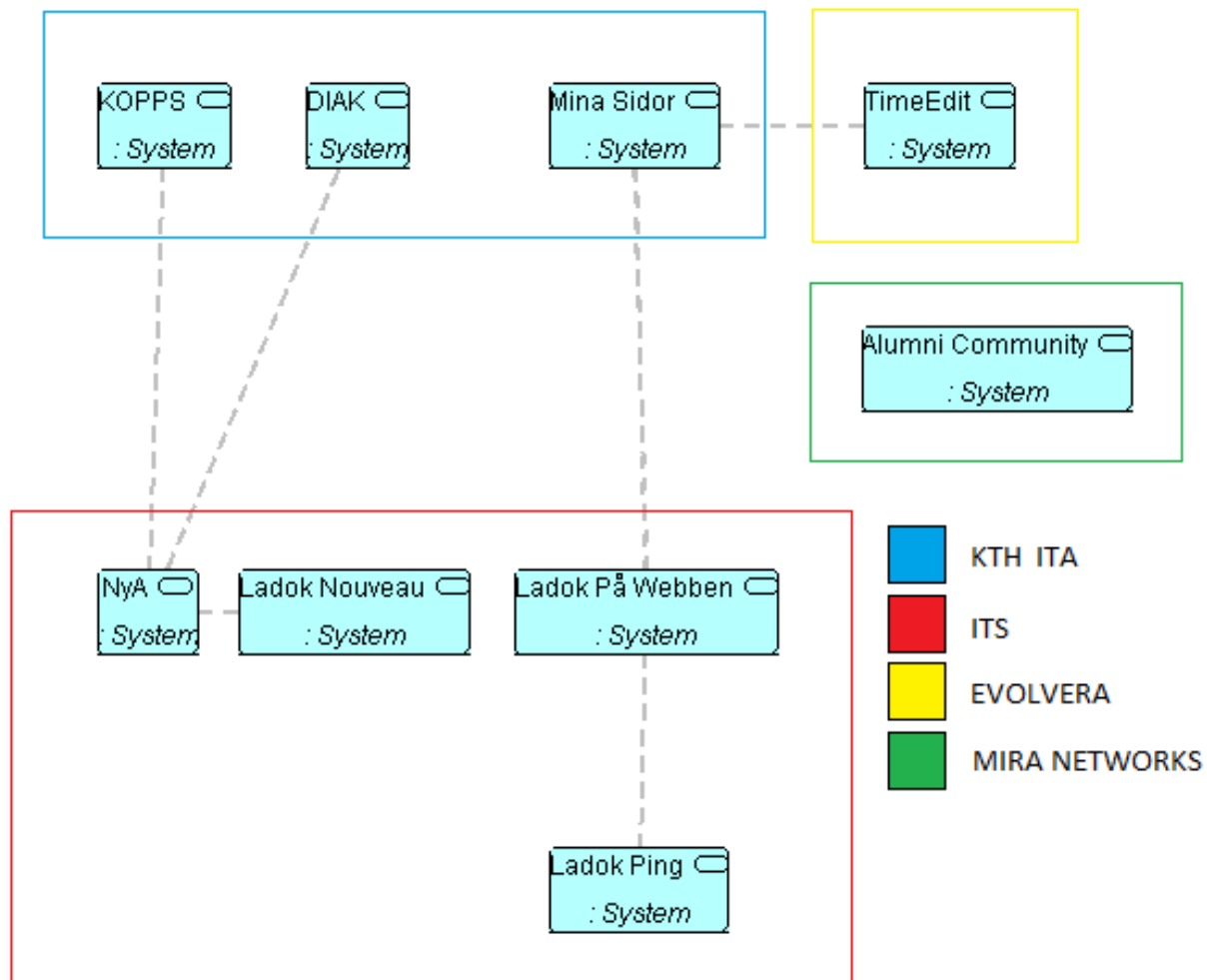


Figure 31 - The developers of the systems are quite diverse.

Mina Sidor is also coupled to TimeEdit, both directly and by an API developed by KTH which is used to present scheduling tasks in Mina Sidor; more information can be found in the TimeEdit section of the report. The coupling and API is dependent on the TimeEdit system and any changes in TimeEdit could thus cause effects in Mina Sidor. As presented in Figure 31 the TimeEdit is also a system which is not developed or owned by KTH, but by the commercial enterprise Evolvera. Therefore KTH's scheduling system is reliant upon another company, and this could further affect the modifiability available in Mina Sidor to some extent.

This is as mentioned previously related to the owner attribute, and while no consideration has been taken to the modifiability when concerning this attribute, mainly because such an investigation was not part of the goals of this study. Also considering the impact the owner has on a system can be quite diverse, and attributes which have been proven to have modifiability impacts which are not used in this study should probably also be included in such an investigation. But there are other things which the owner attribute can be used with, one for instance is to compare the systems developed by KTH and the systems developed by commercial software development companies. The ones present in Figure 31 are only two, Alumni Community and TimeEdit. Compared to these two systems which were concluded to have average-high modifiability, the systems developed by KTH have slightly lower modifiability; average to average-high.

(45) presents case studies of different modifiability analysis where systems of commercial enterprises have been analysed. The modifiability analysis in these papers considers entire enterprise projects, using the entire modifiability meta model presented earlier in full. However it is still possible to compare the different systems with the ones at KTH to some degree. This is an interesting comparison as KTH is a university in the public sector and not a commercial enterprise. The systems in the papers are varying in size and modifiability; some are tightly coupled, large and complex which suggests low modifiability, while some are more loosely coupled and small, suggesting high modifiability, and some systems which have other variations in between. The systems which have been developed at KTH can be concluded to be average leaning towards better in terms of modifiability when compared to commercially developed systems.

What is interesting to notice in this study is the mix of different systems at KTH; there are systems which were developed by KTH to meet their specific needs, systems which were developed to meet the requirements of universities at a national scale, and system which were developed by third party commercial company developers. Due to the varying origins of these systems further Enterprise Architecture Analysis could be interesting to conduct in a pure academic sense for further research, especially in areas as to how they integrate and compliment with each other, and if in house developed systems are better or worse than third party developed systems. In this study modifiability was concerned, and the results provided what could be considered a tie in modifiability between KTH's own developed systems and those bought by commercial developers. Further Enterprise Architecture Analysis active in other areas such as interoperability and security could provide interesting results for research since the system and their development source are so diverse. Comparing systems and solutions between commercial enterprises and the public sector might provide some interesting results, especially as KTH has such a rich foundation of system designs, some bought, some developed and some systems which they have to adapt to such as LADOK.

## 11 CONCLUSION

Using Enterprise Architecture for modelling systems and performing system quality analysis using the models and software metrics, information regarding the systems of KTH's composition and connections was determined together with a modifiability analysis.

The systems are not all connected to each other directly; most of them are only connected to one other system, which makes them less dependent on one another. Two systems were however widely connected to other systems; these were the NyA and LpW systems. Another system indirectly connected to all system was the Nouveau system, due to its core administrative functionality to the LADOK database, which in turn is connected to most systems.

The modifiability analysis showed that most of the systems have average or low modifiability. The low modifiability was also represented in the systems which were mostly connected to other systems, i.e. NyA and LpW, which signals that core systems were to which other systems are connected are also most likely the most costly to modify in terms of resources. Due to circumstances of the Nouveau system regarding high uncertainty this system could in fact be less modifiable than what was reached in the results, thus making three out of four systems closely connected to the LADOK database to have low modifiability.

The critical systems of KTH are closely related to their core processes, where the investigated systems are used to promote mainly the student process and processes related to the management of information related to the students, such as schedules, courses and programmes. Although the systems are all related to the student process, they are almost all also related to the LADOK systems and database, which shows that the systems are quite dependant on LADOK and thus the developers of LADOK. The critical systems of KTH are of quite different origin, some are developed in house, i.e. by KTH themselves, while others are joint efforts of different universities and some are third part developed and purchased by KTH. Comparing them further validates that an institution or university can competitively develop its own quality systems in order to increase their productivity. Rather than relying on third party developers, at least in the area of modifiability.

Although owner does not, or should not affect modifiability of a system in a direct degree, however as shown in this study, system or enterprise modifiability analysis regarding the systems, can be affected by how accessible said system information are, which can affect modification options, depending on the owner's support of the enterprise and system.

## 12 FUTURE WORK

This study focused on the application part of Enterprise Architecture, therefore several aspects of the modifiability meta model which contribute to modifiability was left out. Information regarding the other aspects which relate to modifiability was not really touched upon, information such as documentation, personnel, technical aspects and technical environments. Thus to get a more accurate and more enveloping modifiability analysis, and more viewpoints on what measures can be taken to modify the systems.

Since the goals of this project spanned several different systems, an abstract meta model that could be applied on all of the system was created from the information gathered from the systems. If a more elaborate study of each system was performed then a less abstract meta model could be used. This would increase the detail level of the system models, but would of course require a considerable amount of more time and data. This would provide further insight into e.g. the modifiability of each component. In order to achieve this VoS could adapt a less abstract meta model and analyse one system at a time using for instance more parts of the ArchiMate modelling language.

The ArchiMate language while still very useful, providing information regarding several parts of the enterprise, but how useful is the ArchiMate language to enterprise decision makers? Why does it not use attributes such as those used in this study for Enterprise Architecture Analysis? How would such attributes best be implemented? As touched upon in the discussion, ArchiMate is still a language that not everyone understands, there is of course the TOGAF framework which some Enterprise Architecture practitioners can relate to as they are combinable with each other. However in this study ArchiMate was extended with probabilistic attributes for quality analysis, which is based on software metrics not quite as widely understood by all, is there any ways to improve the understanding of this or “translate” it to appeal to a wider audience?

Can this “translation” then also be reflected in EAAT? By making EAAT use more modelling languages the audience can be widened, as mentioned, using Enterprise Architecture does not exclude using other frameworks for more detailed modelling such as UML for system modelling. Should then EAAT be extended to incorporate such modelling extensions as well?

The uncertainty used in this study was offered and presented in EAAT in two forms, the Existence of different relations and entities and the uncertainty expressed as normal distribution in the attributes. But which of these uncertainty forms have the most impact on uncertainty when combined? Are there better ways to present these uncertainties than as done in this study?

Also by further determining what systems are used in which business activity and for instance how modifications in the systems can modify the different processes at KTH. On the technology side there might be different technological bottlenecks from old technology (e.g. DIAK) that are affecting the systems and thus further the business processes.

The results regarding the systems studied showed that LADOK is used in almost all the processes studied as well, which makes LADOK a part of the core processes at KTH. However is this really necessary? LADOK is a system that is national only in Sweden, and not all universities use the system and what about the rest of Europe and the world? They do not use a similar system for every country. Seeing as the systems with the lowest measured modifiability are actually part of LADOK this does not seem to propose a winning concept from a modifiability perspective. Of course the other benefits of the system might overshadow the modifi-

ability advantages, but it does raise a question regarding the alternatives: What are they, and what would the advantages and disadvantages be? Perhaps if more Enterprise Architecture studies and analysis were to be performed at universities which do not use a national system similar to LADOK such answers could be found. There is another point to consider which might be uncovered with further insight into the systems: Is the results local contained only to KTH? Would there have been better results at another university? Or could they be worse? However, as it is outside of this study such answers can only be lightly speculated at.

Since there is an on-going project to migrate from LADOK to LADOK3, extra concerns should be taken into consideration regarding the LpW system, as this system is critical and several other systems rely on it. These systems will eventually need to integrate with the upcoming upgrades or replacements. Therefore every system which has connections to the LpW system should be investigated and mapped out in preparation for the migration. Similar arguments can be made for LadokPing and Nouveau since they are also part of LADOK.

LADOK 3 is a project for creating the next generation education administrative system for supporting the different universities in Sweden. This system is meant to replace the current LADOK system used today. The project is currently in progress as of this writing, and the conclusions of this project provides a foundation to Enterprise Architecture in a public sector, which is something that will be used in the LADOK 3 project, and thus also in its migration from the current LADOK system. The results can therefore be used in KTH's migration to LADOK 3, determining what systems that needs to be taken into consideration during the migration, and also further estimate which of them that might take more time to migrate.

What contributions does technology have to modifiability? This study focused on the application layer, but what about a similar study focusing only on the technical aspect of Enterprise Architecture? Technology is something that in the recent years have expanded at an almost alarming rate, new frameworks for developing systems and platforms for them to run on are many. Other new technologies such as cloud computing and other distributed system solutions are becoming more and more common. Can these new technological innovations have impact on the modifiability of systems? If say a company wants to move their database to a cloud (see cloud computing), how would this affect modifiability?

The cyclomatic density attribute used in this study provides some weight into two already established attributes of system modifiability. In this study cyclomatic complexity was used as a complexity measure, which is required for the use of cyclomatic density. While cyclomatic complexity does have recommended limits for modules in regards to complexity, there

As the size attribute is a rather subjective attribute and a larger size simply suggests lower modifiability as it increases, it does not have any min or max limit. The cyclomatic density attribute can thereby provide some insight into the relative complexity in regards to size of the module, which can then be considered together in terms of size, complexity and couplings when determining the modifiability of the system. A large sized system can still be considered highly modifiable if the cyclomatic density is low, i.e. size can then also be considered in regards to the complexity of the code rather than regarding them individually.

A full investigation as to how the owner of a system could affect modifiability in Enterprise Architecture is beyond this study, but summarized, in this study it has contributed to insight into how different owners can affect the modifiability analysis through various means. The different system owners of the KTH systems provides an unique scenario of how an enterprise has its core critical system spread out, developed and maintained by different owners. An enterprise having its systems developed and maintained by another company is not unheard of, but it is also something that has emerged during the recent years and with the

development of new technology such as distributed systems. So how much could a different owner of a system affect the modifiability of a system?

Another topic of interest is: How common is the mix of original developers of the systems? How does the mix of systems affect the modifiability of the systems from an Enterprise Architecture perspective? One aspect which is not covered in this study is the diversity of technology and how different systems developed in different languages affect modifiability due to compatibility issues.

With more and more distributed systems and technology such as cloud computing, where the infrastructure is maintained by another company, what options become available for large enterprises that have subsidiaries and use distributed systems? Can Enterprise Architecture be used for mapping similar systems functionality for systems which are not maintained by the mother company but instead by a subsidiary enterprise? What if the company has most of its systems running on cloud services? Does this have any effect on modifiability? In such a sense what can the owner attribute be considered to have more impact if further researched?



## 13 ACKNOWLEDGEMENTS

I would like to thank: Robert Lagerström & Magnus Österlind at ICS for their support and advice during the entire project. Katarina Jonsson Berglund for her assistance and information regarding KTH its systems and processes. The people at the VoS department of KTH for their support and help. The software developers at KTH for their patience, assistance and information regarding the systems modelled and analysed. MIRA for their information and help regarding Alumni Community, and VHS and ITS at Umeå university for their help and assistance with the NyA and LADOK systems. And everyone else who assisted in the project who I might have forgotten to mention, you know who you are!

## 14 REFERENCES

1. **Ullberg, Johan, Johnson, Pontus and Buschle, Markus.** A Modeling Language for Interoperability Assessments. [ed.] Marten Sinderen, et al., et al. *Enterprise Interoperability*. s.l. : Springer Berlin Heidelberg, 2011, Vol. 76, pp. 61-74. 10.1007/978-3-642-19680-5\_7.
2. **Ross, J.W., Weill, P. and Robertson, D.** *Enterprise architecture as strategy: Creating a foundation for business execution*. s.l. : Harvard Business Press, 2006. p. 288. ISBN: 1591398398.
3. *Enterprise architecture: Management tool and blueprint for the organisation.* **Jonkers, H., et al., et al.** 2, s.l. : Springer, 2006, Information Systems Frontiers, Vol. 8, pp. 63-66. ISSN: 13873326.
4. **Lankhorst, M.** *Enterprise architecture at work: Modelling, communication and analysis*. s.l. : Springer-Verlag New York Inc, 2009. p. 345. Vol. 10. ISSN: 01121642 ISBN: 3540243712.
5. *ArchiMate 1.0 Specification.* **Iacob, ME, et al., et al.** 5, s.l. : The Open Group, 2009, Zhonghua er bi yan hou tou jing wai ke za zhi Chinese journal of otorhinolaryngology head and neck surgery, Vol. 42, p. 323. ISSN: 16730860 ISBN: 1931624801.
6. *A Tool for Enterprise Architecture Analysis using the PRM formalism.* **Buschle, M., et al., et al.** 1, s.l. : Springer, 2011, Information Systems Evolution, Vol. 72, pp. 108-121.
7. *The DoDAF Architecture Framework Version 2.02.* **Role, A. and Role, D.** s.l. : US Department of Defence, 2010, DoDAF Journal.
8. **Group, The Open.** *TOGAF Version 9*. s.l. : Van Haren Pub, 2009.
9. *A framework for information systems architecture.* **Zachman, J.** s.l. : IBM, 1999, IBM system Journal 38 (2/3), pp. 454-470.
10. *A tool for enterprise architecture analysis.* **Johnson, P., et al., et al.** s.l. : IEEE, 2007. 11th IEEE International Enterprise Distributed Object Computing Conference EDOC 2007. Vol. 5, pp. 142-142. ISSN: 15417719 ISBN: 0769528910.
11. *A method for creating enterprise architecture metamodels--applied to systems modifiability analysis.* **Lagerström, R., et al., et al.** 5, s.l. : Citeseer, 2009, International Journal of Computer Science and Applications, Vol. 6, pp. 89-120.
12. **Schekkerman, J.** *How to survive in the jungle of enterprise architecture frameworks: creating or choosing an enterprise architecture framework*. s.l. : Trafford Publishing, 2006. p. 268. ISBN: 9781412016070.
13. **Johnson, P. and Ekstedt, M.** *Enterprise Architecture: Models and Analyses for Information Systems Decision Making*. s.l. : Studentlitteratur, 2007. ISBN: 9789144027524.
14. *A method to define an Enterprise Architecture using the Zachman Framework.* **Pereira, C.M. and Sousa, P.** s.l. : ACM Press, 2004. Proceedings of the 2004 ACM symposium on Applied computing SAC 04. pp. 1366-1371. ISBN: 1581138121.

15. *Tailoring DoDAF For Service-Oriented Architectures*. **Ang, Huei-Wan, Dandashi, F. and McFarren, M.** s.l. : IEEE, 2006. Vol. 06, pp. 1-8. 142440617X.
16. **priceless**. Virtually Priceless Thoughts. [Online] 28 February 2010. [Cited: 10 07 2012.] <http://virtuallypriceless.org/blog/tag/health/>.
17. *Developing fully functional E-government: A four stage model*. **Layne, Karen and Lee, Jungwoo.** 2, s.l. : Elsevier, 2001, Government Information Quarterly, Vol. 18, pp. 122-136. ISSN: 0740-624X.
18. *Enterprise Architecture Integration in E-Government*. **Janssen, M. and Cresswell, A.** s.l. : Computer Society Press, 2005. Proceedings of the 38th Annual Hawaii International Conference on System Sciences HICSS05 Track 5. Vol. 00, p. 118b. ISSN: 1530-1605 .
19. *E-government: the realities of using IT to transform the public sector*. **Hazlett, S.A. and Hill, F.** 6, s.l. : MCB UP Ltd, 2003, Managing Service Quality, Vol. 13, pp. 445-452. ISSN: 09604529.
20. *Enterprise architecture implementation and management: A case study on interoperability*. **Hjort-Madsen, K.** s.l. : Computer Society Press, 2006. Proceedings of the 39th Annual Hawaii International Conference on System Sciences HICSS06 Track 6. Vol. 00, pp. 1-10. ISSN: 15301605 ISBN: 0769525075.
21. *Institutional patterns of enterprise architecture adoption in government*. **Hjort-Madsen, K.** 4, s.l. : Emerald Group Publishing Limited, 2007, Transforming Government: People, Process and Policy, Vol. 1, pp. 333-349. ISSN: 17506166.
22. *Towards an enterprise architecture for public administration using a top-down approach*. **Peristeras, V. and Tarabanis, K.** 4, s.l. : Palgrave Macmillan, 2000, European Journal of Information Systems, Vol. 9, pp. 252-260. ISSN: 0960085X.
23. *Interoperability frameworks and enterprise architectures in e-government initiatives in Europe and the United States*. **Luis and Guijarro.** 1, s.l. : Elsevier, 2007, Government Information Quarterly, Vol. 24, pp. 89-101. ISSN: 0740-624X .
24. *Building an enterprise architecture for public administration: a high level data model for strategic planning*. **Tarabanis, K., Peristeras, V. and Fragidis, G.** s.l. : Moderna organizacija, 2001. Proceedings of the Ninth European Conference on Information Systems. pp. 987-998. ISBN: 9612321213.
25. *Analyzing Enterprise Architecture in National Governments: The Cases of Denmark and the Netherlands*. **Janssen, Marijn and Hjort-Madsen, Kristian.** s.l. : IEEE, 2007. 2007 40th Annual Hawaii International Conference on System Sciences HICSS07. p. 218a. ISSN: 1530-1605 ISBN: 0769527558.
26. **Lankhorst, M.** ArchiMate Language Primer--Introduction to the ArchiMate Modelling Language for Enterprise Architecture. Version 1.0. *ArchiMate Language Primer*. s.l. : Telematica Instituut, 2004. p. 27.
27. *ARIS Toolset: A software product is born*. **August-Wilhelm and Scheer.** 8, s.l. : Elsevier, 1994, Information Systems, Vol. 19, pp. 607-624. ISSN: 0306-4379.
28. **Technologies, Troux.** METIS. *METIS*. December 2011.
29. **Metastorm.** BPA - Business Process Analysis. *BPA - Business Process Analysis*. December 2011.

30. **Paul., Mark; Clements.** *ATAM: Method for architecture evaluation*. Software Engineering Institute, Carnegie Mellon University, Pittsburgh. Pittsburgh : s.n., 2000. Tech. rep.
31. *SAAM: a method for analyzing the properties of software architectures*. **Kazman, Rick, et al., et al.** Los Alamitos, CA, USA : IEEE Computer Society Press, 1994. pp. 81-90. ISBN: 0-8186-5855-X.
32. **Jensen, F.V. and Nielsen, T.D.** *Bayesian networks and decision graphs*. s.l. : Springer, 2007. ISBN: 9780387682815 LCCN: 2006938666.
33. *Analysis and application scenarios of enterprise architecture: An exploratory study*. **Bucher, T., et al., et al.** s.l. : IEEE Computer Society, 2006. Journal of Enterprise Architecture. Vol. 3, pp. 28-28. ISBN: 0769527434 DOI: 10.1109/EDOCW.2006.22.
34. **Johansson, Erik.** *Assessment of Enterprise Information Security : How to make it Credible and Efficient*. KTH, Industrial Information and Control Systems, KTH, Industrial Information and Control Systems. s.l. : KTH, 2005. p. 28, Ph.D. dissertation. QC 20101028.
35. *Enterprise architecture models for cyber security analysis*. **Ekstedt, M. and Sommestad, T.** s.l. : IEEE, 2009. IEEE PES Power Systems Conference and Exposition. pp. 1-6. ISBN: 9781424438105.
36. *An AHP-based approach toward enterprise architecture analysis based on enterprise architecture quality attributes*. **Razavi, M., Aliee, F. Shams and Badie, K.** 2, s.l. : Springer, 2011, Knowledge and Information Systems, Vol. 28, pp. 449-472. ISSN: 02191377.
37. *Formalizing analysis of enterprise architecture*. **Johnson, P., Nordström, L. and Lagerström, R.** 2, s.l. : Springer, 2007, Enterprise Interoperability, pp. 35-44.
38. *Change impact analysis of enterprise architectures*. **Boer, FS De, et al., et al.** Las Vegas, NV, USA : IEEE Systems, Man, and Cybernetics Society, 2005. IEEE International Conference on Information Reuse and Integration. pp. 177-181. ISBN: 0780390938 DOI: 10.1109/IRI-05.2005.1506470.
39. *Enterprise architecture analysis with xml*. **Boer, FS De, et al., et al.** Big Island, HI, USA : IEEE Computer Society, 2005. Proceedings of the 38th Annual Hawaii International Conference on System Sciences. Vol. 00, pp. 222b--222b. ISSN: 15301605 ISBN: 0769522688 DOI: 10.1109/HICSS.2005.242.
40. *Enterprise architecture analysis with extended influence diagrams*. **Johnson, P., et al., et al.** 2-3, s.l. : Springer, 2007, Information Systems Frontiers, Vol. 9, pp. 163-180. ISSN: 13873326.
41. *A Tool for Interoperability Analysis of Enterprise Architecture Models using Pi-OCL*. **Ullberg, J., et al., et al.** s.l. : Springer, 2010, Enterprise Interoperability IV: Making the Internet of the Future for the Enterprise, pp. 81-90. ISBN 978-1-84996-256-8.
42. **Närman, Per, Buschle, Markus and Ekstedt, Mathias.** An integrated enterprise architecture framework for information system quality analysis.
43. *Experiences with ALMA: Architecture-Level Modifiability Analysis*. **Lassing, Nico, et al., et al.** 1, 2002, Journal of Systems and Software, Vol. 61, pp. 47-57. ISSN: 0164-1212.
44. *Analyzing Software Architectures for Modifiability*. **Bengtsson, Perlof, et al., et al.** 2000, Science, pp. 1-28.

45. *Software Change Project Cost Estimation--A Bayesian Network and a Method for Expert Elicitation*. **Lagerström, R., et al., et al.** Fraunhofer IESE, Kaiserslautern, Germany : s.n., 2009. Third International Workshop on Software Quality and Maintainability.
46. *Systems and software engineering -- Vocabulary*. **IEEE**. 1, 15 2010, ISO/IEC/IEEE 24765:2010(E), Vol. 2010, pp. 1-418. DOI: 10.1109/IEEESTD.2010.5733835.
47. *Cost models for future software life cycle processes: COCOMO 2.0*. **Boehm, Barry, et al., et al.** 1, s.l. : Springer Netherlands, 1995, Annals of Software Engineering, Vol. 1, pp. 57-94. 10.1007/BF02249046. ISSN: 1022-7091.
48. *Architecture-level modifiability analysis (ALMA)*. **Bengtsson, PerOlof, et al., et al.** 1, s.l. : Elsevier, 2004, Journal of Systems and Software, Vol. 69, pp. 129-147. ISSN: 0164-1212 DOI: 10.1016/S0164-1212(03)00080-3.
49. *PASA: A method for the performance assessment of software architectures*. **Williams, Lloyd and Smith, Connie U.** Rome, Italy : ACM, 2002. Proceedings of the 3rd international workshop on Software and performance. pp. 179-189. 1-58113-563-7.
50. *Comparison of scenario-based software architecture evaluation methods*. **Babar, M.A. and Gorton, I.** s.l. : Software Engineering Conference, 2004. 11th Asia-Pacific, 2004. pp. 600 - 607. 1530-1362.
51. *Metrics for assessing a software system's maintainability*. **Oman, P and Hagemeister, J.** s.l. : IEEE, 1992. Software Maintenance, 1992. Proceedings, Conferens on. pp. 337-344.
52. *A definition and taxonomy for software maintainability*. **Oman, P, Hagemeister, J and Ash, D.** 08, Moscow : Software Engineering Lab, Tech Rep., 1992, Vol. 91. ID83843.
53. *Software engineering--product quality--part1: Quality model*. **Standardization, International Organization for.** s.l. : International Organization for Standardization, 2001. International standard ISO/IEC TR 9126-1:2001(E).
54. *Software engineering--product quality--part2: External metrics*. **Standardization, International Organization for.** s.l. : International Organization for Standardization, 2003. International standard ISO/IEC TR 9126-2:2003(E).
55. *Software engineering--product quality--part 3: Internal metrics*. **Standardization, International Organization for.** s.l. : International Organization for Standardization, 2003. International standard ISO/IEC TR 9126-3:2003(E).
56. *Software engineering--product quality--part 4: Quality in use metrics*. **Standardization, International Organization for.** s.l. : International Organization for Standardization, 2004. International standard ISO/IEC TR 9126-4:2004(E).
57. **Lagerström, Robert.** *Enterprise Systems Modifiability Analysis : An Enterprise Architecture Modeling Approach for Decision Making*. KTH, Industrial Information and Control Systems, KTH, Industrial Information and Control Systems. s.l. : KTH, 2010. pp. x, 29, Ph.D. dissertation. QC20100716.
58. **Yin, R.K.** *Case study research: design and methods*. s.l. : Sage Publications, 2003. ISBN: 9780761925521 LCCN: 2002152696.

59. *Guidelines for conducting and reporting case study research in software engineering*. **Runeson, P. and Höst, M.** 2, s.l. : Springer, 2009, Empirical Software Engineering, Vol. 14, pp. 131-164. ISSN: 13823256 ISBN: 1382325615737616.
60. *Qualitative case study methodology: Study design and implementation for novice researchers*. **Baxter, P. and Jack, S.** 4, s.l. : TQR Community of Nova Southeastern University, 2008, The Qualitative Report, Vol. 13, pp. 544-559.
61. *The case study: a useful research method for information management*. **Smith, N.C.** 3, s.l. : Nature Publishing Group, Palgrave Macmillan, Routledge, 1990, Journal of Information Technology, Vol. 5, pp. 123-133. ISSN: 02683962.
62. *Guidelines for industrially-based multiple case studies in software engineering*. **Verner, JM, et al., et al.** s.l. : IEEE, 2009. Third International Conference on Research Challenges in Information Science. pp. 313-324. ISBN: 9781424428649.
63. *Using enterprise architecture models for system quality analysis*. **Närman, P., et al., et al.** s.l. : IEEE, 2008. 2008 12th International IEEE Enterprise Distributed Object Computing Conference. pp. 14-23. ISSN: 15417719 ISBN: 9780769533735.
64. *Essential layers, artifacts, and dependencies of enterprise architecture*. **Winter, R. and Fischer, R.** 2, s.l. : AEA - Association of Enterprise Architects, 2007, Journal of Enterprise Architecture, Vol. 3, pp. 7-18.
65. *Enterprise Meta Modeling Methods--Combining a Stakeholder-Oriented and a Causality-Based Approach*. **Lagerström, R., et al., et al.** s.l. : Springer, 2009, Enterprise, business-process and information systems modeling, Vol. 29, pp. 381-393. ISBN: 9783642018619.
66. **Morgan, M.G. and Henrion, M.** *Uncertainty: a guide to dealing with uncertainty in quantitative risk and policy analysis*. s.l. : Cambridge University Press, 1992. p. 332. ISBN: 0521427444.
67. **Booch, G., Rumbaugh, J. and Jacobson, I.** *Unified Modeling Language User Guide, The (Addison-Wesley Object Technology Series)*. s.l. : Addison-Wesley Professional, 2005.
68. **L., A. Fuernweger.** UML Tool for Fast UML Diagrams. [Online] December 2011. <http://www.umlet.com>.
69. *Architecture analysis of enterprise systems modifiability--Models, analysis, and validation*. **Lagerström, R., Johnson, P. and Höök, D.** 8, s.l. : Elsevier, 2010, Journal of Systems and Software, Vol. 83, pp. 1387-1403.
70. **Bass, L., Clements, P. and Kazman, R.** *Software architecture in practice*. s.l. : Addison-Wesley, 2003. ISSN: 03008495 ISBN: 0321154959.
71. **Grubb, P. and Takang, A.A.** *Software maintenance: concepts and practice*. s.l. : World Scientific, 2003. ISBN: 9789812384256 LCCN: 2005297903.
72. *Deriving structurally based software measures*. **Fenton, N. and Melton, A.** 3, s.l. : Elsevier, 1990, Journal of Systems and Software, Vol. 12, pp. 177-187. ISSN: 01641212.
73. **Halstead, M.H.** *Elements of software science*. s.l. : Elsevier, 1977. ISBN: 9780444002051 LCCN: 77001321.

74. *A Complexity Measure*. **McCabe, T.J.** 4, dec. 1976, Software Engineering, IEEE Transactions on, Vols. SE-2, pp. 308-320. ISSN: 0098-5589.
75. *Using architectural models to predict the maintainability of enterprise systems*. **Lagerstrom, R. and Johnson, P.** 2008. IEEE. pp. 248-252.
76. *Measuring coupling and cohesion in object-oriented systems*. **Hitz, M. and Montazeri, B.** 1995, Angewandte Informatik, Vol. 50, pp. 1-10. ISSN: 0013-5704.
77. *Design complexity measurement and testing*. **McCabe, T.J. and Butler, C.W.** 12, s.l. : ACM, 1989, Communications of the ACM, Vol. 32, pp. 1415-1425. ISSN: 00010782.
78. *Structured testing: A testing methodology using the cyclomatic complexity metric*. **Watson, A.H., McCabe, T.J. and Wallace, D.R.** 235, s.l. : National Institute of Standards and Technology, 1996, NIST special Publication, Vol. 500, pp. 1-114.
79. *Cyclomatic complexity density and software maintenance productivity*. **Gill, G.K. and Kemerer, C.F.** 12, s.l. : IEEE Computer Society, 1991, Software Engineering, IEEE Transactions on, Vol. 17, pp. 1284-1288. ISSN: 00985589 DOI: 10.1109/32.106988.
80. **Gilb, T.** *Software metrics*. s.l. : Winthrop Publishers, 1977. ISBN: 9780876268551 LCCN: 76040163.
81. *Software Metrics for Predicting Maintainability*. **Frappier, M., Matwin, S. and Mili, A.** 3, s.l. : Virginia Polytechnic, 1994, Software Metrics Study: Tech. Memo, Vol. 2, pp. 129–143. ISSN: 1096908X.
82. **Software, McCabe.** McCabe Software Outsourcing Development Software Metrics. *McCabe Software Web site*. [Online] McCabe Software, 29 04 2012. [Cited: 29 April 2012.] <http://www.mccabe.com/pdf/McCabeCodeQualityMetrics-OutsourcedDev.pdf>.
83. *Concepts for modeling enterprise architectures*. **Jonkers, H., et al., et al.** 3, s.l. : World Scientific Publishing Co. Pty. Ltd, 5 Toh Tuck Link, 596224, Singapore, 2004, International Journal of Cooperative Information Systems, Vol. 13, pp. 257-287. ISSN: 02188430.
84. **Allweyer, T.** *Bpmn 2.0*. s.l. : Books on Demand, 2010.
85. *Enterprise architecture modelling - the issue of integration*. **Lankhorst, M.M.** 4, s.l. : Elsevier, 2004, Advanced Engineering Informatics, Vol. 18, pp. 205-216.
86. **O.M.G.A.** *Object Constraint Language Specification*. s.l. : May, 2010.
87. **Bort, Guillaume.** *Play Framework*. [Online] Typesafe, Zenexity. [Cited: 29 April 2012.] <http://www.playframework.org>.
88. **Leijdekkers, Bas.** MetricsReloaded. *IntelliJ IDEA Plugins*. [Online] IntelliJ IDEA. [Cited: 29 April 2012.] <http://plugins.intellij.net/plugin/?idea&id=93>.
89. **Foundation, Apache.** *Apache Tomcat*. [Online] Apache Foundation. [Cited: 29 April 2012.] <http://tomcat.apache.org/>.

90. **Lagerström, Robert.** *Enterprise Systems Modifiability Analysis : An Enterprise Architecture Modeling Approach for Decision Making*. KTH, Industrial Information and Control Systems, KTH, Industrial Information and Control Systems. s.l. : KTH, 2010. pp. x, 29, Ph.D. dissertation. QC20100716.
91. **CommunityZero.** *Uniface Global Network Online Community*. [Online] Ramius® Corporation. [Cited: 29 April 2012.] <http://www.uniface.info>.
92. **RedHat.** *JBoss Community*. [Online] JBoss Community. [Cited: 29 April 2012.] <http://www.jboss.org>.
93. **Foundation, Apache.** *Apache HTTP Server Project*. [Online] Apache Foundation. [Cited: 29 April 2012.] <http://httpd.apache.org/>.
94. **Source, Open.** *Community developed LDAP software*. [Online] OpenLDAP. [Cited: 29 April 2012.] <http://www.openldap.org>.
95. **Telerik.** *Telerik .NET Frameworks*. [Online] Telerik. [Cited: 29 April 2012.] <http://www.telerik.com/>.
96. **Aspose.** *Aspose - Your file format experts*. [Online] Aspose. [Cited: 29 April 2012.] <http://www.aspose.com/>.
97. **Intellect, Advanced.** *Asp.NET Email*. [Online] Advanced Intellect. [Cited: 29 April 2012.] <http://www.aspnetemail.com/>.
98. **Google.** *Google Maps API*. [Online] Google Inc. [Cited: 29 April 2012.] <https://developers.google.com/maps/>.
99. **Microsoft.** *Visual Studio Metrics Plugin*. [Online] Microsoft. [Cited: 29 April 2012.] <http://msdn.microsoft.com/en-us/library/bb385914.aspx>.
100. **Putman, J.** *Architecting with RM-ODP*. s.l. : Prentice Hall, 2001. ISBN: 9780130191168 LCCN: 00058866.
101. **Plazaola, Jos{\\'e}; Leonel.** *Strategic Business and IT Alignment Assessment : A Modeling Approach Associated with Enterprise Architecture*. KTH, Industrial Information and Control Systems, KTH, Industrial Information and Control Systems. s.l. : KTH, 2009. pp. iv, 41, Ph.D. dissertation. QC 20100805.
102. **Lindstr{\\\"o}m, {\\AA}sa.** *Using Architectural Principles to make the IT-Strategy come true : focusing on the electric power ndustry*. KTH, Industrial Information and Control Systems, KTH, Industrial Information and Control Systems. s.l. : KTH, 2006. p. 41, Ph.D. dissertation. QC 20100908.
103. **Jaring, Michel and Bosch, Jan.** *Representing Variability in Software Product Lines: A Case Study*. [ed.] Gary Chastek. *Software Product Lines*. s.l. : Springer Berlin / Heidelberg, 2002, Vol. 2379, pp. 219-245. 10.1007/3-540-45652-X\_2.
104. *Quantitative analysis of enterprise architectures*. **Iacob, M.E. and Jonkers, H.** s.l. : Springer, 2006, Interoperability of enterprise software and applications, pp. 239-252.
105. *Modeling the IT Impact on Organizational Structure : 2009 IEEE INTERNATIONAL ENTERPRISE DISTRIBUTED OBJECT COMPUTING CONFERENCE*. **Gustafsson, Pia, et al., et al.** s.l. : IEEE COMPUTER SOC, 2009. pp. 14-23. QC 20110228.



106. *Enterprise architecture : A framework supporting organizational performance analysis*. **Gustafsson, Pia, Johnson, Pontus and Nordström, Lars**. 2009. QC 20111202.
107. *Analyzing IT Impact on Organizational Structure : A Case Study*. **Gustafsson, Pia, et al., et al.** s.l. : IEEE, 2009. pp. 3113-3126. QC 20110222.
108. *A method for estimating maintenance cost in a software project: a case study*. **Granja-Alvarez, J.C. and Barranco-García, M.J.** 3, s.l. : Chichester, Sussex, England: Wiley, c1989-c2000., 1997, Journal of Software Maintenance, Vol. 9, pp. 161-176.
109. **Gammelgård, Magnus**. *Business Value Assessment of IT Investments : An Evaluation Method Applied to the Electric Power*. KTH, Industrial Information and Control Systems, KTH, Industrial Information and Control Systems. s.l. : KTH, 2007. Ph.D. dissertation. QC 20100608.
110. *The Zachman Framework and the OMG's Model Driven Architecture*. **Frankel, D.S., et al., et al.** 2003, Business Process Trends.
111. *Trends in Enterprise Architecture Practice : A Survey*. **Franke, Ulrik, et al., et al.** 2010. Vol. 70, pp. 16-29. QC 20110316.
112. *EAF(2) - A Framework for Categorizing Enterprise Architecture Frameworks*. **Franke, Ulrik, et al., et al.** s.l. : IEEE COMPUTER SOC, 2009. pp. 327-332. QC 20110303.
113. *Decision Support oriented Enterprise Architecture Metamodel Management using Classification Trees*. **Franke, Ulrik, et al., et al.** s.l. : IEEE, 2009. pp. 328-335. QC 20110301.
114. *An Enterprise Architecture Framework for Application Consolidation in the Swedish Armed Forces*. **Franke, Ulrik and Johnson, Pontus**. s.l. : IEEE, 2009. pp. 264-273. QC 20110301.
115. *Enterprise Architecture Models for Cyber Security Analysis*. **Ekstedt, Mathias and Sommestad, Teodor**. s.l. : IEEE, 2009. pp. 832-837. QC 20110301.
116. *A Tool for Enterprise Architecture Analysis of Maintainability*. **Ekstedt, M., et al., et al.** 2009. IEEE. pp. 327-328.
117. *Writing case studies in information systems research*. **Der, H. Van.** 1, s.l. : Taylor & Francis, 2003, Journal of Information Technology, Vol. 18, pp. 45-52.
118. *Successfully completing case study research: combining rigour, relevance and pragmatism*. **Darke, P., Shanks, G. and Broadbent, M.** 4, s.l. : Wiley Online Library, 1998, Information systems journal, Vol. 8, pp. 273-289.



## 15 APPENDIX - SYSTEMS

As presented in the ArchiMate segment every layer has an internal and external view, which is used to represent the overview and the individual system models. The System overview is created using the external view of the application layer of ArchiMate, i.e. no internal parts of the systems are presented. In the individual systems view both the internal and the external views are represented. In the component view however only the internal view is present.

**User Groups** – A thorough explanation of the UG system is not provided in the project as it is not part of it; however a basic explanation is included as there are different couplings connected from the systems. UG, short for User & Groups and is the authorization system used by almost all KTH systems. This system is generally in the models considered comprised of two components, the login server and the UG database.

Using a Central Authentication Service(CAS), CAS login through which all login is handled. This service is connected to the central User and Group(UG) system. UG stores and handles all authorization settings available in KTH. This is also based on information which is available in LADOK. In the Mina Sidor system model, there are some BehörighetsKontrollSystem(BKS) files, i.e. authorization validation files together with LpWAuth, the different roles of people who are defined in LADOK are translated into the different roles available in UG, e.g. student, researcher or employee etc.

Using the student or employee portal the system users can also access the UG system to change different settings to their own profile. Through the student portal one can only change settings to one's own account, such as contact information and email settings. The employee section allows access to change different settings of other employees or a student's account, such as access to different systems.

## 15.1 OVERVIEW

The overview provides an overview of how the different systems are connected with each other, the processes and the different databases they use. The dotted lines are the lines derived from communications between different system components, as presented in the meta model section.

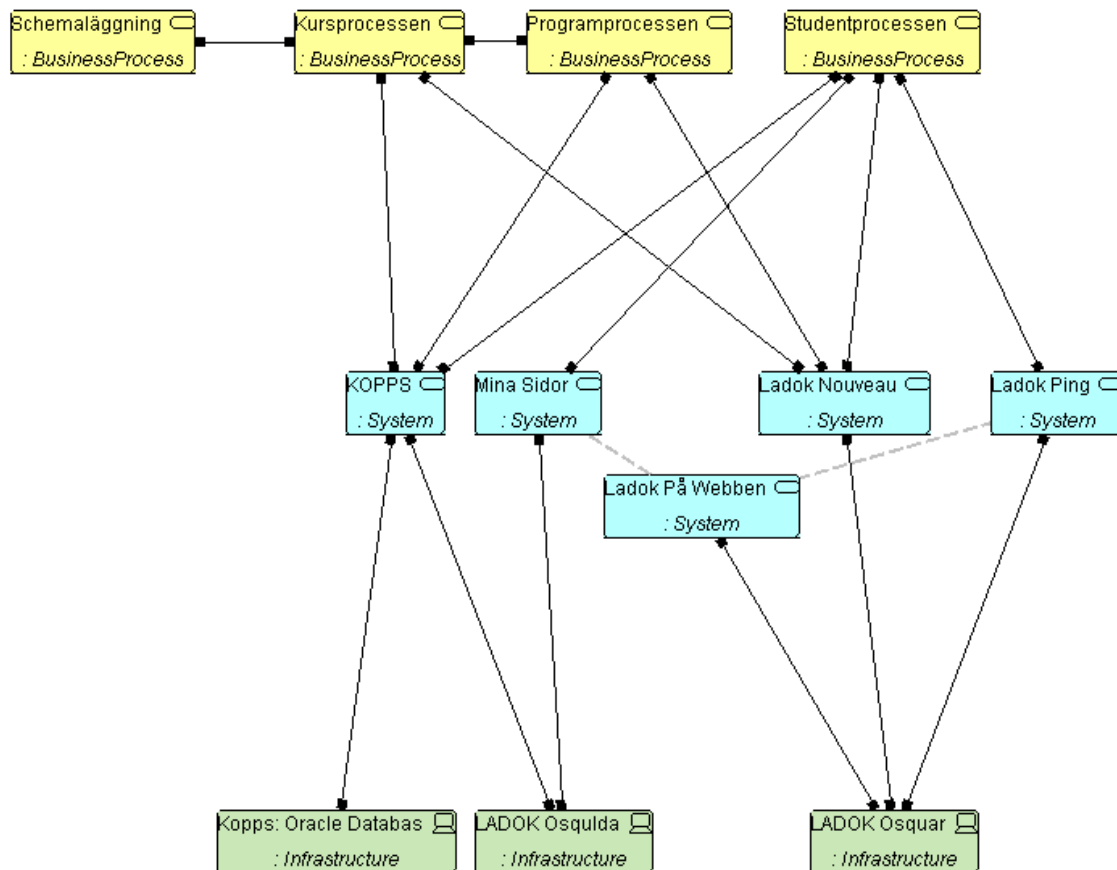


Figure 32: The connection of some of the systems which were studied and which processes and databases they connect to. The dotted line represents a connection between systems.

As can be seen in Figure 32, Mina Sidor does not have a direct connection to LADOK Osquar database, but instead has a derived connection to Ladok På Webben which is used by Mina Sidor to access LADOK. This is further explained in the Mina Sidor segment.

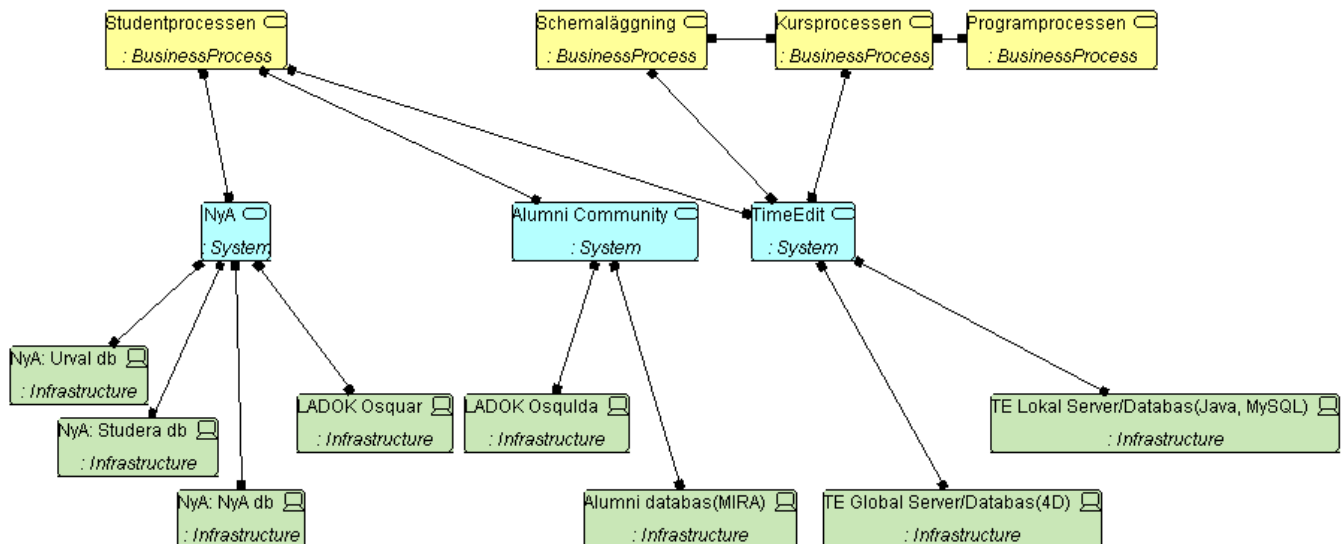


Figure 33: The rest of the systems and their connections.

The Figure 33 shows here as well that the connections to LADOK are quite many.

## 15.2 LADOK

Explained in this sub segment is the reason for how the LADOK system was modelled. Different scenarios were considered, but in the end one method was chosen.

LADOK is a national system solution developed at Umeå University and requested by Ladokkonsortiet and the 39 different universities that exist in Sweden. Every university has its own LADOK system installations complete with its own database. Using the same system and database implementation in all universities in Sweden achieves a standardization of data between universities. The data of each database is of course different, but the database tables and columns are all the same.

In this degree project the LADOK system is represented by three different applications and are considered and studied as individual systems. The three systems realize and represent what is considered to be the system “LADOK” and the reason for this is explained in the discussion section. The three different LADOK systems are:

- Ladok Ping (LadokPing System) realizes a connection between the different LADOK systems of the universities in Sweden. The system will be explained in its own section.
- Ladok på Webben (LpW) is a portal which realizes mostly read but also some write access to the LADOK database through a set of fixed SOAP services. This is represented as a component of the Mina Sidor system and will be explained in that segment.
- Ladok Nouveau (Nouveau) is the administrative GUI client for which is used for all administrative tasks to LADOK. This system is explained in its own segment.

The LADOK database is available in two instances. The LADOK Osquar, which is the actual database which allows read and write to the data. Write access to this database is restricted only to the Nouveau system and to a smaller extent the LpW portal. The second database which is a replica of Osquar is LADOK Osquilda, also called LADOK Open, which is used as a read only instance of LADOK. This is to get information which

might not be available through services in LpW. The LADOK database for KTH is hosted by ITS at Umeå University.

### 15.2.1 NOUVEAU

Nouveau is the only system used for administering the LADOK database. The system is developed in Uniface (91), maintenance and development is handled by ITS at Umeå university. At KTH it is a critical part of the institution as which can be seen as most of the core processes use the system (Figure 34):

- The student process – The student information and their accomplishments are kept in the LADOK database.
- The course process – The many different courses available at KTH are all kept in LADOK.
- The program process – The many different programs available at KTH are also kept in LADOK.

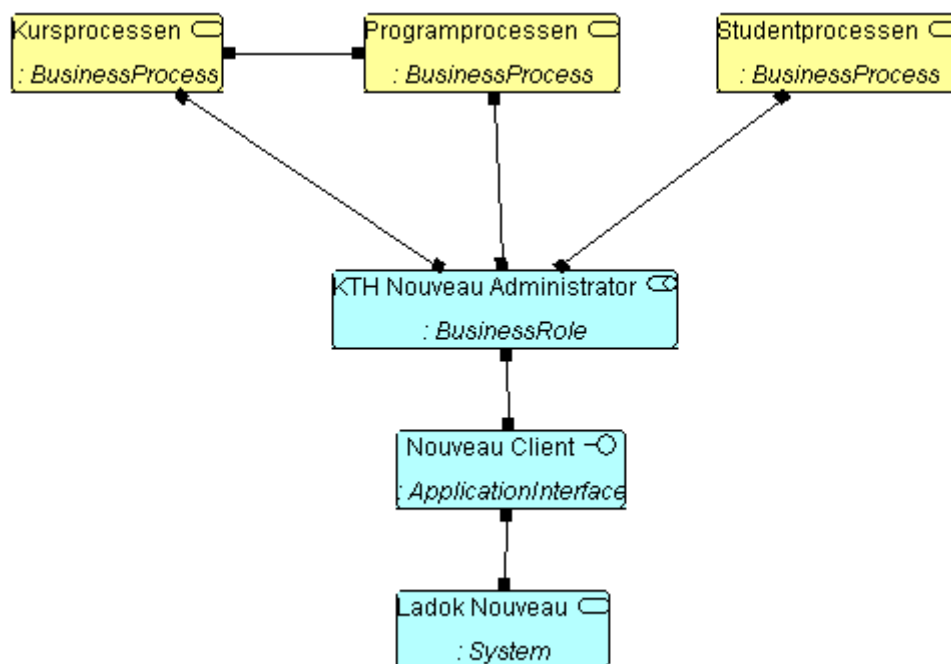


Figure 34: The processes which use Nouveau.

The system is a client – server setup, and the only interface to the system is the Nouveau client. The system is realized by in the model by five different components (Figure 35), two of which are part of the Uniface setup, and one represents the JavaBatches:

- urouter – “Uniface Router, The Uniface Router is a multithreaded process that performs a number of tasks for the deployed application. It sits between the Uniface client applications and the Uniface Servers that execute the client requests.” - According to a web presentation of Uniface v8.4.
- Javabatch – These are java batches which numbers over 20. Every batch is invoked from the client to perform different sets of functions on the database.
- Client - This is the large client of Nouveau, the GUI which contains all the forms and services used for maintaining the LADOK database. This client contains all the business logic as well.
- userver – The Uniface Server which works as the application server for a uniface based system, it receives the client requests from the urouter sent from the client.

- Flexlm – Present in the documentation was a component called flex-lm, the existence, its purpose and its use was very uncertain and no further explanation was found regarding it in the LADOK documentation, thus this was added with high uncertainty to its existence. The FLEXlm User end manual however explains Flexlm as a license management tool: “FLEXlm provides utilities for the license administrator to help manage the licensing activities on the network.”

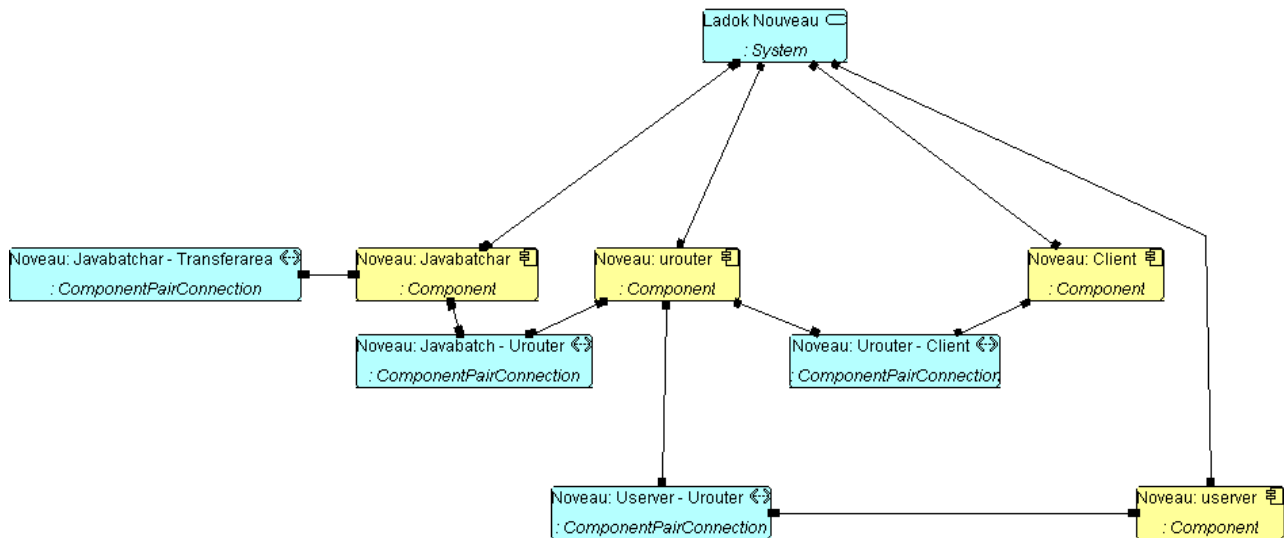


Figure 35: The components of the Nouveau system together with the connections (flex-lm not modelled due to its existence uncertainty).

According to the documentation, the userver and urouter run on Uniface 4GL application server (Figure 36). The client is installed at each user end in order to access the LADOK database, and requires a third party C++ runtime library to work.

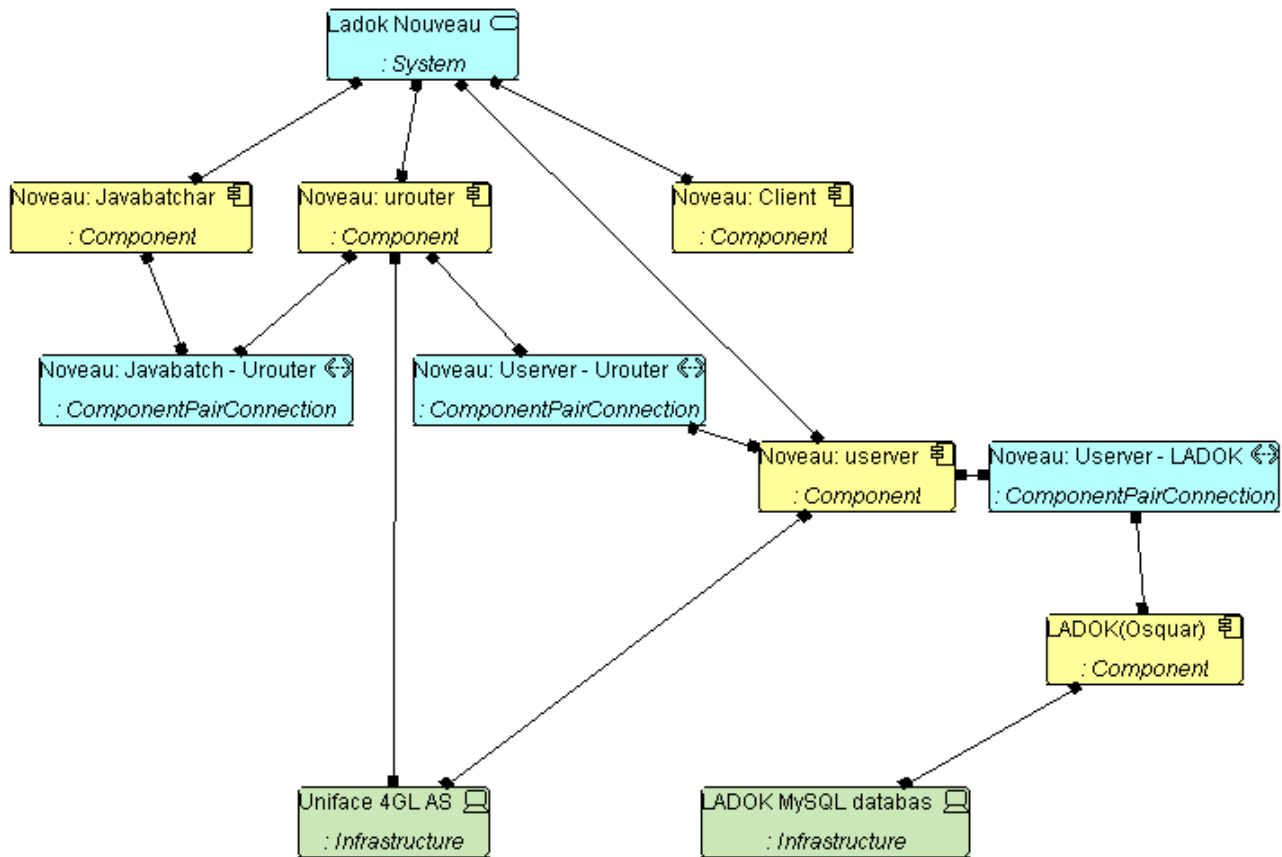


Figure 36: The technology which the Nouveau system runs on.

Information regarding this system was scarce and out of date. No metrics were obtained in order to perform the modifiability analysis, and were thus estimated:

- The existence of the flexlm component was quite uncertain as the LADOK documentation does draw the component but does not further mention its use, and was thus estimated to (0.25).
- The coupling estimations were quite uncertain and thus given a coefficient of variance of 75% in order to express this probability.

Component	Metric	Value
Client	CouplingAVG	3,50
	CouplingMAX	3,54
urouter	CouplingAVG	2,97
	CouplingMAX	3,81
userver	CouplingAVG	3,16
	CouplingMAX	4,40

The client, urouter and userver are linked to each other as a client – server, which was described above, which is here reflected in the coupling values. However urouter is also connected to the JavaBatches which lowers its average coupling value a bit as this is merely batch invocation.

Component	Metric	Value
JavaBatchchar	CouplingAVG	2,10
	CouplingMAX	3,35



The concluded modifiability for the entire Nouveau system was:

System	Metric	Value
Nouveau	CouplingAVG	3,16
	CouplingMAX	4,40
	Cyclomatic Complexity	2,75

The modifiability was determined to be average with hints towards greater modifiability, this mainly due to the estimated couplings value which were determined to be “average” i.e. a mean value of: 3,16. Further the components of the systems are rather few with quite few connections, both internally and externally, which thus gives the system a rather low cyclomatic complexity in the calculation, which suggests better modifiability.

This can also be supported by the lack of connections to other systems, i.e. the only system which it has a connection to is NyA through the Java batches. A system which is similar in that aspect is for instance KOPPS which has an average to high modifiability, and also only has a connection to NyA which slightly strengthens the similarity.

### 15.2.2 MINA SIDOR

The only process that uses the Mina Sidor system is the student process, as it is the portal from which students access their personal information regarding KTH. The program and course processes do not use this system in their activities, and neither does the scheduling process. However as the student process is the core process and the Mina Sidor system is the students main access point to KTH and the information available here, the TimeEdit system is presented through its API in Mina Sidor, and links are available for accessing for instance course information in KOPPS.

Mina Sidor is accessed through a web interface by both employees and different kinds of students, as can be seen in Figure 37. This is the portal used by student, research students and employees at KTH for accessing restricted information in LADOK. The Mina Sidor system consists of four different components (Figure 38).

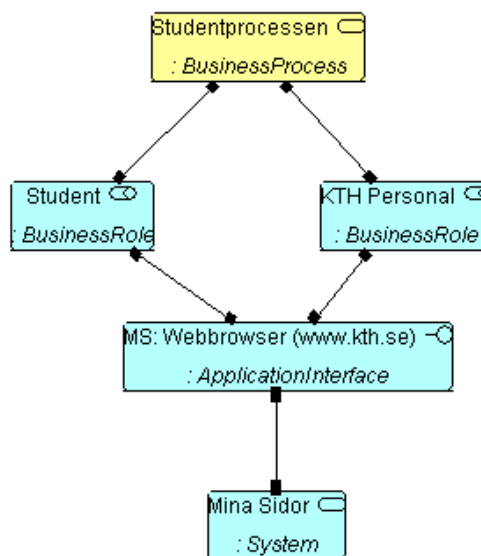


Figure 37: The process which use the Mina Sidor system.

- Mina Sidor Students - The first is the student portal “Mina Sidor för studenter”. This part of the system is used by student to access information and options that are specific for the students, e.g. information about what courses they are taking the current semester, or an option such as registration for an exam. The part for research students “Mina Sidor för forskarstuderande” is an extension to the student part of the system, and simply modified by authorization to separate the two.
- Studentdokumentation API - The Studentdokumentation API is also a part of the Mina Sidor system. It is a restful API for accessing information such as courses, programmes and staff using JSON.
- Mina Sidor Anställda - Third is the employee section “Mina Sidor för anställda”, this is used by people employed at KTH, e.g. teachers, assistants and professors. This part of the system is used to manage students, exams and grades etc.
- LpWAuth – The LpWAuth component is a simple java code plugin which is directly loaded into the LpW system, the component is not part of the original LpW delivery and thus considered a part of Mina Sidor instead for LpW. The component is used for helping to converting different user roles which are defined in LADOK to the ones in the UG system used by KTH.

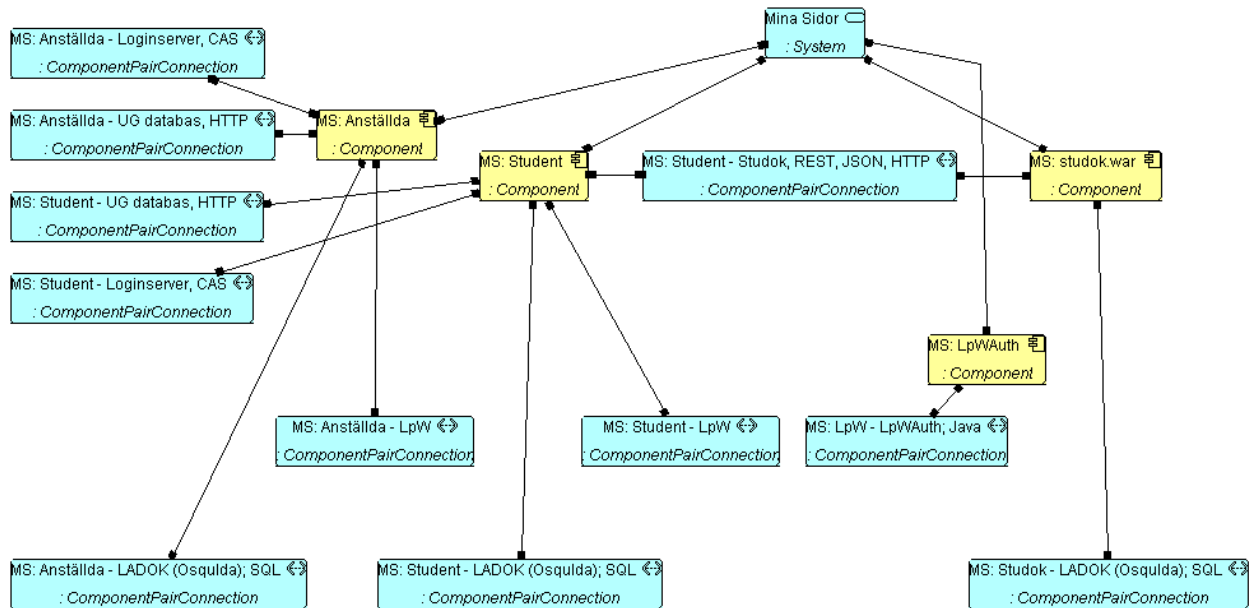


Figure 38: The different components that realizes the Mina system, and the external connections to UG.

The measured metrics regarding each of these components was delivered from the developers. Mina Sidor Student is comprised of a single archive called *student.war*. Since the system is developed and maintained by ITA at KTH these metrics were also provided by the same measurement tool as KOPPS. The archive file size for the student portal was measured to be 6895 lines of code, of which 5915 were actual source lines of code. The cyclomatic complexity for these components was also provided, and was measured to be an average of 1.44. Following are the coupling types that were provided and the calculation for this component in EAAT:

Component	Metric	Value
MS: Student	CouplingAVG	3.88
	CouplingMAX	4.67

The employee part of the system, Mina Sidor Anställda, is comprised of an archive file called *intern.war*. This archive has a measured size of 5046 lines of code, and the actual source lines of code are 3997. The cyclomatic complexity average measurement of the source code for the *intern.war* was measured by the developers to be 1.50. The coupling values provided were calculated by EAAT to be:

Component	Metric	Value
MS: Anställda	CouplingAVG	3.93
	CouplingMAX	4.67

The student documentation API is realised by an archive called *studok.war*. A rather small archive only measured to be 1220 lines of code, of which 1094 are actual source lines of code. This small component also has a small cyclomatic complexity average measurement of 1.26 and its coupling measurements from EAAT were:

Component	Metric	Value
MS: Studok	CouplingAVG	3.67
	CouplingMAX	3.67

The smallest component in the system was LpWAuth. This component *lpwauth.war* only has a size of 436 lines of code, from which 315 are actual source lines of code. The cyclomatic complexity average was measured to be 2.92. And last but not least the coupling calculations from EAAT were:

Component	Metric	Value
MS: LpWAuth	CouplingAVG	3.67
	CouplingMAX	3.67

The assumption made for Mina Sidor is regarding the number of couplings between LpW and LADOK, and also between Anställda, Student and their connections to LpW. The connection numbers were found in the LpW documentation; however they were not that exact as to how many, only the number of exposed methods used by the exposed services. Thus for the number of services a coefficient of variation at 10% was used, as there are only that many services, but the documentation for them were a bit outdated and vague as to which component used which service. The LpW to LADOK coupling were estimated from the documentation through exposed methods, however these could be more or fewer than found in the documentations, also information regarding the type of coupling was not provided and therefore estimated, and thus a coefficient of variation at 25% was used for these values.

The calculated values in EAAT for the Mina Sidor systems were (The size value is lines of code without comments):

System	Metric	Value
Mina Sidor	Size	11323
	Cyclomatic Complexity	12,0
	CouplingAVG	3,79
	CouplingMAX	4,67
	Cyclomatic Density	1,06

Mina Sidor – The modifiability of Mina Sidor was determined to be average. The cyclomatic complexity value of the system is right on the limit of what is considered acceptable. However since the system is of small size and the complexity is rather large, this gives a high cyclomatic complexity density, i.e. a lot of logic considering the amount of code which is outside of what is recommended values. The system is rather loosely coupled which suggests that any modifications made to the system should with low probability affect the connected systems. The Mina Sidor system was developed by KTH.

As mentioned earlier these are SOAP services which are invoked by the “Mina Sidor Student” and “Mina Sidor Anställda” parts of the system in order to read and write information from and to the database. LpW (Figure 39) uses the Spring framework ([www.springframework.org](http://www.springframework.org)) to map the LADOK database to its entity layer. In order to access information which is not available as different SOAP services, the three different components Student, Anställda and Studok API have direct read access to the read version of LADOK at KTH. LpW and LpWAuth run on a JBoss applications server, whereas, the Student, Anställda and Studok API all runs on a Tomcat Servlet.

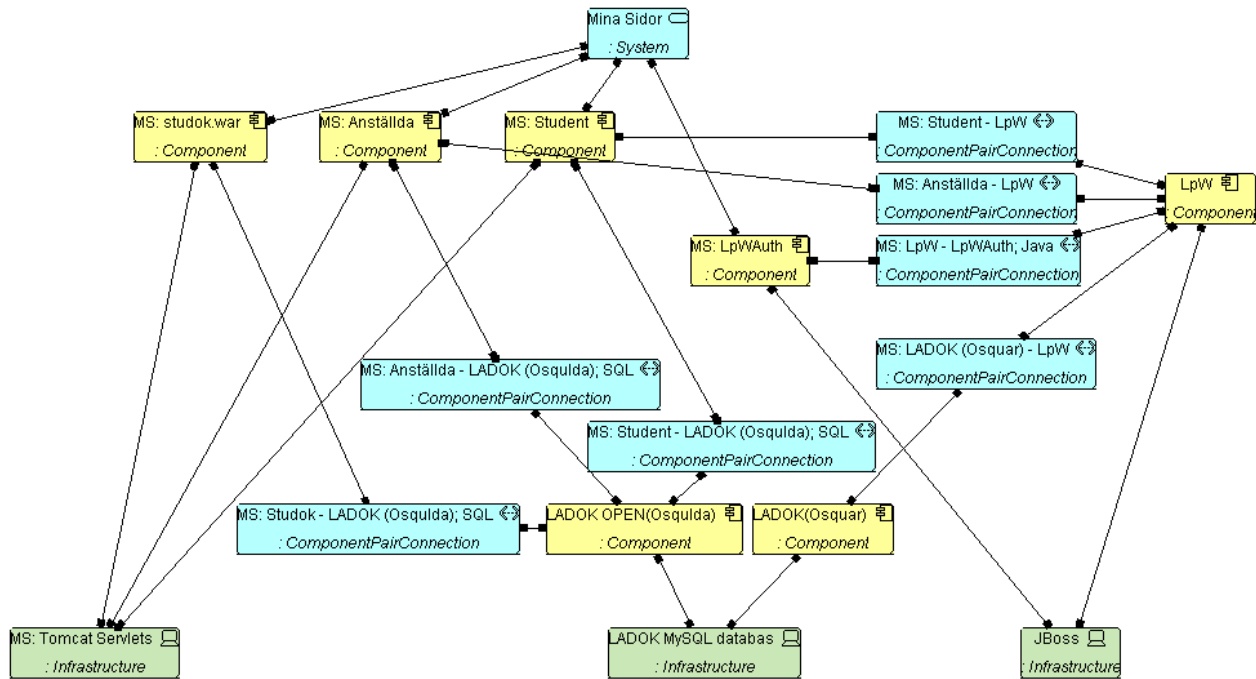


Figure 39: The technology used by the Mina Sidor system. The picture also shows the connection to the LpW system through its component.

- Ladok på Webben (LpW) - In order to facilitate access to the LADOK database the Mina Sidor system uses the SOAP services which are exposed by the LpW system of LADOK.

System	Metric	Value
Ladok på Webben	Size	178758
	Cyclomatic Complexity	6,0
	CouplingAVG	4,10
	CouplingMAX	4,99
	Cyclomatic Density	0,03

Mina Sidor is however highly dependent on the LpW system, even though it is not tightly coupled to it, many of the functions available in the Mina Sidor system uses the services exposed by LpW. The LpW system however has rather low modifiability. The density value suggests that the system has too little logic compared to the size of the system. The coupling value of the system is very high which means that any changes to this system would most likely require changes to other systems. The size of the system is rather large compared to other system, but still not the largest of the systems investigated, however this further suggests low modifiability. The average complexity of the system is rather low, which suggests easy internal modifiability. However, considering all of the different values the general modifiability would be considered rather low, probably easier to modify internally but the effects of the change would most likely affect other systems.

### 15.2.3 LADOKPING

The LadokPing system is generally used mostly as an extension to LpW, however there is a GUI which is used by guidance councillors in order to extract course and program information regarding an individual student. The only process that uses this system is the student process as well Figure 40, directly by councillors and indirectly by students through Mina Sidor.

The LadokPing system is a smaller system which is also a part of LADOK and is mainly used together with the LpW system. The reason that LadokPing stands out from all the other parts of the LpW is because it is used for communication between the different LADOK installations of different universities whereas the other parts are used only with its own database. The LadokPing system consists of a single package file, the ping.war which can be seen in Figure 40. This package consists of several classes and frameworks and the connection to the LADOK database.

The LadokPing system is used to create certificates for students through Mina Sidor, and during the time of this writing KTH was testing the degree application functionality in the system. The system also has a web GUI which is mainly used by guidance councillors at KTH for helping and checking on students.

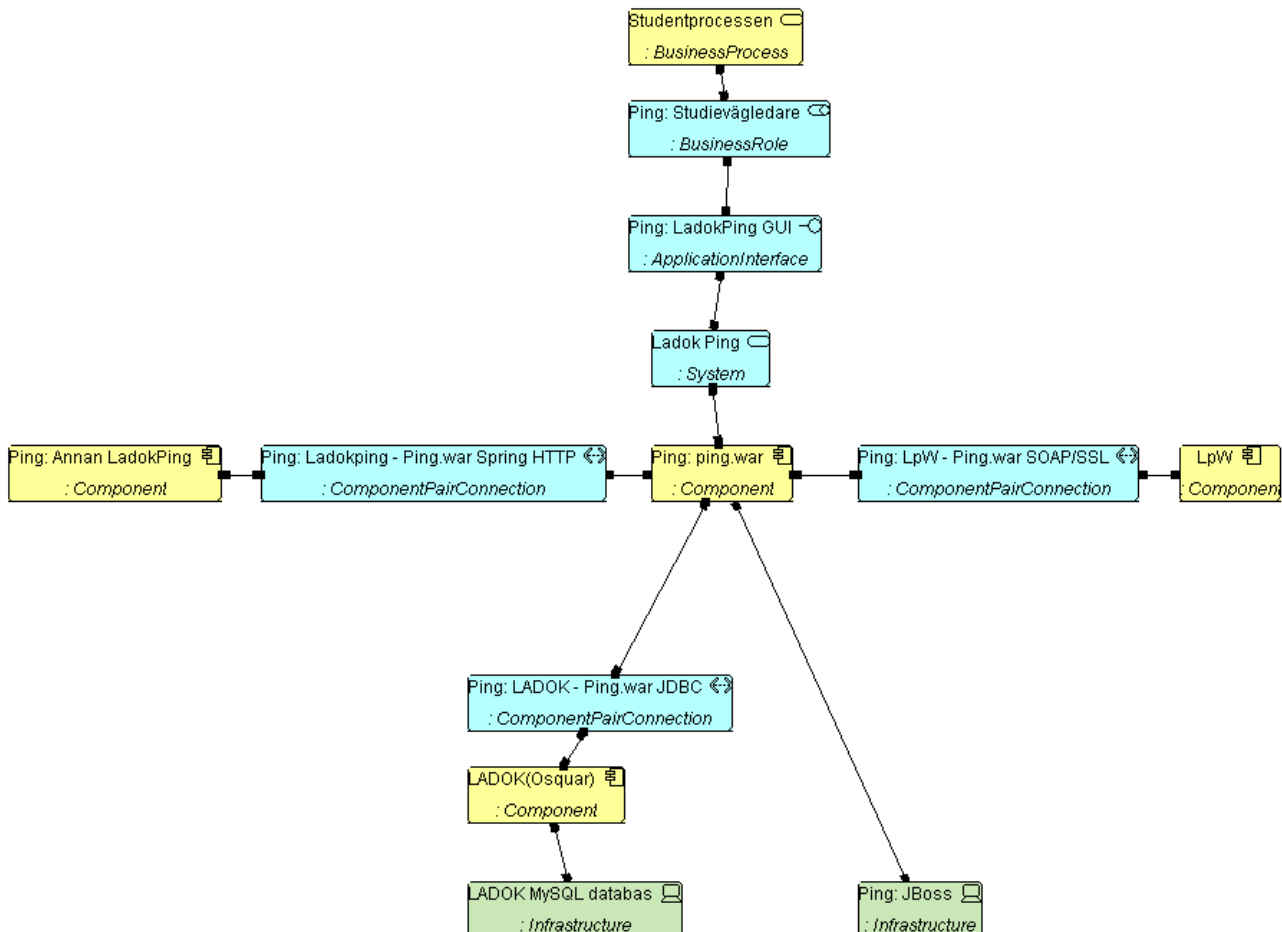


Figure 40: As the ping system is quite small in comparison with other systems, all the levels are available in one figure.

As can be seen in Figure 40 the ping archive is run upon a JBoss (92) applications server. Measurements for this system were done with Sonar and the software metrics obtained regarding the LadokPing component were:

- Lines of code: 45266, of 77684 lines total.
- The average Cyclomatic Complexity of the source code is 4.8 for the methods and 24.8 for the classes in the system.

While the method cyclomatic complexity is within the recommended limits, the average class complexity is way above the recommended limits, i.e. 10. This suggests that the component is very complex and can contribute to trouble when modifying the component.

As there was no specific information provided regarding the type of coupling between the components, these were estimated by using the documentation and the information found regarding the communication and components. But as these are still estimations, a level of uncertainty was thus applied to the values.

From the LADOK documentation there was mentioned four methods exposed through LpW services which had a connection to LadokPing. However the information regarding the methods was vague and a coefficient of variation at 25% was used to express this uncertainty. The same was done regarding the coupling to the other LadokPing systems at other institutions. As there are a 39 different institutions including KTH the estimated values were thus 38 other couplings to other LadokPing systems, and then applied a coefficient of variation at 25%.

As there is only one component realizing the system its values are the same in EAAT as the system, and using EAAT the following values were calculated regarding the system:

System	Metric	Value
LadokPing	Size	45266
	Cyclomatic Complexity	4,0
	CouplingAVG	3,79
	CouplingMAX	4,5
	Cyclomatic Density	0,09

LadokPing – The modifiability of LadokPing is rather high. The only measurement which suggests otherwise was the density, which is considered too low, i.e. the logic for the size of the module is considered rather low. The coupling average was rather low which however speaks for loose coupling and thus a lower chance of a ripple effect should modifications be performed. The size of the system was small, and the complexity well within the limit.

### 15.3 NYA

**Nya Antagningen(NyA)**, translated into “the new admission”, is the national admission system for higher learning and is thus used for all applications for higher education in Sweden. This system was developed by ITS at Umeå University to standardize the admission for every institution in Sweden. International students also use this system for applying to a Swedish education, however further investigations to eligibility is then performed at each individual institution.

The NyA system is used in only one of the core processes at KTH:

- Student process – As national admission to higher education is mostly handled through the NyA system in Sweden, this is no exception at KTH.

The people at KTH that access the system are students which are applying for education and admission experts, i.e. people who take care of admission. There are two different interfaces for accessing the NyA system, first is the web interface, which is where students search and apply for different programs or courses that they wish to take. The same web interface provides login for the experts in order to access the system (Figure 41). There is a second interface available only to the experts however, this is a Java Swing client called “Expertclient” which allows the experts to access functions which are otherwise not available. The client communicates with the NyA system via SOAP messages.

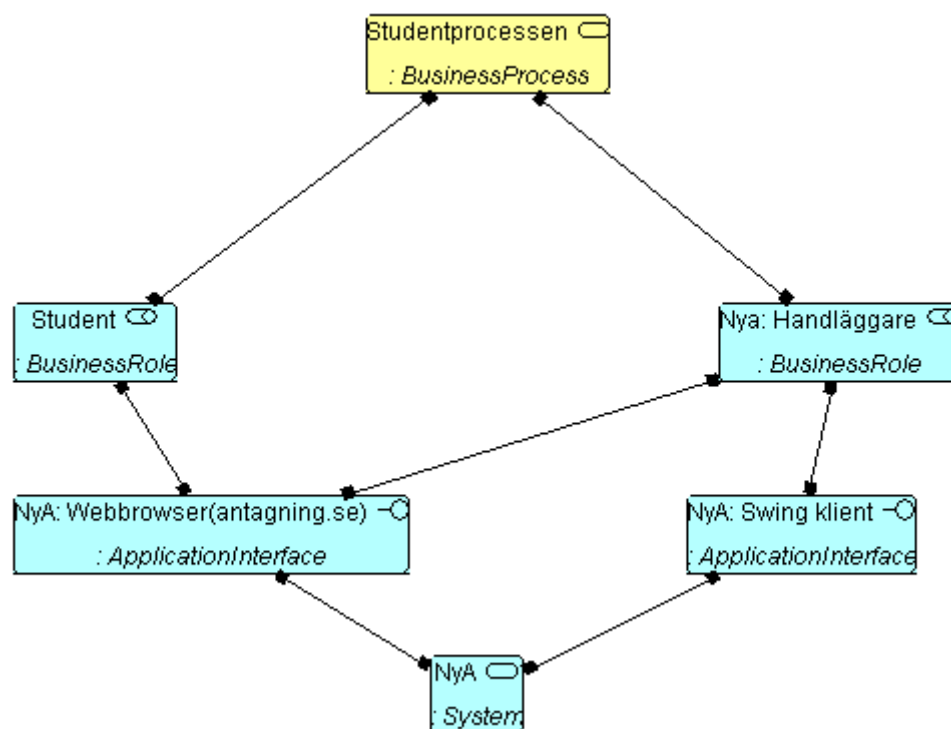


Figure 41: The process which use the NyA system and through which interfaces.

The components that realize the NyA system are quite many; the system is in general divided into two different parts, the applicant part and the expert part. Unlike the other system, due to its size the NyA system is here presented divided into four parts, the web layer (Figure 42), the logic layer which was quite large and thus had to be divided into several pictures (Figure 43, Figure 44 & Figure 45), the other communication (Figure 46) and the database layer (Figure 47):



- Web layer

- Http applicants – This component contains the different packets and modules for realising the student application website.
- Episerver – A third party component used by the applicant web, together with the WAS applicants cluster, for presenting the different webpages on the website.
- Http experts – This is the component which contains the packets and modules for the expert's parts of the website which is realised by NyA. This component also represents the Swing client which is used by the experts.

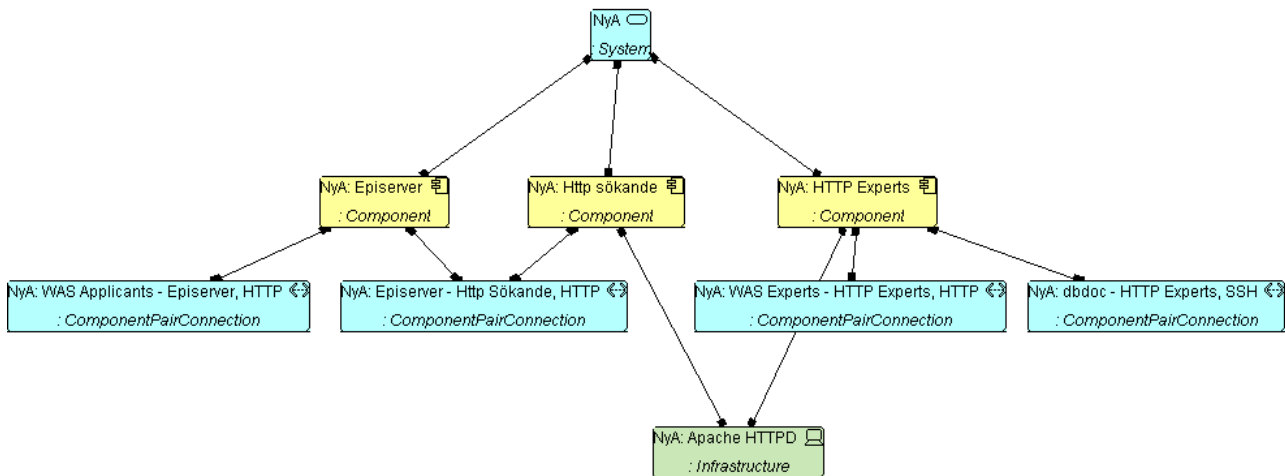


Figure 42: The web layer of the NyA system.

- Logic layer

- WAS cluster applicants – The WebSphere Application Server cluster which is used for handling the system logic in the applicant part of the system.
- WAS cluster experts – The WebSphere Application Server cluster for the expert part of the system. This module contains the logic modules and functions used by the experts, both the web part and the swing client.
- LDAP applicants – This is the Lightweight Directory Access Protocol used in order to access the applicants parts of the system.
- LDAP experts – The same as LDAP applicants but for the experts part of the system.
- NDM (Network Deployment Manager) – This is a part of the WebSphere Application Server used for setting up, configuring and managing the WAS clusters.
- NyAQueueCache – This component is a dedicated WAS cluster that holds the cache file queues in NyA. This is according to the document only a small part of the system.
- Batch – The batch environment runs on the selection/batch machine, JMS services and access to the WAS cluster. Most of these batches are java-based and are initiated on the selection/batch machine which then continues to communicate with functionality which are located on the WAS clusters. Different files are read and written, the machine also communicates with other system, mostly through the transfer area.
- (NFS)batchroot – This is the actual batch machine. Some components use this machine as a Network File System for sharing files.
- Dbdoc – This is a sub product of the database documentation.

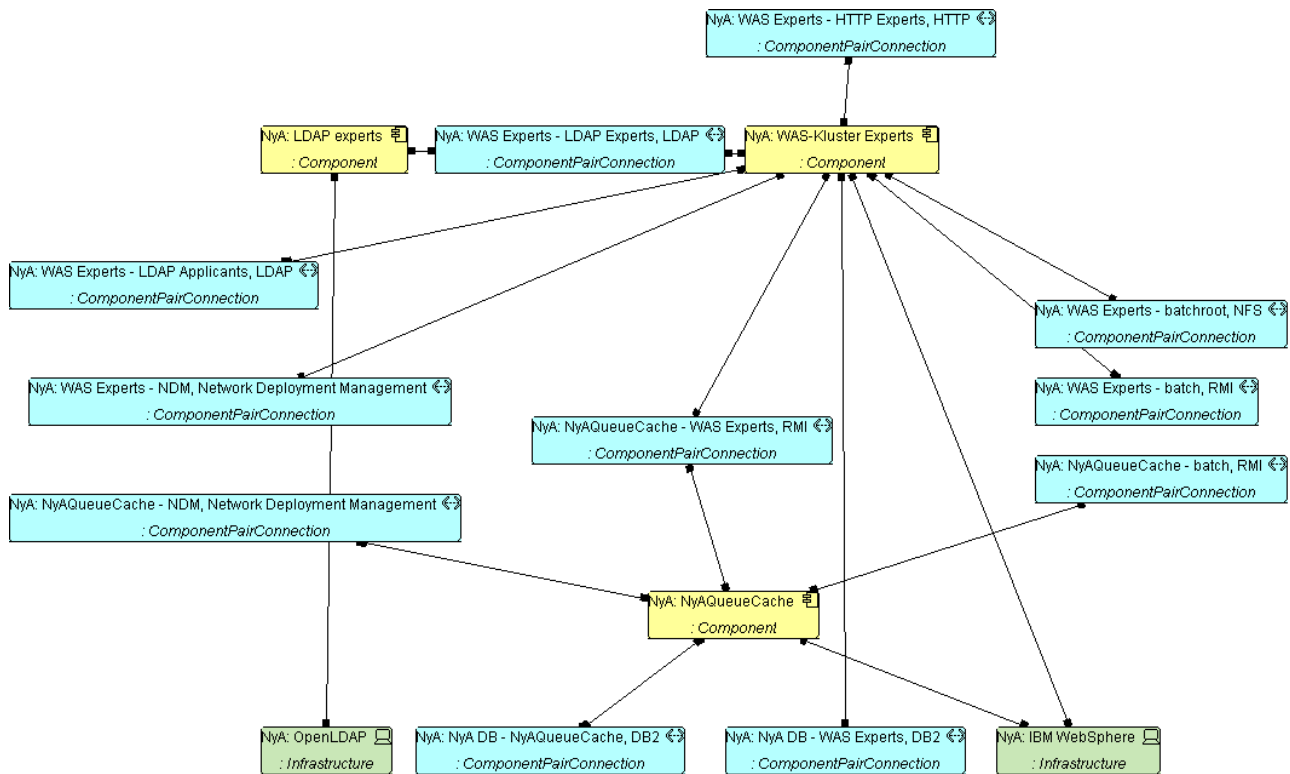


Figure 43: The first of the pictures which represent the logic components and their couplings in NyA.

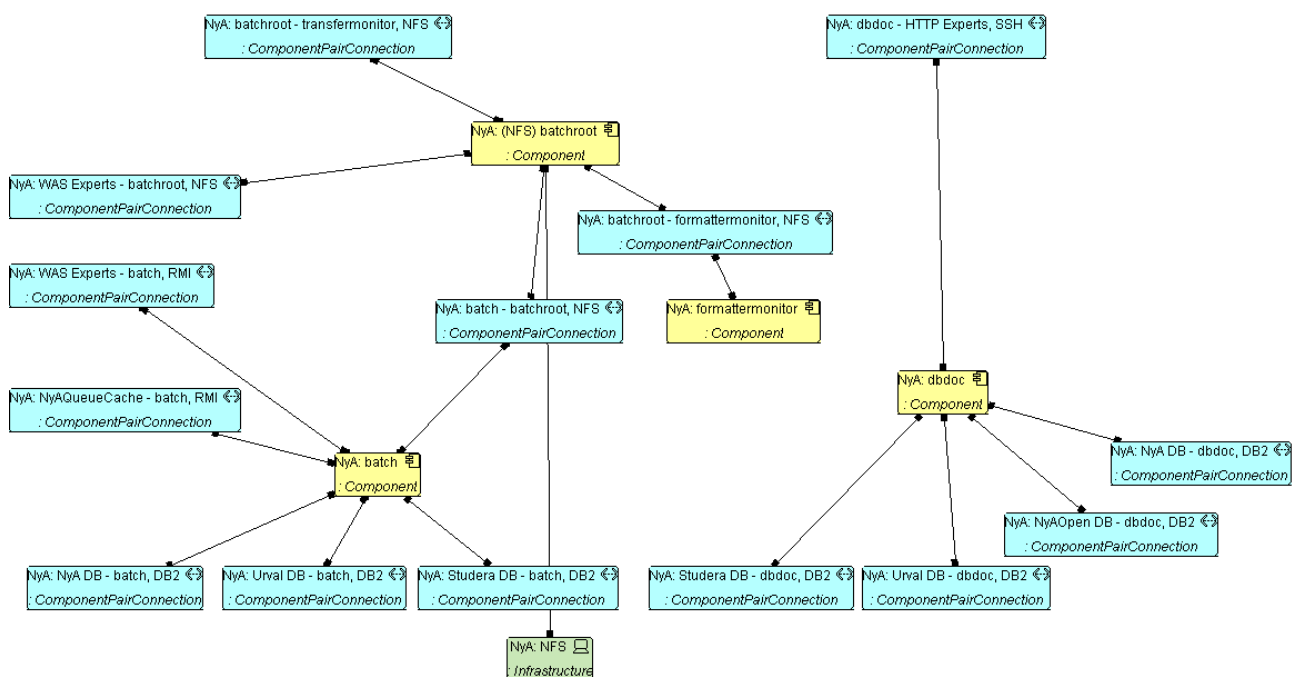


Figure 44: The second part of the logic layer and the couplings.

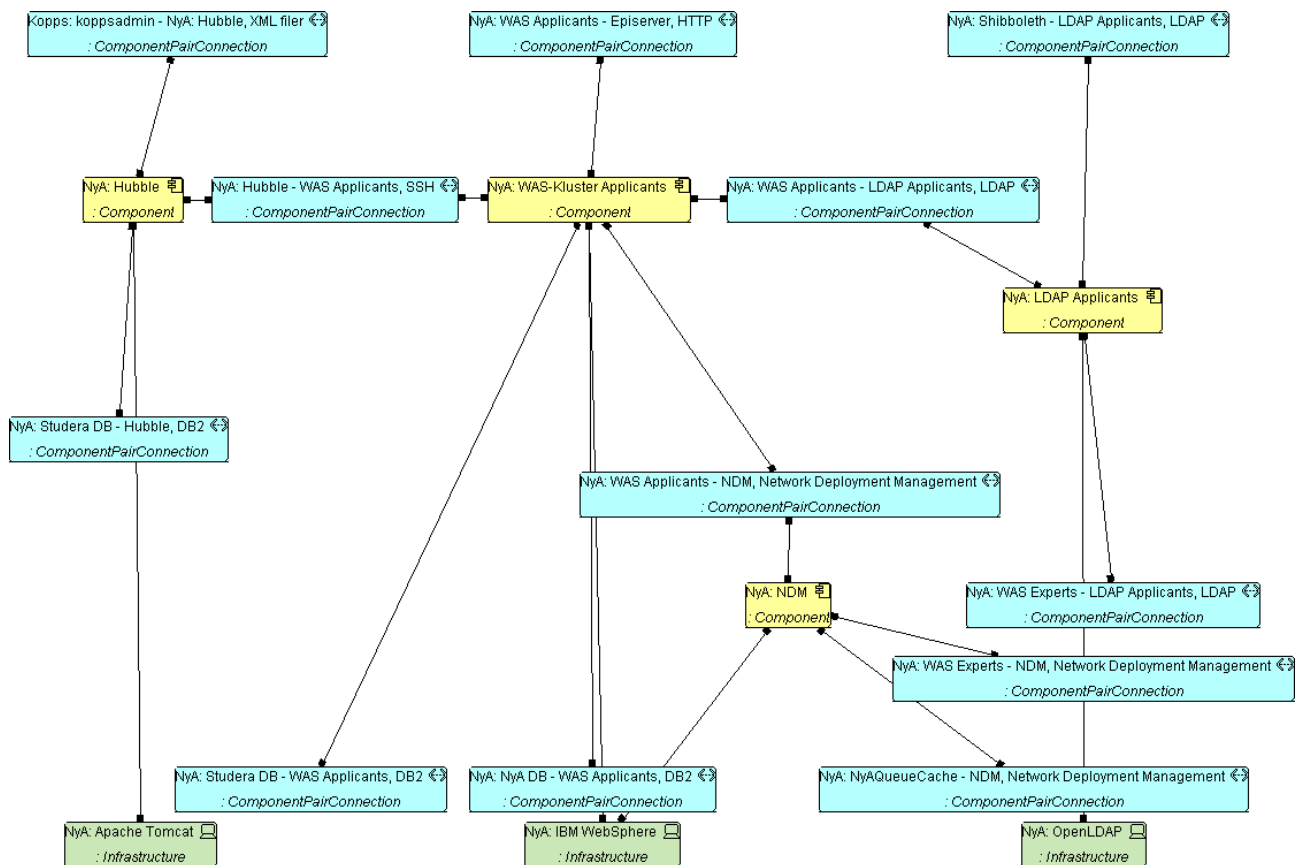


Figure 45: The third picture of the NyA logic components and their couplings.

- Other Communication

- Transfermonitor - The communications machine consists of a SFTP server and a transfer application. The transfer application handles the file transfers to other systems, mainly through sftp, ssh and ftp.
- Transferarea - This is part of the communications machine and works together with the transfer monitor. Through this transfer area different files are transferred to different systems, e.g. LADOK.
- Shibboleth – Shibboleth is used for federated login to NyA via the institutions own login system.
- dw-idp – Sub product of shibboleth.
- dw-ldap – Part of the LDAP for Shibboleth.
- Hubble – This component is used for by different institutions to upload their course and program information to NyA, in order for students to be able to view it on the public NyA website.

However NyA uses some external components as well in order to realize some other functions used in this large system.

- Formattermonitor – This component represents the Antenna House XSL formatter and a propriety java application which generates pdf documents from XML and XSTL.
- Archiveserver – There are two instances of ArchiveServer, one master and one cache, they are used for document management in the NyA system. The master and cache installations are on different machines at different locations. The master contains all the documents, and if someone tries to ac-

cess a document at the Cache and it cannot be reached, then it contacts the master in order to get the document. Cache also systematically synchronizes with the master in order to minimize the amount of requests.

- Archivecache – The cache part of the archive components. See Archiveserver.

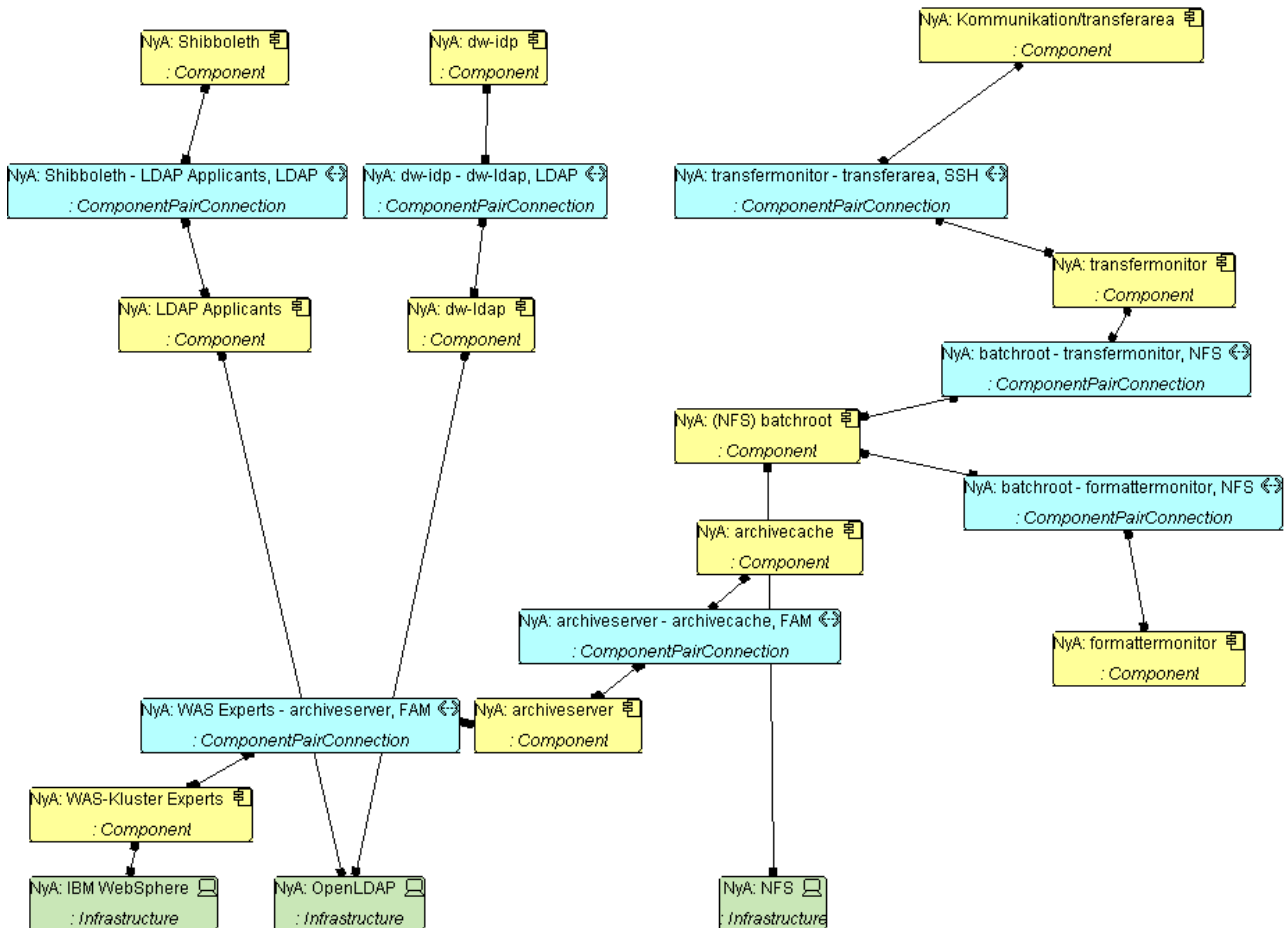


Figure 46: This picture shows the other communications that NyA has and also the different external components employed.

NyA uses several different databases for different purposes, there are three of them but there is also a read version as a replica of one of the databases:

- Urval Database – This is the selection database, it is generally only used when the selection for admissions to institutions are being done.
- Studera Database – This database contains all the different courses and programs which are available at the different universities in Sweden.
- NyA Database – The NyA database is the main database for the NyA system.
- NyAOpen – This is the replica and read version of the NyA database, available for different users with authorization to make read commands to the database.

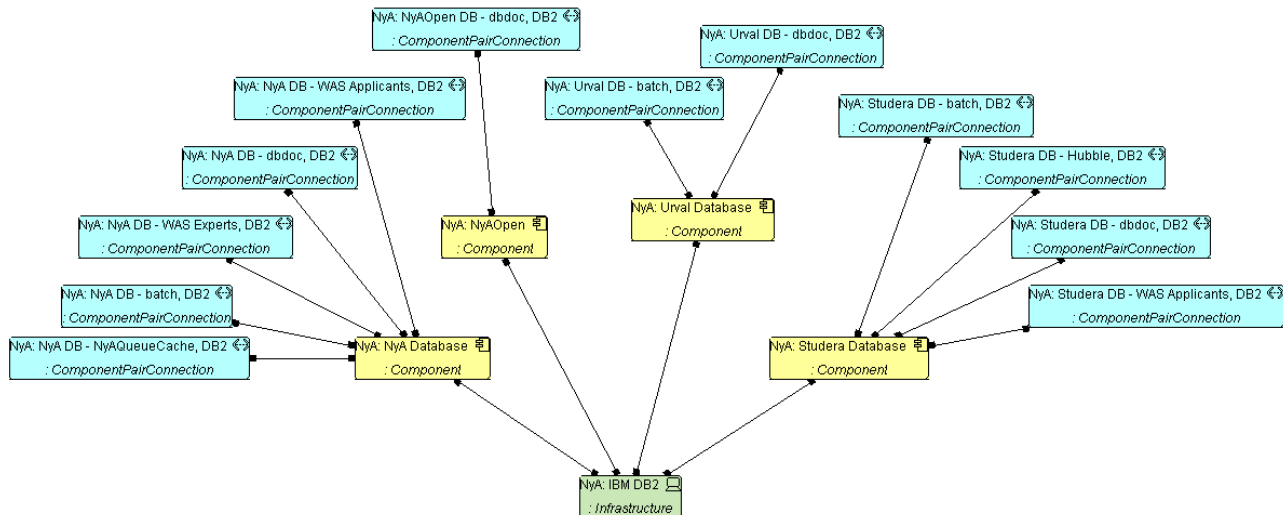


Figure 47: The four databases which are used in NyA, one of which is the read only replication NyA Open. All of them are IBM DB2 installations.

A large system NyA runs on quite many different kinds of infrastructure. These are the infrastructure types which realize most of the system:

- APACHE HTTPD – The web layer runs on Apache webservers. (93)
- APACHE TOMCAT – Some parts of the system, like Hubble runs on a Tomcat. (89)
- OPENLDAP – This is what runs the LDAP components. (94)
- IBM WS – IBM WebSphere, the commercially successful application server developed by the commercial international company IBM.
- IBM DB2 – All the NyA databases use the same database provider, DB2 from IBM.

Some components like batch & dbdoc are simple java applications and run on the JVM at the machine they are deployed at. Others are commercial products like Episerver which run their own environment. NyA transfers information to LADOK via transferarea.

The metrics acquired for the NyA system were rather abstract, i.e. the developers who sent them said that they were from the top level of statistics, and the picture provided can be found in APPENDIX – DATA. The application used for doing these calculations was Sonar.

- Although software metrics were given, these were not in-depth and rather given abstractly, i.e. a quick measure without any specific information regarding which parts but rather on the system as a whole. The NyA system source lines of code were thus estimated to be 819 341 source lines of code, from a total of 1 371 250 lines.
- The average Cyclomatic complexity of the source code was calculated to be 3,6 for the methods and 21,1 as an average for the classes.

The information regarding the couplings was estimated from the documentation that was obtained together with email responses regarding questions. However as these were still estimations, a level of uncertainty was applied:

- As the documentation was out of date, the information which was not commented on by VHS was therefore considered still being actively used today. Even so there was a level of uncertainty re-

garding this out of date information, thus a coefficient of variation at 25% was used for these values.

- As the NyA system is a very large system, the information regarding couplings had to be estimated from the documentation and how the components interact with each other. This however were still estimations from out of date documents and therefore applied together with a level of uncertainty, a coefficient of variation of either 10% or 25%.
- The information that was corrected or commented on was not given any level of uncertainty.

The results following the EAAT calculations were:

System	Metric	Value
NyA	Size	819341
	Cyclomatic Complexity	18,0
	CouplingAVG	3,53
	CouplingMAX	4,72
	Cyclomatic Density	0,02

NyA – This systems modifiability was concluded to be rather low. The density value was measured to be too low, suggesting too much code compared to logic. The coupling measurement for the NyA system was of an average value, this suggests that there is an average chance that any modifications in a module of NyA might affect other modules or systems. NyA is the largest of the systems investigated and thus compared to the other systems; the size suggests that this would be the hardest system to understand, which contributes to lower modifiability. The complexity of the system was measured to be very high, almost twice the recommended value, further contributing to low modifiability.

## 15.4 ALUMNI COMMUNITY

The Alumni Community is used for maintaining and handling Alumni; students who have ended their education, and that's considered part of the student process. This system is used to maintain contact and increase the social contacts of those members in the system. Although the system is mostly directed towards Alumni, people who are currently employed or students at KTH are also welcome. Alumni Community system provides a means to maintain communication and establish new contacts with students after ones studies are complete. The two roles that access the system are alumni and alumni administrators, both through a web interface Figure 48.

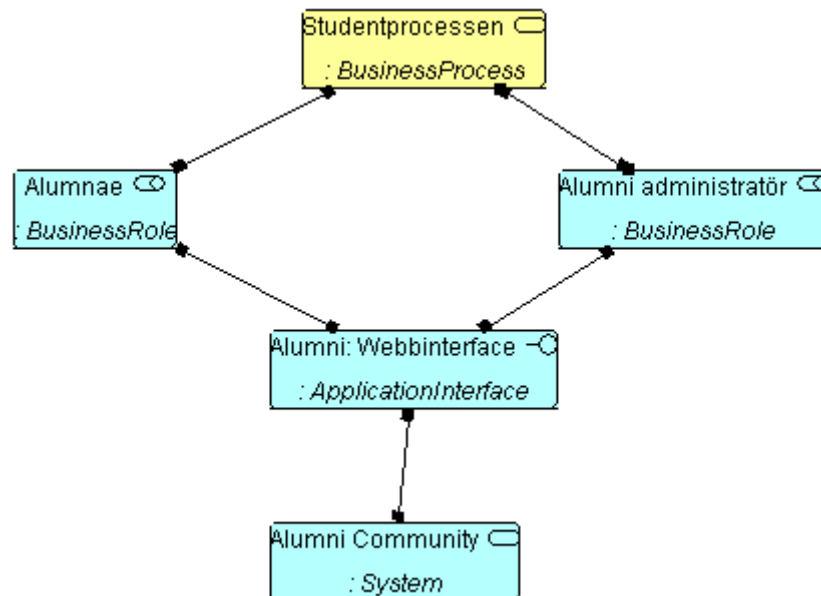


Figure 48: The processes, roles and interfaces for the Alumni Community system.

The Alumni Community system has a connection to LADOK for withdrawing information regarding the student or alumni. Verification to the student's authenticity is done via LADOK through the information available in LADOK.

MIRA networks provide alumni solutions for a number of higher education's institutions in the Nordic region. The solution that MIRA provide is a basic structure complete with a connection to LADOK; the Alumni Community system is then added as a unique extension to this structure in order to give KTH the system that they require.

The information collected suggests that the system consists of one primary component which is in the model called web components (Figure 49). This component is actually three smaller components called Ne5.dll, WebControls.dll and Core.dll.

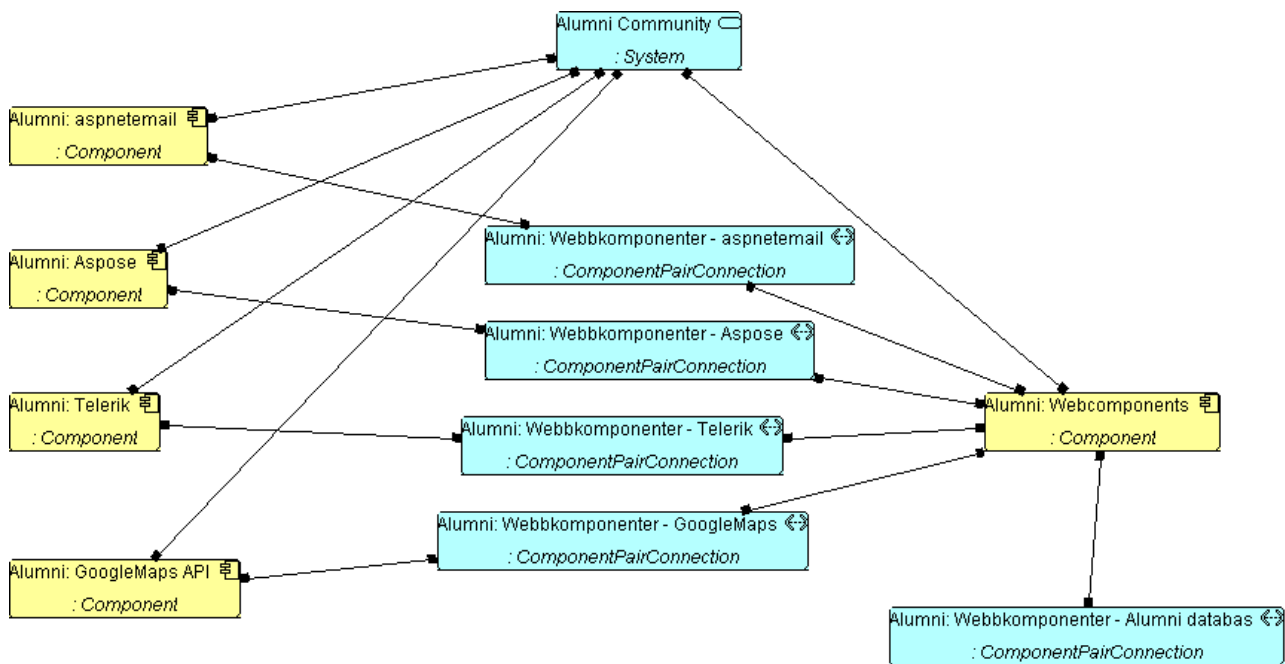


Figure 49: The components that realize the Alumni Community system.

- Webcomponents
  - Ne5.dll – This part handles the database access and security layer etc.
  - WebControls.dll – Web forms controllers.
  - Core.dll – This component contains the business logic and interfaces.

However in order to maintain the abstraction similar to the other models they are here drawn as one single web component. The web component together with the other components in the model realizes the Alumni Community system. The other components are external components not developed by MIRA themselves but rather implemented into the system:

- Telerik.dll (95) this is a web controls component and was developed by a company named telerik.
- Aspose.cells.dll & aspose.words.dll, (96) this is a components to generate different Microsoft Excel and Word documents. Modelled only as one component called Aspose.
- AspNetemail.dll (97) a component for sending email with asp.net applications.
- GoogleMaps API (98) used for coordinating of alumni locations and implementing maps.

This is all hosted on two Windows 2008 servers, one which hosts the IIS, i.e. the web and application server, and another which hosts the Microsoft SQL 2005 database (Figure 50).



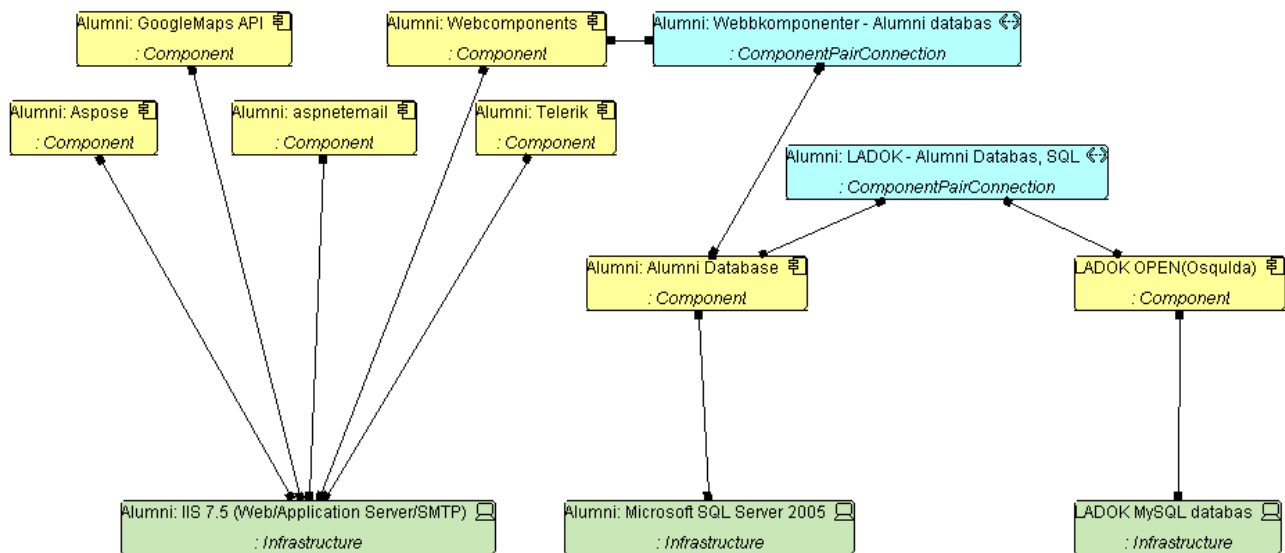


Figure 50: The technology which the Alumni Community system runs on.

Although MIRA did provide information about what external components that Alumni System uses, such as Telerik, information regarding how these were coupled was not provided.

- However as they are external plugin components, it is assumed that they have a control coupling and a data coupling; that the input provides some form of control over the component, and that the data that is returned is all used.
- GoogleMaps is available via an API which is then implemented into the webpage. This implementation is then sent information regarding coordinates to draw upon the map via the GoogleMaps program. This is therefore assumed to be a data coupling.
- However as these are all assumptions based on the fact that they are third part external components, a measurement of uncertainty was also added to the value, a coefficient of variation at 50%.

Using this information the following values were obtained regarding the components:

Component	Metric	Value
Webcomponents	CouplingAVG	2,95847055
	CouplingMAX	4.5
Telerik	CouplingAVG	3,598858878
	CouplingMAX	3,598858878
Aspose	CouplingAVG	3,598398791
	CouplingMAX	3,598638419
Aspnetemail	CouplingAVG	3,591100586
	CouplingMAX	3,591119324
GoogleMaps API	CouplingAVG	1,461279598
	CouplingMAX	1,462105764

The three external components Telerik, Aspose and Aspnetemail all have the pretty much the same coupling values, which can be understandable as they are all only connected to the webcomponents and are all

assumed to have the same type of coupling; control and data. The GoogleMaps API is only considered to have a data coupling however, as it is an API, and therefore has a slightly loose coupling.

The complexity value given from MIRA was impossible to interpret, although MIRA gave the information about what tools they used for developing Alumni Community, Visual Studio, they didn't provide information regarding what plugin to Visual Studio they used to perform the measurements. Therefore the default Visual Studio code analysis program is assumed; as the output format of the measurements in the file provided by MIRA matches the output as described in Visual Studio (99). From the measurements given size was readable however and the size of the *webcomponents* was 23803 lines of code. The code lines measured in the tool is intermediate language however and not the actual source code, thus it cannot be used in comparison with the rest of the systems. Since there was no confirmation regarding as to which metric analysis tool was used the metrics are not used in the conclusion regarding the system, and only presented in Table 8 as cursive in case it can be used for future work.

However using the information obtained together with the uncertainty towards it, calculations with EAAT reached the following results:

System	Metric	Value
Alumni Community	Cyclomatic Complexity	3,00
	CouplingAVG	3,05
	CouplingMAX	4,50
	Size	23807
	<i>Cyclomatic Density</i>	<i>0,126013357</i>

Table 8: The results of the Alumni Community system.

The Alumni Community modifiability can be summed up as high, the cyclomatic complexity is low and the coupling values are low. The highest coupling value is the one to the database which is shown in the couplingMAX attribute. If the size and cyclomatic density values were to be taken into consideration as well the modifiability would still be considered low as the cyclomatic density is just slight below the lowest limit and the size is rather small.

## 15.5 TIMEEDIT

The TimeEdit System (TE) is the scheduling system used by employees and students at KTH. The TE is a system solutions developed and sold by the commercial company Evolvera(www.evolvera.com). The current system TE is realized by two parts, TimeEdit Global and TimeEdit Lokal. There are two processes which use the TE system (Figure 51):

- The student process – This system is used to present the different seminars and lectures for the students, and is thus used by the students during the “study” activity of the student process.
- The scheduling process – The TE system is obviously used in the scheduling process in order to present the different objects for the students. In this process the different objects are created, planned and presented in the TE system.

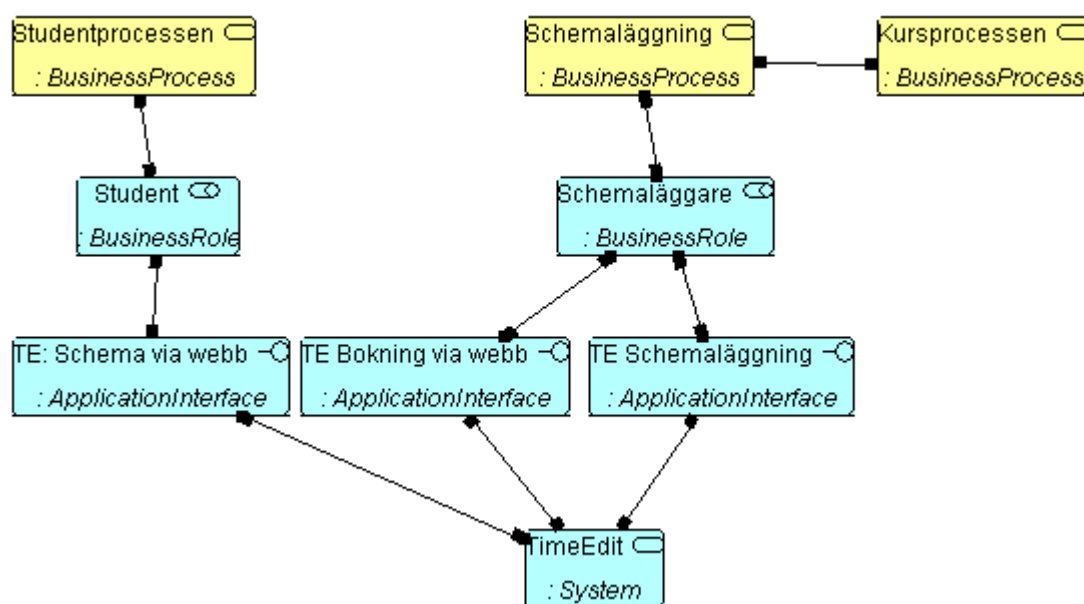


Figure 51: The processes, roles and interfaces that are used with the TimeEdit system.

There are a few interfaces in order to access the TE system; first is the schedule presentation on the web interface, which is used by students in order to find information regarding their schedules. The schedule planner uses the TE Global client for creating schedules, and the TE Lokal web client is also used to some regard.

According to the information obtained, the system is as mentioned divided into two parts:

- TimeEdit Global – This is the part which contains the schedule part; this is where the timeslots are booked.
- TimeEdit Lokal – This is the locations part of the system, which is to book facilities.

Component	Metric	Value
TimeEdit Lokal	CouplingAVG	3,41
	CouplingMAX	3,41

The values calculated for TE Global were:

Component	Metric	Value
TimeEdit Global	CouplingAVG	3,58
	CouplingMAX	3,67

Both Lokal and Global can be considered to have an average coupling value and any component coupled to it has an average chance of being affected.

The TE Global communicates with the Lokal component in order to establish at what time that a certain facility is booked. In addition to this there is another component considered part of this system (Figure 52), which was not developed by Evolvera:

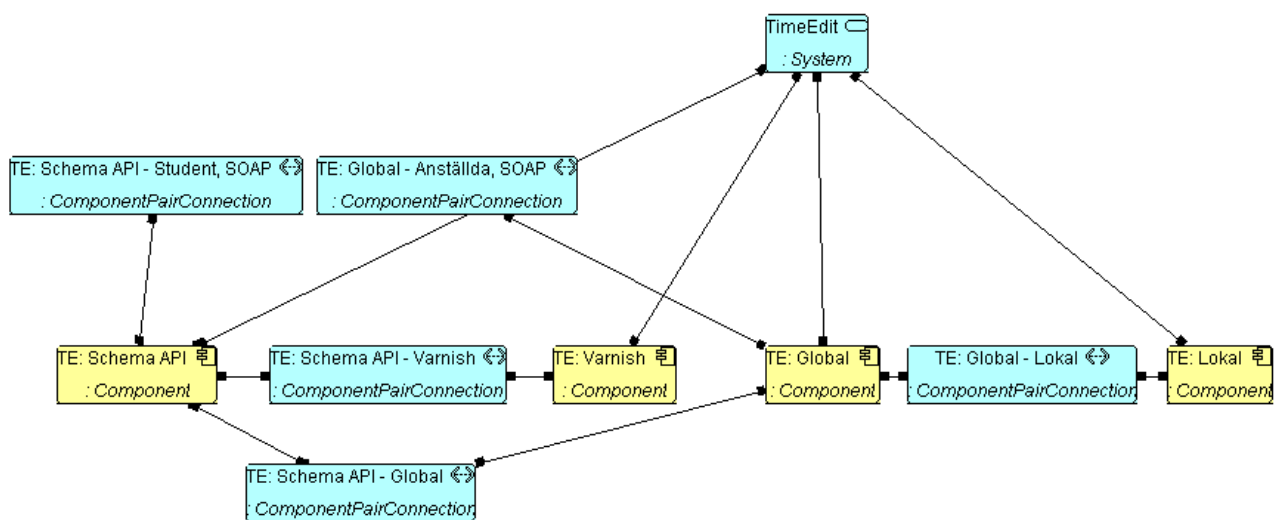


Figure 52: The components that realize the TimeEdit system, also the connections to other systems, in this case Mina Sidor.

- TimeEdit API – This component was developed at KTH in order to be able to integrate certain scheduling services with other systems available at KTH, such as Mina Sidor.
- Varnish – A component realized by an external program called varnish, an instance of the same product used in the KOPPS system for caching static information.

Component	Metric	Value
TimeEdit API	CouplingAVG	4,00
	CouplingMAX	4.66
Varnish	CouplingAVG	4,00
	CouplingMAX	4.66

The Varnish and API components are rather tightly coupled to each other and to Global, which can be explained as any changes to Global would most likely affect the API and any changes to API would have to be made to Varnish.

And although metrics about the TimeEdit system was not available from Evolvera, some information regarding this API which was developed at KTH was provided by the developers at KTH. The TimeEdit system is hosted at KTH and runs on a Windows environment, the API is run on a Tomcat Servlet. According to the documentation the TE Global server and database is run on a 4D application server (Figure 53). The TE Lokal is a Java application with MySQL database.

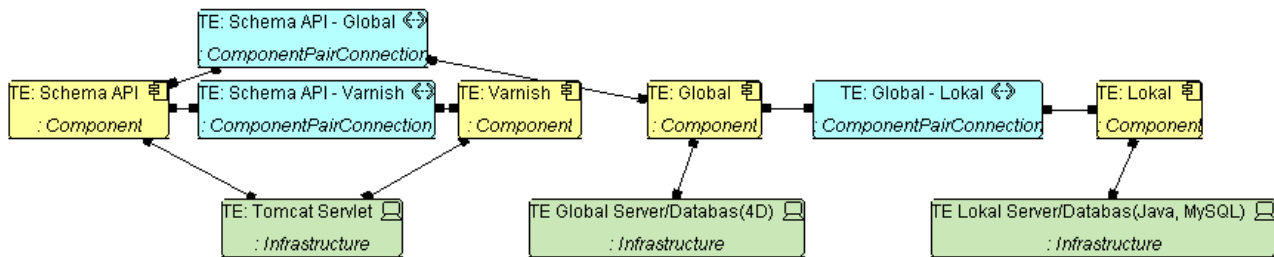


Figure 53: The technology which the TimeEdit system runs on.

As the information regarding the TimeEdit system was very scarce, especially from Evolvera, high levels of probability were used regarding the couplings estimations which were not provided by the ITA developers: a coefficient of variation at 50%. The coupling between TE Global and TE Lokal was estimated to be a control coupling as it is assumed that the different components have a form of control relation regarding each other. When it comes to the varnish system the same uncertainty rate was used as in KOPPS.

The results of the analysis were:

System	Metric	Value
TimeEdit	Cyclomatic Complexity	3,0
	CouplingAVG	3,91
	CouplingMAX	4,66

TimeEdit – The modifiability of the TimeEdit system, based on the information that was available, suggests it being average, it is tightly coupled and thus any modifications made would most likely cause ripple effects, and modifications to any related systems would most likely be required. The complexity value was low which hints suggests that the system is easy to understand, and thus easy to modify. These values together, one high, and one low thus combined suggests a more average modifiable system.

## 15.6 DIAK

Database for International Activities at KTH (DIAK) is a system used for handling exchange students that are applying and studying at KTH, and is also used for handling student housing at KTH. The only ones that use this system are the people at KTH's admission department, which use the web interface in order to handle the exchange students (Figure 54).

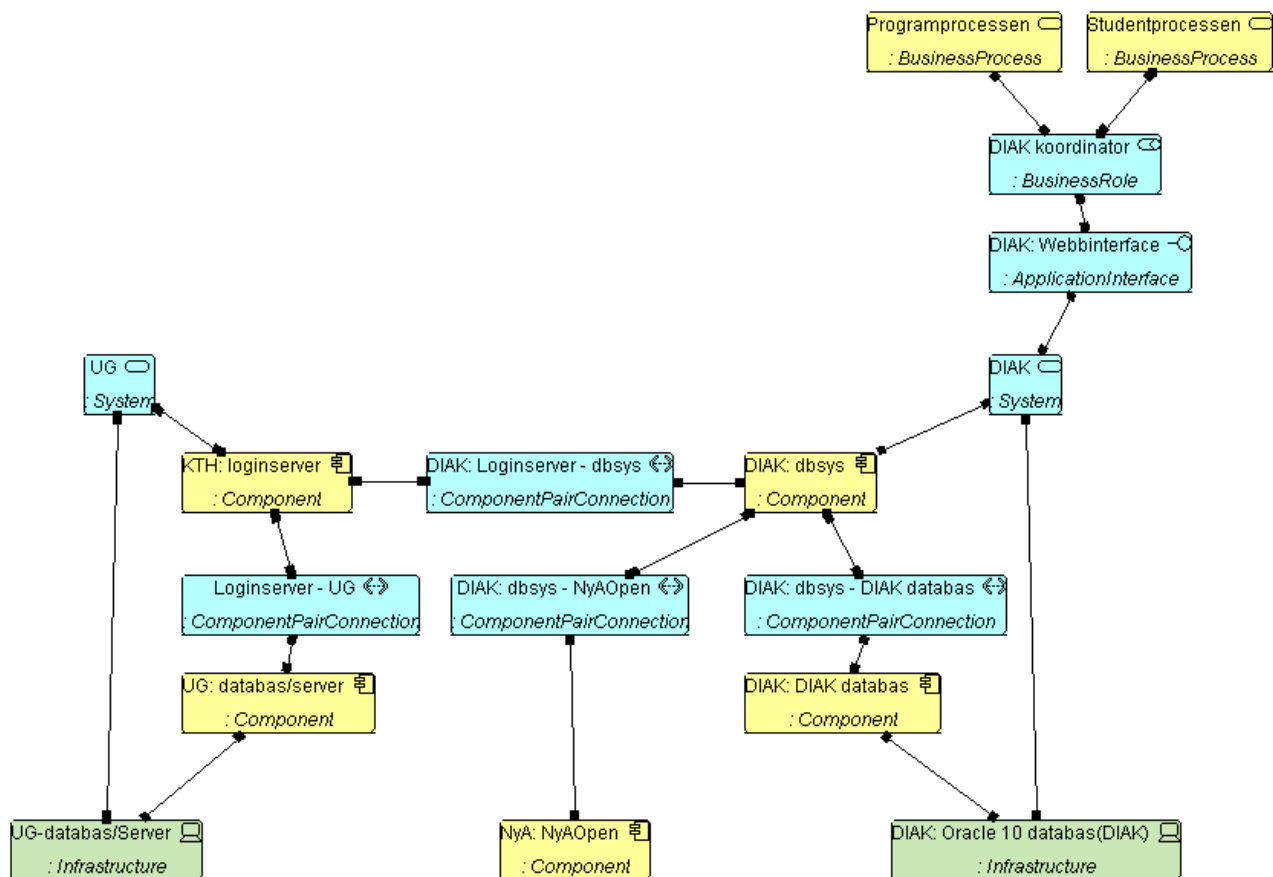


Figure 54: The DIAK system together with connections to UG.

The processes which use the DIAK systems are:

- The student process – As the exchange students are also students, the exchange student programs are used in order to handle them.
- The program process – Used for determining which Master's program that exchange students are to attend.

The DIAK system contains several main areas which are used for administrating among other things the exchange students. The main areas are:

- Inresande (Incoming students)
- Magisterprogram (Master's Program)
- Utresande (Outgoing students)
- Bostäder (Housing)
- Avtal (Agreement)
- Statistik (Statistics)

Unique to the DIAK system is the Housing part as there is currently no system at KTH which can handle it or replace its function, which can be used as an argument why DIAK has been kept around all these years.

DIAK is built around a database, which is an old Oracle10 database and DIAK is tightly coupled to and was developed by one person in ASP and is currently being hosted on an IIS server at KTH. There is only one component which realises this system (Figure 54):

- Dbdsys – This component was described as different ASP-scripts which present the many different tables in DIAK.

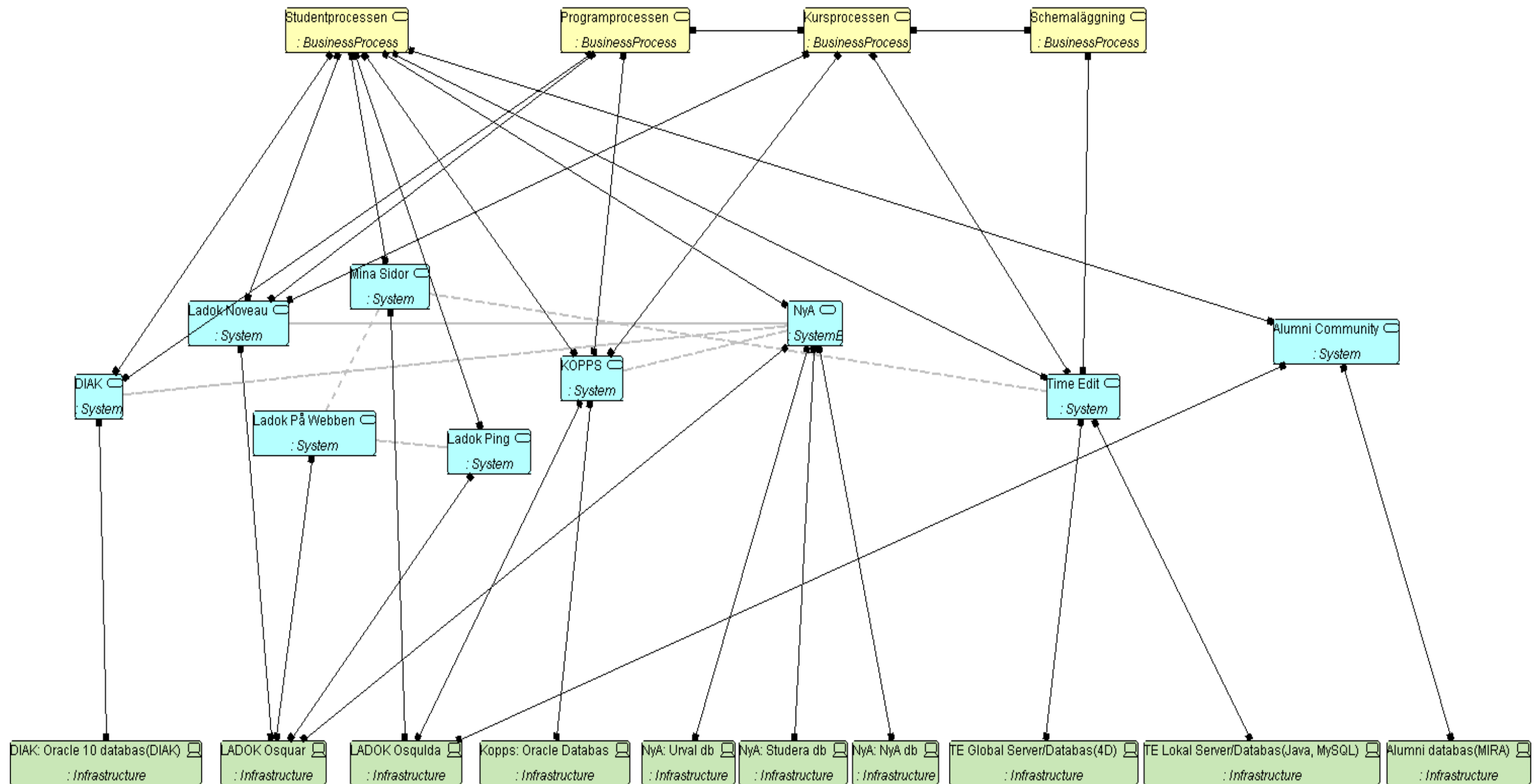
Some estimated information regarding the couplings was given. The developers expressed some uncertainty to these measurements though as there was no measurements tool available for the programming language which the system was developed in. This provided a degree of uncertainty to the measurements which was then defined as coefficient of variation at 25%. As there is only one component which thus realises the DIAK system, the calculation in EAAT for that component is the same as the System measurements:

System	Metric	Value
DIAK	Cyclomatic Complexity	4,0
	CouplingAVG	3,66
	CouplingMAX	3,69

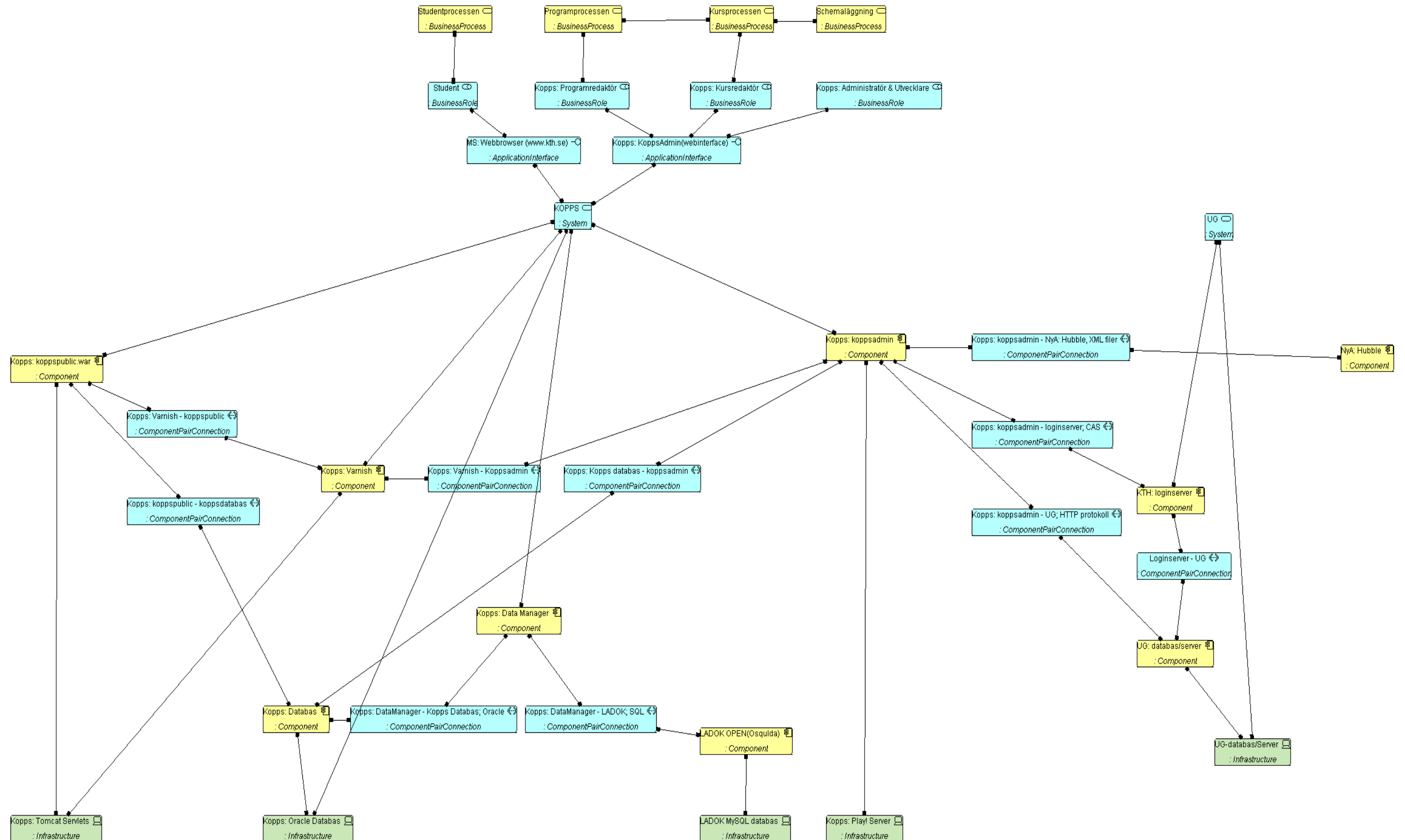
The DIAK system, based on the information acquired suggests a system that has average to high modifiability. As size and thus also density values were not available, the coupling values were given and the complexity calculation from the model was available however. The coupling calculation was measured to be of average value, suggesting that there is an average chance of being affecting by modification in other systems or modules. Since the system is connected to NyAOpen this suggests that any changes made to the NyA database could affect DIAK. The complexity was measured to be low, providing a sense of simple understanding. This suggests thus that the modifiability should be average or high.

The fact that DIAK is developed in ASP, a script language, and there seems to be no tool available for measuring size or complexity. This proposes that performing any kind of software analysis on this system could be quite a challenge, and the scarcity of the tools further suggests that the language is not quite supported for analysis either.

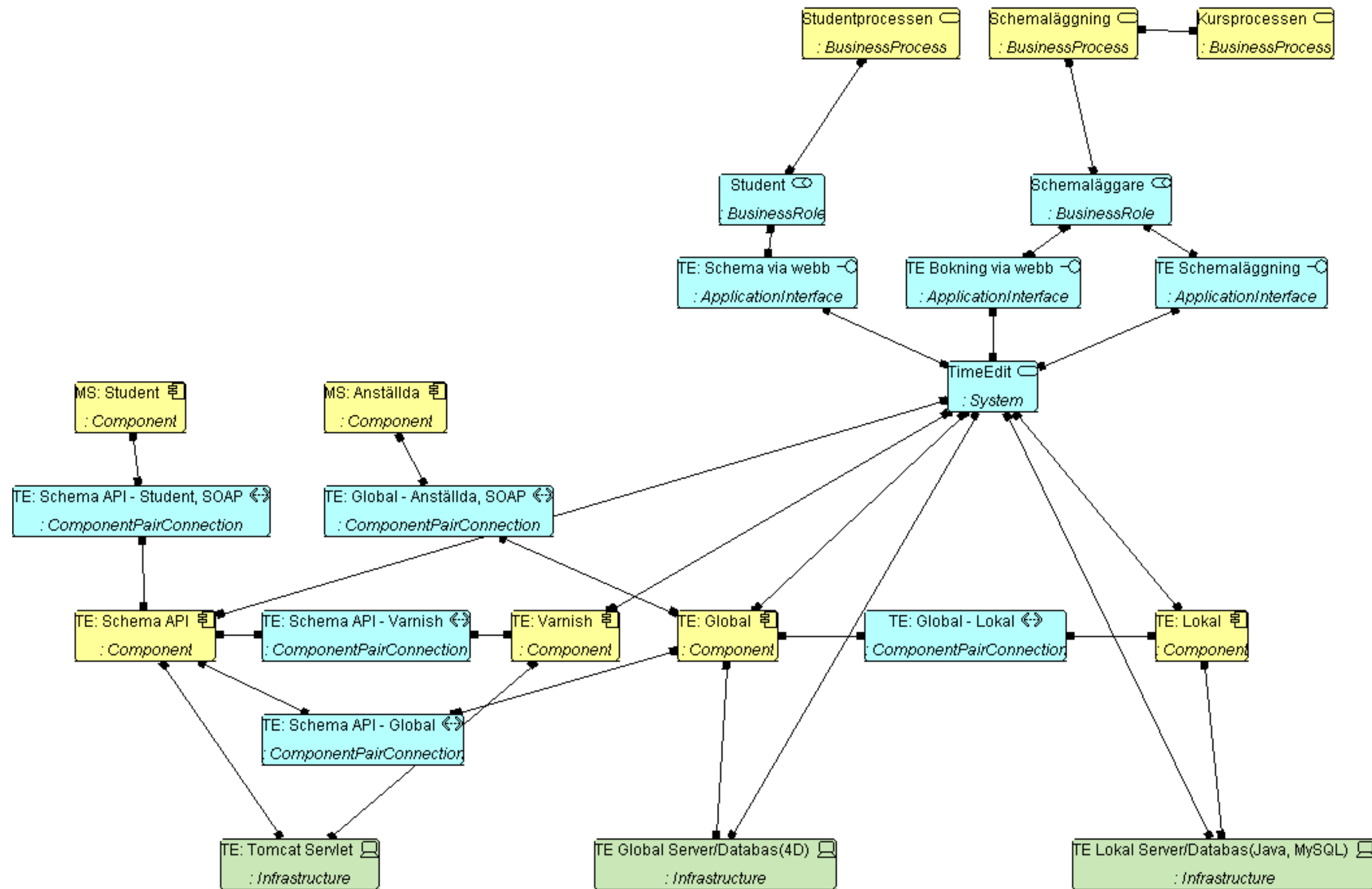
## 16 APPENDIX – MODELS



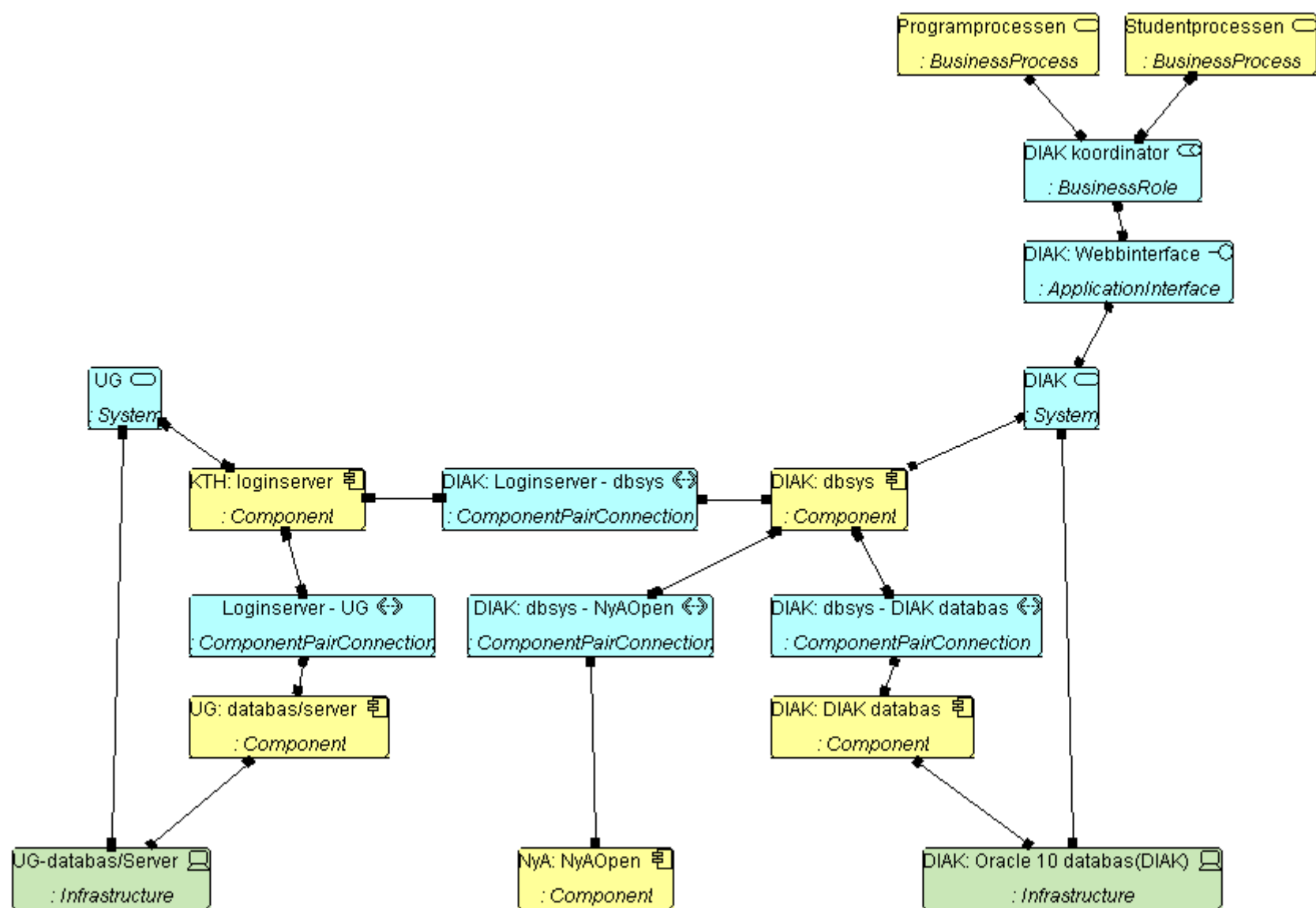


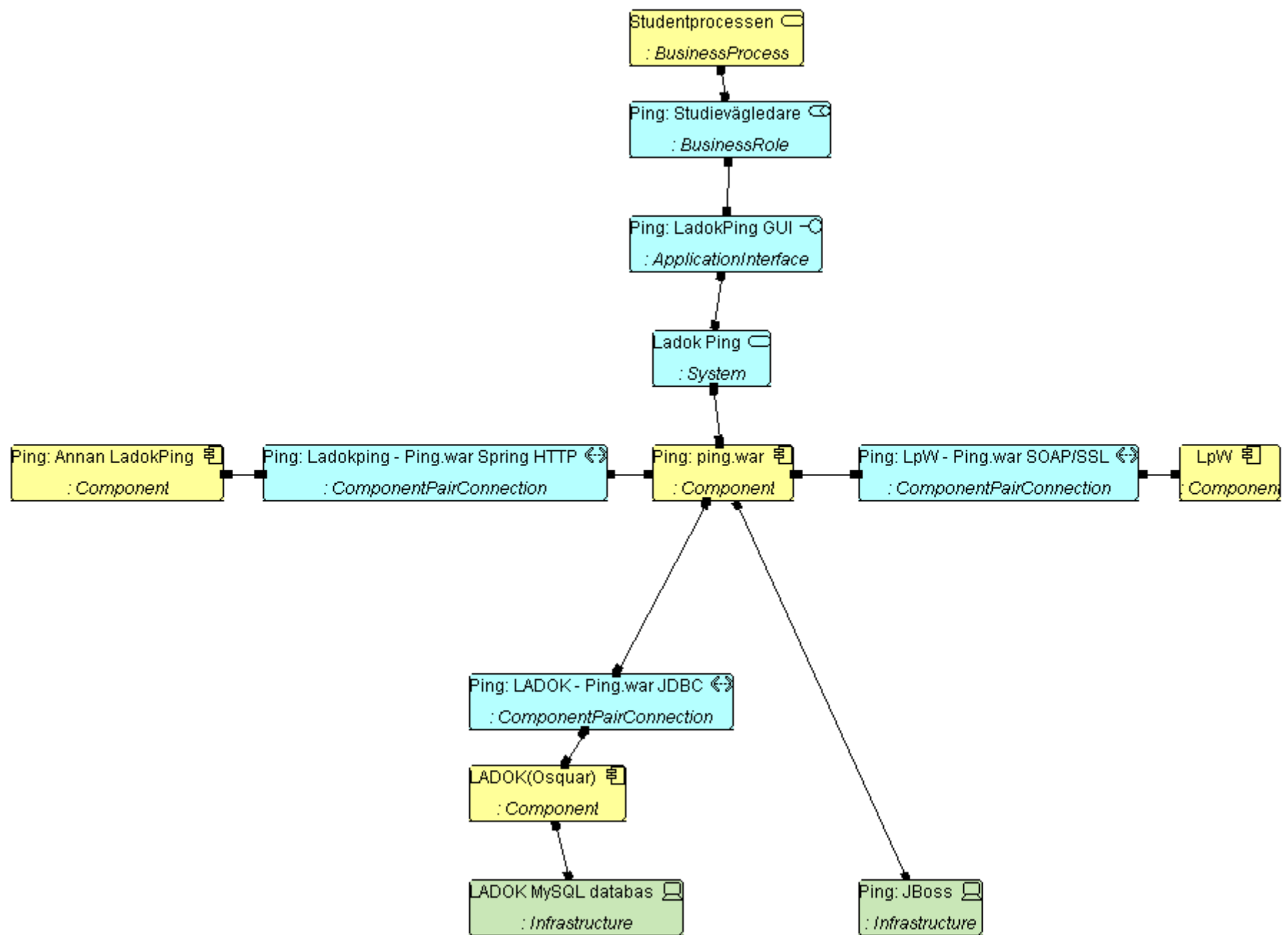


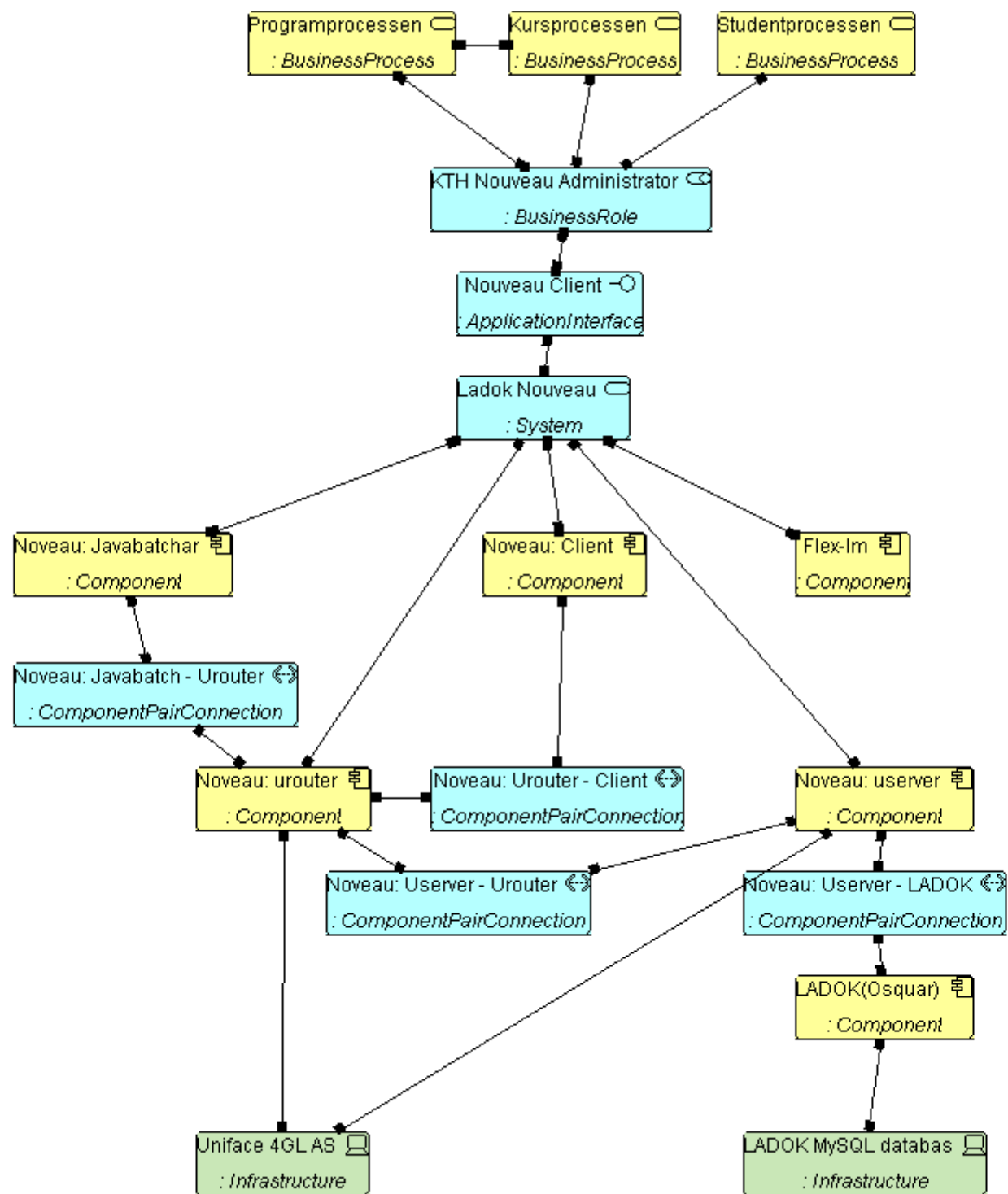


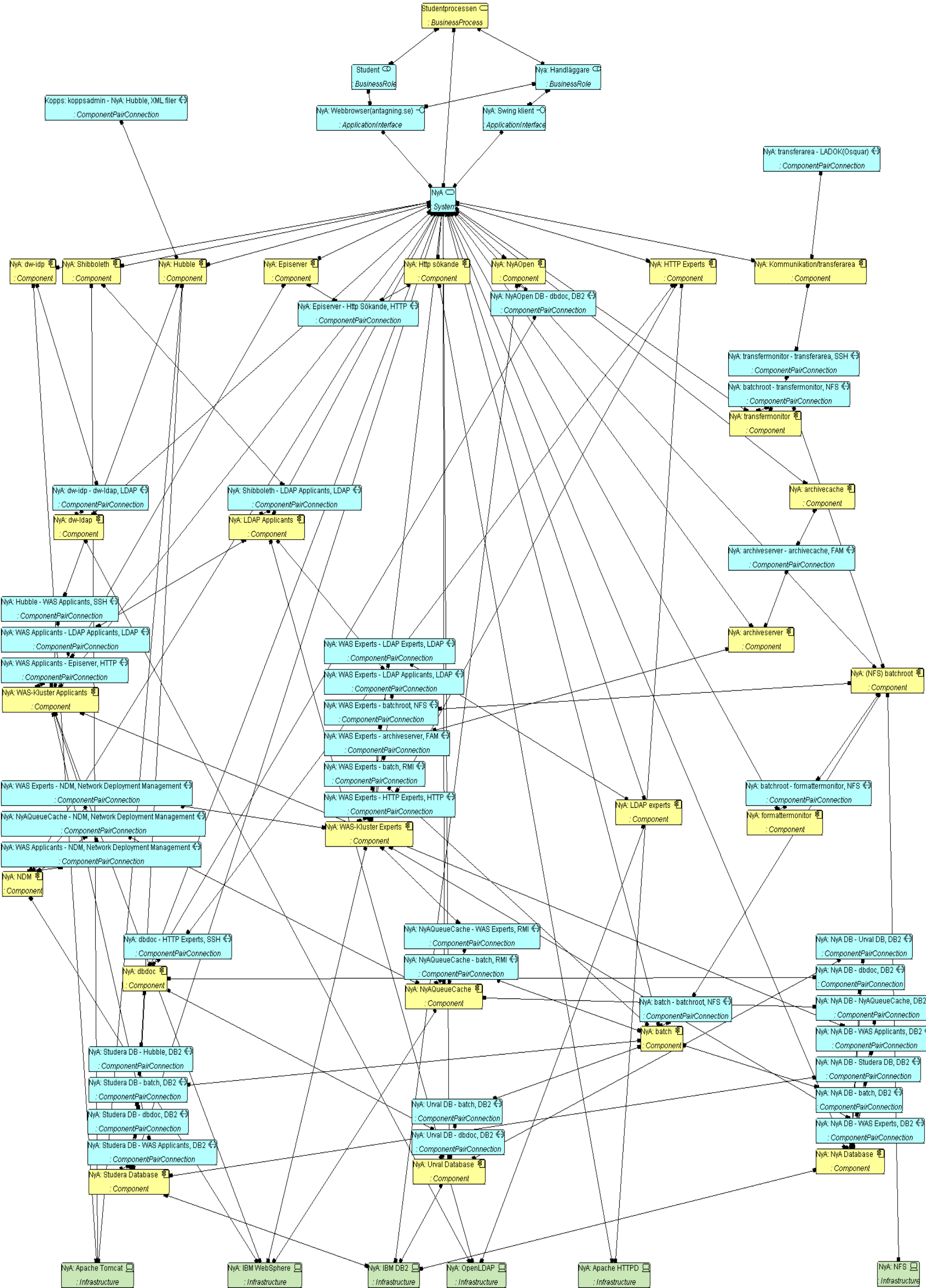














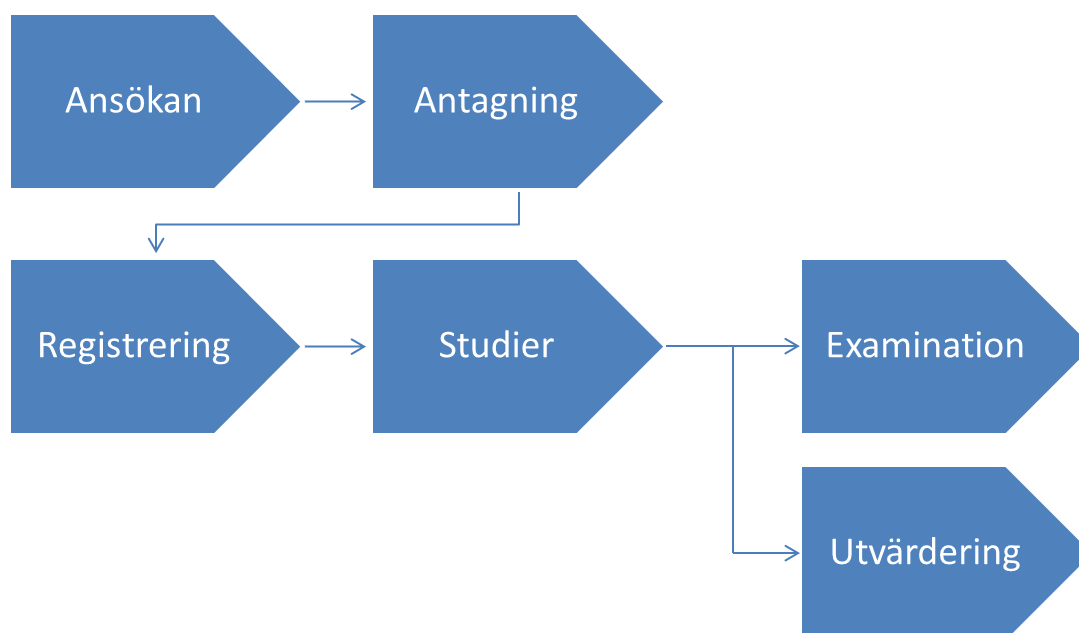
## 17 APPENDIX – UPDATED PROCESSES

These are the updated processes; the models were updated and recreated. As the processes were not the main focus of the study, limited time was used for investigating them. The information that was gathered was mostly put to use for updating and provide a short description of the different processes and their activities. The figures were adapted from out of date documents, but run by current personnel at VoS in order to update them and confirm their validity.

The processes which were updated still retained a lot of the activities and structures from several years ago. Although this degree project does not further investigate the process activities, it does suggest that the processes remain while the systems, such as Mina Sidor are modified, and new systems such as KOPPS are created to meet the process' needs better. The student process was updated and compared with the results of the student process of other universities, and the results proved that the processes are very similar to other universities. The other processes were updated and proved to be mostly the same, but in conclusion it shows that the systems are built around the students, i.e. the product.

### 17.1 STUDENT PROCESS

The student process, this represents the process of handling a student's time at a university from admission to graduation. This process is the core of all the universities in Sweden, and similar processes are available at all of them according to the documents from the LADOK3 project.



**Ansökan(Application)** – Information exported from KOPPS is made publically available on the web. Using this information an applicant decides what program or course he or she wishes to study. An application is made using the studera.nu and NyA.

**Antagning(Admission)** – From autumn 2007 all new admissions are made through antagning.se and the NyA system. Eligibility is determined using the eligibility models which can be constructed in the system. International student admissions are also handled through NyA at first but further qualification tests are done at each individual institution.

Registrering(Registration) – Registration of new students are handled by experts and inputted into LADOK using Nouveau.

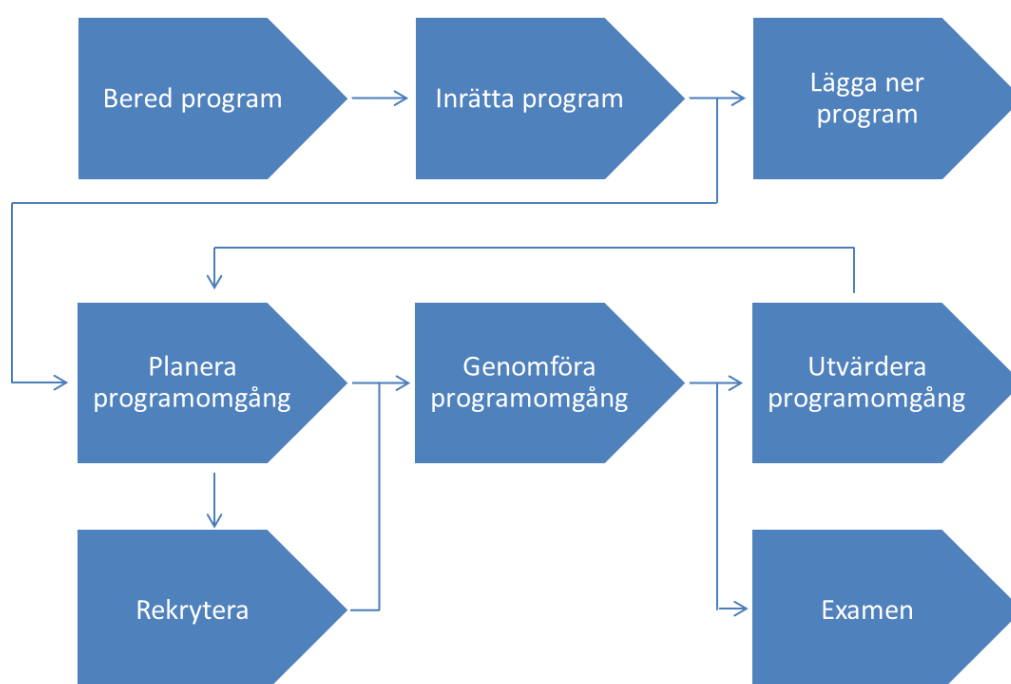
Studier(Studies) – The time that the student spend studying, from admission to graduation. During this time the student performs term registration, course registration, exams etc. Most of these functions are made available through the Mina Sidor system.

Examination(Graduation) – Students who are considered to fulfill the requirements for graduation hands in an application for graduation. This phase also considers handling alumnae, and thus has a connection to the Alumni Community system.

Utvärdering(Evaluation) – Courses and programs are evaluated once they are finished. Feedback is possible through surveys both oral and written.

## 17.2 PROGRAM PROCESS

The program process is for establishing, maintaining and terminating programs. A program cannot exist without courses, thus the program process is dependent on the course process. KTH is a program university, focused on programs.



Bered program (Prepare program) – A program is prepared and planned in order to determine its purpose, subject and contents.

Inrätta program (Establish program) – A program is established, this is finalized by a decision from the university board (universitetsstyrelsen).

Lägga ner program (Close/dismantle program) – Once a program no longer fills its purpose it is dismantled. This requires a decision from the university board.

Planera programomgång (Plan program edition) – The plan for the program is determined each year.

Rekrytering (Recruitment) – Recruitment necessary for carrying out the program such as teachers and other administrative personnel.

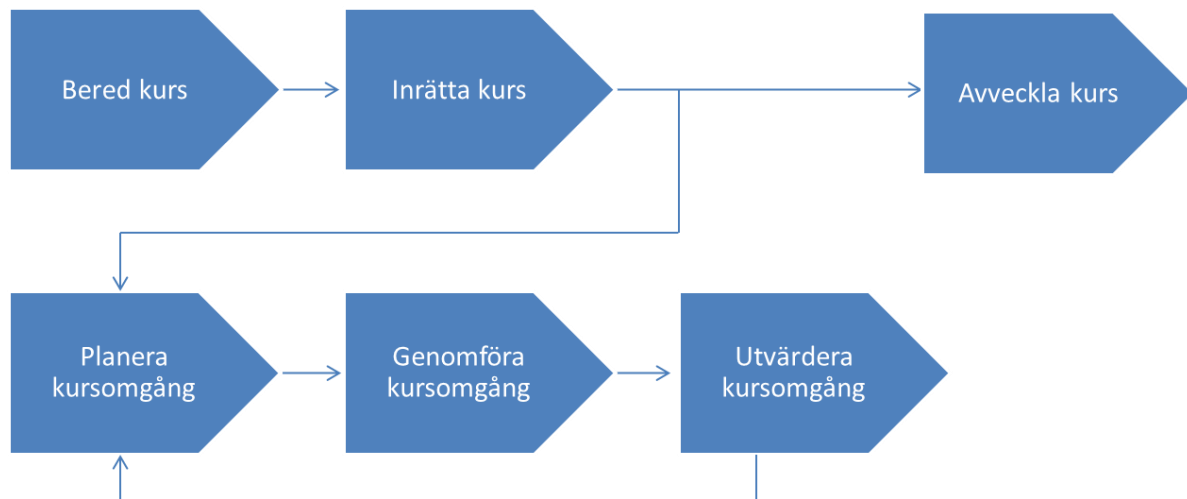
Genomföra program (Implement program) – This is the part which programs are implemented and students study.

Utvärdera programomgångar (Evaluate program edition) – Evaluation of the program that year in order to provide feedback to improve the program for next year.

Examen (Graduation) – Students graduate once they have finished their programs.

### 17.3 COURSE PROCESS

The course process is for establishing, maintaining and terminating courses. A program consists of different courses and cannot exist without them, but courses can exist without programs.



Bereda kurs (Prepare course) – The phase in a course is prepared.

Inrätta kurs (Establish course) – The course is established when the previous phase is finished and a decision is made by the school dean.

Avveckla kurs (Dismantle course) – This is the sub process of dismantling a course, when it no longer to be implemented. This requires another decision from the school dean.

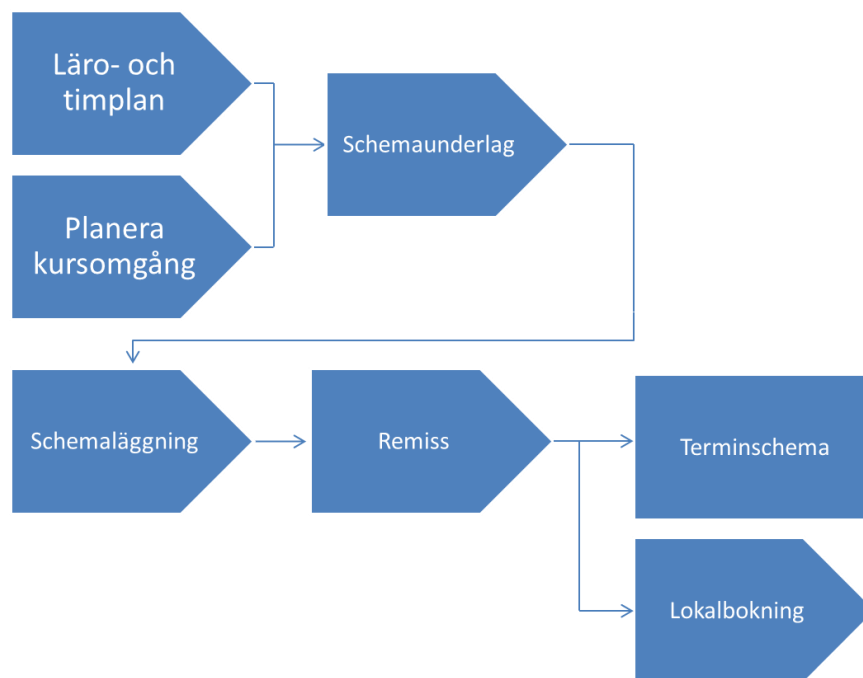
Planera kursomgång (Plan course edition) – The planning of the course for the coming year's implementation of the course. The phase takes in the feedback from the evaluation phase.

Genomföra kursomgång (Implement course edition) – Implementing the courses, teaching the student and following the planning of the course.

Utvärdera kursomgång (Evaluate course edition) – Collecting feedback about the course, through either oral or written surveys, this is then used in the planning for the next course implementation.

## 17.4 SCHEDULE PROCESS

The scheduling process for creating schedules for different courses. Schedules cannot exist without courses.



Planera kursomgång (Plan course edition) – This is done individually by each teacher responsible for each course.

Läro- och timplan (Curriculum and Timetable) – This is used for updating the program relations in TimeEdit.

Schema underlag (Schedule foundation) – A foundation for a schedule is prepared in KOPPS and then used by TimeEdit for creating the schedule. This is done manually and not through a system connection between KOPPS and TimeEdit.

Schemaläggning (Scheduling) – This is where the course schedules are created to match the different programs and times.

Remiss (Referral) – This is where any last changes can be made to the schedule.

Terminschema (Term schedule) - The finished schedule for the term is published and made available for the students and employees on the web.

Lokalbokning (Locale booking/Room reservation) – This is of booking a facility or reserving a room for when they aren't occupied, i.e. not used during the regular curriculum.

## 18 APPENDIX – DATA

KOPPS and DIAK are systems developed and hosted at KTH. Information gathering for modelling the KOPPS system was done by going through an old system model from 2007 and a workshop with a developer at IT-Avdelningen(ITA) at KTH. The only documentation available for DIAK was an old, coffee-stained database table document and some process description papers. A quick demonstration and explanation from a different developer at ITA was also given about the DIAK system explaining the basic functionality and technology of DIAK.

In addendum to the emails and, photos and notes below, information was gathered from documents available at various internal websites which requires login acquired from the VoS department. Posting it here would seem to defeat the idea of keeping them at the internal site since this is a public document.

### 18.1 EMAILS

#### LPW DATA EMAIL

Metrics Summary For Checkpoint 'Checkpoint1'

Parameter =====	Value =====
Project Directory	C:\Documents and Settings\mada0108\My
Documents\Temp\	
Project Name	LPW
Checkpoint Name	Checkpoint1
Created On	7 Mar 2012, 08:03:20
Files	1015
Lines	178 758
Statements	87 703
Percent Branch Statements	12,1
Method Call Statements	59 546
Percent Lines with Comments	22,1
Classes and Interfaces	1 072
Methods per Class	10,66
Average Statements per Method	5,24
Line Number of Most Complex Method	{undefined}
Name of Most Complex Method	CourseRegistrationVO.equals()
Maximum Complexity	102
Line Number of Deepest Block	{undefined}
Maximum Block Depth	9+
Average Block Depth	1,98
Average Complexity	1,88

*Extract from an email received from ITS at Umeå university, whom performed the metrics.*

**NyA**

SAD & PK

MAIL 2

Loggning? CAS? BKS? Används det fortfarande, de är ej med i produktkartan?

Om de inte används längre, sker då autorizeringen bara genom LDAP?

CAS och BKS är borttagna ur arkitekturen. Auktorisering sker genom en enklare inbyggd funktion i systemet.

I SAD finns en LADOK, vilken LADOK? Dokumentet säger alla tre driftcentraler, stämmer det fortfarande?

NyA går mot alla LADOK instanser på alla driftscentraler.

Vad är Transfermonitor och transferarea? Är det den generella kopplingen till externa system, filöverföringar osv?

Ja

Fortfarande incoming och outbound för filöverföringar? Eller nya funktioner i det området?

Utåt mot högskolorna ser det ut så.

Filöverföringar till Posten via transfermonitor också?

Ja

RSV kopplingen SHS, om fortfarande aktuell genom batchen? Eller genom transfermonitorn som alla andra?

Nej det är en speciallösning som går via https.

Netuniv nämns i SAD, är det fortfarande i aktuell användning?

Se svar ovan

Det är tre saker som jag inte riktigt kan hitta någon benämning om i SAD filen:

Dbdoc är något som dyker upp i PK, vad är det?

Delprodukt som innehåller databasdokumentation (namn på RMP paket vid leverans)

Dw-idp och dw-ldap, är något som dyker upp i PK, vad används det för?

Delprodukt för Idp (Shibboleth) och LDAP (namn på RMP paket vid leverans) som är specifika för NyA webben

Sker extern autorizering fortfarande genom CWAA? Nej

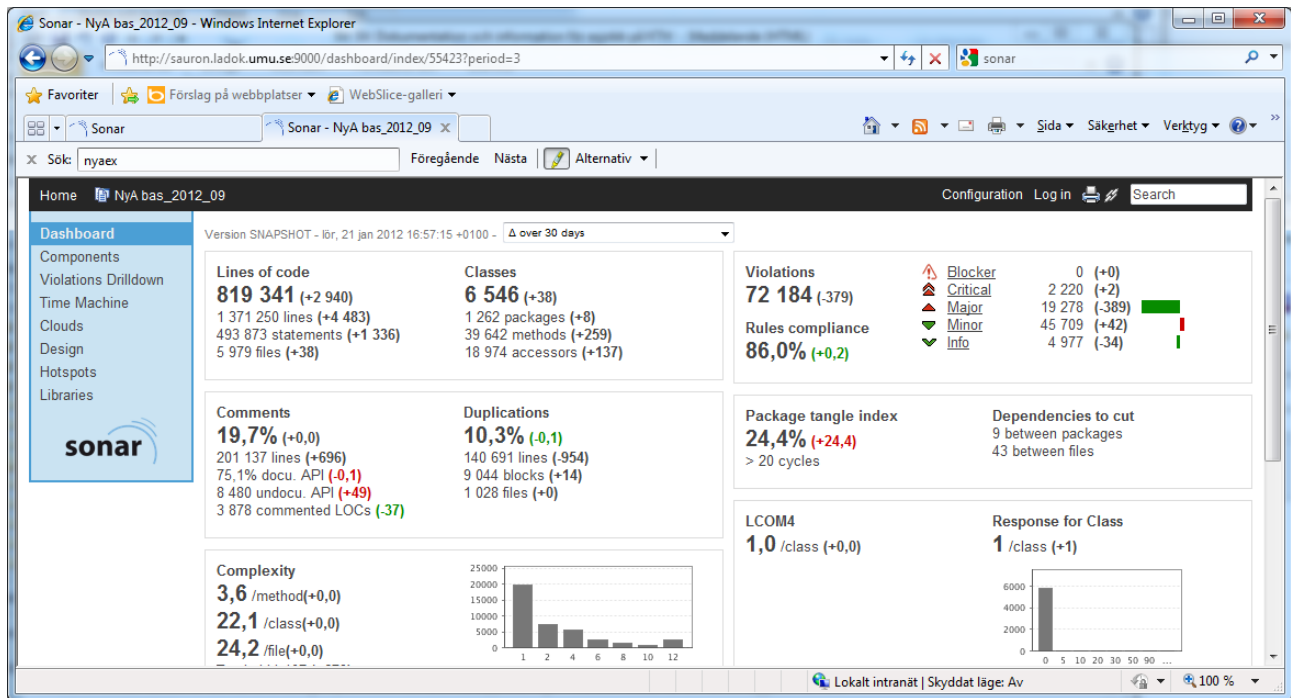
De olika inringade delarna av PK, är det olika paket eller web arkiv? (EAR) I SAD nämns att på sikt ska EAR filerna minskas, har antalet ear filer blivit minimerade, eller fortfarande samma struktur som i SAD filen?

Allt paketeras med RPM.

EAR strukturen stämmer inte som det ser ut i SAD det är färre EAR filer idag.

Finns det möjlighet för tillgång till information angående mer systemspecifika såsom Cyclomatic Complexity(ett sätt att beräkna exekveringsvägar i ett program) och Kodrader?

Vi använder Sonar som bla tar fram den informationen. Dock tyvärr ej åtkomlig utifrån i dagsläget. Skickar med en bild som visar en översta nivån av statistik.



### MAIL3

Mätningarna som ni gav mig kallar ni översta nivån av statistik? Vad menas med det? En mätning av hela systemet som det är, eller bara webblagret?

Vilka EAR paket finns det idag om de har minskat? Vilka är de?

ApplicantWebbserver  
 ExcptClientServer  
 QueueCacheServer

Hur paketeras och körs EAR paketen, allt paketeras i RPM men vilka EAR i vilka RPM paket?

ApplicantWebbserver->aws-ear  
 ExcptClientServer->ecs-ear  
 QueueCacheServer->qcs-ear

PK visar tre externa delar, Sitevision är CMS, är det DRIFTINFO och AWEB också? Eller är det andra

CMS=Episerver

DRIFTINFO=har vi inte så mycket info om.

AWEB=Sökandewebben

Dbdoc-www, dbro-www, dw-www, rpm-paketen är det informationen om databaserna, NyAopen och shibboleth som visas på användarwebben?

Dbdoc-www, dbro-www är dokumentation till databaserna NyA samt NyA OpenMS=episerver.

Sen behöver jag lite mer systemdetaljer.

De mätningar jag är ute efter är samma som förut; kodrader och cyclomatic complexity som jag redan fått. Men jag är även intresserad av vad för olika kopplingar som finns mellan systemen, både rent tekniskt, exempelvis SOAP eller RMI vilket jag kan se i PK (produktkartan) men behöver gärna veta hur många kopplingar mellan komponenterna. Men om möjligt skulle jag även behöva information för att kunna ta reda på enligt följande specifikation, som ni säkert har hört talas om förut (Coupling & Cohesion). Om det inte är möjligt att ge sådan specifik information så svara gärna efter de två olika kategorierna som jag grupperat de i, beroenden och kommunikation. Om även det inte går så förklara gärna så bra ni kan.

### MAIL4



När det gäller Batch och NyAQueueCache så går det att ta fram då det är få anrop ca 5 stycken. Men när det gäller övriga anrop tex Batch till WAS cluster och alla DB2 anrop så är det ett mycket stort antal som olika metod anrop och här skulle man nog behöva någon form av analys script för att ta fram detta.

*Kopplingen till LDAP:en används ju exvis både till enkel autentisering, att söka ut en lista med entries som matchar ett uttryck och att lägga till/uppdatera attribut. Protokollet bör ge en bra bild av vad man kan förvänta sig att kopplingen används till.*

## MINA SIDOR

Hej!"

Här kommer kopplingskallificeringarna...

Mina sidor:

-----

MS: Student – LpW; SOAP = stamp, control  
 MS: Student - LADOK (Osqulda); SQL = stamp, common  
 MS: Student – Studok; REST, JSON, HTTP = stamp, control  
 MS: Student - UG databas; HTTP = stamp, control  
 MS: Student - Loginserver; CAS = data, control  
 MS: Anställda - Loginserver; CAS = data, control  
 MS: Anställda - LADOK (Osqulda); SQL = stamp, common  
 MS: Anställda – LpW; SOAP = stamp, control  
 MS: Anställda - UG databas; HTTP = stamp, control  
 MS: Studok - LADOK (Osqulda); SQL = stamp, control  
 MS: LpWAuth - UG BKSfiler; HTTP = data, control  
 MS: LpW – LpWAuth; Java = data, control

OBS; LPW-LPWAuth är ren java-runtime kommunikation (LPWAuth är en plugin med kod som laddas direkt in i LPW)

Coupling	Connection Type	Metric
MS: Student – LpW	SOAP	stamp, control
MS: Student – LADOK (Osqulda)	SQL	stamp, common
MS: Student – Studok API	REST, JSON, HTTP	stamp, control
MS: Student – UG databas	HTTP	stamp, control
MS: Student – Loginserver	CAS	data, control
Coupling	Connection Type	Metric
MS: Anställda – LpW	SOAP	stamp, control
MS: Anställda – LADOK (Osqulda)	SQL	stamp, common
MS: Anställda – UG databas	HTTP	stamp, control
MS: Anställda – Loginserver	CAS	data, control

Coupling	Connection Type	Metric
MS: LpWAuth - UG BKSfiler	HTTP	data, control
MS: Studok - LADOK (Osqulda)	SQL	stamp, control
MS: LpW – LpWAuth <sup>4</sup>	Java	data, control

KOPPS:

-----

KOPPS: Koppsadmin – UG; HTTP protokoll. = stamp, control

KOPPS: KOPPS databas – koppsadmin; SQL. = stamp, common

KOPPS: Koppsadmin – loginserver; CAS = data, control

KOPPS: KOPPS: koppsadmin – Hubble; XML filer = stamp, control

KOPPS: koppspublic – koppsdatabas; SQL. = stamp, common

KOPPS: DataManager – KOPPS Databas; pratar Oracle = stamp, common

KOPPS: DataManager – LADOK; pratar MySQL = stamp, common

Varnish är en transparent HTTP proxy/cache, både koppsadmin och koppspublic fungerar utan varnish, så jag vet inte hur relevant klassificeringen blir, men drar till med ngt i alla fall.

KOPPS: Varnish – koppspublic = data, common

KOPPS: Varnish – Koppsadmin = data, common

Coupling	Connection Type	Metric
KoppsAdmin – UG	HTTP	stamp, control
KoppsAdmin – KOPPS database	SQL	stamp, common
KoppsAdmin – Loginserver	CAS	data, control
KoppsAdmin – NyA: Hubble	XML files	stamp, control
KoppsAdmin – Varnish	HTTP proxy/cache	data, common
Coupling	Connection Type	Metric
KoppsPublic – Varnish	HTTP proxy/cache	data, common
KoppsPublic – KOPPS database	SQL	stamp, common
Coupling	Connection Type	Metric
DataManager – LADOK	MySQL	stamp, common
DataManager – KOPPS database	Oracle	stamp, common

TimeEdit/Schema:

-----

TE: Global – MS Anställda; SOAP = stamp, control

TE: Schema API – TE: Global = stamp, control

TE: Schema API – MS Student; SOAP = stamp, control

OBS! Samma varnish resonemang som för kopps applicerar här

TE: Schema API - Varnish = data, common

<sup>4</sup> LpW-LpWAuth is a pure java-runtime communication where LpW is a plugin which is loaded directly into LpW.

```
-<METRIC abbreviation="LOC" name="Lines of code" category="Module"> <VALUE value="436.0" measured="Lpw-kth-authplugin"/> </METRIC> -<METRIC abbreviation="SLOC" name="Source lines of code" category="Module"> <VALUE value="315.0" measured="Lpw-kth-authplugin"/>
```

```
-<METRIC abbreviation="CCavg" name="Average cyclomatic complexity" category="Module"> <VALUE value="2.923076923076923" measured="Lpw-kth-authplugin"/> </METRIC>
```

DIAK:

-----

Informationen runt DIAK är tyvärr lite vag och desutom är det väl tänkt att försvinna så du får ta detta med en nypa salt.

Koppling till NyAOpen som hämtar information efter urval. = stamp, control

Koppling till DIAK databasen(Oracle 10) = stamp, control

Koppling till loginservern = data, control

Hoppas jag fått med svar på alla dina frågor nu, hör av dig annars.

/MVH Jimmy"

Till:

Peter Rosell

Master Exjobb

den 27 december 2011 16:34

Komponent: Kurs- och programkatalogen (kataloginformation) och KoppsAdmin (administratörsverktyg)

Applikations Interface: Webbapplikation, Läsaccess via DB, Öppet API med kataloginformation, Stängt "API" som Kurs och programkatalogen använder för att prata med KOPPS. Dessutom genereras varje natt kataloginformation ut som XML (hämtas av NyA)

Infrastruktur: DB; lokal DB i Oracle samt kopia av delar av LadokDB som hämtas ner och sparas lokalt varje natt. Vi använder Hiberante för att mappa mot databasen. KoppsAdmin utvecklas i ramverket Play med java i botten. Kurs och programkatalogen använder JSP. Både apparna har cacheramverket Varnish som front framför Apachen. Apachen pekar ner till lokal appserver (Tomcat) i fallet för Kurs- och programkatalogen. Inlogning i KOPPS sker via CAS och behörigheten stäms av mot UG via ugklient.

**ALUMNI COMMUNITY**

Hej Peter,

här kommer en liten sammanställning. Glömde att vi använder en komponent för att generera rapporter till word och excel, den är med i listan nu.

**Driftsmiljö**

Server 1:

- Windows 2008 R2
- IIS 7.5 (Web/Applikations server/SMTP)

Server 2:

- Windows 2008 R2
- Microsoft SQL Server 2005 (Byts snart till Microsoft SQL Server 2008) (SQL jobb mot Ladok)

**Kod**

All kod är skriven i C# och kompilerad för .NET framework 2.0, vi ha även beroenden på .NET framework 3.0/3.5

Ne5.dll (Webb applikationen, allt gränssnitt och logik)

- Utvecklad av Mira

Mira.Core.dll (Hantering av anslutning till databas, funktioner för säkerhet, m.m.)

- Utvecklad av Mira

Mira.WebControls.dll (Web forms kontroller)

- Utvecklad av Mira

Telerik.dll (Web forms kontroller)

- Utvecklad av [www.telerik.com](http://www.telerik.com)

aspNetEmail.dll (Komponent för att skicka e-post via smtp)

- Utvecklad av <http://www.aspnetemail.com/>

aspose.cells.dll och aspose.words.dll (Komponenter för att generera MS Excel och MS Word filer för rapporter)

- Utvecklad av <http://www.aspose.com/>

**Code Metrics**

Hittade en plugin till Visual Studio för att ta fram Cyclomatic Complexity, bifogar en rapport.

Du får gärna skicka en kopia på arbetet när du är klar.

Lycka till!

Hälsningar

Per

Hej Peter!

Ursäkta dröjsmål med svar.

Vi använder i nuläget både timeedit global och lokal, dock inte bokning via webb i global.

Lokalbeställning via webb i lokal används i mindre utsträckning.

Din bild stämmer ganska bra tycker jag. LoT används fortfarande för att uppdatera timeedit med programrelationer.

På schemasidorna för lärare har vi en uppdaterad processbeskrivning:

<http://intra.kth.se/utbildning/tentamen-och-schema/om-schema-1.279132>

Återkom gärna om du har någon ytterligare fråga.

Mvh,

Aksel

### **LadokPing**

Hej!

Jag heter Peter Rosell och jag gör ett exjobb på VoS avdelningen på KTH där jag gör systemmodeller och analyserar modifierbarheten i dessa system. Ett av dessa system som jag kollar på är LadokPing, och utifrån den information jag har fått tillgång till ifrån Ladok hemsidan så har jag lyckats rita upp den bifogade modellen som är generellt baserad på dokumentationen. Jag har dock lite frågor som du gärna får svara på om du kan annars gärna hänvisa mig till någon som kan.

Jag läste i ett av dokumenten att det i slutet av 2011 bytte ut EJB-bönorna till spring-bönor, påverkade det själva strukturen på systemet eller så?

Nej inte direkt. Vi använder springbönorna till samma sak som EJB (transaktioner).

Axis läste jag också att ni hade tagit bort, använder ni bara JWS då för SOAP generering? Eller har ni någon annan lösning som ersätter den LadokPing komponenten som realiserar systemet?

Hela webservice-lagret är bortrationaliserat. Gui:et anropar springböerna med lokala javaanrop på servern. Allt ligger i samma war-fil.

Det nämndes att tidigare var systemet uppdelat i en sar + war + jar fil, var de filerna de olika komponenterna som finns i modellen bifogad, HTTP, Axis och GUI? Och om det nu då är bara en War fil istället hur ser det ut då?

Du får ta bort Axis-lådan i modellen och dra anropen från gui:et till springböne-lådan. Sen så verkar du ha ett externt system ("Mina sidor") som går mot webservice-lagret vilket det inte kan då det inte finns. Eller är detta LPW?

I så fall pratar det systemet RMI med springböne-lådan.

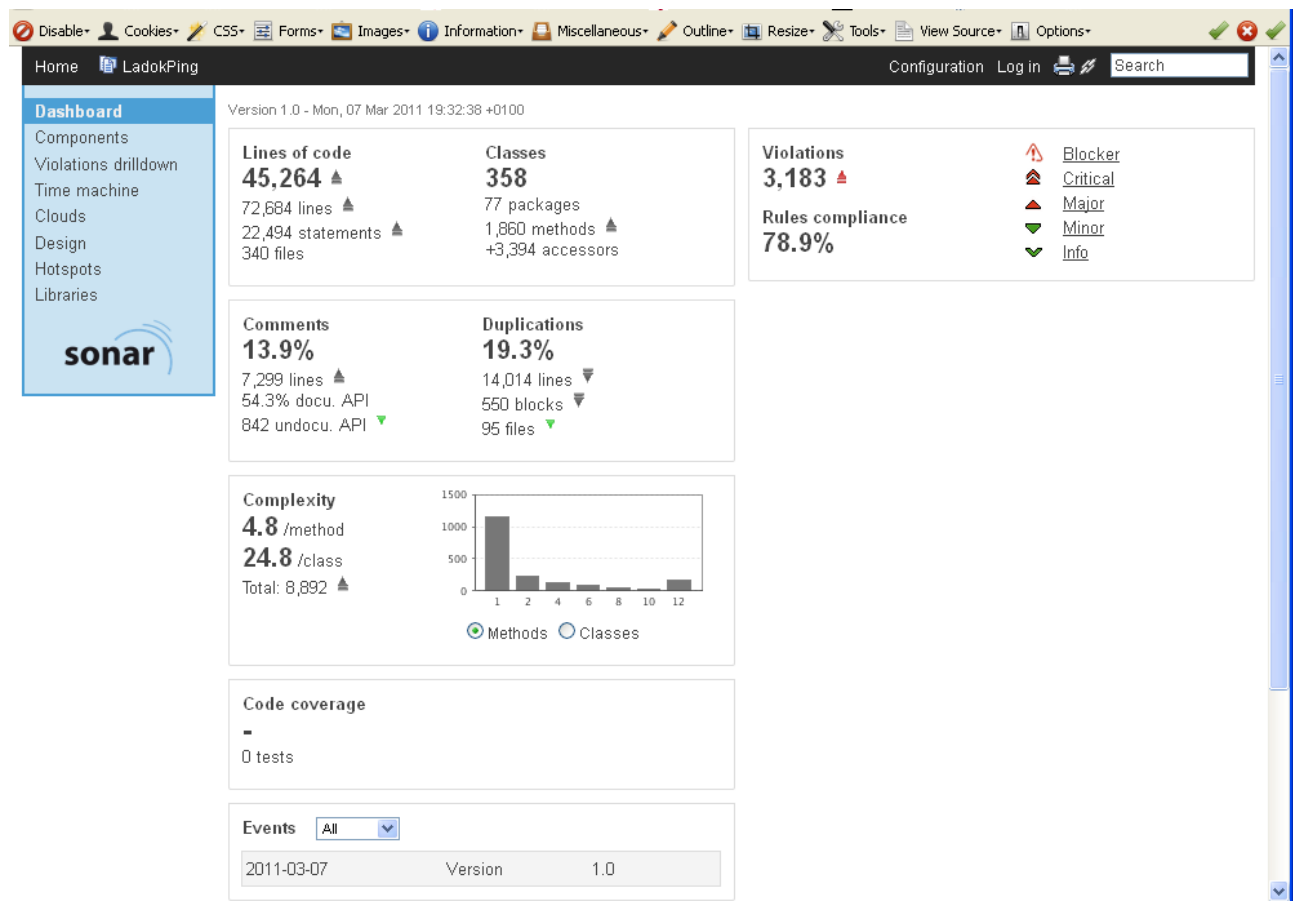
Utöver det undrar jag om det är möjligt att få tag i viss specifik information om Systemet, så som storlek och komplexitet(Cyclomatic Complexity) på komponenterna eller systemet?

Vi har ingen sådan aktuell statistik. Jag bifogar en skärmdump från sonar som kördes i våras innan konverteringen. LadokPing använder en hel del kod från ett annat system(LPW) så det är väl inte hela sanningen.

Om du vill ha aktuella siffror kan du säkert få tillgång till källkoden hos portalutvecklarna på KTH.

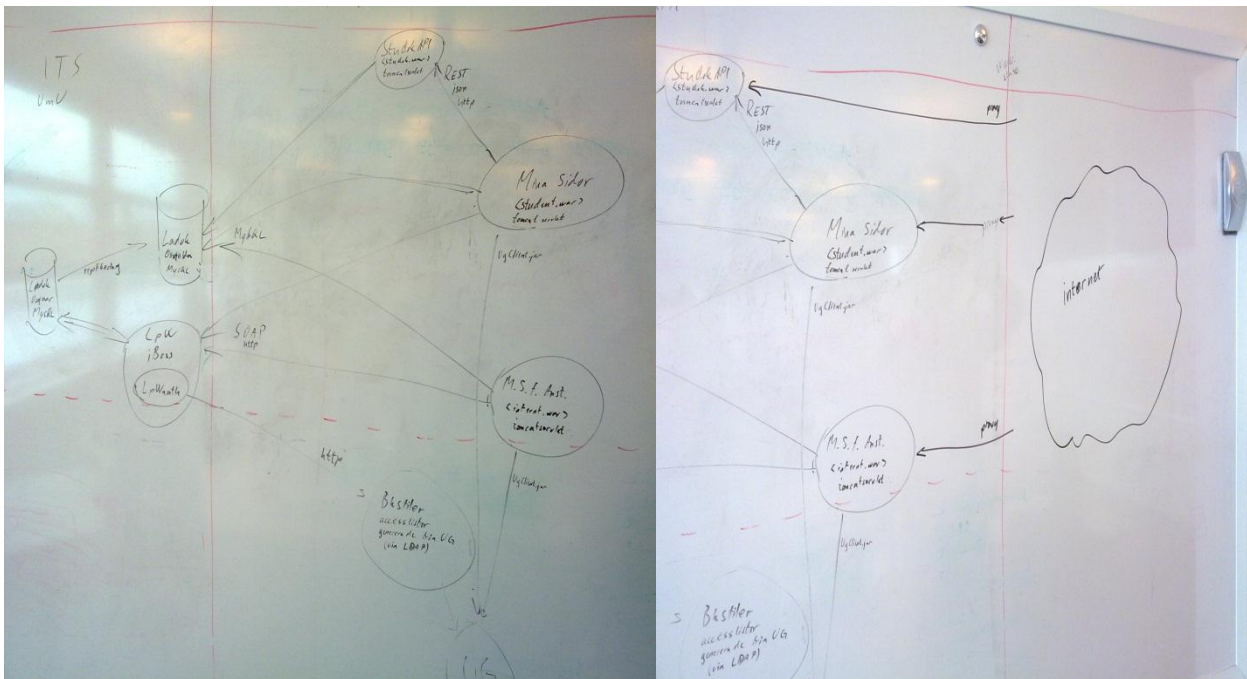
Eller kan vi skicka den Mats?

/Anders

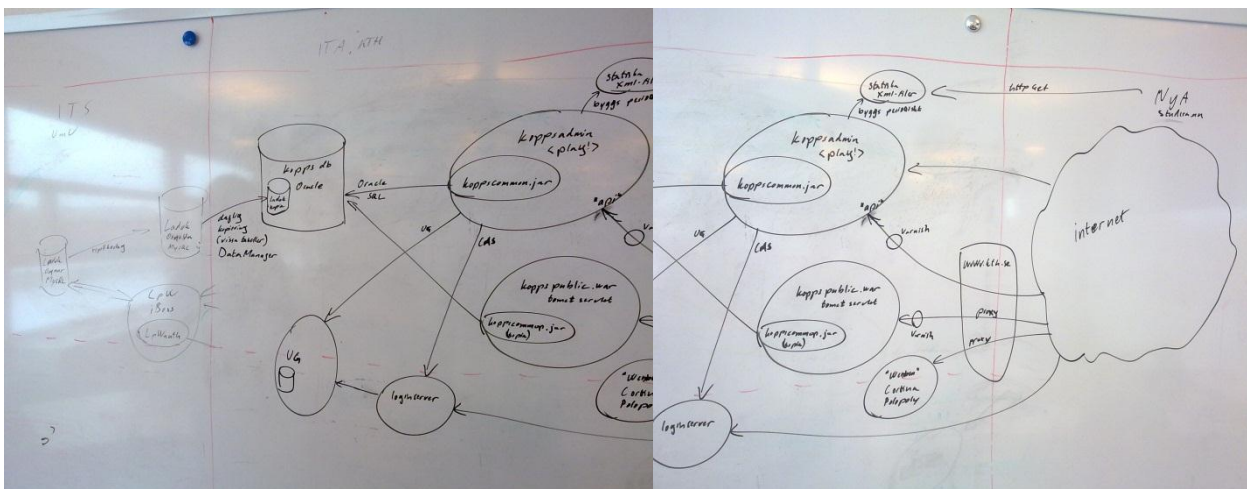


## 18.2 WORKSHOP

## Mina Sidor



**KOPPS**



KOPPS: Koppsadmin – UG; HTTP protokoll. = stamp, control

KOPPS: KOPPS databas – koppsadmin; SQL. = stamp, common

KOPPS: Koppsadmin – loginserver; CAS = data, control

KOPPS: KOPPS: koppsadmin – Hubble; XML filer = stamp, control

KOPPS: KoppsPublic – koppsdatabas; SQL. = stamp, common

KOPPS: DataManager – KOPPS Databas; pratar Oracle = stamp, common

KOPPS: DataManager – LADOK; pratar MySQL = stamp, common

Varnish är en transparent HTTP proxy/cache, både kopsadmin och KoppsPublic fungerar utan varnish, så jag vet inte hur relevant

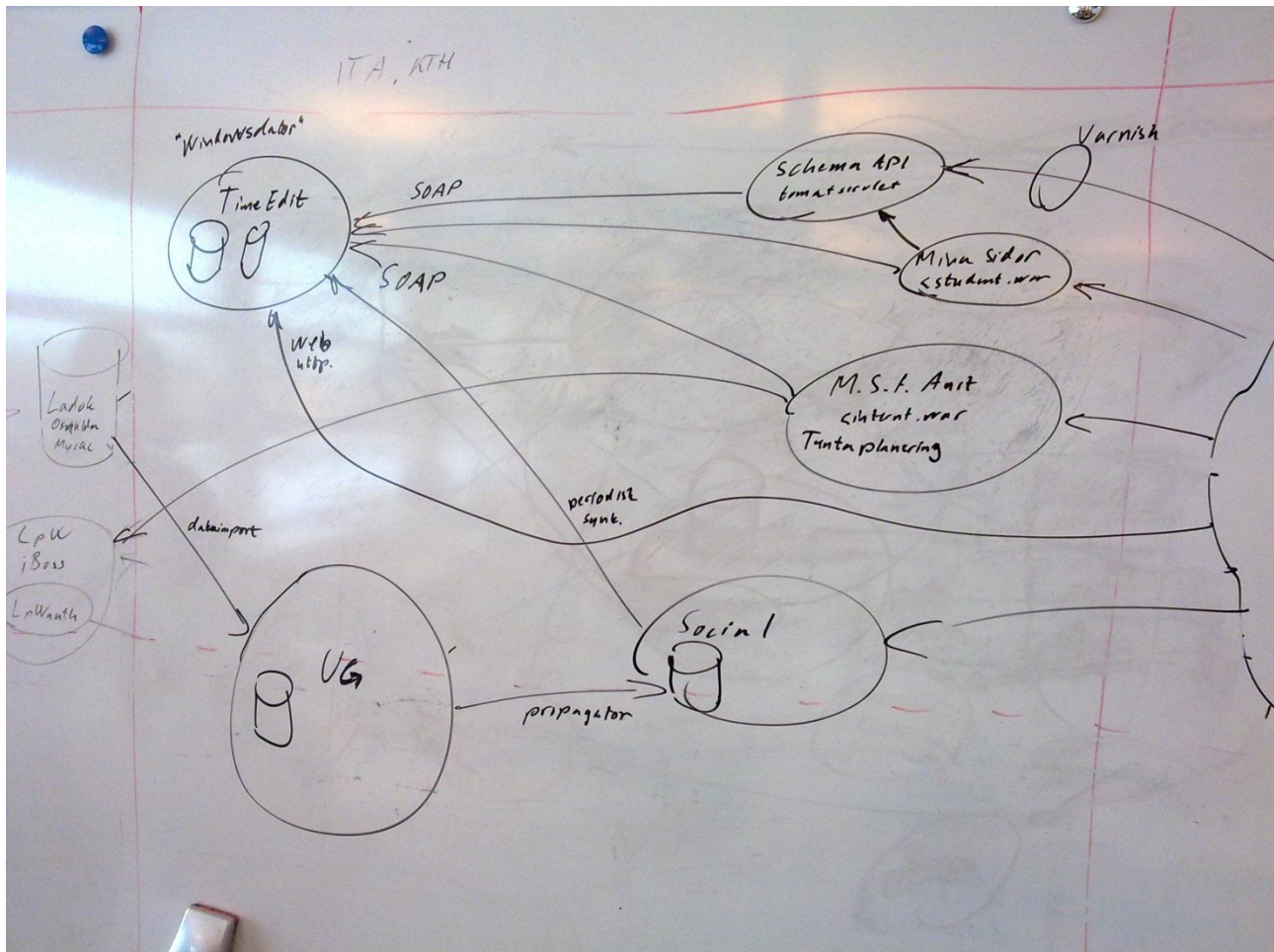


klassificeringen blir, men drar till med ngt i alla fall.

KOPPS: Varnish – KoppsPublic = data, common

KOPPS: Varnish – Koppsadmin = data, common

### TimeEdit (API)



There was a workshop with a developer at ITA at KTH, where he drew models of the different systems at KTH; KOPPS, Mina Sidor & TimeEdit.

### 18.3 INTERVIEW NOTES

#### KOPPS – FRÅGOR

[1] Endast KTH? Eller ett system för flera skolor?

Endast KTH.

Utvecklat av KTH? Eller endast delutvecklare?

KTH

Support? Ägare?

Antagligen KTH.

#### Processer(Business)

Vilka använder KOPPS? VoS? Framtiden?

KTH skolorna använder KOPPS, VoS utvecklar och administrerar.

I vilket samband använder man KOPPS? Bara programkatalog sökande? Syftet?

Inlägg av kurser och program via LADOK.

Används också som schemaunderlag till TimeEdit.

Används då endast programkatalog uppdaterande?

JA.

Används KOPPS ofta i samband med andra system? T ex mina sidor?

Inte processmässigt, men finns några tekniska samband.

Om isf hur då? Arbetsprocesser?

[1] Detta olika beroende på olika skolor?

#### Applikation

Vilka system hämtar den information ifrån? Endast LADOK?

LADOK

Program, programplaner, kurser(provuppsättning, kursplan, kursinfo), läsårsplaner.

Detta hämtas ifrån LADOK och läggs in i KOPPS databasen, via t ex kurskod.(1 dag roundtrip)

Vilka system skaffar den information åt? Tjänster? Typ VHS eller studera.nu?

NyA, antagningen.se, studera.nu(jämförelse) > Samma fil skickas till dessa system.

Convertus för översättning av kursplaner.

Mina sidor har en koppling för kurser av periodöversikt.

CSN utnyttjar kursinformation för deras räkning.

Innebörden av att Administrera kopps, uppdatera programkatalogen?

JA.

Polopoly endast för webbhanteringen?

JA, endast utseende för läswebben.

Tillgång åt KOPPS, lokal klient skillnad ifrån webb delen? Tillgång via webben också(Admin)?

"Lokal klient", webbaserad klient för adminstration. Tillgänglig via VPN.

"Webben" är den läsvänliga webdelen på KTHs hemsida om kurser.

#### **Tekniska:**

Data skickas via XML.

Daglig uppdatering för allting.

De olika databaserna, en för replikering eller olika funktionalitet?

Databas hanteraren endast för att uppdatera KOPPS databaserna med LADOK info?

Beroenden mellan systemen? KOPPS av ladok, vilka av KOPPS? (NyA)

NyA innan KOPPS, LADOK? SHB?

**Övriga frågor:**

Hur ser framtiden ut för KOPPS? Vad är tanken med det?

Utvecklas konternuerligt, ser inget närliggande slut på utvecklingen.

Noteringar:

Har egen testmiljö mot LADOK test.

Antagningsenheten(1 tr ned) har en koppling till kopps.(NyA?)

NyA använder (kurs) informationen i KOPPS för att bygga modeller.

De olika behörighetsnivåerna:

Lägst → Högst (Högre behörighet kan göra allt de lägre kan, topp tar hand om behörighet)

Kursredaktörer, programredaktörer, centralredaktörer(VoS personal), Systemredaktör(Thomas, utvecklarna)

**NYA – KTH FRÅGOR****Förfrågor**

Vilka är utvecklare/drift?

VHS anlitat ITS i Umeå. Kravställt av alla lärosäten.

Vem är systemägare?

VHS

**Processer**

Vilka är det som använder NyA systemet?

Alla universitet och högskolor. Myndigheter, polis och myndigheten för beredskap. HSV.

I vilka samband används systemet?(Processer) Främst i vilka syften?

Antagning, avgiftstatus, anmälan och urval.

Hur används det av VoS? Om det går att svara på?

Används till statistik, även koppling till VIS och KOPPS. Thomas fixar xml för import till NyA.

Används systemet i samband med några andra system?(Processmässigt)

LADOK för avgiftstatus.

I sådana fall hur då?

Varierar detta på andra skolor?

ERAMUS MUNDUS (LADOK) Meritvärdering hos andra skolor kanske med något annat system.

#### **Teknisk:**

Finns det möjlighet för tillgång till information angående mer systemspecifika detaljer när det gäller det tekniska? (Cyclomatic Complexity, Lines of Code)

Prata med utvecklarna isf.

Har också en koppling till UG, en annan till Migrationsverket.

#### **ALUMNI COMMUNITY**

#### **Förfrågor**

Endast KTH? Eller även andra skolor?[1]

Ja, ett eget gränssnitt, huvudsystemet är utvecklat av MIRA och anpassat till KTH, finns andra gränssnitt till samma huvudsystem hos andra skolor.

Vilka är det som utvecklar och driver Alumni?

MIRA

För mer teknisk information om systemet, om du inte kan svara på det, vem ska jag prata med då?

Martin@mira.se?

Systemägare?

KTH äger systemet som de använder (gränssnittet).

## Processer

Vilka är det som använder Alumni systemet?

Alumner, personal och student.

Även indelat i kategorier: Administratör, fundraising, CRM och reklam.

I vilka samband används systemet?(Processer) Främst i vilka syften?

Hålla kontakter med alumner samt de med sig själva. I examen samband ifrån kth. Den tekniska termen är att lämna lärosätet, så prat om att även ta med folk som läst enstaka kurser på kth.

Hur används det av VoS? Om det går att svara på?

Används inte av VoS, Elisiv(och Ida) var med och kravställde däremot tillsammans med Mikael. Elisiv var med och pratade om vilka tabeller som behövdes ifrån LADOK.

Används systemet i samband med några andra system?(Processmässigt)

Nej.

Isf hur då?

Varierar detta på andra skolor?[1]

Behörigheten kommer kunna variera på andra KTH skolor, om det är personal där som behöver lägga in studenter som tagit examen(på något mer speciellt sätt).

## System

Vad för system hämtar Alumni information ifrån? LADOK?

Endast LADOK, även också den interna MIRA databasen.

Vad för system hämtar information ifrån Alumni? Tjänster? T ex VHS eller SCB?

Inga andra kopplingar.

Vad är innebörden av att använda alumni? Dvs hur påverkar det andra system, om det nu gör det?

Tillgångar till alumni? Interface? Webbklient och/eller applikation?

Webbgränssnitt för både studenter och alumner. Ett administrativt webbgränssnitt för administratörer(Adela och Emma).

### **Teknisk**

Databaser som används? Bara LADOK? Egen databas?

MIRA interna db och även LADOK.

De tekniska kopplingarna mellan olika systemen nämnda tidigare?

Beroende av något annat system?

LADOK

Tekniska komponenter som bygger upp systemet?

Webbdelen? Lager och systemmodell?

Applikationsdelen? Systemmodell?

Vad är det för bakliggande system och/eller miljö?

Skulle jag kunna få tillgång till information angående mer systemspecifika detaljer när det gäller det tekniska? (Complexity, LoC & coupling)

**Övriga frågor:**

Framtiden för alumni? Några utökade funktioner eller processer?

Bättre funktionalitet, nytt system så förväntar sig tillväxt, t ex facebook integrering.

UG koppling?

Egna noteringar:

CAS för login. (UG?)

Information om konton sparas på egen databas hos leverantören, MIRA.

Används också som Jobbportal för företag, annonserar tjänster.

Tillgängligt via [www.kth.se/alumni](http://www.kth.se/alumni)

Tidigare koppling till SPAR systemet för adresser inom Sverige.

Uppdatering till LADOK sker en gång om dagen.

**PROCESSER**

Prata med Elisiv. Check

Process till Process?

**Studentprocessen**

NyA? Preliminärt Ja: Antagning.

TimeEdit? Preliminärt Ja: Föreläsningsplanering.

Ping? Preliminärt Ja: Nationellt intyg.

Mina Sidor? Preliminärt Ja: Studentens information ifrån ladok.

KOPPS? Preliminärt Ja: Studentens information om kurser (antagning).

Alumni? Preliminärt Ja: Examen.

DIAK? Preliminärt Ja: Internationella studenter.



**Kursprocessen**

NyA?

TimeEdit? Preliminärt ja, schemaunderlag?

Ping? Ja, ladok. Men används inte lokalt på kth.

Mina Sidor?

KOPPS? Preliminärt Ja. 2005

Alumni? Ja.

DIAK?

**Programprocessen**

NyA?

Ping?

Mina Sidor?

KOPPS? Preliminärt Ja. 2005

Alumni?

DIAK? Preliminärt Ja. 2005

TimeEdit? fortfarande en del av programprocessen? 2005 Osäker. Nej beroende av kursprocessen.

**Schemaläggning**

NyA?

TimeEdit? Preliminärt Ja. 2005

Ping?

Mina Sidor?

KOPPS? Schemaunderlag? 2005

Alumni?

DIAK?

Noteringar:

Ping och studentprocessen är ej knytta, utan nationellt intyg används via Mina Sidor, som utnyttjar Ping.

Ping är generellt inte använt av någon process(av de utvalda), utan används bara av andra system. Kan användas av studievägledare under tillgodoräknande (studentprocessen).

Kurs och programprocessen är relaterade, program kan inte existera utan kurs, så beroende. Kurs kan existera utan program dock. Kursprocessen är också relaterad till schemalägningsprocessen i att scheman inte finns utan kurser som går.

Studievägledning del av studentprocessen? Om isf kan LadokPing tillhöra en process.

#### 18.4 TELEPHONE INTERVIEW NOTES

Utvecklas i .Net 2.0 och 3.5

Arkitekturen är ganska enkelt, det är en IIS webserver(7.5) som driver hela applikationspoolen med komponenterna som realiserar systemet. En av dessa komponenter är databaslagret som är SQL till en Maxus databas (2005 till 2008).

Det är fyra mindre komponenter som skapar skapar webbkomponenter som realiserar systemet. De andra komponenterna förutom databaslagret är webbkomponenten, teleric(teleric.com) och aspnetemail(email funktioner, använder inbyggda smtp servern). Dessa skapar själva huvudkomponenten kan man säga vilket realiserar systemet. Storleken på den här komponenten är en cirka ~~77~~ tusen rader. KTH lösningen använder inte allt av vad lösningen har utan bara den delen som är skräddarsydd för KTH, då produkten är generell och säljs till många andra högskolor och universitet.

Det är inte Mira som driftar själva systemet utan DGC som är en hosting partner, de har en så som jag förstod det garanti att hålla systemet uppe hela tiden och byter endast kluster om servern går ned. Driftas på en windows 2008 server.

Det finns en CAS koppling, så att CAS inloggning fungerar.(UG)?

Ladok kopplingen består av en del SQL stored procedures in deras SQL databas, där informationen hämtas och skrivs(?) ifrån LADOK beroende på informationen som krävs(och de har tillgång till).

Det finns också en koppling till Google maps, som utnyttjas för koordination av alumner och visning av kartor. De skulle kunna använda andra karttjänster men finner att google är lätt att använda.

## 18.5 ESTIMATIONS

These are the estimations done where information was not obtained.

Coupling	Estimation	Reason
Noveau: Javabatch – Urouter	Control	Initiering av JavaBatchar insinuerar någon form av kontroll över batcharna.
Noveau: Urouter - Klienten	Stamp, control	Klienten använder sig av kontroll som input och får ut data som den inte använder helt, därav stamp.
Noveau: Userver - Urouter	Stamp	En massa data skickas imellan userver och urouter, antagligen används inte all data som skickas, därav stamp.
Noveau: Userver - LADOK	Common	En databas koppling, därmed en common koppling.

Coupling	Estimation	Reason
Ping: Ladokping - Ping.war HTTP Spring	38 Stamp Couplings	Coupling: 38 andra högskolor, antagande: 38 stamp, då det är antaget att LadokPing hämtar mer information än vad som används för intyget. Alternativt data då den anropar en egen instans av sig själv och borde veta exakt vad som behövs.
Ping: LpW - Ping.war SOAP/SSL	4 Control; 4 Stamp	LpW dokumentationen beskriver totalt 4 metod anrop till LadokPing ifrån LpW. Antagande: 4 control input, och fyra stamp output, då det är antaget att inte all data som hämtas används.
Ping: LADOK - Ping.war JDBC	1 Common	Antagande: 1 common coupling till databasen.

Coupling	Estimation	Reason
Alumni: Webcomponents - Telerik	Control;Data	Telerik erbjuder tre komponenter som är tillägnade för ASP.NET och klassificeras som webbkontroller på telerik hemsidan. MVC, AJAX eller Silverlight. I varje fall så består det av en produkt som implementeras via referenser (Bibliotek). Exakt hur de implementeras eller vilket bibliotek som används är okänt men antagande om alla tre kan göras. Komponenterna är produktlösningar, vilket innebär att kunden inte har tillgång till hur modulen är konstruerad, i och med detta så måste det vara stamp eller data. Då de andra tre kräver kunskap om logiken (i.e. white box). Specifika instruktioner om användning och implementering av modulen brukar ingå i produktlösningen, därmed kan man utgå ifrån att modulen endast får den data den behöver, i.e. data koppling. Men informationen

		förespråkar någon form av kontroll i modulen, därav också control.
Alumni: Webbkomponenter - Aspose	Control;Data	Samma som innan.
Alumni: Webbkomponenter - aspnetemail	Control;Data	Samma som innan.
Alumni: Webbkomponenter - Loginserver	Control;Data	Antagande: Control, data, samma som loginserverkoppling för KOPPS och Mina Sidor
Alumni: Webbkomponenter - Alumni databas	Common	Antagande: Common coupling, då det är en databas koppling.
Alumni: Webbkomponenter - GoogleMaps	Data	Endast ett API som används för att ta fram koordinater, därav data.

Coupling	Estimation	Reason
TE: Global - Lokal	Control	En koppling beskrivs i TimeEdit pärmén. Antagligen system eller databas relaterad.

Coupling	Estimation	Reason
NyA: transferarea - LA-DOK(Osquar)	Control;Stamp	Överföringen av antagna studenter till Ladok centralerna.
NyA: NyAOpen DB - dbdoc, DB2	Common;Stamp	En databaskoppling innebär att den ena modulen har en förståelse för hur databasen ser ut och använder sig av denna globala yta för att prata: Common. Datan som skickas används antagligen inte helt: Data.
NyA: Urval DB - dbdoc, DB2	Common;Stamp	Samma som ovan.
NyA: Urval DB - batch, DB2	Common;Stamp	Samma som ovan.
NyA: Studera DB - Hubble, DB2	Common;Stamp	Samma som ovan.
NyA: Studera DB - dbdoc, DB2	Common;Stamp	Samma som ovan.
NyA: Studera DB - batch, DB2	Common;Stamp	Samma som ovan.
NyA: Studera DB - WAS Applicants, DB2	Common;Stamp	Samma som ovan.
NyA: NyA DB - Urval DB, DB2	Common;Stamp	Samma som ovan.
NyA: NyA DB - Studera DB, DB2	Common;Stamp	Samma som ovan.
NyA: NyA DB - dbdoc, DB2	Common;Stamp	Samma som ovan.
NyA: NyA DB - NyAQueueCache, DB2	Common;Stamp	Samma som ovan.
NyA: NyA DB - batch, DB2	Common;Stamp	Samma som ovan.
NyA: NyA DB - WAS Applicants,	Common;Stamp	Samma som ovan.

DB2		
NyA: NyA DB - WAS Experts, DB2	Common;Stamp	Samma som ovan.
NyA: dw-idp - dw-ldap, LDAP	Control;Stamp	Enligt mailet skickat så används LDAP till en massa olika funktioner, dock inte specificerat exakt hur men den verkar ha ett kontrollerande syfte, därmed antagande: Control. Datan som skickas antas inte användas helt: Stamp
NyA: Shibboleth - LDAP Applicants, LDAP	Control;Stamp	Samma som ovan.
NyA: WAS Experts - LDAP Applicants, LDAP	Control;Stamp	Samma som ovan.
NyA: WAS Applicants - LDAP Applicants, LDAP	Control;Stamp	Samma som ovan.
NyA: WAS Experts - LDAP Experts, LDAP	Control;Stamp	Samma som ovan.
NyA: transfermonitor - transfer-area, SSH	Control;Stamp	Transfermonitor övervakar transferarea och kontrollerar aktiviteterna. Antagande: Control, stamp
NyA: batchroot - transfermonitor, NFS	Common	Antagande: common, NFS innebär användande delande av samma globala filsystem, därmed common
NyA: batchroot - formattermonitor, NFS	Common	Se ovan om NFS.
NyA: batch - batchroot, NFS	Common	Se ovan om NFS.
NyA: WAS Experts - batchroot, NFS	Common	Se ovan om NFS.
NyA: dbdoc - HTTP Experts, SSH	Data	Databasdokumentationen, antagande att den bara läser information om databasen, antagande: Data
NyA: archiveserver - archivecache, FAM	Data	Antagande: Kommerciell tredjeparts produkt, därmed antas data koppling.
NyA: NyAQueueCache - WAS Experts, RMI	Control	Jcrontab schemalägger batchar i kön. Därmed Antagande: Control
NyA: NyAQueueCache - batch, RMI	Control	NyAQueueCache är kön för batchar, antagande: control
NyA: WAS Experts - NDM, Network Deployment Management	Control	Utan att i större del sätta sig in i exakt hur NDM fungerar, men det används för att administrera och hantera WAS klusterna, och antas därför vara: Control
NyA: NyAQueueCache - NDM, Network Deployment Management	Control	Se ovan om NDM.

NyA: WAS Applicants - NDM, Network Deployment Management	Control	Se ovan om NDM.
NyA: WAS Experts - HTTP Experts, HTTP	Control;Stamp	Expertklienten använder sig av SOAP tjänster för att komma åt logiken i was klustret, därmed ett antagande: control, stamp
NyA: WAS Experts - archiveserver, FAM	Data	Antagande: Kommerciell tredjeparts produkt, därmed antas data koppling.
NyA: WAS Experts - batch, RMI	Control;Stamp	WAS kallar på de olika batcharna i systemet, som schemaläggs av Jcrontab: Control. Data skickas som antagligen inte används helt: stamp.
NyA: Hubble - WAS Applicants, SSH	Stamp	Information om data som överförs ifrån olika lärosäten, antagligen används inte all data, antagande: stamp
NyA: WAS Applicants - Episerver, HTTP	Stamp	Informationen som presenteras ifrån databasen i Episerver, Antagande: Stamp koppling.

## 19 APPENDIX – P-OCL STATEMENTS

### SYSTEM.COUPLINGAVG

```
-- The sum of all Component.coupling realizing the System divided
by the number of Components

(self.realizedBy ->iterate( sc : Component ; sum : Real=0 | sum+
sc.couplingAVG ))/(self.realizedBy ->size())
```

### SYSTEM.COUPLINGMAX

```
self.realizedBy ->iterate( sc : Component ; max : Real=0 |
if sc.couplingMAX > max
then sc.couplingMAX
else max
endif
)
```

### CYCLOMATIC COMPLEXITY

```
-- Cyclomatic complecity by McCabe. Based on the number of Applica-
tionFunctions as nodes and class relations as edges. This implemen-
tation does consider relations to ApplicationFunctions outside of
the owning ApplicationProcess

(numberOfConnectionWithinModule()
+numberOfConnectionsToOtherModules() ) - (self.realizedBy -
>size() ) +2
```

### CYCLOMATIC DENSITY

```
if size<1000 then -1
else cyclomaticComplexity / (size /1000)
endif

-- CMAXDENS \cite{gill1991cyclomatic}. RLC
\cite{frappier1994software}
```

### SIZE

```
self.realizedBy .size ->sum()
```

### COMPONENT.COUPLINGAVG

```
self.communication ->iterate( con : ComponentPairConnection ;
max : Real=0 |
if con.CouplingInPair() > max
then con.CouplingInPair()
else max
endif
)
```

### COMPONENT.COUPLINGMAX

```
-- Fenton and Melton Software Metric as a sum fo all pair wise cou-
pling measures / number of pairs

(self.communication ->iterate( con : ComponentPairConnection ;
sum : Real=0 | sum+ con.CouplingInPair() ) ) / (self.communication
->size())
```

## COUPLINGINPAIR

```
-- Fenton and Melton Software Metric
let n : Integer =numberOfConnections()
in
getWorstType() + n/(n+1)
```

## GETWORSTTYPE

```
if R5_ContentCoupling >0 then 5
else if R4_CommonCoupling >0 then 4
else if R3_ControlCoupling >0 then 3
else if R2_StampCoupling >0 then 2
else 1
endif
endif
endif
endif
```

## NUMBEROFCONNECTIONS

```
R5_ContentCoupling + R4_CommonCoupling + R3_ControlCoupling +
R2_StampCoupling + R1_DataCoupling
```

## NUMBEROFCONNECTIONSWITHINMODULE

```
(self.realizedBy.communication->iterate( con : ComponentPairConnec-
tion ; sum : Real=0 |
if self.realizedBy.communication->count(con)=2 then sum+1
else sum
endif
)) /2
```

## NUMBEROFCONNECTIONSTOOTHERMODULE

```
(self.realizedBy.communication->iterate( con : ComponentPairConnec-
tion ; sum : Real=0 |
if self.realizedBy.communication->count(con)=1 then sum+1
else sum
endif
))
```

## SYSTEM.COMMUNICATES

```
self.realizedBy .communication .communicates ->asSet() .realizes -
>asBag()
```

## COMPONENT.INTERCOMPONENT

```
self.communication .communicates -> asBag()
```



20 APPENDIX - EAAT CALCULATION HISTOGRAMS

Nouveau

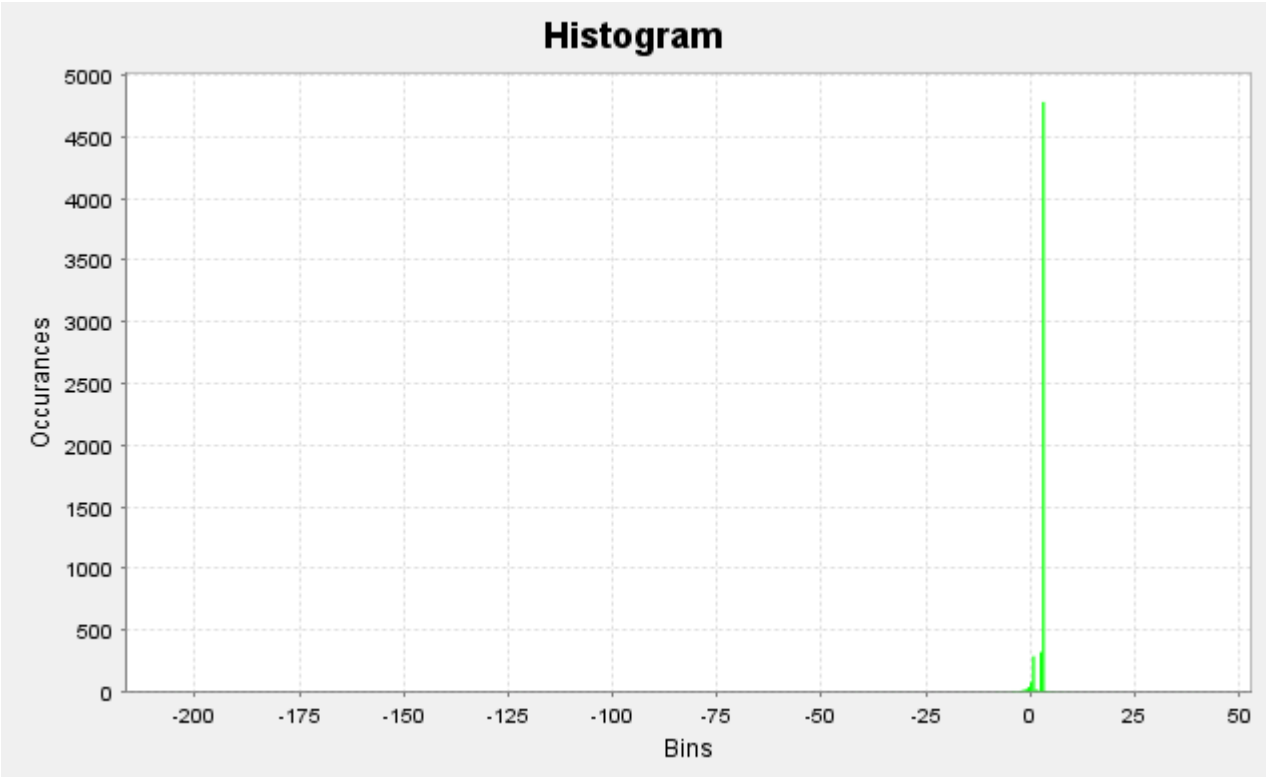


Figure 55: The Nouveau System CouplingAVG Histogram.

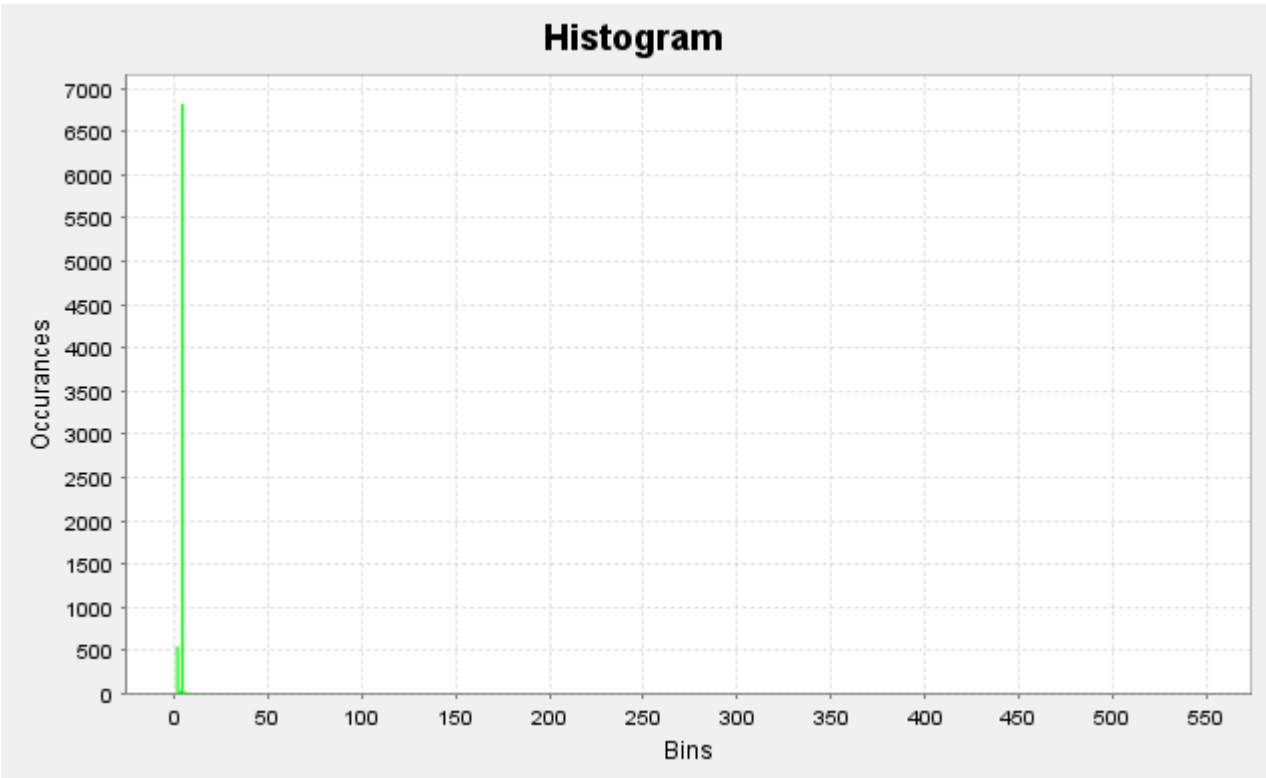


Figure 56: The Nouveau System CouplingMAX Histogram.

## Mina Sidor

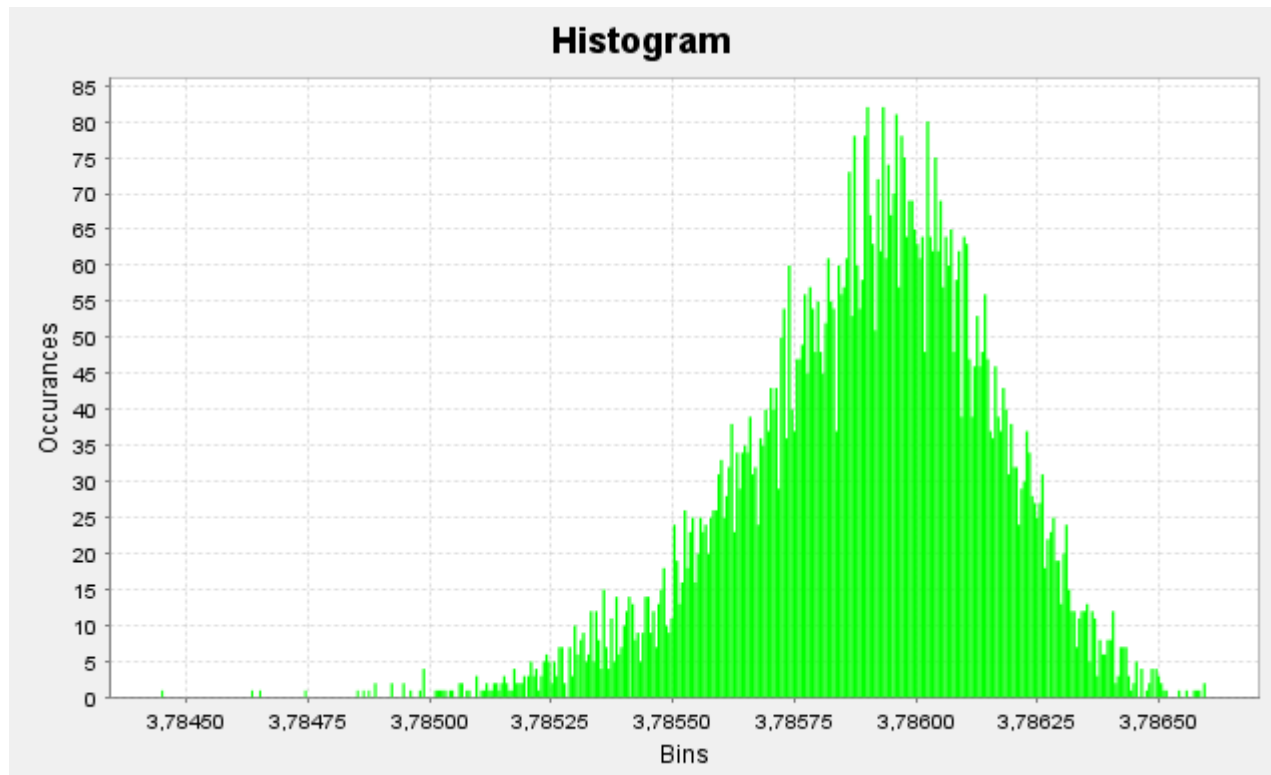


Figure 57: The Mina Sidor System CouplingAVG Histogram.

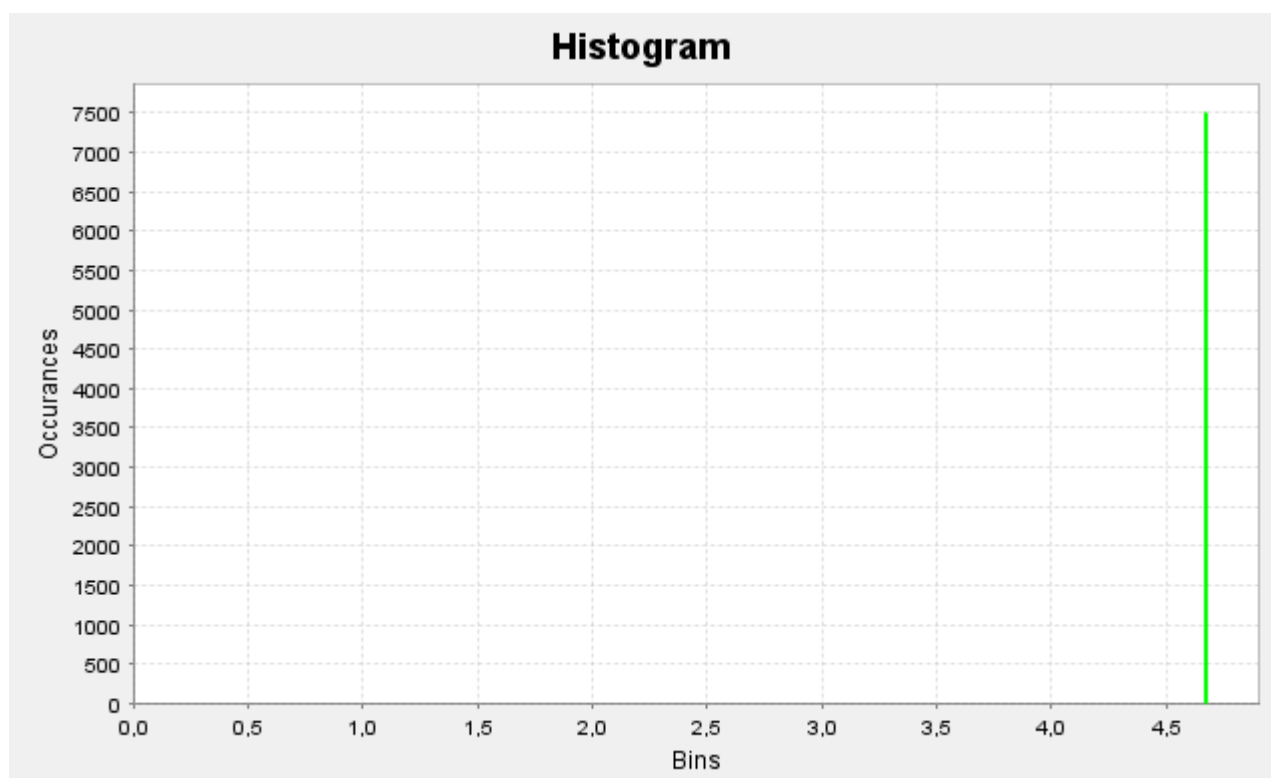


Figure 58: The Mina Sidor System CouplingMAX Histogram.

LpW

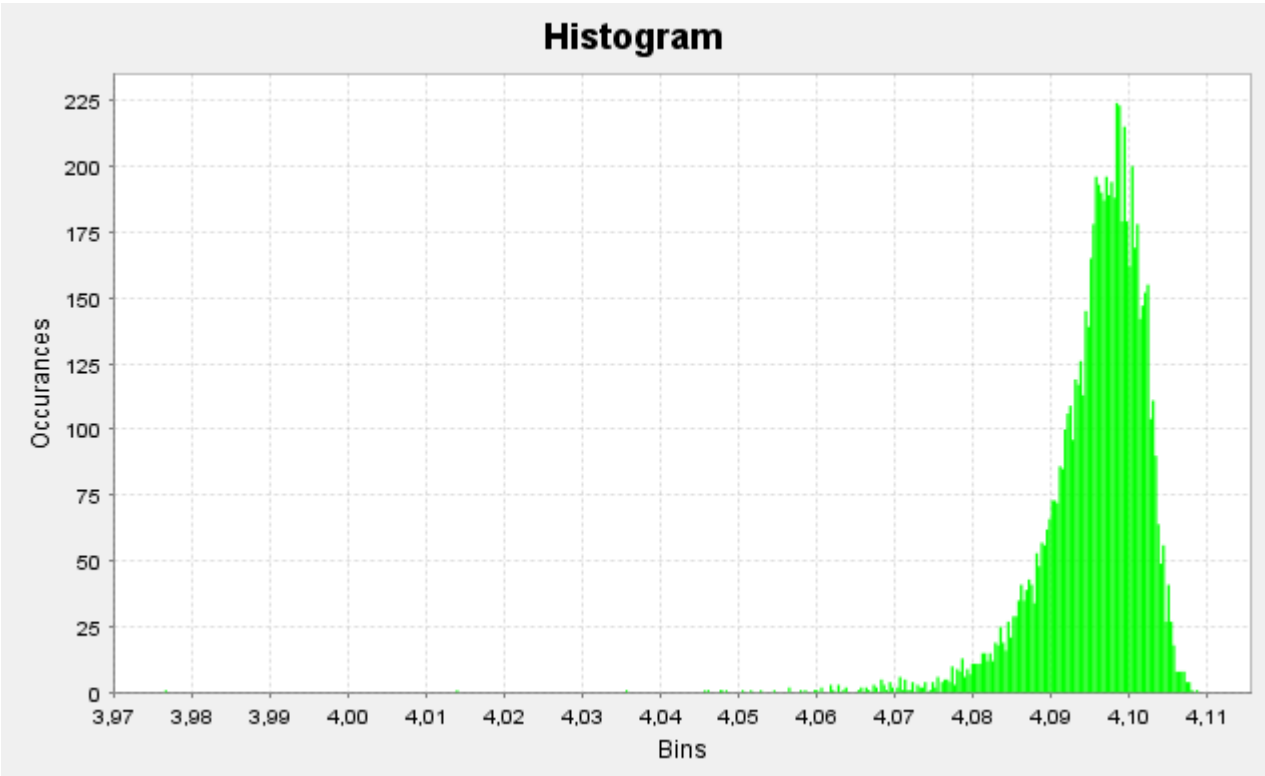


Figure 59: The LpW System CouplingAVG Histogram.

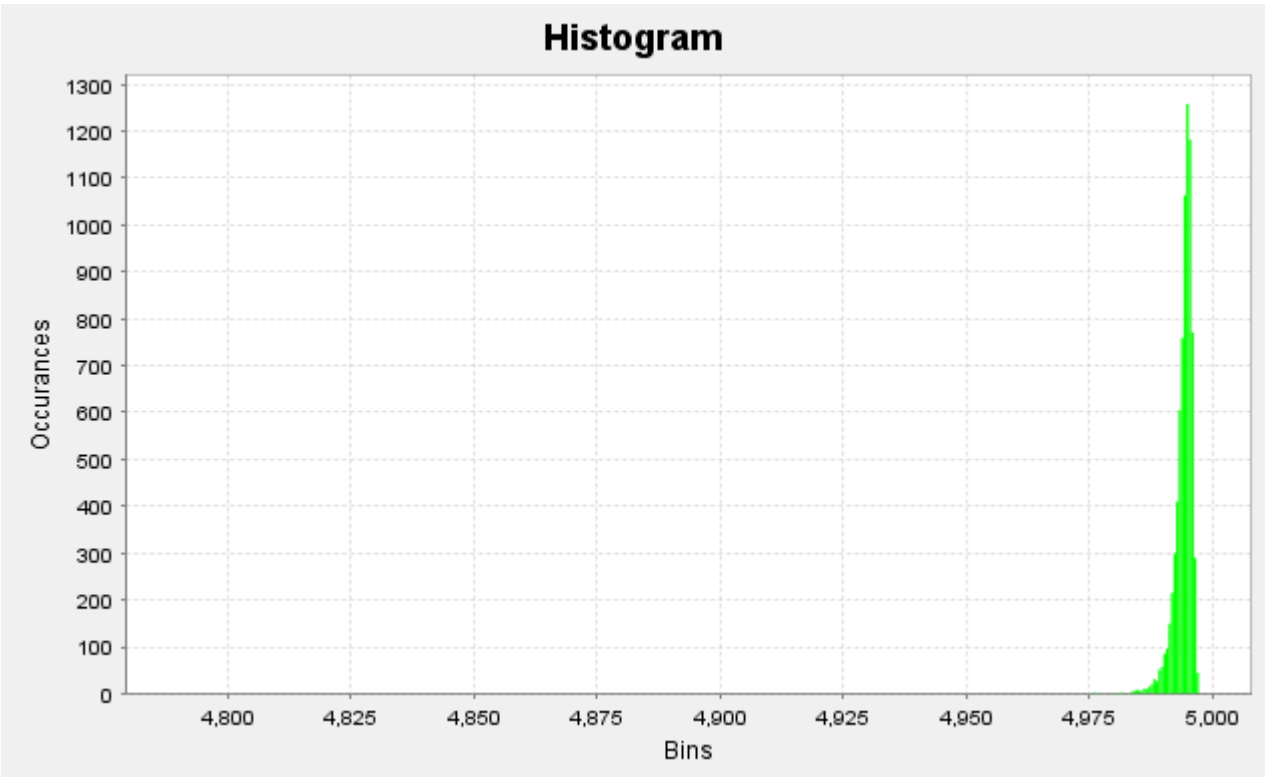


Figure 60: The LpW System CouplingMAX Histogram.

LadokPing

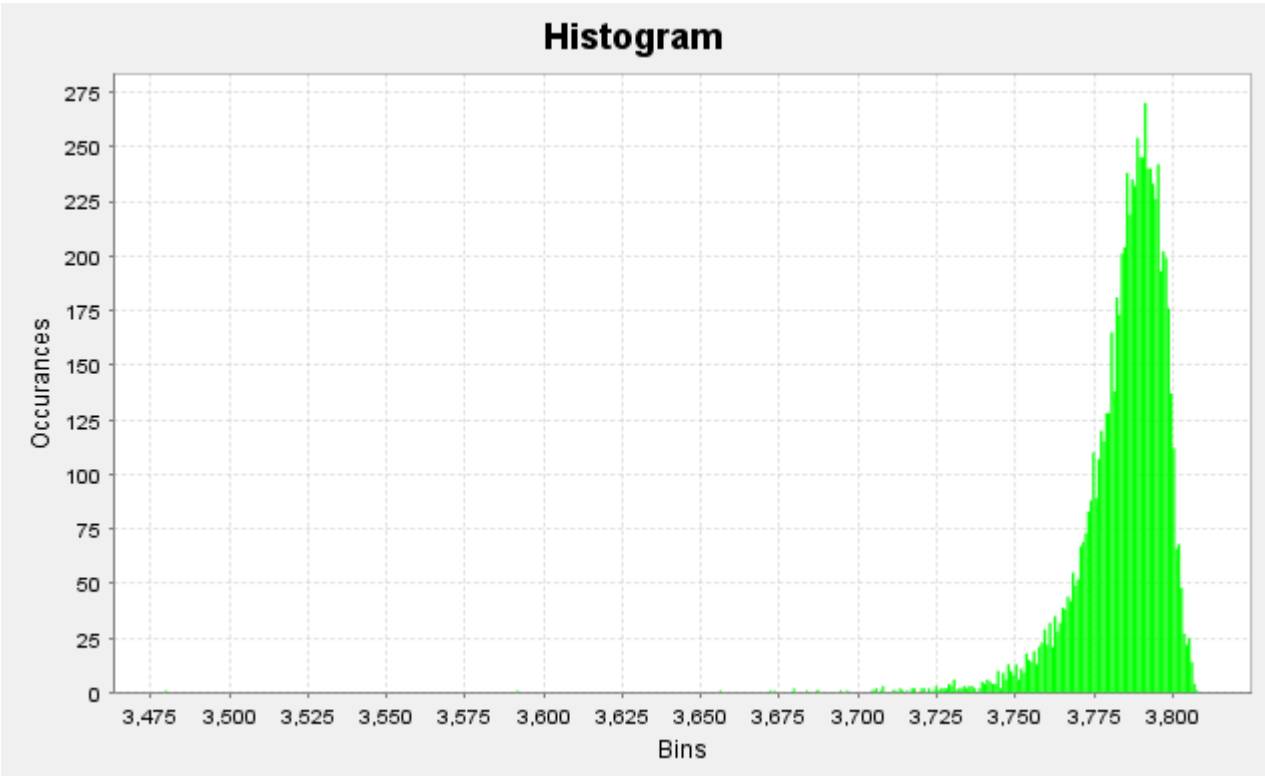


Figure 61: The LadokPing System CouplingAVG Histogram.

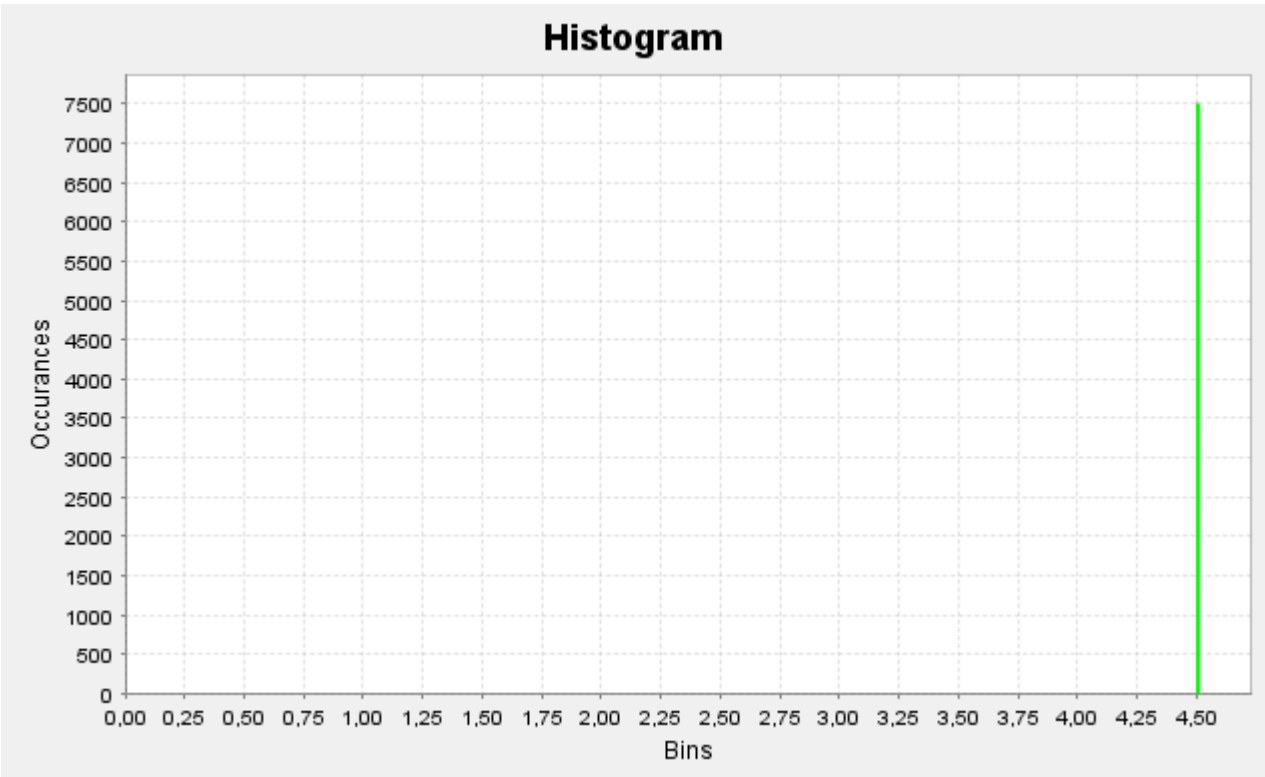


Figure 62: The LadokPing System CouplingMAX Histogram.

NyA

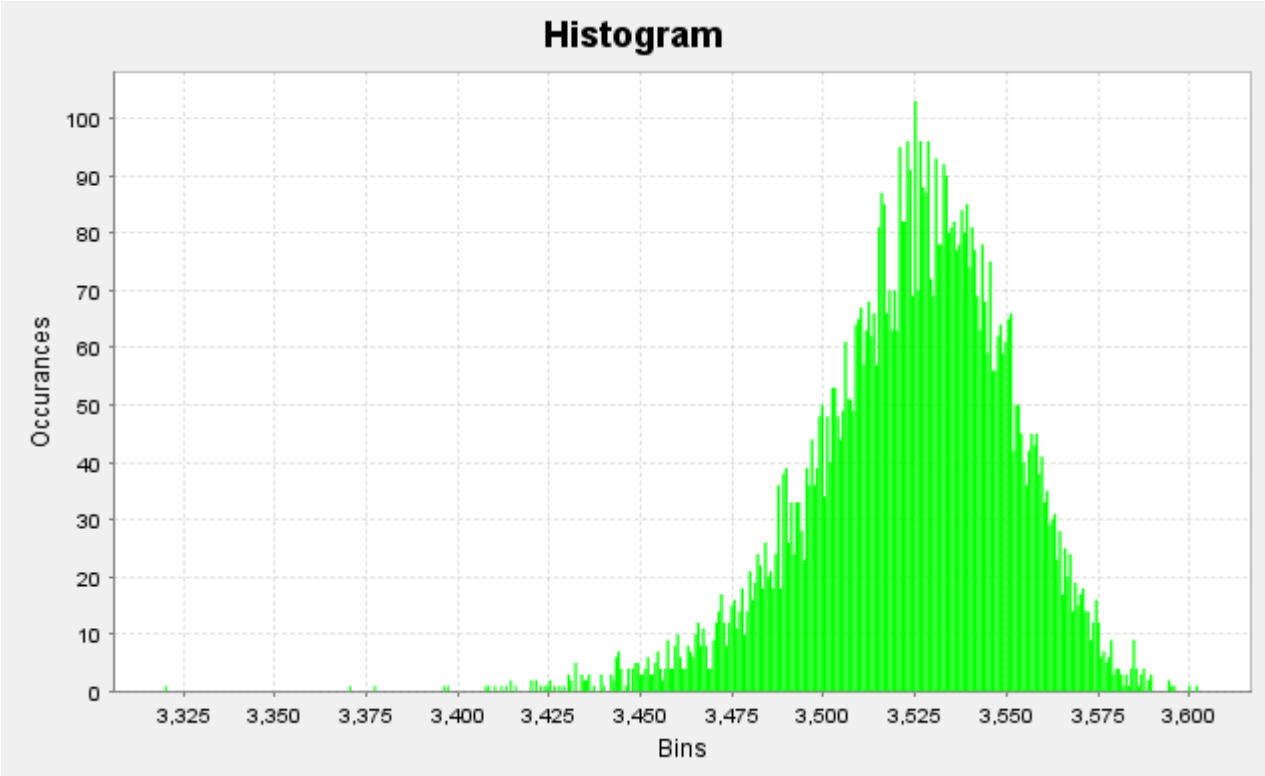


Figure 63: NyA System CouplingAVG Histogram.

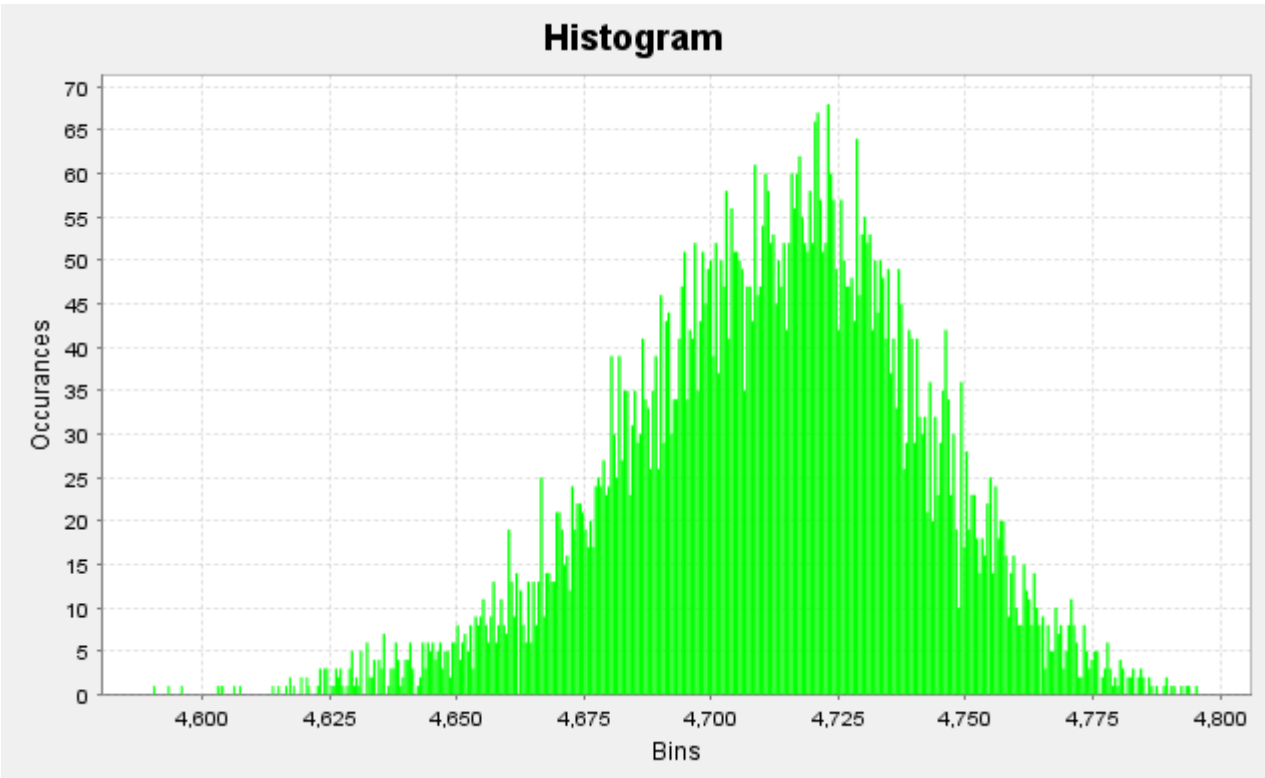


Figure 64: NyA System CouplingMAX Histogram.

Alumni

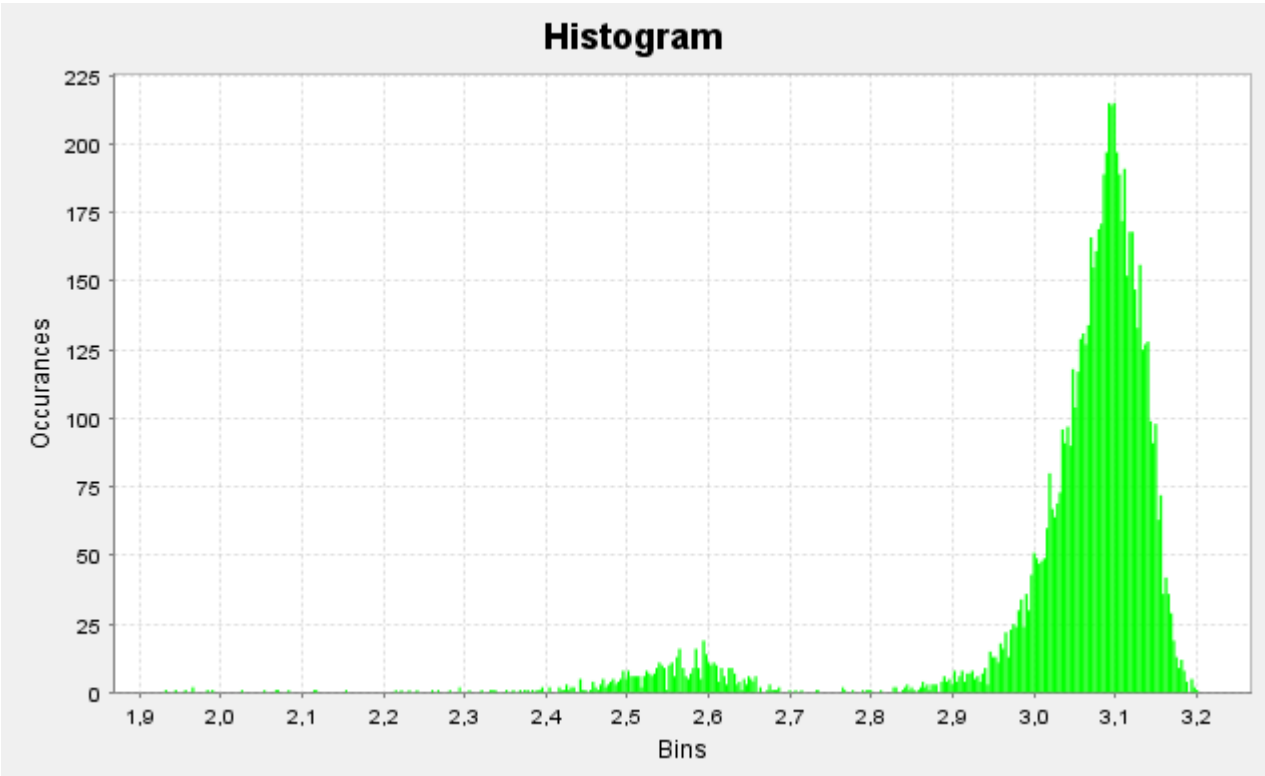


Figure 65: The Alumni Community System CouplingAVG Histogram.

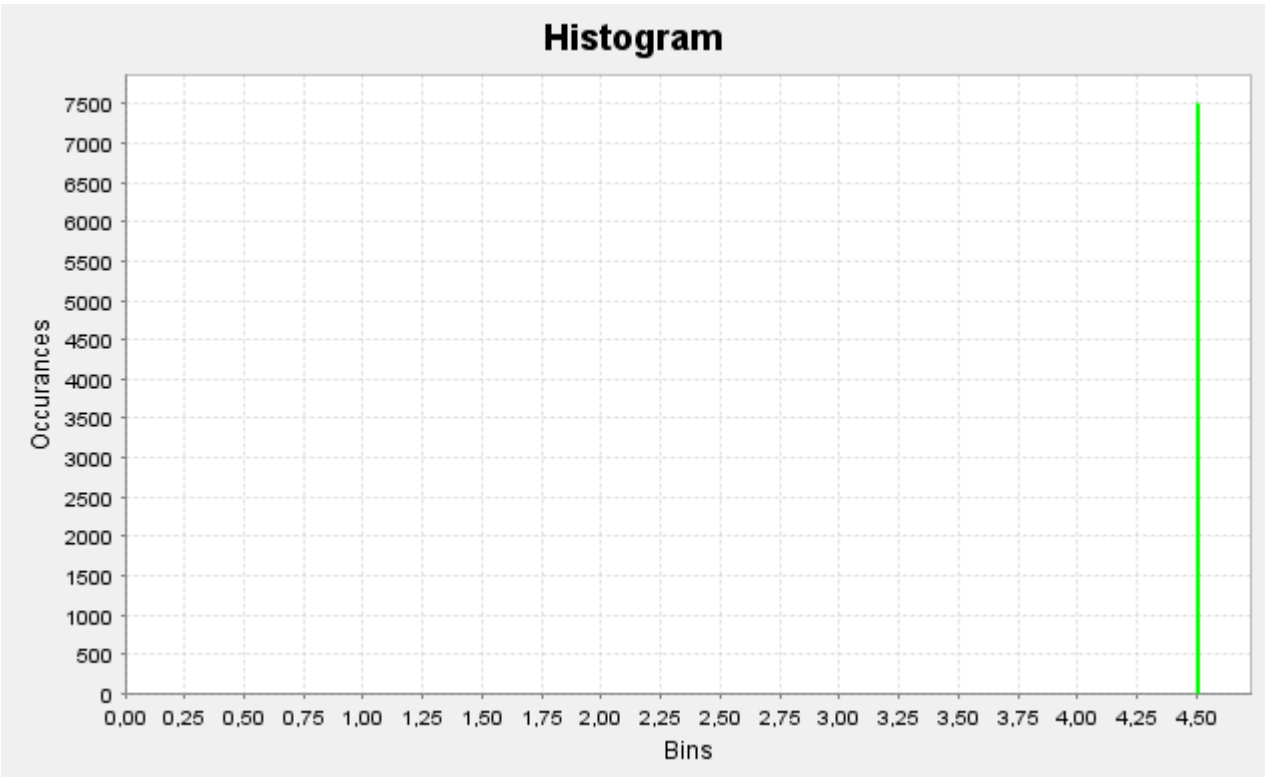


Figure 66: The Alumni Community System CouplingMAX Histogram.

TimeEdit

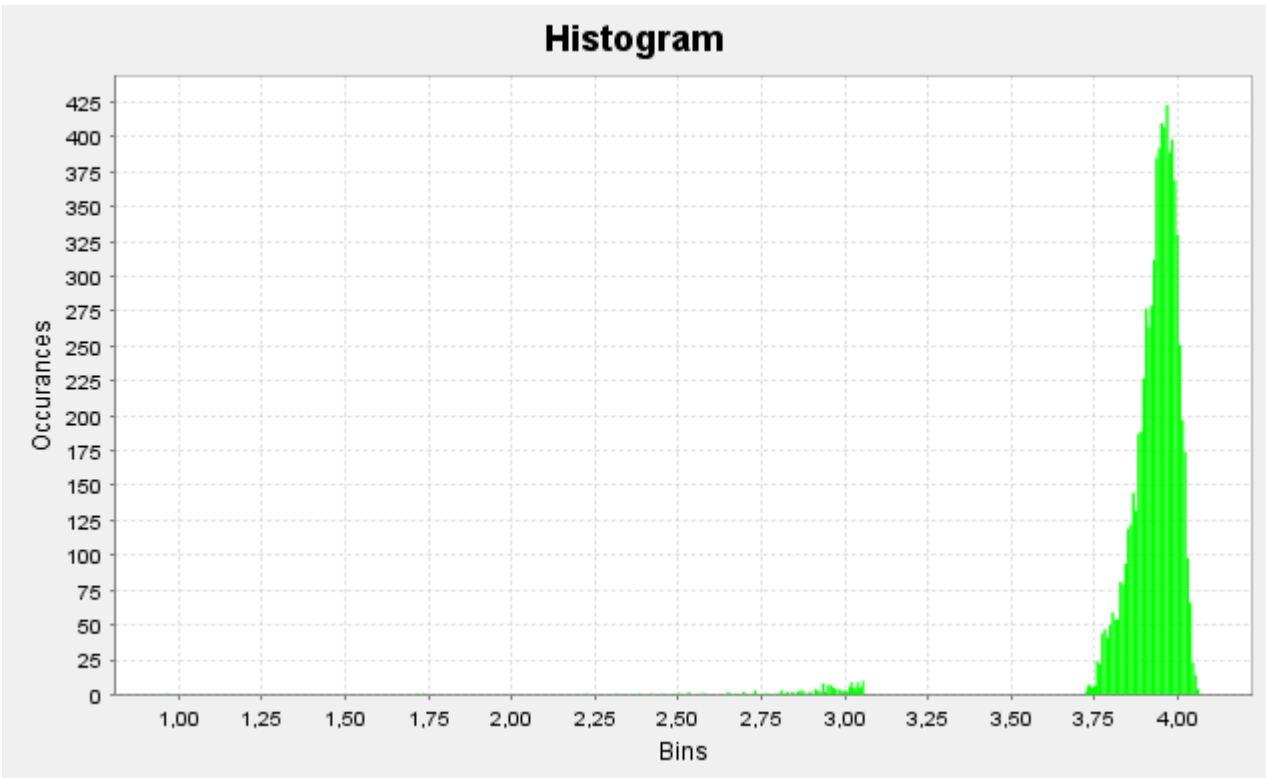


Figure 67: The TimeEdit System CouplingAVG Histogram.

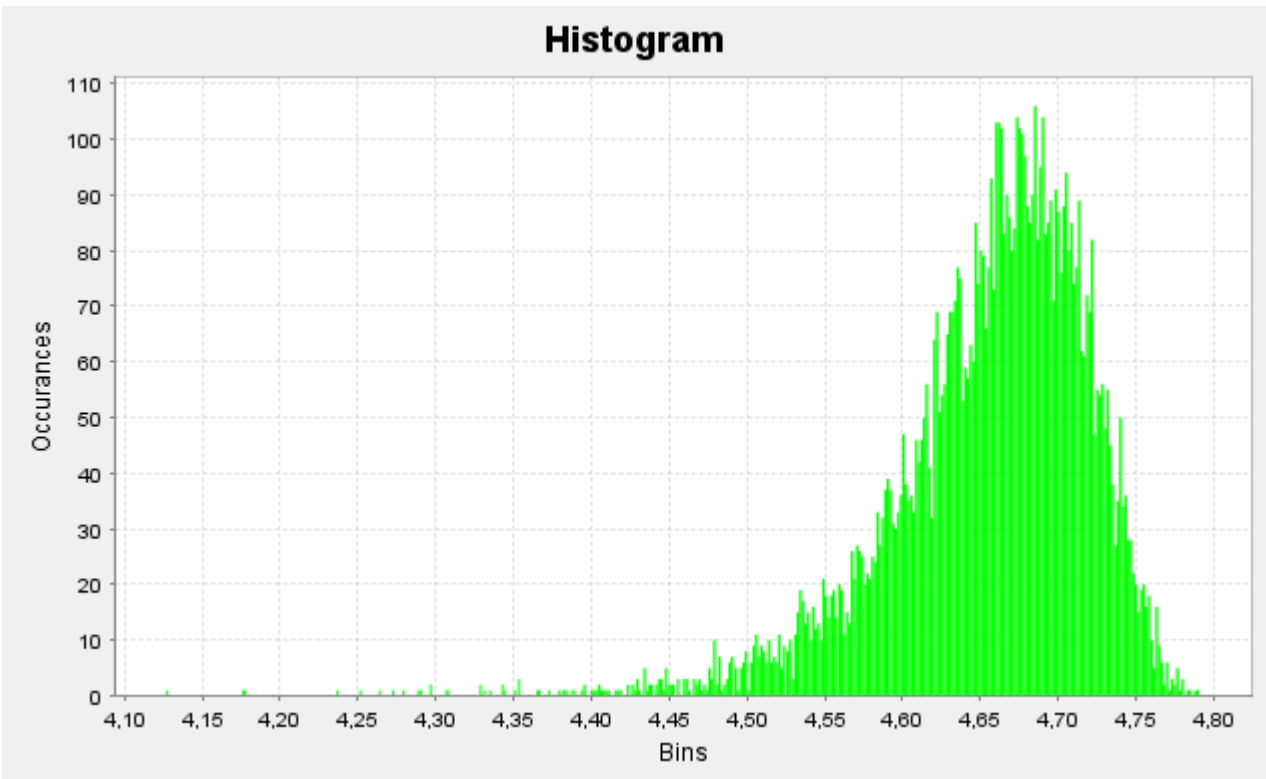


Figure 68: The TimeEdit System CouplingMAX Histogram.

DIAK

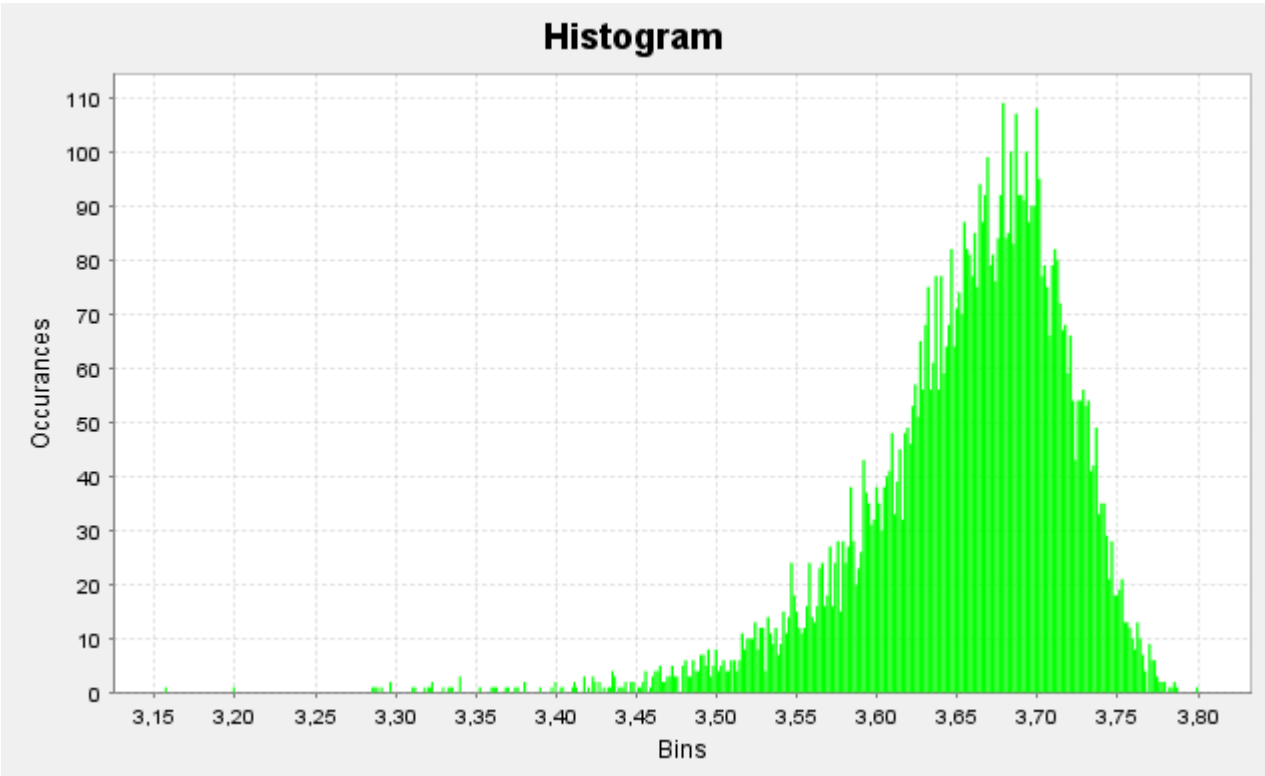


Figure 69: The DIAK System CouplingAVG Histogram.

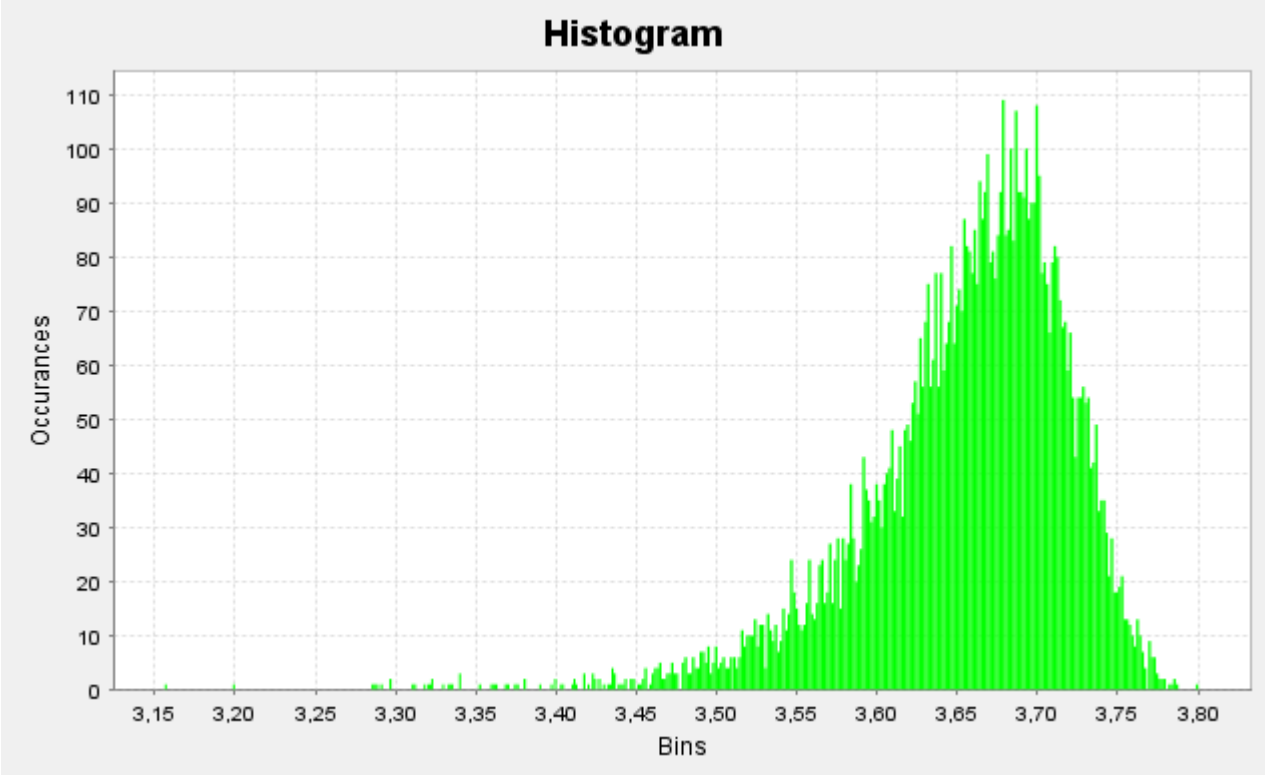


Figure 70: The DIAK System CouplingMAX Histogram.