Language assessment of an Interoperability Assessment Language

Johannes Edelstam

Degree project in Ind. Inf. and Control Systems

XR-EE-ICS 2011:021
This page is intentionally left blank.
ABSTRACT
In order to assure the usability, validity and reliability of an enterprise architecture analysis framework, it needs to be tested and evaluated. This report shows how such a test and evaluation can be performed using a real world scenario.

The study will investigate a language for interoperability assessment proposed by Johan Ullberg et. al. in a construction project in Stockholm called the Royal Seaport project. The language is first used to evaluate the interoperability in the future IT-architecture in this project. Then the use of the language is evaluated according to best practices in a case study.

Finally improvements and changes are proposed to the language, which would enhance its use in the earlier mentioned project.
# Table of Contents

- **Table of Contents**
- 1 **Background**
  - 1.1 **Stockholm Royal Seaport project**
  - 1.2 **Goals and purpose**
  - 1.3 **Disposition**
  - 1.4 **Acknowledgements**
- 2 **Method**
  - 2.1 **Validity & Reliability**
    - Construct validity
    - Internal validity
    - External validity
    - Reliability
  - 2.2 **Data collection**
    - Identifying Communication Needs and Actors
    - Interviews
    - Documentation
  - 2.3 **Pre-analysis**
  - 2.4 **Analysis**
    - Assessing construct validity
    - Assessing internal validity
    - Assessing external validity
  - 2.5 **Presentation**
  - 2.6 **Alternatives methods**
- 3 **Theory**
  - 3.1 **Using models**
  - 3.2 **Probabilistic relational models**
  - 3.3 **A modeling language for assessing interoperability**
- 4 **Data collection**
  - 4.1 **Prioritization**
  - 4.2 **Actors and communication needs**
  - 4.3 **Technical specifications regarding the systems and components**
    - Meters
IT-systems 27
Translations 27
4.4 Other aspects of communication 27
Legal 27
Public Relations 28
Political 28
4.5 Summary of findings 28
5 Pre-Analysis 28
Constructing the model 28
Conclusions 34
6 Analysis 34
6.1 Naming conventions 35
6.2 Probability of existence 35
6.3 Aspects of the language for interoperability assessment 36
6.4 New meta-model 40
6.5 Comparison 45
6.6 Import learning’s to the original language 47
6.7 Visualization 49
6.8 Other considerations and further research 51
Data influence on Transformation 51
Acceptable thresholds 52
More or less important Needs 52
Resolution 52
6.9 Case study principles fulfillment 52
Construct validity assessment 53
Internal validity assessment 53
Reliability assessment 53
7 Discussion 54
7.1 How the questions where answered 54
7.2 Improvement areas 54
7.3 Study design 55
8 Conclusion 55
9 REFERENCES 56
10 Appendicies 57
Appendix A. Questionnaire 57
1 Background

This master thesis aims in short to evaluate a specific language for modeling of interoperability. The language is developed by the supervisor of this thesis, Johan Ullberg, who is currently a PhD student at the department for Industrial Information and Control Systems (ICS) at KTH Royal Institute of Technology. The evaluation was done by using the language in a real world scenario; a construction project in Stockholm called “Stockholm Royal Seaport”.

1.1 Stockholm Royal Seaport project

The project aims to build a “climate positive” neighborhood in a part of Stockholm called Hjorthagen. To be able to do this, a research project has been initiated by the city of Stockholm with the goal to determine what criteria that must be fulfilled to be able to call a neighborhood climate positive and evaluate if the needed data can be collected, merged and presented to the inhabitants in real time.

The project is divided into several “Work Packages” with different responsibilities in the overall project. This study has been a part of Work Package 5 – “Royal Seaport Information Management System” which has the task to prepare for an information system that is able to collect the data needed in order to ensure that the neighborhood actually is climate positive. The vision is doing this by installing a lot of different meters in every apartment and building and then is able to collect these values and together with impact data then be able to calculate the consumers environmental impact.

The project also has the vision of giving the habitants feedback on their consumption patterns and also giving them the opportunity to tweak their behavior with help of these benchmarks.

To define what counts as climate positive one part of the project has been to identify “indicators” which are key figures that will be checked in order to measure how climate positive the neighborhood is. One indicator could be measuring the power consumption and then convert it to environmental impact using figures on how the power used was generated.

1.2 Goals and purpose

The main goal of the project was to evaluate the proposed modeling and assessment language for interoperability proposed by Ullberg et. al. The language is constructed using theory about “probabilistic relational models” and “meta models”. To make the evaluation as realistic as possible the modeling language was applied to a real world project, namely the “Royal Seaport” project. As stated earlier, that project’s goal is to construct a “climate positive” neighborhood in Stockholm. By using the modeling language for interoperability, this master thesis project has constructed a model of the different systems that need to interoperate with a future Information Management System (IMS) in order to be able to measure to what degree the
neighborhood is climate positive or not in real time. Therefore a sub goal of this master thesis project was to provide useful information as decision basis to what systems that need to interoperate, and how likely they are to, in order to measure climate positivity. The main purpose of this master thesis project is however to increase the validity of the proposed modeling language for interoperability. To sum up; this master thesis project will answer the question *How does the modeling and assessment language for interoperability proposed by Ullberg et. al. performs in the Stockholm Royal Seaport project, and can any changes or additions be made to the language to improve the models usefulness and/or accuracy?*

In order to answer this main question, some sub-questions had to be answered. The project can be divided into two main stages. One stage is the case study of the interoperability in the Royal Seaport. The case study will point out risk areas in the future interoperability. The next stage analyzed the first case study and evaluated the use of the modeling language for interoperability. Basically it is a case study of a case study, where both case studies will concern single cases and be conducted in the same master thesis project. The case study concerning the implementation of the language was aimed to answer the question *How interoperable is the Royal seaport likely to be?* When this question had a substantial answer in the form of an analyzed model constructed with the interoperability language, the next stage could proceed.

The stage concerned with analyzing of the first single case study had to answer the following sub-questions in order to be able to answer how well the language performed and how it can be improved:

*What other models/viewpoints could be used?*

*What would make this language easier/better to use?*

*Could the instantiation of the model been done any different?*

*Could the data collection been done any different?*

This master thesis project has mainly focused on the language evaluation, and secondly being focused on delivering results to the Royal Seaport project. The evaluation was made with respect to the scenario analyzed and does not focus on covering aspects of the modeling language that does not affect this scenario. Another limitation is that the analysis of the modeling language will concern this specific case, i.e. the generalization aspect of any proposed changes to the language will just briefly be covered. The study will aim to investigate the general validity of the language, as this study will aim to show if the language is not insufficient in the modeling of interoperability, as compared to showing that the language is sufficient in the general case. The most important shortcoming is probably that the case in question is a to-be case, not an as-is. This means that the systems that are analyzed do not exist yet, but some of their specifications. The modeling language on the other hand is designed to analyze as-is system architectures, why the lowest level of detail of the languages capabilities cannot be tested in a realistic way.
1.3 Disposition

The thesis is divided into Method, Theory, Pre-Analysis, Analysis, Discussion and Conclusion. The Method chapter will describe how the study was conducted and the thoughts behind the analysis. In the Theory chapter will terms and concepts be explained. The Pre-Analysis chapter is the usage of the interoperability language in practice while the actual Analysis chapter will analyze how well the language performed its task and compare its results to the goals in the previous section. The Discussion chapter will focus on weaknesses in the study itself, lessons learned and what could be done differently. The Conclusion chapter will sum up the study.

1.4 Abbreviations used

This is a list of some abbreviations used. The list is not exhaustive.

**SRS – Stockholm Royal Seaport project**
The project used to test the interoperability language.

**IMS – Information Management System**
A system responsible for the management of some information. Often used to refer to the central future information system in the SRS project.

**MPS – MessagePassingSystem**
An entity in the interoperability language, which represents for instance a router.

**ESI – Energy Services Interface**
A computerized small central used to collect metering values.

**PRM – Probabilistic Relational Model**
A meta-model with defined relationships between the attributes in form of a Bayesian network.

1.5 Acknowledgements

First of all I would like to give a big thanks to Johan Ullberg who has supervised this thesis and helped out in many aspects of this work. Without his interest and devotion this thesis would not be the same.

I would also like to thank Hossein Sharokni together with the Industrial Ecology department, who has given me great support in the data collection phase and helped with getting in touch with various key persons.

The project members in Work Package 5 in SRS also deserve many thanks for their support and time they have spent on answering my questions. In particular Henrik Evers, Stefan Schreiter and Pia Stoll.
I would also wish to send a thanks to some of the persons I have been interviewing; Erik Hjelm at Fortum, Jenny Dahlberg at Fortum Värme, KTC and Envac for their time and interest.

2 Method

This section will aim to describe the design of this study by explaining the goals and purpose and then proceed with the different steps in the master thesis project.

2.1 Validity & Reliability

The concepts presented by Yin, 2009, for assuring validity and reliability is already presented in section 2.4 Analysis. These are not only important aspects in the analysis of the interoperability model, but also as guidelines in how to conduct this whole study. Below each of the four concepts will be considered and measures to ensure them will be suggested.

Construct validity
Construct validity is a challenging test for most studies. The idea is to be sure that no “subjective” judgments are used to collect data. Important to fulfill this is to use multiple sources of evidence for the findings, and to relate changes in concepts or procedures to the study’s original objectives.

In practice this means that a thorough plan for investigation should be determined before the study commences, and that this plan is followed. If you for instance are investigating changes in a neighborhood over time, a measure could be reported crime. If this is decided on beforehand it is easy for the reader to understand the conclusions and see that they are valid, as opposed to just investigating the neighborhood and then after the data collection is made see that there has been a change to reported crime and then choose it as a measure.

A common strategy to achieve this is to use multiple sources of evidence for the conclusions. In this study two main sources of evidence will be used; interviews and documentation. However by also try to interview several persons related to the same actor, the evidence base can be strengthen further. Another important aspect is to maintain a chain of evidence. It means that a reviewer of the study shall be able to trace the conclusions drawn back to the initial sources and why they were chosen. The method is similar to what can be expected from a forensic investigation. One more measure to ensure construct validity is to let key informants review a draft of the findings, in this case the result summary report.

Internal validity
Internal validity is mostly a concern for explanatory studies where the research aims to explain how and why event $x$ leads to event $y$. If the researcher establishes a casual relationship without knowing of a third event $z$ that may affect the occurrence of $y$, the research design has failed to deal with internal validity.
Since this is mostly a concern for studies trying to explain why event $x$ leads up to event $y$, and this analysis is not trying to show such a connection, but rather to explore the modeling language in question, internal validity is not of high interest in the study as a whole.

**External validity**

External validity deals with knowing whether the case study's results are generalizable beyond the immediate proximity of the case studied. Case studies are sometimes said to be hard to generalize since the results may just be occurring in the very case studied. In this aspect case studies are sometimes compared to survey studies, and it is being said that surveys are intended to be generalizable to a larger universe. This is however not certainty the case, since survey studies relies on statistical generalization, whereas case studies rely on analytical generalization. One way of ensuring external validity in a case study is to test the theory by replicating the findings on two or three other cases, a so called multiple case study.

The best setting for this study would to conduct multiple single case studies where the modeling language where tested and evaluated and then to sum up the most important findings and generalize them to proposed changes and additions to the modeling language. However due to time constraints it is not realistic to plan such a large study as of now. Nonetheless this study can very well be the first step in such a study so that later studies can together with this one create validity for proposed changes.

**Reliability**

Reliability means that the study should result in the same findings and conclusions if it were to be performed again by another investigator. To ensure this it is of course very important to document the steps that led up to the findings and conclusions. A good way of ensuring reliability is to conduct research as if somebody always were looking over the shoulder.

To ensure reliability, all interviews, documents etc. will be saved so that a future reviewer can follow the chain of evidence. All steps taken will also be carefully documented and presented in the final report. This is very important since a reviewer should be able to do the exact same study and come to the same conclusions.

**2.2 Data collection**

The method for data collection consisted of four somewhat iterative steps as shown in Figure 1. The first step was to identify the communication needs, i.e. what information that must be exchanged in order for the communication to function. This will be an essential part of instantiating the model. If there is no communication need, there is no point of investigation the interoperability. The next step was to identify actors that will be part of the communication chain. An actor can for instance be a system or a sensor of some kind. This step also involved finding out whom is responsible for the identified actor. When these people had been identified
the next step was to interview them. From such an interview it is very likely to end up with:

- documentation regarding the actor
- a new communication need not included already
- a known communication need with a new actor that was not included earlier or
- another person that has to be interviewed in order for the picture of the actor to be complete.

The documentation may in its turn lead to even more questions, why a new interview or more documentation can be needed in order to identify even more actors and so on.

![Diagram](image)

**Figure 1 The data collection method**

Both the interviews and the documentation studies were documented in order to preserve validity and reliability.

The goal of the data collection was to gather enough data to instantiate an extensive model using the interoperability language to be evaluated. Since the main purpose of the study is to evaluate the interoperability language, and a sub goal is to provide useful input to the Royal Seaport project, the limit in extent was done in the number of participants (companies and authorities), since it is more meaningful in the analysis of the language to have a limited number of participants with well investigated conditions and variables rather then a lot of participants with less time spent per participant, and therefore less information about individual variables and settings.

The interoperability language has some primary entities that the data collection will be designed around. The Actor entity is central, it represents a system, a person or
similar which actively participates (with altering the data for example). Another type of entity is the MessagePassingSystem (MPS), which is similar to the Actor in terms of also being a participant in the communication, but only by passing it on. The language the participants use to communicate is also represented in the language by an entity called Language. A fourth entity important for the data collection is the Language Translation, which represents that some Actor translates a language into another. A language could for instance be XML and a translation could for instance be between XML and JSON.

![Diagram](image)

**Figure 2 The levels of data needed**

To be able to instantiate the model three main levels of data were collected, see Figure 2. The first level was which Actors and Message Passing Systems that were of interest for the interoperability analysis. This was completed with what languages and translations that are used to communicate. The next level was to establish how these relate to each other and the future IMS. The third step was to investigate attributes related to Actors, MPSs and language translations. For every Actor and MPS, probabilities for how likely it is to distort a message, drop a message and be available will have to be discovered. For every language translation a correctness probability should be determined. This information was elicited foremost by interviews guided by the questionnaire in Appendix A. When all the above was found (or estimated), the model was instantiated and the interoperability analysis could be performed.

**Identifying Communication Needs and Actors**

As a starting point for the data collection phase initial communication needs was identified followed by the identification of their actors and responsible persons for the actors. This identification step had its outset in a list of important environmental aspects in the Royal Seaport project. This list focused mainly on so called “indicators” which are a kind of environmental requirements on different parts of the neighborhood. The indicators can be divided into two main groups. One group concerns aspects of the neighborhoods effect on the environment. The effect on the environment is derived from these “indicators” which have measure points. These measure points are then converted into kg CO$_2$. 
Figure 3 Example of how indicators relate to the interoperability analysis

An example could be an indicator stating that the individual household's electrical power consumption should be low. A measure point could then be the electrical power consumption in kWh. The communication need in this case would be the need of getting the power consumption and the other actor (one is the IMS) would be the electrical power consumption meter in the households, see Figure 3. Another measure point for this indicator would be the current power source shares (how much of the current electricity derived from nuclear power etc.). The communication need would be the need to get the power source shares and an actor could be an IT-system at “Svenska Kraftnät” (responsible for the Swedish power grid). If this communication need is satisfied, a conversion from kWh to kg CO₂ can be made since the amount of CO₂ for every produced kWh for every power source can be derived through documentation and research.

Since there are a lot of indicators and actors involved, the indicators will be prioritized in importance to the whole picture of assessing climate positivity in the Royal Seaport neighborhood. This prioritization was done in consultation with the Institution for Industrial Ecology at KTH. When prioritizing, a lot of different factors could be considered. One factor is how reliable the data available is, this may however be hard to assert on beforehand without talking to the involved parties, and since the purpose with the prioritization was to limit the number of interviews it does not seem very relevant. Another factor could be to focus on actors involved with a lot of indicators, that may limit the number of interviews, but not necessarily, since it is probably different persons responsible for data regarding different measuring points in the organization. A perhaps more suitable prioritization is to try to predict how likely every indicator is to be of high importance in the future. That was basically determined of how easy it would be to estimate, and how high effect on the total environmental impact it has. So these two factors was predicted and combined, which resulted in a priority order.

Interviews

When conducting interviews it is important that the interview rather become a guided conversation than structured queries to be able to get as much useful information as possible. It requires the interviewer to be sure to pursue the goals of the interview so that no needed evidence is left out, at the same time as having a conversation with the interviewee that probably will lead to evidence that wasn’t
thought of, as well as other sources of information not previously known to the interviewer. The main type of interviews that will be conducted throughout this study will be so called focused interviews. These interviews are time limited, 1 hour for example, and follow a set of predefined questions that will guide the conversation. The interviewer can however still ask open-ended questions to find new angles and perspectives. When formulating the questions it is important to be unbiased as well as thinking over the phrasing. There is a difference between asking “why” and “how” questions. “How” questions are more likely to get more information from the interviewee. (Yin, 2009)

The interviews where preferably performed in person, but since this wasn't logistically doable in many cases some interviews were performed over telephone and email.

**Documentation**

A part from interviews, documentation was thought to be an important source for evidence in this study. The documentation that was relevant was manuals, requirement specifications etc. regarding the IT systems that are to be investigated. This proved however to be hard to get hold of. (Yin, 2009)

### 2.3 Pre-analysis

The analysis is divided into two parts. The first part is the instantiation of the models by using the language for interoperability analysis. Since the main analysis phase will serve to evaluate the language, rather then evaluate the raw data that has been collected in the previous phase, a middle phase in between the data collection and the analysis is required.

The instantiation of the model was a parallel process to the data collection phase. The model work done in the pre-analysis phase was rather to sharpen the model and make adjustments to better mirror the big picture that the data collection phase will result in.

When the models where created the original thought was to perform the interoperability analysis using software tools available. However since the tools where in such early stages of development the most straightforward solution was to develop a simple analysis tool made especially to be able to do the calculations required by the interoperability language. A graphical representation function was also added in order to visualize the models with the added calculations. The tool has the development name “Interoop” and is publically available and open source1. The outcome is probabilities that interoperability can occur which helps in identification of potential problem areas in the communication chain.

During the development some observations regarding the calculations was made that led to a deeper understanding of how the framework could be used.

The outcome of this phase was a preliminary results summary. A draft of the models presented in this document was shared with the stakeholders for the different actors

---

1 [https://github.com/jede/interoop](https://github.com/jede/interoop)
analyzed. This provided some useful feedback, which was incorporated in the document.

2.4 Analysis

The main analysis was based on the outcomes from the pre-analysis, namely the Results summary report. The analysis will be done with starting point in Yin’s theories for what characterizes a good case study. In other words will these principles used twofold in this study. Any deviations will be discussed regarding the cause, and if the modeling language could be designed any different to avoid similar problems in the future.

There are four commonly used tests do determine the quality of empirical social research, which case studies are. These tests are: (Yin, 2009)

Assessing construct validity
As explained earlier construct validity deals with the data collection being done in an objective manner.
In this case this will correspond to that the pre-analysis is done in an objective matter. That means that the pre-analysis phase should investigate interoperability in the system architecture in question, without designing a solution or exaggerate any potential problems. This will be investigated by looking closely on how the modeling language handles lack of data when building the model; does the modeler have to make assumptions similar to designing the system?

Another way of asserting this test will be to investigate if the same data could result in a very different model. If that is the case the language could potentially open for the modeler to choose how to build the model depending on the results he/she wants to achieve.

Assessing internal validity
The study’s pre-analysis is concerned with concluding the presumed state of the interoperability in the Royal Seaport project. Therein lies particularly to conclude what problem areas there are in the communication, i.e. not only where problems are likely to occur, but where they originate. In that sense it is of high importance that no “third event” is left out in the model, all relevant information collected should be utilized in the model. To check this the collected data will be compared to what was able to fit in the model. This will be done by, during the data collection, evaluate the collected material in how important it is likely to be in an interoperability perspective. Then the finished model will be compared to this data and any discrepancies in what data that was regarded as important and is not represented in the model will be noted.

Assessing external validity
Cities have unique IT architectures, why it is hard to generalize the findings in the pre-analysis phase, and especially since this is a very special urban area which aims to become climate positive.
Assessing reliability
In this case it will be partly about studying if the model analysis could give another result using the same data, just as described above in the construct validity section. The other part is about assuring that the process was clear and comprehensible so that another party would end up with the same model. An important part is to discuss how clear the definition of the language is, if it is open to interpretation, and then it is unlikely that another person would use it in the same way.

This boils down to three tests that will have to be asserted: Construct validity, Internal validity and reliability. The output from the discussions from these tests will then serve as a starting point for a discussion regarding changes and add-ons to the modeling language. The goal of the discussion is to propose modifications that would lead to minimizing the potential problems found when asserting the tests.

2.5 Presentation

The findings from the analysis phase will be summed up in a final report document, which together with a presentation held at KTH will serve as the presentation of the project.

2.6 Alternatives methods

This study could be carried out in many different ways. I can identify two main rivaling methods; doing a survey study or doing an experimental study. (Yin, 2009) A survey study could be conducted as such as handing out surveys to different domain experts with questions regarding how to assess interoperability, and ask them to rate some quality aspects of models which where build with the language in question. A clear risk with this kind of study is the inflexibility in that the survey questions cannot be changed on what the reaction of the person that gets the survey. This could be troublesome since it is a very complex matter to investigate, why the same question can have different interpretations depending on background knowledge and assumptions of the respondent, which would lead to no or very low significance and data appearing as random.

The other alternative would be to conduct an experiment. This could be conducted by using a fictive case where interoperability is an issue and then model this fictional scenario with the model. The apparent risk with this approach would be that the made up scenario would lack any non-predictable problems that can occur in the real world, making the result not as interesting as if based on a real world scenario. An advantage would however be that the fictive case could easier contain all aspects that the language is designed for, which as stated also would be the methods weakness since it is the aspects that will occur that the language was not designed for that are really interesting to investigate further.
3 Theory

This section will describe the theories used in the analysis chapters later on.

3.1 Using models

The field of enterprise architecture promotes the use of documentation of the business and IT architecture of enterprises. The approach is intended to document the business in a way that applies to all stakeholders. The board and the developers have different views on what is important in the organization, however one could not function without the other why it is important that all stakeholders have a shared vision of the organization. If an organization fails to convey the vision, people will likely work in different directions, which will lead to unnecessary power struggles and internal politics disagreements.

A good way of gaining and communicating a shared vision is through models. A model is a representation of the enterprise from some viewpoint. A meta-model defines the rules of a modeling language. From a meta-model, viewpoints can be derived, as sub meta-models which is a subset of the whole meta-model. A viewpoint can then be used to instantiate the model. This is illustrated in Figure 4.

![Figure 4 How meta-models and models represent the enterprise](image)

The meta-model can be described as a filter that the viewer looks on the reality through. Depending on which filter you choose and what parts of the organization you chose to focus on, you will get a different representation of the enterprise. An important aspect when using models is that the models is supposed to help achieve a shared vision, there is no value in modeling itself, the models are mere a path to some other goal, not the actual goal.

There are a lot of different methodologies, frameworks, taxonomies etc. to deal with enterprise architecture. To mention some there are the Zachman framework, TOGAF, FEA and the Gartner methodology. All have different ideas and strategies on how to gain a shared vision throughout the organization. The Zachman
framework focuses on what to model, TOGAF focuses on how to model and so on. To infer enterprise architecture can indeed be the solution to diverging visions, but not a universal solution. Enterprise architecture should be viewed upon as the city plan when building a city. Do you need to develop such a plan if you are out camping and about to raise the tents? Probably not. The idea of enterprise architecture is to be a tool, a way of structuring the visions in an understandable manner, not to be in the way. (Sessions, 2009)

One definition of Enterprise Architecture is:

“A coherent whole of principles, methods, and models that are used in the design and realization of an enterprise’s organizational structure, business processes, information systems, and infrastructure.” (Lankhorst, 2009)

In addition it should also strive to give a holistic view of the enterprise. This holistic view is very important in order to make the correct decisions in an organization. But based on what should these decisions be made? And what decisions are there to be made? This is where enterprise architecture is pushed further to enterprise analysis, which depends on the enterprise architecture to be modeled. One such analysis, depending on the modeling language used, can be made using probabilistic relational models.

3.2 Probabilistic relational models

To be able to express uncertainty or many possible solutions in a model, probabilities often is needed. Otherwise models would express a fixed state which would be a incomplete representation of the reality. There are numerous ways of inferring probabilities into a mathematical model.

When constructing statistical models for relational data Bayesian networks are used quite frequently. In addition there is an extended variant of Bayesian networks, namely Probabilistic Relational Models (PRMs). The modeling language for interoperability that this thesis will focus on is constructed as a PRM and therefore the theories behind PRMs are quite essential.

A Bayesian network is used for modeling how different attributes influence each other. As an example we can look at Figure 5, which shows a Bayesian network for some disease. It states that the age and occupation for a person has influence over how likely it is to get a certain disease, and if you where to have the disease then how likely it would be for you to get some symptoms, and maybe how severe they would be.
The same concept can be used to model more complex scenarios that require multiple instances of similar entities. For those cases PRMs are more suitable. A PRM defines relations between different classes of objects. So instead of starting modeling with a blank sheet, you start with a metamodel that states which objects that can be included (scoping). The metamodel also knows how instances of different classes attributes affect related objects attributes. In the example in Figure 6 we could for instance know that a course’s difficulty is affected by the registered students intelligence. If we have a couple of courses and some hundred students, a corresponding bayesian network is of course possible to create, however it would be very hard to understand, and even harder to analyze. (Getoor & Taskar, 2007)

3.3 A modeling language for assessing interoperability

This study aims to investigate the modeling language designed for interoperability analysis presented here, and evaluate the performance of the language in a real world scenario. The term interoperability is used as defined by IEEE as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” (The Institute of Electrical and Electronics Engineers, 1990). The article also presumes that a central ingredient in interoperability is communication need, i.e. two or more systems, organizations etc. that need to communicate with each other. By definition: if no communication need exists, interoperability is not of interest to be improved or investigated.

The language is a meta model constructed as a probabilistic relational model. Hence the instantiated model contains defined dependencies between the different objects that are modeled. There are two main parts of the meta-model. One simplified used to model the basic infrastructure for interoperability, and one more fine grained that
is intended to use for modeling specific conversations, or information exchange. This study will however only use the simplified version since the detailed version is hard to use on a system that doesn’t exist yet.

The benefits with this modeling language is twofold. First of all it provides a suitable modeling language of interoperability. The workflow of constructing the model is shown in Figure 7. The starting point is the PRM that will be described more thoroughly below. This PRM serves together with the attribute dependency structure as the metamodel. However the beauty of PRMs instead of Bayesian networks is that the user doesn’t have to be concerned with how different attributes affects each other, that is defined in the PRM. The user only have to define the relations between the objects to create the instansiated PRM together with some basic attributes. When this is done a Bayesian network for the attribute dependencies can be generated automatically, and the unknow attributes of interest can be calculated. Even though the generation is done automatically, the created Bayesian network will still be far to complex to be analyzed “manually” in most cases. Therefor the modeling language supports evaluation through Probabilistic Object Constraint Language (P-OCL) statements.

![Figure 7 The work order of the modeling language](image)

P-OCL serves as a type of query language in the PRM. So instead of manually investigate if there is a communication chain between two Actors in the instansiated model, a P-OCL statement can be executed and will then answer the query. The P-OCL statements can of course be used to answer more complex queries, like the probability that the communication between two Actors will work. This makes the modeling language very powerful, since it has a way to deal with a vast number of actors, without doing a tradeoff on level of detail.

The suggested simplified meta model is shown in Figure 8. The different object types will now be described.
An actor is anyone actively participating in the chain of communication. It could be a person, an information system or an electrical component. An actor that passively takes part of a communication chain, i.e. just passes information on, is modeled as a Message Passing System. Both the Actor and Message Passing System inherits from Abstract Actor, which represents attributes and relations shared between Message Passing Systems and Actors.

A central part for the meta model (and eventually the models) is the Communication Need. As stated earlier, without a communication need, interoperability becomes a non-issue. The whole purpose with the interoperability analysis is to find different needs for communication and determine how likely they are to be satisfied.

A Language represents any language that can be used to exchange information. It could be Swedish, HTML or even human body language. A language can inherit from its reference language, for example HTML has XML as its reference language, meaning that any potential limitations or defects in XML may also affect HTML. Languages can also be translated by an Actor using a Language Translation.

Figure 9 shows the detailed PRM with the attributes. (Ullberg, Buschle, & Johnson, 2010)
Besides the relation structure, the attributes in the meta-model are an important part of the PRM. These will be described below. All values are of the type "boolean". That indicates that in the reality at a given point an attribute can be either true or false, nothing else. Compare for instance the hypothetical attributes ‘available’ and ‘voltage level’. The attribute available indicates if a system is available or not, at a given point in time. This value can only be true or false, since there is nothing in between, as the system is either up or down in this sense. Voltage level however could represent the voltage output from some device at a specific point in time. This value would rather be something like 4.5 Volt then true or false.

In the models however boolean values are represented as probabilities to take into account the dimension of time. I.e. instead of saying in the model that a system is available all the time or not at all, we say that a system is available for instance 99.9% of the time.

The different attributes are described in Table 1, except from one. The MessagePassingSystem has a special attribute named fixed. The fixed attribute is not probabilistic, it describes that the MessagePassingSystem is of such a simple nature that it doesn’t use addressing. An example of a MessagePassingSystem with the attribute fixed = true would be a serial cable connected between two computers.

**Abstract Actor**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>distortsMessage : boolean</td>
<td>Symbolizes the probability that a message’s content is garbled, given that a message has arrived.</td>
</tr>
<tr>
<td>dropsMessage : boolean</td>
<td>The probability of a message to be dropped by the actor.</td>
</tr>
<tr>
<td>isAvailable : boolean</td>
<td>The probability of a system to be available at a given time.</td>
</tr>
<tr>
<td><strong>Language Translation</strong></td>
<td><strong>Communication Need</strong></td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>correct : boolean</td>
<td>Symbolizes the probability that a message is translated correctly using this translation.</td>
</tr>
<tr>
<td>syntacticDistortion : boolean</td>
<td>The probability that a message gets distorted somewhere in the communication.</td>
</tr>
<tr>
<td>semanticDistortion : boolean</td>
<td>The probability that a message is translated in a way that the meaning is lost in the communication.</td>
</tr>
<tr>
<td>addressingFailure : boolean</td>
<td>True if the addressingNeed (another optional communication need) is satisfied.</td>
</tr>
<tr>
<td>pathAvailable : boolean</td>
<td>Symbolizes that the communication path is available.</td>
</tr>
<tr>
<td>noPath : boolean</td>
<td>True if a path between the actors in the communication need does not exist.</td>
</tr>
<tr>
<td>noCommonLanguage : boolean</td>
<td>Indicates if there is a shared language across the communication path.</td>
</tr>
<tr>
<td>noLanguageChain : boolean</td>
<td>The probability that there is a chain of languages and translations that enables the communication.</td>
</tr>
<tr>
<td>languageFailure : boolean</td>
<td>If there is noCommonLanguage and noLanguageChain, then a languageFailure is imminent.</td>
</tr>
<tr>
<td>dropsMessage : boolean</td>
<td>The probability that a message is dropped somewhere in the communication path.</td>
</tr>
</tbody>
</table>

**Table 1 Descriptions of the different attributes**

To be able to use the interoperability language and perform the calculations some additions has to be made. During the evaluation process of this interoperability language one important aspect has been identified. There can occur troublesome situations where two paths exist between two Actors that share a CommunicationNeed. One example is illustrated in Figure 10, where we have two actors connected by two independent MessagePassingSystems. One MessagePassingSystem distorts a message with a probability of 80% and the other drops a message with a probability of 80%. If we do the calculations in the most naïve way, where every calculation is done independent of the others, and we choose the “best” value (the lowest values in this example) it results in a miss guiding result; a communication need with both the lowest values.
This would mean that the probability of a message to be dropped between Actor A and Actor B would be 0% and the probability of a message to be distorted would also be 0%. This would then indicate that there exists a way which makes this possible, that is not the case according to the figure.

What is lacking here is a way to indicate that the results differ depending on which way is chosen. One way would be to always calculate the values along the way that is most likely to be chosen in the actual implementation. Of course strategies for this varies, but one assumption would be that the path that is most likely to be available is preferred. In Figure 10 that would lead to any of the ways being used, so either the result would be that messages are distorted in 80% of the cases or it would lead to messages being dropped in 80% of the cases.

That strategy would correspond to saying that if one way is more available that way is chosen with 100% probability. Another strategy would be to say that if way A is available 90% of the times and way B 70%, then we choose A with 90% probability and B with 10%. This algorithm could even incorporate other attributes, like distorts message or drops message, to calculate probabilities corresponding to the likelihood of a specific way to become the actual implementation.

This also implies that CommunicationNeeds with more than two actors needs an interpretation. Since it is now important to choose one path to analyze, two or more actors would yield a preferred set of paths to analyze. The question is only how to interpret many actors. There are basically three options: All actors have to communicate with all other, all actors must be connected or one actor needs to communicate with all others. Since this is a matter of choice, the last interpretation will be used in this report since that definition proves to be the most useful in practice. That definition also leads to the attributes being aggregated at the CommunicationNeed as the product of all the paths’ attributes since the interpretation says that the main actor needs to communicate with all underlying actors in order to be satisfied.
4 Data collection

The data collection and the results that it yielded are described below. This will then be analyzed in the Pre-analysis section.

4.1 Prioritization

To be able to ensure that the focus of the analysis was right, a prioritization of different groups, or clusters, of indicators where made.

<table>
<thead>
<tr>
<th>System för:</th>
<th>Ska leverera data:</th>
<th>Svärestimerbarhet (1-5)</th>
<th>WPEs Behov (1-5)</th>
<th>Prioritet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual delivery points</td>
<td>Elproduktion</td>
<td>5</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Elhandel</td>
<td>Köpt/såld el per fastighet</td>
<td>5</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Elmätare</td>
<td>Elkonsumtion</td>
<td>5</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Transportsystem</td>
<td>Hur boende transporterar sig</td>
<td></td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Building information management</td>
<td>Luftfuktighet, elkonsumtion + vattenkonsumtion fastighet</td>
<td>5</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Elkraftkällfördelning</td>
<td>Hur elmixen ser ut</td>
<td>5</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Vattarmätare</td>
<td>Vattenkonsumtion + temp</td>
<td>5</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>Värme/klimathandel</td>
<td>Köpt/såld värme per fastighet</td>
<td></td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Avfällskvarnar</td>
<td>Växt malt avfall</td>
<td>3</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>Sopnedlottar</td>
<td>Väkt/avfallstyp</td>
<td>3</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>Avfallshårdadta</td>
<td>Avfall som hämtas, vikter</td>
<td>3</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>Energiförbrukning - avfallshämtning</td>
<td>Hur mycket energi som går åt för att hämta avfall</td>
<td>3</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>Vattendebitering</td>
<td>Vattenkonsumtion</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>AVC</td>
<td>Avfallsfördelningen på ÅVC</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Avloppsvattenrenning</td>
<td>Återvunnen energi, hälsoskadaliga ämnen</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Bevattningmätare</td>
<td>Hur mycket som vattnas privat/fastighet</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Massbalanser</td>
<td>Föröreningar</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2 The prioritization of indicator clusters (in Swedish)

Table 2 shows the prioritized clusters. These prioritizations led to a much better understanding of where the effort was going to make the most use. With regards to the result the decision was made to analyze the clusters with a priority of 12 or above. Taking these clusters and grouping them by the plausible organizations that will be able to deliver the data the diagram in Figure 11 was achieved.
4.2 Actors and communication needs

To be able to show the results from the data collection to the stakeholders, a temporary model was created with a very informal meta-model. The purpose of the exercise was to ensure that the data collected was interpreted the correct way. Therefore this model differs from the later ones, since this was created before the analysis of the collected data. It was created on the go and double-checked with the stakeholders before proceeding.
Figure 12 The presumed system architecture at this stage
Figure 12 illustrates the structural findings in the data collection. It illustrates what systems will be located where in the actual buildings and what larger systems that the IMS will have to communicate with in order to be fore-fill the highest prioritized metering point clusters. The question marks illustrates that the details about the circumstances are not known. A question mark at a line indicates that the protocol used is unknown. A question mark after a system description indicates that the exact system that will be the endpoint of the communication is unknown.

4.3 Technical specifications regarding the systems and components

This section is intended to summarize the data that was found regarding to different systems. In general it was quite hard to actually get hold of reliable and unbiased data for the systems regarding availability, likelihood to drop a message and to distort a message. This is due to either companies don’t want to release such figures because it could damage their reputation, or due to interviewees being too positive like for instance saying that a system is always available when it is obvious that a power outage or similar could make it go down. A third reason is people not actually knowing, often because it seems not to be a sales argument, it is rather focus on for example advanced features.

Meters
Meters are the common term for the equipment that measures different values either from an apartment or a building. It could be a water meter or an electricity meter for instance. The data concerning the meters comes from interviews with the companies:

- Fortum
- KTC
- ABB

KTC works with equipping buildings with monitoring equipment for the residents, much what the SRS project wants to accomplish (amongst other things). Therefore KTC has a lot of knowledge in what metering equipment that works and how it is used. ABB and Fortum are participants in the SRS project and responsible for a lot of the technology why they had some input and information regarding the meters and the related equipment. As shown in Figure 12 every meter is connected to an Energy Services Interface (ESI). These ESI stations can be fed with metering data and then aggregate it in for instance XML and then pass them on to other systems. The meters will most likely use the standard protocol M-Bus (Meter-Bus), defined by the standards **EN 1434** and **EN 13757** (M-Bus Usergroup). The ESI has an availability of 99.9% according to ABB. An electrical meter does not distort a message in 99.9% of the cases and does not drop messages in 99.9% of the cases according to Fortum, hence all other meters will be estimated to have the same values. Since an ESI is of roughly the same technical complexity, it will also be given the same values for dropsMessage and distorsMessage. It was hard to get an accurate number of how available a meter is, but a fair estimate would be that a meter is probably more available than the ESI, since it is of a more simple kind. However
when a meter breaks it takes time to repair it why an average of one-day downtime every year is a fair estimate, i.e. $364/365 = 99.7\%$.

The M-Bus components connect to the ESIs with a serial cable. This cable is modeled as a fixed MessagePassingSystem since it just connects two nodes without any addressing. Since the system actually is a cable, the availability is estimated to 100%. However potential disturbances can cause the message to be distorted or dropped why probabilities of 0.1% are inferred on drops message and distorts message. This is illustrated in Table 3.

**IT-systems**

When it comes to the IT-systems availability is a measure that is hard to get hold of. It is a complex figure that can vary in different contexts. Therefore a general estimation will be done based on the virtual server services that Amazon provide, called EC2. In the SLA Amazon guarantees an uptime of at least 99.95% on a yearly basis, why this figure will be used for the general modern IT-system. Since the communication between these and the IMS will be realized using TCP/IP messages that are dropped are resent, and therefore not dropped in the sense meant in the interoperability. The same goes for distortion in the TCP/IP protocol, why both these will have a 0% possibility.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Meter</th>
<th>ESI</th>
<th>Serial cable</th>
<th>Other system</th>
</tr>
</thead>
<tbody>
<tr>
<td>isAvailable</td>
<td>99.7%</td>
<td>99.9%</td>
<td>100%</td>
<td>99.95%</td>
</tr>
<tr>
<td>dropsMessage</td>
<td>0.1%</td>
<td>0.1%</td>
<td>0.1%</td>
<td>0%</td>
</tr>
<tr>
<td>distortsMessage</td>
<td>0.1%</td>
<td>0.1%</td>
<td>0.1%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 3 The attributes of the different Actors

**Translations**

The language translations made (M-Bus to XML) are assumed to be correct and free of bugs when the system has been in service for some time. Therefore the correctness of these will be estimated to 100%.

### 4.4 Other aspects of communication

The effort has been focused at investigating the technical prerequisites and eventual problems for the communication in the future IMS. However besides the technical aspects, other aspects have also arisen during the analysis. There are basically three other major areas that can cause substantial problems when implementing the IMS, which described below.

**Legal**

First of all there is the legal aspect on measuring, collecting and storing data about individuals. In Sweden there are strict laws about what one can measure and store without the permits of the individual being the subject whose data is being stored. However since the IMS won’t collect and store information that is regarded as sensitive in a legal meaning a permit to collect and store the data can be included in
the buying/rental contract that the individual signs before moving in. But even though this problem seems solved, it is important not to miss the aspect when moving forward with the IMS project.

**Public Relations**
With the legal aspects settled, the same basic issue, that is the individual is being measured and the data stored, can cause troubles at the public relations side. There has already been some media attention to the projects goal of data collection.

**Political**
There seems to be some cases of data exchange being technical possible, but the data is for some reason to sensitive to the organization owning it to share it with the IMS.

### 4.5 Summary of findings

The technical prerequisites for the communication to work seem to be in place at this early stage. There can still show to be problematic when the analysis is complete, but as of now, no major problems are imminent. However there are other problems that can lead to severe consequences, because of their uncertainty and their unpredictability, like the political aspects.

### 5 Pre-Analysis

This section will cover the analysis of the data using the interoperability assessment framework presented previously in this report. The pre-analysis is performed using the framework as it is described to strive for reproducible results. In the following Analysis section will this pre-analysis be evaluated and changes to the framework that could render different results will be discussed.

**Constructing the model**
The starting point for the construction of the model was the architecture draft in Figure 12. The idea was to have it as guidance when creating the relationships.

The meters and their infrastructure were modeled first. Every meter is connected to an ESI. This ESI can belong to a building or to an apartment. In Figure 13 an example is shown where an electrical billing meter is connected through a serial bus to the Home ESI, which in its turn sends data through the building network to the IMS. There is also a CommunicationNeed between the meter and the IMS. The communication need is named “Power consumption impact” which refers to an indicator mentioned earlier. The IMS shall be able to calculate how much environmental impact the consumers power consumption has, and for that to happen the IMS has to communicate with certain actors, i.e. the billing meters.
Figure 13 Illustration of one meter connected to its ESI in an apartment
The same principle is applied when instantiating the other meters in an apartment or in a building.

For the IT systems that the IMS needs to communicate with the model looks slightly different. Figure 14 shows an example of how a system that knows the mix of sources that produces electrical power at the moment needs to communicate with the IMS.

Figure 14 Example of an IT system that needs to communicate with the IMS
This information is vital in order to convert used kWh to CO₂. The SvK system connects to the IMS via the Internet. Internet is estimated to be available 99.95%,
just as an IT-system, and always delivers its messages otherwise thanks to TCP. The Internet not being available means that an Internet Service Provider fails to deliver access for some reason in any of the parts of the communication.

In both the examples above the languages of the systems are not visualized. They will not be in the following examples either, since languages always will exist to make the communication work in this case. However it is possible to verify this by checking the language attributes in the communication needs.

Figure 15 The instantiated model for the IT-systems

Figure 15 shows how the model looks with all IT-systems in place, without any apartments or buildings with meters. The model grows rapidly and becomes hard to understand just by looking on it as a whole. Heat consumption impact corresponds to Power consumption impact, but for heating power instead of electrical power. Another one is Transportation impact, which is supposed to represent the needed data to be able to analyze how much environmental impact transportation has in the neighborhood, per inhabitant. To be able to retrieve these values a minimum is to communicate with the system measuring the overall transports in the neighborhood and also the system at the Recycling entrepreneur measuring how much energy that
is consumed when taking care of the waste. This could for instance be energy spent on vacuum suction of the waste. This is followed by the Waste impact communication need. This represents the communication needed to be able to calculate how much energy that is spent on recycling the waste per household. That needs data from disposer scales in the households as well as data from the recycling entrepreneur showing how much that is thrown away and where. The Price forecast communication need is because the IMS needs to know how much the electrical power is supposed to cost the next day so that effective scheduling of laundry etc. can be performed. The last added communication need is the CO₂ Analysis. This represents the data that needs to be sent to the Active House system so that it can be displayed to the inhabitants.

**Note** that the satisfied attributes are high (>90%) all over the communication needs in this instantiation.

In Figure 16 the instantiated model for one building with one apartment is shown. This instantiation is done without any IT-systems involved. The apartment contains the following meters:

- 2 water meters (since there are several water outlets in an apartment)
- 1 billing electricity meter
- 2 electricity sub meters (could measure outlets for instance, also several)
- 1 thermometer (could be several, but one is probably sufficient)
- 1 disposer scale (weighs organic waste disposed in the sink)

In every building there is a solar panel, which sends its values using M-Bus the home ESIs. The building also contains meters (to measure common areas). They are as follows:

- 2 water meters (there are several water outlets in the common areas)
- 1 billing electricity meter
- 2 electricity sub meters (could measure outlets for instance)
- 2 thermometers (there are several spaces that needs to be measured in the building)
Figure 16 An example of one apartment in one building with meters

Even here it is noteworthy that the communication needs have all over high probability of satisfaction. There is also another communication need that was not present earlier; Water consumption. This communication need only states that the measure points from the water meters should be forwarded to the IMS.

The next step in the model creation is to join the two examples above so that a model with both the IT-systems and the meters are achieved. This is illustrated in Figure 17, but without the meters and ESIs visualized.
The probabilities of the communication needs being satisfied are still quite high. This means that the interoperability conditions are quite good according to the model in the simplest case, i.e. one building with only one apartment. This is however not the actual case, since in reality it would be a large neighborhood with around 400 apartments. In Figure 18 the model is shown for 2 buildings containing 10 apartments each. Then we see that some of the satisfactory probabilities drop vastly. This is especially true for the Power consumption impact and the Water consumption communication needs.
Figure 18 An example with 2 buildings containing 10 apartments each

Conclusions
The vast drop is due to each of these being dependent on several meters in every ESI, which also means a lot of more points of failure, because if one of these meters fail the totals will not match the billing information and hence result in a lot of problems in the calculations.

In order to approve this, an obvious way is to get better meters, or fewer meters in order to decrease the possible points of failure. Another possibility would be to introduce estimation of the Power consumption and the Water consumption in the cases where the data is unreliable.

6 Analysis
This chapter will evaluate how well the language worked out as a tool for assessing the interoperability in the Pre-analysis chapter.
6.1 Naming conventions

In the language for interoperability the calculated attributes on a communication need is named after barriers of communication. For instance noPath is true when there is no path, and false when there is. This can be confusing when trying to understand the language since the interpretation of noPath = false is not obvious if you do not know the definition. A better name would be pathExists, since pathExists = true is much easier to grasp. (Steve McConnell, 2009)

6.2 Probability of existence

According to Ullberg, one important feature of the interoperability language is the possibility to assign probabilities of existence for every entity and relation in the models. The purpose of using these existence probabilities is to be able to determine not only which is the best scenario in a communication need, but also to include how likely that way is to exist. So for instance if we have two paths \( w_1 \) and \( w_2 \) between the Actors A and B, and the availability expressed as \( a(w_i) \) and the probability of existence as \( e(w_i) \).

Given these values we would now like to be able to introduce a communication need between actor A and B with availability expressed as a stochastic variable \( C_a \). The expected value of the outcome of a discrete sample space (we have two different possibilities for the availability) is defined in mathematical statistics as (Blom, Enger, Englund, Gardell, & Holst, 2005):

\[
E(X) = \sum_k k p_X(k)
\]

Where \( k \) the possible values for \( X \) and \( p_X(k) \) is the probability for \( k \) to occur. In our case that would correspond to:

\[
E(C_a) = \sum_w a(w)P(w)
\]

Where \( P(w) \) is the probability that way \( w \) is used for communicating. The problem is now to determine the function \( p \). By using the law of total probability we obtain (Blom, Enger, Englund, Gardell, & Holst, 2005):

\[
P(w_1) = P(w_2)P(w_1|w_2) + P(\neg w_2)P(w_1|\neg w_2)
\]

If the case were that \( w_1 \) and \( w_2 \) where two independent events, then this expression could be simplified into \( P(w_i) = e(w_i) \). However \( w_1 \) and \( w_2 \) is not independent in most cases of interoperability. In reality there are several plausible communication links available between say two persons. It could for instance be phone calls, meetings, letters and emails. If the two persons choose to communicate with email in 70% of the cases and phone in 30%, then it doesn’t mean that if one person has sent an email containing some information he or she will still make a phone call telling the exact same information in 30% of the cases. In other words:

\[
P(w_{email}) = 0.7
\]
Hence if an excepted value of availability shall be calculated based on probability of existence of two ways, the user must submit probabilities for every possible combination of existing ways. That leads to three problems:

1) It is not enough to specify a value for every entity and relationship and their probability of existence, this must be specified per path.
2) The number of probabilities that must be specified will increase quadratic (i.e. 4 possible ways needs 16 different probabilities). This will probably be very hard to find out from a company.
3) To get hold of all possible paths, even very unlikely/bad/expensive solutions has to be specified, which will make understanding of the models almost impossible, and the number of needed probabilities increase even more. The possible communication paths between two persons for example, they are almost uncountable.

A better way is probably to create one model for a handful of interesting scenarios and benchmark these, instead of spending time on estimating innumerable amounts of abstract probabilities.

6.3 Aspects of the language for interoperability assessment

Since the experience from last section is that it is not viable to create a language that will predict an actual outcome, but rather to aim at creating several plausible models and then compare them, some changes to the interoperability language is introduced.

When modeling specific scenarios it was found during this study that the language in communication is not as interesting as the data being sent. There are few times where you have a general XML to JSON translator, it is rather often the case that two systems send data in formats A and B which a third system passes on in an aggregated form in format C. If the third system where to receive some unrelated data in format A, it would not be able to do anything with it. It is only because the third system knows what is sent that it is able to do something with it. To only regard languages can therefore lead to problems with ways that seams to exist although they are not. This will be exemplified below.
Figure 19 Cell phone with limited functionality

Figure 19 shows a diagram of how a phone with limited functionality can communicate with computers and the Internet. The phone is far from today’s smartphones, so it only handles some WAP communication and syncing contacts using a cable and custom software.

The next part of the example model is shown in Figure 20.

Figure 20 Two computers exchanging contacts

Figure 20 shows how the first computer that receives the contacts from the cell phone is able to email these to another computer. Now we would like to model this very simple scenario from the figures using the language for interoperability. To start of it is done without languages in the model to easier understand the model.
Figure 21 A hypothetical scenario with two computers, one phone and the Internet

Figure 21 shows the instantiated model with some inserted attributes. The most interesting attributes are the availability of the GPRS system, which is only 20%. This is of course exaggerated; the point is that the availability should be lower than the product of the others. Because when doing the analysis it results in the attributes shown in Figure 21 in the rightmost Communication Need between the phone and the Internet. This shows that the phone reaches the Internet in 99.5% of the cases, not 20% as mentioned earlier. This is because the analysis founds a way from the phone to Laptop 1 and from Laptop 1 to the Internet. This path is not valid in practice since the phone can only synchronize contacts with the computer, not connect to the Internet.
Figure 22 The same model with languages introduced

Figure 22 shows the same model with added languages. The idea is that the phone speaks a WAP language with the GPRS system that translates it to some other format that we call GPRS. The phone exchanges contacts through the serial cable. They are translated from the internal VCF format to a serial format, which is read by the computer and again translated to an internal format, which is sent by email.

A problem is however that this doesn’t change the outcome of the analysis. Even though we specified exactly which language is spoken where, it seems that our model still has a path between the Internet and the Old phone through Laptop 1. This is because our model states that our phone and the laptop can communicate, and therefore data can be sent that route. An idea of how to fix this with minimal change to the language is to add domains to the communication needs. A domain would be the native language the communication has to occur in. It is not enough that two actors share the same format if they cannot in any way get a translated version of the domain language. Even though this solves the case illustrated in Figure 22, it is unchallenging to alter the scenario so that it still doesn’t work. If the phone can save contacts from its WAP communication (i.e. some provider has downloadable contacts on its WAP page) then the phone can suddenly translate from WAP to VCF and we are back to the same situation. This is illustrated by Figure 23.
We see that even though we strictly specify what language we want to communicate with, it is not enough. This shows that modeling languages cannot easily replace the need of modeling the actual data.

Therefore a new meta-model constructed for the language for interoperability is proposed in the next section.

6.4 New meta-model

The focus of the meta-model is to create a language that is designed to analyze the interoperability in one specific case so that the user can build several rivaling models and compare their results and from that pick one as a desired setup. As shown in the previous section it is good to focus on the data being sent, rather than the formats of it. Hence the core of the new meta-model will be data and actors shown in Figure 24.

![Diagram of Actors and Data](image)

**Figure 23** The same situation, even with added domains

The Data is at some point created, and always by an actor. It is important to model this relation. In practice it could feel superfluous since the actor that first exchanges some data, can be presumed to have created it. However there can be missing links in the path from the creator to the needing actor. It is also probably useful to keep track of which data that inherits from where in order to easier understand the communication when looking at the visualization. The Data can only be created by one Actor, and must always have a creator. The Actor however does not necessarily need to be creating any Data to be interesting in the analysis. The Actor entity
corresponds to the AbstractActor in the original language. It has also three attributes; correct, delivers and available. The attribute available corresponds to the former isAvailable. It is now called only available since it was regarded inconsistent to name for instance the attribute ‘correct’ of a LanguageTranslation to ‘correct’, and not ‘isCorrect’. The correct attribute of the new Actor entity corresponds to the old distortsMessage. This change is done since its meaning is similar to the correct attribute of a LanguageTranslation, they both tell if the message is made unreadable. It was also inconsistent to have distortsMessage, which would correspond to NOT correct, while having the correct attribute on the LanguageTranslation still be the non-negated form. To make it more consistent these two was decided to have the same name, but for different entities. The last attribute, delivers, corresponds to the old attribute dropsMessage. This attribute was also a negation, meaning that ‘true’ meant that the message was NOT delivered. Therefore this was renamed as well.

The next important elements in the meta-model are the Exchange and Transformation. This is shown in Figure 25.

**Figure 25 Added Exchange and Transformation entities**

The exchange entity corresponds to data being sent, i.e. the relation between message passing system and actor in the former meta-model. An Exchange can only occur with one Data between two actors. It is important to note that the Exchange is directed; i.e. the data flows from one actor to another. If the Data is sent back another Exchange is needed.

The Transformation entity corresponds to the old LanguageTranslation, but with the difference that it was two different languages that where being translated, in this model it is more abstracted. A transformation could be a translation between two languages; it could also be an aggregation or some alteration of the Data.
Just as the original language, this variant also has two variations of Actors (AbstractActors in the former). One is RelayActor and is the same as a MessagePassingSystem, but renamed to reflect its inheritance. An ActiveActor is the same as the Actor in the original language. This name change was made since the AbstractActor entity was better to name just Actor, since “Abstract” can be confusing. Just as the original language, an ActiveActor has addresses.

The last part is the part needed for analysis, the former CommunicationNeed. In this version the Communication need is split up into DataNeeds. These two inherit from the same base class, Need as shown in Figure 27.

A DataNeed is the need that an ActiveActor has for some Data. This is similar to the CommunicationNeed in the original language, apart from it being limited to only one Actor and one Data per DataNeed. Several DataNeeds then constitute a CommunicationNeed. A CommunicationNeed can however also be made up out of other CommunicationNeeds, or a combination of DataNeeds and CommunicationNeeds. Both the CommunicationNeed and the DataNeed inherits from the Need entity. This means that they have the same set of attributes. There is a difference however. A DataNeeds attributes is calculated similar to the original
language, but a CommunicationNeeds attributes is an aggregation of its aggregated Needs.

The analysis is performed in several steps. First the DataNeeds are evaluated. This starts with identifying the most likely route for communication. In a case where, as in the original language, the result of interest would be some kind of future prediction one would want this to calculate weights for all possible routes and then calculate an expected value for each attribute. That would be a rather complicated very much like the one discussed in section 6.2. The user would basically have to specify probabilities for each path that it is chosen (notice the difference from that it exists) for all combination of paths in every analyzed need. This would as shown before result in a plethora of probability assignments. However this meta-model is only trying to analyze one fixed case. Therefore it is no need to make combinations of attributes, just to provide a strategy for how a path is chosen.

For every path $W$ of length $n$ between two nodes we can calculate the attributes in a Need. We say that a path consist of Actors $A_1 ... A_n$ where $A_1$ is the starting Actor and $A_n$ is the end Actor, i.e. the creator of the needed data. It also consists of a set of translations $T_1 ... T_q$ where $q \leq n - 1$ since there cannot be more translations than there are actors in the path. We also say that $a_* (Q)$ denotes the * attribute (for instance available or delivers etc.) for entity $Q$.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>satisfied</td>
<td>$p_{satisfied}(W) = p_{exists}(W) \times p_{semantic}(W) \times p_{available}(W) \times p_{delivers}(W) \times p_{syntactic}(W) \times p_{addressing}(W)$</td>
</tr>
<tr>
<td>addressingSuccess</td>
<td>$p_{addressing}(W) = \begin{cases} 1, &amp; \text{addresses known} \ 0, &amp; \text{otherwise} \end{cases}$</td>
</tr>
<tr>
<td>pathExists</td>
<td>$p_{exists}(W) = \begin{cases} 0, &amp; W \subset \emptyset \ 1, &amp; \text{otherwise} \end{cases}$</td>
</tr>
<tr>
<td>semanticCorrect</td>
<td>$p_{semantic}(W) = \prod_{i=1}^{q} a_{correct}(T_i)$</td>
</tr>
<tr>
<td>pathAvailable</td>
<td>$p_{available}(W) = \prod_{i=1}^{n} a_{available}(A_i)$</td>
</tr>
<tr>
<td>messageDelivers</td>
<td>$p_{delivers}(W) = \prod_{i=1}^{n} a_{delivers}(A_i)$</td>
</tr>
<tr>
<td>syntacticCorrect</td>
<td>$p_{syntactic}(W) = \prod_{i=1}^{n} a_{correct}(A_i)$</td>
</tr>
</tbody>
</table>

| Table 4 Attributes of the DataNeed |

The attribute addressingSuccess is not defined completely, since it will not actually be used in the models in this case. The concept is meant to be the same as in the original language.

Since Table 4 defines how to calculate all attributes needed for a path we can now define strategies for choosing the best path. This strategy corresponds to a simulated
brain of the software architect who will make the decisions in the real world. A strategy function shall return a value when provided a path. The path that yields the highest value is chosen. So a strategy function that always choses the most available path would look like:

\[ s(W) = p_{\text{available}}(W) \]

Another example of a more complex strategy function would be the following:

\[ s(W) = 2 \cdot p_{\text{available}}(W) + p_{\text{delivers}}(W) \]

Which is a weighted strategy function saying that availability is twice as important as if the message is delivered. Since the strategy function aims to simulate a software engineer the function should be the same for the whole model. The path for a DataNeed that maximizes the strategy function is called \( W_{\text{max}} \). That leads to the definition of any attribute \( * \) for a DataNeed \( D \):

\[ a_*(D) = \prod_{i=1}^{n} a_*(W_{\text{max}}) \]

When analyzing a CommunicationNeed the calculations are done the same regardless of attribute, since a CommunicationNeed needs for instance all the other Needs below to be available to be classed as available itself. That means that any attribute \( * \) for a CommunicationNeed \( C \) with aggregated Needs \( D_1 \ldots D_n \) is calculated as:

\[ a_*(C) = \prod_{i=1}^{n} a_*(D_i) \]

The complete meta-model is shown in Figure 28.
6.5 Comparison

To compare the two meta-models the same example is described as in section 6.3, but with the new meta-model proposal. The instantiation begins with adding Actors and their exchanges; the result is shown in Figure 29.

![Diagram of example with new meta-model](image)

Figure 29 The example with the new meta-model

The largest difference from the original language is so far that this model contains much more entities since the exchange entity was not needed previously. However this changes when the Data and Transformations is added as shown in Figure 30.
Figure 30 The new model with added Data entities and Transformations

If a comparison is made to Figure 23 we can now see that the biggest difference is the lack of language names in the later. This could make the model easier to grasp since it from the beginning was the actual data that had the focus, the languages had to be investigated in order to make the first model. The last step is to add the DataNeeds, the former CommunicationNeeds. This is shown in Figure 31.
Figure 31 The complete model

Note that the DataNeeds automatically has the ‘domain’ that was introduced earlier since it connect a Data with an Actor, and by that also specifying the other party in the communication. However this model looks quite similar to the one used earlier. How come this one is giving correct values? The answer is that a big difference is that both the Exchange end the Transformation is directed, meaning that if the Transformation or Exchange is made both ways a contra entity needs to be added. This makes sure that only real paths are possible to use.

6.6 Comparison in the SRS case

In the Pre-analysis section the original interoperability language was used to analyze the SRS projects future IMS and the interoperability around it. Now that some improvements and problematic areas has been discussed and somewhat solved, this section will show how the improved meta-model influences the analysis.

The basics of the languages are quite the same. Therefore now big differences are to wait. The improvements has only affected areas where there is two or more possible ways to communicate, and often there is only one way in the SRS project (often an external system through the internet and to the IMS). There is however one case where this is of interest. The solar panel values are not sent directly to the IMS as the other values. The solar panel data in M-Bus is first sent to the home ESI, which then forwards it to the IMS. This may sound like an unnecessary step, but that is how it is designed, so we want the model to reflect that. The solar panel data is a part of the
CommunicationNeed “Power consumption impact” since we want to subtract the generated power from the consumed before converting it into kg CO$_2$.

A comparison of the satisfied attributes in the different CommunicationNeeds in the two versions of the languages is shown below in the two first columns in Table 5. The comparison is based on the model with all IT-systems, 2 buildings with 10 apartments each. The model is unfortunately very large, why it is hard to show it in this report.

<table>
<thead>
<tr>
<th>Satisfied attribute of</th>
<th>Original</th>
<th>Improved (home)</th>
<th>Improved (building)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power consumption impact</td>
<td>0,560</td>
<td>0,556</td>
<td>0,560</td>
</tr>
<tr>
<td>Price forecast</td>
<td>0,999</td>
<td>0,999</td>
<td>0,999</td>
</tr>
<tr>
<td>Heat consumption impact</td>
<td>0,814</td>
<td>0,814</td>
<td>0,814</td>
</tr>
<tr>
<td>Waste impact</td>
<td>0,843</td>
<td>0,843</td>
<td>0,843</td>
</tr>
<tr>
<td>Transportation impact</td>
<td>0,998</td>
<td>0,998</td>
<td>0,998</td>
</tr>
<tr>
<td>CO2 Analysis</td>
<td>0,999</td>
<td>0,999</td>
<td>0,999</td>
</tr>
<tr>
<td>Water consumption</td>
<td>0,688</td>
<td>0,688</td>
<td>0,999</td>
</tr>
</tbody>
</table>

Table 5 Comparison of the satisfied attributes

This shows that when it is possible to specify routes for paths we see that the satisfied level drops. This is because we now have the tools to make the solar panel data to go through the home ESI instead of directly to the IMS like the original language forced it to. This makes it possible to create two comparable scenarios in the improved language; one where the data is sent via the home ESI, shown in the ‘Improved (home)’ column, and one where the data is sent directly to the IMS, shown the ‘Improved (building)’ column.

This shows that we can create models that show the same results as the original language with the improved one. The difference is that the user is now aware of how these differences occur, and how to investigate them.

6.7 Import learnings to the original language

In the last section it was found that modeling data instead of languages could lead to a different model. It was also found that directions of the entities Transformation and Exchange solved the problem discussed in section 6.3. Since the models are a bit alike it would be ideal if these adjustments where made to the original instead of creating a new language. The change that has to be made is adding directions to Translations, so that a translation translates from one language to another. CommunicationNeeds will also have to be modified to have one starting point and one endpoint, otherwise it is pointless to direct Translations since it cannot be determined from which way the translations should go. Since the Exchange entity does not exist in the original language, these cannot be directed. However in practice this would only work as long as a system does not translate two different types of data expressed in that same language in different directions. A system may for instance send events in XML one way, and contacts in XML the other way. That
would bring back the discussed situation again. That is not an issue in the new meta-model, since it is concerned with the actual data.

It could be argued that just saying that a data is expressed in ‘XML’ is not precise enough, and that these kinds of problems could be solved by modeling that two systems exchange ‘Phone contact XML’ with each other. However that would be misusing the Language entity as a Data entity. Then it would be wiser to use the new meta-model and focus on that exact communication.

6.8 Visualization

Even if the models are correct from an analyzing point of view, another very important aspects of modeling are the communicability of a model (Lankhorst, 2009). Therefore this section suggests a visualization of the model that is easier for a stakeholder to understand. The visualization aims to visualize the new meta-model proposal in an easier way. The first step is to visualize Actors and Their Exchanges. The Exchange is an entity, but it serves as a relation as well.

![Figure 32 Better visualization of Actors and Exchanges](image)

Figure 32 shows how the Exchange is visualized as a semi entity. The ActiveActor and RelayActor are quite similar to point out that they have the same inheritance. The next step is to visualize data.

![Figure 33 The Data and Transformation visualization](image)

The Data and the Transformation is bot connected to an ActiveActor to show where it is performed/created with grey lines since it is shown for reference purpose. The more important Data transformation path is however solid lines to state its importance for the model. The Data is also attached to Exchanges to show what is exchanged. Last we have the DataNeed visualization.
The DataNeed visualization features arrows to show that the data is supposed to go to the Actor. This makes it easier to see the desired flows. A simplified meta-model showing the meta-model is shown in Figure 35.

As an example the situation used earlier in this chapter is used in the visualization shown in Figure 36.
6.9 Other considerations and further research

Besides what has already been discussed there are some other shortcomings found during the analysis. These have no direct solution today, but rather is mentioned as potential future enhancements to the language.

Data influence on Transformation
When a Transformation between one Data and another is made there can be a problem when the Data is transformed in the matter that it is merged with another Data, rather then creating a new data set. This is the case in the example earlier when the phone saves contacts from the WAP page. Then just a few of the contacts on the phone are from the WAP page, and if they are distorted that doesn’t mean that all contacts saved on the phone is distorted. Therefore it could be useful to be able to estimate how much impact the different Data entities have on one Transformation to get a better analysis.
Acceptable thresholds
In the case of the Royal Seaport project a lot of metering data shall be collected. All metering data is not however crucial to get. If the system for instance retrieves 99 metering values, but 1 fails, then the data is still useful. That is probably the case even if just 80 meters delivers values; however somewhere there lies a threshold when the data is not useful anymore. Maybe 20 meter failures once a year is acceptable, but not constantly. Hence it would be useful to gain another dimension of the analysis showing plausible outcomes.

This could be done by aggregating say for instance 100 DataNeeds for some metering values in one CommunicationNeed. Then it would be possible to calculate a normal distribution describing how many system that is, for instance, available in a given moment.

More or less important Needs
When aggregating Needs into CommunicationNeeds a useful feature would be to, as with the Transformations, specify the importance of the different Needs for the whole CommunicationNeed. In the Royal Seaport project this would reflect as some Data values are easier to estimate then others and hence less important for the CommunicationNeed to become more satisfied.

Resolution
A common aspect of the communication that was heavily discussed when collecting the information about the Royal Seaport project was the time resolution of the data. That means that if two data sets needs to be aggregated (i.e. calculate how much CO$_2$ that the electricity consumption corresponds to for one household) it can arise problems when the CO$_2$ data is retrieved on a monthly basis, while the electrical power consumption data is measured per hour. As of now there is now way to indicate time, or other types of resolutions, in the interoperability language.

Assumption of independence
Though some of the problems with incorrect assumptions of independence are resolved, some still remain. Say for instance that a system A has DataNeeds to two independent systems B and C. If this data were sent using the same MPS, then a CommunicationNeed aggregating the DataNeeds would simply take the product of the attributes of the underlying DataNeeds. This will however assume that the two paths are independent, which they could be said not to be since they both use the same MPS, and are thereby both affected if that MPS fails. However if the probabilities are regarded as “per transaction” and the communication is synchronous, the paths can still be said to be independent. This needs however to be more thoroughly investigated and defined for the language.

6.10 Case study principles fulfillment

There was decided in the method chapter that the analysis of the interoperability language should be based on Yin’s principles for case studies and using these to validate the usefulness and accuracy of the interoperability language.
Construct validity assessment
The threats identified against construct validity earlier was the risk that the usage of the language would force the user to design a solution on beforehand or that problems got exaggerated. The language does not force a design in the fashion that the language is intended to only model potential data flows. However since it has proven to be difficult to make that kind of analysis work, the user will probably be forced to design one or multiple solutions to be able to perform the analysis.

The other risk was problems being exaggerated by using the interoperability language. This is somewhat fore-filled due to not being able to specify thresholds for instance. This namely leads to the interoperability language pointing on a problem, which would not be a problem if the user were able to specify thresholds.

Internal validity assessment
The main concern for internal validity was identified to be that not all data collected was able to fit in the analysis. As discussed in the previous section this was partly the case since the interoperability language does not handle time resolution of data. Another factor was that it is not possible to specify that a communication path already is set, that it form some reason is not possible to route the data any other way due to politics for instance.

This is also related to which data that was collected. The data collection was affected by the author’s knowledge about the interoperability language. Therefore the data collection could have had a different focus if another language was used, or not known at the point of data collection, and therefore end up with another result. However, data collection without knowing how to analyze the data would probably be a tedious task with little or no reward.

Reliability assessment
The reliability part was supposed to be assessed by how unequivocally the result of the interoperability language analysis can be said to be. It means that the same results should be achieved whoever constructed the models using the data collected.

The modeling language is open for quite broad and general uses of the entities. A language for instance can represent everything from a broad family of languages to almost represent the data that is being sent. This leads to many possible models given the same data. However since this scenario concerned a future situation it would probably be hard to be even more specific, and the analysis would be harder to do if the model where at a even higher level, since it would be hard to get some actual results beside the model. At this level the attributes for the different entities where assessed why the analysis could be performed.


7 Discussion

7.1 How the questions where answered

This chapter will evaluate the analysis by the initial goals and methods chosen for the analysis. The analysis had its outset in the question presented in the Goals and Purpose section. The main question was:

*How does the modeling and assessment language for interoperability proposed by Ullberg et. al. performs in the Stockholm Royal Seaport project, and can any changes or additions be made to the language to improve the models usefulness and/or accuracy?*

To answer this question 4 sub-questions where defined. How well these are answered will be analyzed below.

*Could the data collection been done any different?*

This topic was discussed in the last section in the analysis. The conclusion was that the data probably could be collected differently, but that it would not be very realistic to collect data without knowing the purpose.

*Could the instantiation of the model been done any different?*

How the model was instantiated was also a topic discussed in the last section of the analysis. The conclusion about this was that it probably could not been done very differently, at least not on the same level of detail.

*What other models/viewpoints could be used?*

The analysis mainly aims to investigate how well the language served its purpose. It became quite clear during the theory study that languages that touch upon interoperability are quite few. Therefore it seemed reasonable to focus on how the current language could be improved.

*What would make this language easier/better to use?*

This question is the question that was the main theme for the analysis chapter. In the analysis some ideas where proposed in how the interoperability language could be modified to better be able to model the scenario studied.

7.2 Improvement areas

To ensure the validity of the Result Summary a draft should have been shared with key informants that have been involved in the data collection. This could have been people that had been interviewed about a particular system, or perhaps colleagues to an interviewee. This should be to ensure the correctness of the data collection, and to minimalize misunderstandings. The optimal method would probably be to arrange a workshop where all the interviewees can sit together and discussed the finalized material, this can however be hard to arrange in reality. This was not done with the finalized model, but with the model shown in Figure 12.

On top of this the results should probably have been discussed with an interoperability expert in order to better understand how the could be improved and if it identifies the correct problem areas.

- 54 -
7.3 Study design

The results obtained in this study can of course be discussed. The outcome of the analysis was basically a new meta-model to use instead of the one proposed originally, even though they share a lot of common thoughts. This conclusion can be seen as rather radical. This section aims to discuss what could have been done differently in order to not have to make such drastic changes in the original language.

One area that can lead to obtaining different results is the data collection. For instance: the problem that the interoperability language aims to solve was misinterpreted. In that case the data collected will not be the optimal data for constructing the model with the language, and hence it will seem like the modeling language is insufficient. If this is the case, then there is a problem in the documentation of the interoperability language, since it could mean that a user does not understand the intended usage of the language and therefore misuses it.

Another problem in the data collection could be that the wrong stakeholders where interviewed in the data collection phase. If the people interviewed was not technical or involved enough it is possible that their answers where incorrect which then led to a false model. In this case however the interviews has been conducted with as technical people as possible available.

There is also a possibility that there are frameworks for assessing interoperability, which is unknown for the author. This would then affect the study in the way that there can be more refined ways of performing the analysis, and other aspects that is important to take into account when designing a meta-model for interoperability.

8 Conclusion

The language for interoperability has proven to be a valuable tool in the enterprise architecture toolbox. It has provided insights in the design of a future large-scale information system and its environment in the Royal Seaport project, where this language has been used. However it has also been shown that even though the language is very competent, it may need a change in concept for some scenarios to better being able to handle and fit in the information available in the analyzed architecture.

This master thesis has therefore suggested a variation of the language which instead of focusing on designing a language that is able to analyze an unbiased version of the reality and based on this provide answers in how future scenarios will look like analyzes interoperability with a fixed perspective. This fixed perspective means that different solution alternatives are chosen and these are then used to create comparable models to be able to see which solution is likely the perform the best.

Besides the variation of the language, a visualization of the language was also proposed in order to easier communicate the models to the stakeholders since communicability is a very important aspect of modeling.
9 REFERENCES


Appendix A. Questionnaire

- The future IMS will need data X from your organization. Which system is responsible for that data?
- Where does the data come from, i.e. what other systems does this rely on?
- What technologies does these systems communicate through, and how?
- How certain are you that this information is correct?

These questions should be answered for the systems identified above (incl. message passing systems):
- How likely is the system in question to distort a message?
- How likely is the system in question to drop a message?
- How available is the system?

For every language translation that occurs in the chain:
- How likely is this language translation to translate a message correct?

This will be concluded with:
- Any other information regarding the discussed systems that is of interest for the interoperability aspect?
- Can you think of any other actors of relevance for this type of data interchange?