**Specifying abstract requirements of buildings using Knowledge-Based Systems**

***Towards automated requirement verification using Open BIM***


By: R.A. (Ruben) Bultman

0813842


Master Thesis

Construction Management and Engineering

2018-2019

Eindhoven University of Technology

*February 2019*

*Supervisors:*

Prof. Dr. Ir. B. (Bauke) de Vries – TU/e
Ir. J.J. (Jakko) Heinen – TU/e and BIM-Connected
Ir. Ing. A. (Aant) van der Zee – TU/e

Date of final presentation: February 20, 2019

## Colophon

*General*
Report : Specifying abstract requirements of buildings using Knowledge-Based Systems, towards automated requirement verification using Open BIM

Date : Monday 18th February 2019
Date Presentation : Wednesday 20th February 2019
Place : Eindhoven, The Netherlands

*Student*
Author : Ruben Arthur Bultman
Student Number : 0813842
E-mail : r.a.bultman@student.tue.nl
rubenbultman@hotmail.com
University : Eindhoven University of Technology
Master Track : Construction Management and Engineering (CME)
Chair : Information Systems in the Built Environment (ISBE)

*Graduation Company*
Company : BIM-Connected

*Supervisors*

*Chairman*
Prof. Dr. Ir. B. (Bauke) de Vries
TU/e

*First Supervisor*
Ir. J.J. (Jakko) Heinen
TU/e and BIM-Connected

*Second Supervisor*
Ir. Ing. A. (Aant) van der Zee
TU/e

Construction Management and Engineering
Eindhoven University of Technology
De Wielen
Postbus 513, 5600 MA Eindhoven
Tel. +31 (0)40 247 5051

BIM-Connected
Gebouw Anton, eerste verdieping
Torenallee 62-22
5617 BD, Eindhoven
Tel. +31 (0)6 5333 9382

## Preface

This master thesis represents the end to successfully finish the 4TU Master Construction Management and Engineering at the University of Technology in Eindhoven. The results during my Master career including this thesis report make me proud and determined to continue my career in the domain of Architecture, Engineering and Construction. After finishing my Bachelor in Architecture, Urbanism and Building Sciences I continued with this Master to develop skills in the field of process-, project- and information management. During several internships and followed courses I found the broadness and variety of knowledge within this field that interested me.

This graduation research had been conducted under the supervision of Jakko Heinen, as founder of BIM-Connected and lecturer at the University of Technology. I would like to thank him for his appreciation, support, feedback, guidance and the opportunity to use his connections and knowledge as implementation of this master thesis. Also for the extensive advice to conduct the next step again and again and helping me with programming to complete this master thesis.

Next I want to thank Prof. Dr. Ir. Bauke de Vries for his feedback and support during the meetings at the University. For his inspiring and challenging conversations. And even more for his innovative and up-to-date providing of possibilities that could help me further with my research. Also I want to thank Ir. Ing. Aant van der Zee for the time invested in helping and thinking with me. Even for the discussions about the concepts used during this research on which we disagreed. And I am also grateful for all the supervisors that believe in me about this difficult and abstract topic.

Malco van den Eijnde from HEVO B.V. and the interviewees, including the ones who had arranged several interviews for me, are highly appreciated for their contribution to this thesis. Without them it would not be possible to finish my Master and develop and test my research against a Building Information Model and collecting data from practice. But certainly also all my friends and family for their support.

Finally, I want to thank my other colleagues from BIM-Connected and all others from StrijpCS for their interest in my research during my graduation period. And perhaps even more for the good time these six months with lunches, parties and the table soccer games which remained challenging till the end.

I hope this thesis contributes, however small, and provides more insight to everyone who deals with these developments in the future, and that every reader reads with pleasure and learns as much as I have learned.

Ruben Arthur Bultman

## Table of contents

# Abstract

**Keywords:** Building Information Modeling (BIM), Requirement Processing, Experts, Validation and Verification process, Abstract Requirements, Knowledge-Based Systems (KBS), Expert Systems (ES), Proof of Concept (PoC), automated verification.

**Abstract:** Enhanced by the development of Building Information Modelling (BIM) and related to more data that is captured, an international shift from the traditional building process to a more integrated way of working takes place. The traditional construction process resulted in many ambiguities and inconsistent information transfers. Problems in buildings can often be traced back to the requirements processing in the design phases. In here the requirement specification must be validated with the client in order to make sure the contractor and the client are on the same level to reduce any discussion about it in a later stage (Hoeber, 2012). These requirements are often just a list in a word processor program and are not explicit formulated, but having a lack of consistency and unambiguity and no standard procedure of dealing is formulated. The interpretation and implementation of these abstract requirements is based on the combination of the available information and reasoning of experts. This results in a continue iterative process of defining, validating and verifying requirements. This process with human experts has several difficulties in which knowledge-Based Systems (KBS) could possibly play an important role. It is worth investigating if these systems indeed can improve the specification process of abstract requirements, by developing an Expert System (ES) as Proof of Concept (PoC). A model will be developed to show a new procedure and implementation via a pilot case. The development is based on the objective in this research to investigate if it is possible and if this way can improve the requirement specification in the design process. Implementation extends the research into automated verification to further improve the quality of building designs and innovation without sacrificing these qualities (Solihin & Eastman, 2015).

## Summary

Since the introduction of integrated contracts in the construction sector a shift of responsibilities has occurred. Leading factors in this are reducing (failure) costs and provide a higher quality in the designs. The different forms these integrated contracts know can provide, via functional specification, parallel activities. This can result in parallel execution of the activities. Functional specification finds its origin in the method of Systems Engineering (SE) at which a functional analysis has to lead to functional requirements. The drawing and using of a functional specification is complicated. Important is mapping out the problem and the need of end users, stakeholders and own organization. Often is worked with interdisciplinary design teams who need to keep in mind the final result. Without thinking in concrete solutions and the corresponding technical specifications. However, it remains important that the functional specification must be sufficiently detailed described to guarantee unambiguity, consistence and measurability. By using the method of SE the design is created in a systematic way and makes use of top-down reasoning.

Describing requirements and functions starts often early in the development phases of a project and has a peak during the design process. The different requirements will become more and more specific during a design. Various requirements will be formulated at which is worked from client specific requirements via system requirements towards the final demand specification. Out of the method of functional specification a distinction is usually made between objects, spaces and requirements. This package of requirements represent often requirements that are difficult to design, to make measurable and are undefined. It is often about certain qualities that are imposed to a design that cannot unambiguously formulated without discussing about. A distinction can be made between abstract, undefined and not-computable requirements and specific, unambiguously and objective measured requirements. During the theoretical research there will be looked in-depth what an abstract requirement entails and where potential challenges are during specification. A requirement to which a certain value is connected will raise fewer questions than a certain intangible quality that is connected to a certain function, object or as requirement. This raises often questions about what is really meant or the precise meaning to a certain end user. It is important to manage the process of requirement specification properly in order to guarantee the design assumptions. The translation of abstract formulated requirements into a more specific interpretation of that requirement is about something different than the real signification of a requirement. In this research, abstraction is defined as a certain quality which is desired from a client, but contains no specific and usable data. Such as the requirement of comfort, security or sustainability to a certain space or building. Often because the client does not know yet why or what his needs are. The translation of this abstract formulated requirement towards a specific requirement often causes a lot of discussion and ambiguity between contractor and client which causes errors in a later stage with exceedances of budget and time. The problem can be found in the fact that these descriptions have too little hold to make decisions within the design. Out of the literature several articles are dedicated to what abstract requirements include and what conditions they must met. But what are the possibilities to optimize this process and have a correct result of the translation? When requirements are formulated that are clear and measurable the execution of validation and verification is easier. The process of validation examines if the right thing is made and if this corresponds the need of the client. This therefore involves the mutual understanding of contractor and client regarding the requirements which is not always the case with abstract formulated requirements. This research has

worked towards a way to improve this process and optimize the translation of these requirements. Currently experts with extensive experience are the ones who are going into conversation with clients to specify requirements. All these experts together have a huge database filled with knowledge, experience and expertise. But every expert and client is different even as every project they work on. This often causes slightly different results which influence the satisfaction of the client. By bundling this knowledge and using it within this process a clearer way can be found to work towards a correct specification and more successful process of validation and verification. In this research is looked to ES's which are concerned, in general, with taking decisions, something that is always an issue with requirements specifications. Of course it is not that simple to capture all possible decisions with corresponding results and all the experts knowledge. But nevertheless the use of such systems can sharpen the process and ensure less failures during these requirement specification process within the design phases. Out of this idea a model is developed to investigate if knowledge-based systems can contribute to the requirement specification process to result in computable values. The objective of this model is to show with the use of an example how the translation and specification can take place. Out of the interviews an overview is presented in which advantages and disadvantages are given of the current approach and possibilities of knowledge-based systems.

In existing research into automated verification until now only investigation have taken place regarding measurable requirements. With increasing developments and the use of BIM and the connected information streams, verification of the abstract formulated requirements becomes more important and are worth more. Though, the requirement must contain a value including an unit to use it for verification via BIM. Out of the developed model the result is used to show via a case how automated verification can take place. This automated verification is based on a query which can check a measurable value in a BIM model. The idea of specification using Knowledge-Based Systems gives the reason to work towards a Proof of Concept. With this the possibility is investigated how to deal with abstract requirements in the construction sector. Knowledge about the current approach and the possibility as well as the information for the input of the Knowledge-Based System (KBS) are the result of interviews with experts. This results in a research on the current dealing with abstract requirements and possible improvements. These improvements are framed by presenting how abstract requirements can be translated to a description which is more consistent and usable for automated verification. Which in turn can lead to a positive result during practice regarding budget, time and less errors as due to the requirement specification process. The research shows that a KBS indeed can be used to accelerate, clarify and ensure valuable results that lead to less questions and changes in later stages of the design process. However, it is of interest before actual implementation, more research and an extensive development must be conducted.

## Samenvatting

Sinds de introductie van geïntegreerde contracten in de constructiesector heeft er een verschuiving plaatsgevonden van verantwoordelijkheden. Leidende factoren hierin zijn het reduceren van (faal)kosten en zorgen voor een hogere kwaliteit binnen de ontwerpen. De verschillende vormen die geïntegreerde contracten kent, kunnen via het functioneel specificeren zorgen voor parallel aan elkaar lopende activiteiten. Deze kunnen hierdoor ook parallel aan elkaar worden uitgevoerd. Het functioneel specificeren vindt zijn oorsprong in de methode SE waarbij een functionele analyse moet leiden tot functionele eisen. Het opstellen en gebruiken van een functionele specificatie is ingewikkeld. Van belang is het in kaart brengen van het probleem en de behoefte van eindgebruikers, stakeholders en de eigen organisatie. Vaak wordt hierbij gewerkt met interdisciplinaire projectteams welke het uiteindelijke resultaat in gedachten moeten houden. Zonder daarbij te denken in concrete oplossingen en de bijbehorende technische specificaties. Echter is het wel van belang dat de functionele eisen voldoende gedetailleerd zijn omschreven om eenduidigheid, consistentie en meetbaarheid te waarborgen. Via de methode SE wordt op een systematische werkwijze ontworpen en top-down geredeneerd.

Het beschrijven van eisen en functies begint al vroeg in de ontwikkelingsfase van het project, en kent zijn piek gedurende het ontwerpproces. De verschillende eisen zullen gedurende het project steeds concreter worden. Verschillende eisen zullen worden opgesteld gedurende het specificeren waarbij van klanteisen via systeemeisen naar de uiteindelijke vraagspecificatie wordt gewerkt. Vanuit de methode van functioneel specificeren word er meestal onderscheid gemaakt tussen objecten, ruimten en eisen. Binnen deze eisen zijn er veel abstracte omschrijvingen van vaak moeilijk te ontwerpen, meetbaar te maken en ongedefinieerde eisen. Het gaat vaak om bepaalde kwaliteiten die worden gesteld aan een ontwerp welke niet eenduidig zijn zonder hier op in te gaan en deze om te schrijven naar eisen die bruikbaarder zijn in deze ontwerpen en voor uiteindelijke verificatie. Er wordt een tweedeling gemaakt in abstract, ongedefinieerde en niet-berekenbare eisen en specifiek, eenduidig en objectief gemeten en geformuleerde eisen. Tijdens het theoretisch onderzoek zal er gekeken worden wat een abstracte eis inhoudt en waar moeilijkheden zitten tijdens de specificatie. Een eis waarbij een bepaalde waarde wordt gesteld aan een duidelijk omschreven functie, object of eis zal voor iedereen minder vragen oproepen dan een bepaalde ontastbare kwaliteit die aan eenzelfde functie, object of eis wordt gesteld. Vaak roept dit vragen op wat er precies mee wordt bedoeld of voor de eindgebruiker nu echt betekent. Het is van belang om het proces van eisenspecificatie van abstracte eisen in goede banen te leiden om zo de project uitgangspunten te waarborgen. Het omzetten van een abstracte eis welke nog geen diepgang heeft aan het begin van het project naar een concrete invulling van deze eis is daardoor zeer belangrijk. In dit onderzoek wordt naar abstractie gekeken als een eis die een bepaalde kwaliteit omschrijft die vanuit de opdrachtgever gewenst wordt maar geen specifieke en dus bruikbare gegevens bevat. Zoals de eis van comfort, veiligheid of duurzaamheid aan een bepaalde ruimte of gebouw. Vaak doordat de opdrachtgever ook (nog) niet weet waarom hij dit wil of wat er gewenst is. Het vertalen van deze abstracte eis naar de specifieke eis zorgt veelal voor veel discussie en onduidelijkheid tussen opdrachtnemer en opdrachtgever wat fouten in een latere fase kan opleveren met overschrijdingen van budget en tijd tot gevolg. Het probleem is te vinden in het feit dat deze omschrijvingen te weinig houvast bieden om beslissingen te nemen ten aanzien van het ontwerp. Vanuit de literatuur worden vele artikelen gewijd aan wat abstracte eisen inhouden en aan welke voorwaarden ze moeten

voldoen. Maar wat zijn nu de mogelijkheden om dit proces te optimaliseren en bij een juiste vertaling uit te komen? Wanneer eisen wel duidelijk en meetbaar geformuleerd zijn verloopt de validatie en verificatie hiervan een stuk gemakkelijker. Het validatieproces waarbij het gaat om dat het juiste is gemaakt, gaat in op de behoefte van de klant waaraan voldaan moet zijn. Hier gaat het dus om het feit dat opdrachtnemer en opdrachtgever elkaar begrijpen. Gedurende dit onderzoek is toegewerkt naar een manier om dit proces te kunnen verbeteren en de vertaling van deze eisen te kunnen optimaliseren. Op dit moment zijn het experts met ruime ervaring die vaak samen met de opdrachtgever in gesprek gaan om deze eisen te specificeren. Al deze experts samen hebben een enorme database aan kennis, ervaring en expertise bij zich. Maar iedere expert en opdrachtgever is anders evenals ieder project. Dit zorgt vaak voor wisselende uitkomsten en zet de tevredenheid van de klant onder spanning. Door deze kennis te bundelen en te gebruiken binnen dit proces kan duidelijker en overzichtelijker worden toegewerkt naar een juiste specificatie en daarmee ook een succesvoller validatie & verificatie proces. In dit onderzoek wordt gekeken naar expert systemen welke zich, in algemene zin, bezig houden met beslissingen nemen, iets wat bij de eisenspecificatie ook altijd aan de orde is. Natuurlijk is het niet zo eenvoudig om alle mogelijke beslissingen en bijbehorende resultaten gebaseerd op alle kennis te vangen in een systeem. Maar desalniettemin kan de inzet van dergelijke systemen wel het proces verscherpen en zorgen voor minder fouten en onduidelijkheden gedurende eisenspecificatieprocessen tijdens de ontwerpfasen. Vanuit dit idee wordt er een model opgezet om te onderzoeken of kennis-gebaseerde systemen een bijdrage kunnen leveren aan het eisenspecifictieproces om naar meetbare waarden toe te werken. Dit model heeft als inzet om door middel van een voorbeeld te laten zien hoe de vertaling en specificatie kan plaatsvinden. Vanuit interviews wordt een uitgebreid overzicht gepresenteerd waar voor- en nadelen van het huidige proces zitten en waar kennis-gebaseerde systemen mogelijk van waarde kunnen zijn.

In aansluiting op al bestaande onderzoeken wordt er ingesprongen op automatische verificatie, waar tot dusver alleen maar naar al meetbaar omschreven eisen is gekeken. Met de toenemende ontwikkelingen en het gebruik van BIM modellen en de informatiestromen die hier bij horen, wordt het verifiëren van de geformuleerde eisen steeds belangrijker en meer waard. Hiervoor moet de eis echter wel een waarde en eenheid bevatten om ze te kunnen verifiëren middels BIM. Vanuit het opgezette model wordt het resultaat gebruikt om via een case te laten zien hoe dit automatisch geverifieerd zou kunnen worden. Deze automatische verificatie is gebaseerd op een query dat meetbare waarden checkt in een BIM model. Het idee van specificatie op grond van kennis-gebaseerde systemen, geeft de aanleiding om naar een Proof of Concept toe te werken. Waarmee de mogelijkheid wordt onderzocht hoe om te gaan met abstracte eisen in de bouwsector. Kennis over de huidige aanpak en de mogelijke inzet alsmede informatie als input voor het systeem zelf komen vanuit interviews met experts. Dit resulteert in een onderzoek naar de huidige omgang met abstracte eisen en hoe dit mogelijkerwijs kan worden verbeterd. De verbetering zit in het feit dat dit onderzoek laat zien hoe abstracte eisen te herleiden zijn tot een omschrijving die consistenter is en bruikbaar voor automatische verificatie. Wat tot positief resultaat kan leiden in de praktijk ten aanzien van budget, tijd en minder fouten die het gevolg zijn van het eisenspecificatieproces. Het onderzoek laat mede zien dat kennis-gebaseerde systemen wel degelijk bruikbaar zijn om specificatieprocessen te versnellen, verduidelijken en te zorgen voor waardevolle uitkomsten die tot minder vragen en wijzigingen leiden in latere fasen van het ontwerpproces. Echter is het wel van belang dat voor dit daadwerkelijk geïmplementeerd kan worden er meer onderzoek en uitgebreidere ontwikkeling nodig is.

# 1.0 Glossary

## 1.1 List of figures

## 1.2 List of tables



## 1.3 List of listings

### 1.4 List of Abbreviations

| | |
|---|---|
| 3D | Three Dimensional |
| AEC | Architecture, Engineering and Construction |
| AI | Artificial Intelligence |
| BIM | Building Information Modeling |
| BMC | BIM-based Model Checking |
| BNA | Brancheverenging voor Nederlandse Architectenbureaus |
| CAD | Computer Aided Design |
| CE | Conditional Element |
| CLIPS | C Language Integrated Production System |
| CRS | Customer Relationship System |
| DNR-STB | De Nieuwe Regeling - StandaardTaakBeschrijving |
| ES | Expert System |
| FAQ | Frequently Asked Questions |
| GUID | Global Unique IDentifier |
| ICT | Information and Communication Technology |
| IEC | International Electro technical Commission |
| IEEE | Institute for Electrical and Electronics Engineers |
| IFC | Industry Foundation Classes |
| INCOSE | International Council On Systems Engineering |
| ISO | International Organization for Standardization |
| KBS | Knowledge-Based System |
| KM | Knowledge Management |
| KR | Knowledge Representation |
| LHS | Left-Hand Side |
| LOD | Level Of Detail |
| NEN | Normalisatie En Normen |
| NFR | Non-Functional Requirement |
| NL | Natural Language |
| PoC | Proof of Concept |
| QAS | Question Answer System |
| RBS | Rule-Based Systems |
| RE | Requirements Engineering |
| RES | Rule-based Expert Systems |
| RHS | Right-Hand Side |
| RM | Requirement Management |
| ROI | Return On Investment |
| SE | Systems Engineering |
| SMART | Specific, Measurable, Attainable, Realizable and Traceable |
| SOR | Statement of Requirements |

## 2.0 Introduction

In the past, different approaches have been implemented to involve the user in the design process. Approaches such as *Multiple Choice Housing* and *User-Driven Design* (Niemeijer, 2011) are just two approaches that become impractical due to the variants that have to be designed. Another variant that has been used is the traditional one-on-one meeting with the architect to design a house. This resulted in time-consuming meetings within large and complex projects with many stakeholders and even more wishes and different visions.

To allow clients in nowadays design processes, as non-experts in most cases, to design a building introduces a new problem; being unware of all the regulations and the architect's vision. Blueprints are not well suited to this task, as they are very labor-intensive to check for errors (Niemeijer, 2011). These errors consist among other things of undesirable results. Ideally, a large part of the design verification could be done by computers. A computer can check many of the building regulations because these prescribe criteria that can be easily and objectively measured, such as the maximum height of a building element. A computer cannot asses the (e.g. aesthetical) quality or practicality of a design. The process of validation and verification in the design phase are a project's largest costs. Failing to adequately plan for validation and verification from the beginning of the project places the project at high risk for cost overruns and schedule slips (Wheatcraft, 2012). The verification process proves that the designed and built system meets its requirements. These requirements that are written can be translated into rule sets for checking if the system meets its requirements. The process of verification early on in the design phase can contribute significantly in reducing the risks concerning costs and schedules.

New criteria continuously emerge in the Architecture, Engineering and Construction (AEC) industry, ranging from building codes to all different kinds of other requirements (Eastman, Lee, Jeong, & Lee, 2009). These requirements of construction projects are derived from institutions, users and other stakeholders. Some are imposed and others based on preferences and wishes. The interpretation and dealing with these requirements is (partly) a process of cognitivism, and is therefore difficult to describe a valid method or technique for. Clients describe their "requirements" often in a way of abstract, undefined and non-computable statements that must be interpreted and translated by experts to make these usable for verification.

Currently, checking whether all of these rules have been satisfied is, in most cases, still done manually. Due to the large amount of rules in most projects this is very labor-intensive. A lot of research has been done to automate this checking process via formalizing these rules into constraints, and would greatly benefit this phase. Although a large class of rules can be formalized, there are exceptions. Most of these exceptions are having no single accepted definition and resulting in a lack of formalization.

Compared to other industries, the building industry has seen little adoption of constraints, at least not in the sense that they are automatically checked (Niemeijer, 2011). Building designs have to comply with a multitude of constraints, such as building codes and functional and technical requirements that follow from a client's brief, where verifying these is still a manual process in most cases. According to some efforts in checking requirements automatically, little efforts have been initiated in associate abstract, undefined and non-computable statements, and automated verification.

## 2.1 Motivation

The building industry is not entirely devoid of examples of constraint usage, but the adoption of constraints lags significantly behind that of other industries. This can be attributed to different reasons (Niemeijer, 2011); *first* is the slow adoption of Computer Aided Design (CAD) in architectural design in general. BIM, used exclusively in the design and construction of commercial buildings, can make automated constraint checking a lot simpler due to its information processing capabilities. *Second* is the fact that in architecture, more than in other domains, the design is treated holistically rather than as distinct components that are designed by separate teams. And the *third* issue is described as the existence of relative prevalence of non-functional rules in architecture due to the stronger focus on for example aesthetics and other qualitative attributes.

Regarding the first reason it is difficult to accelerate the adoption process of CAD. The second reason is the reason that makes the building industry projects very complex. However, the third reason is specific to the building industry and underexposed in scientific literature. And that while automated verification relating to BIM can really be improved by extending the rules that can be verified automatically. Which results in a more effective, less-time consuming process that also decrease the amount of errors, and in that way also can tackle the first and second issue partly. This makes it worth doing research into this topic.

The management of the abstract requirements during design processes of building projects are still very difficult. This is because of the fact that a client does not completely know what he or she wants in the beginning of a project. Next to this, these requirements have an extra difficulty, which lies in the fact that everyone has his or hers own thoughts, visions and values regarding these requirements. And on the other side there are the experts that have to deal with these kind of requirements and also justify, verify and implement these in the design. Also these experts have their own thoughts, visions and values regarding these requirements as result of their experience and expertise. This difference in cognitivism about these requirements cause the difficulty of managing these abstract formulated requirements.

There is supposed to use a Knowledge-Based System (KBS) which probably makes it possible to gather all this experience and expertise of experts and connect this knowledge base to a client for specifying his or her individual meaning regarding such abstract requirements. Also current automated checkers are focusing mainly on specific domains and elements and not on the total design process (Moonen, 2016). This approach that will be presented marks the start of automated verifying abstract requirements.

In this thesis an approach is presented to cope with these abstract requirements. In order to this approach a stepwise method is described that can serve as support in specifying abstract formulated client requirements and make these usable for automated verification. Investigated is how a KBS can be deployed and used in order to improve the design process and expand the development of automated verification.

## 2.2 Problem Statement

One of the problems that exists in the construction industry is how to cope with the abstract, undefined and non-computable rules in architectural designs. The requirements that follow from a client's brief consist of many different formulated goals, wishes and requirements. These are all formulated differently, even when holding on to predetermined attributes of a requirement, and can be interpreted in various ways by several experts. Some of these are objectively and easily interpreted as a measurable and computable rule, while others are not. A major problem is that of dealing with the lack of well-defined requirements, a topic where just a few studies has been conducted. Among these predetermined attributes the ambiguity of a requirement can be diminished when searching for objectively formulated components of the overarching formulated requirement. In addition to this, verifying these not well-defined requirements, called as *abstract* requirements in this research, is a domain where research is needed.

In the Requirements Engineering (RE) process, the processes of elicitation, analysis, specification and validation of requirements takes place. In here all the requirements are gathered and formulated according to the contractor's insight. Subsequently these are verified via different methods and techniques, often manually (with support of different software). During the building projects lifecycle these are continuously changed and updated, which makes keeping it clear and structured a hard task to manage. Different approaches, such as SE found guidelines to deal with these processes. However, this does not provide an approach to specifically deal with the abstract requirements.

The core problem in here is the specification of abstract requirements. This comes from the fact that these kind of requirements are not well-defined and cannot easily and objectively be measured. Here lies an opportunity to improve the specification of these requirements in the design process by using experts input to support specification. When knowledge of the experts is gathered and abstract requirements are computed based on this knowledge, clients answers regarding these requirements can be translated into specific values. These specific values make abstract requirements also more consistent and objectively measured by combining a KBS in the form of an Expert System (ES), which is knowledge based, and clients answering the questions that are generated by the system. This thesis will present how to deal with abstract requirements and how these can be translated in such a way that these also can be usable for automated verification.

## 2.3 Research Scope

The problem can be divided in several stages according to the design process. Different processes can be distinguished; retrieving abstract requirements from the client, reformulate the requirements to computable statements or verifying abstract requirements with BIM. None of these stages is a standalone topic and can be approached without taking into account the other ones.

This research is handling the reformulation of *abstract* requirements from the client's brief into computable rules that are usable to verify with a corresponding BIM model. This research starts with evaluating the problem in a broad way by reviewing corresponding and related topics regarding abstract requirements. Within this broad overview the research focuses on the design process and more specifically on the RE process within this phase. During the interviews is focused on knowledge about abstract requirements which will be the base for the model development of an ES. A test case as Proof of Concept (PoC) has to show the validity, viability and usability of the developed ES and corresponding method and in here also the reusability and expandability must be evaluated. Therefore one example of an abstract requirement will be used for a certain building space to show the use of an ES in combination with automated verification.

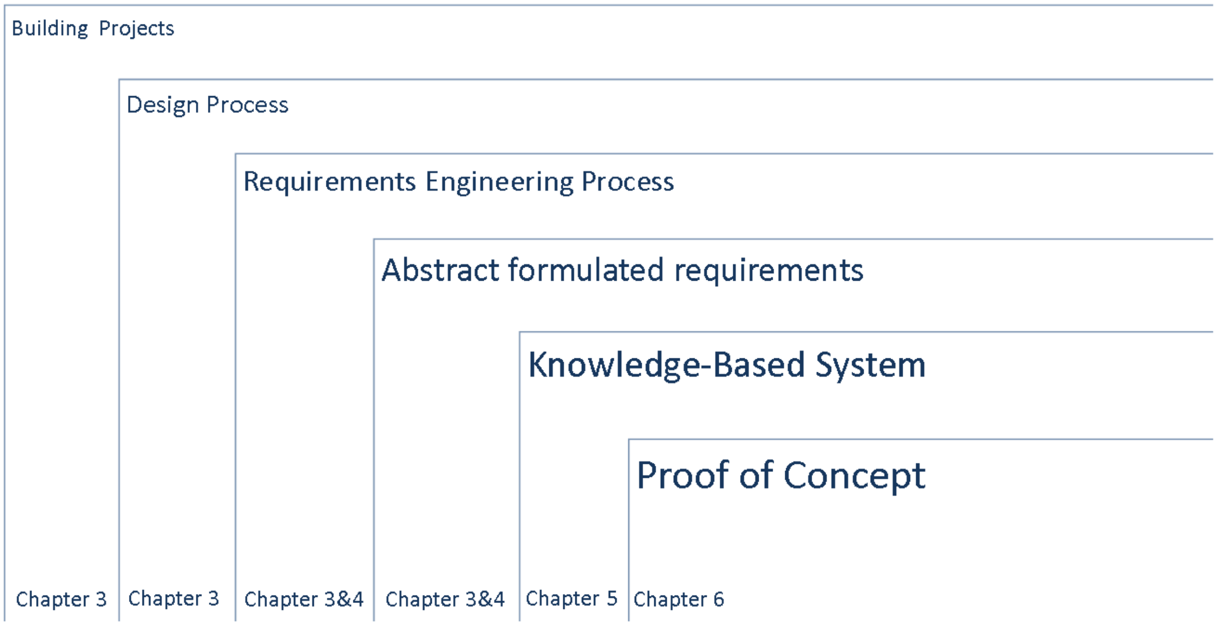



*Figure 1. Research Scope.*

## 2.4 Related Work

Automated verification of designs has been used already for many years in several industries to automate the design verification. For example in the industry of Electrical Engineering, Mechanical Engineering and Software Engineering (Niemeijer, 2011).

In the building industry several graduate reports, professionals and others have put effort into this same topic. The translation of ambiguous client requirements into product specification (Delghandi, 2018) is focusing on the same non-defined requirements of a client's brief and how these can be written down according to very specific product specifications, but differs in the connection to automated verification. Also R. Niemeijer (Niemeijer, 2011) has elaborated on these requirements in a broader perspective during his 4 year PhD. research project. The implications of automated verification of already defined client specific requirements is reviewed by L. Moonen (Moonen, 2016). Also surveying different rule checking systems that assess building designs is examined by C. Eastman (Eastman et al., 2009)(Solihin & Eastman, 2015) and describes different systems.

This thesis is focusing on the demarcated concept of specifying abstract requirements by setting up a KBS and connect the result with automated verification. Regarding this topic it differs in the way the specification process is approached and fills the gap of handling abstract requirements for automated verification.

## 2.5 Objective

The research objective is not to show *how* abstract requirements must be defined, this will always be subject to discussion, but to show *if* it is possible to translate also abstract requirements in an objectively measured way. To prove if this is possible, also on a reoccurring base in building projects, a KBS is developed and the result is used in a test case. The objective of this PoC  is to show if it works and what the implications can be. Researched is the use of KBS in the requirements specification process of abstract requirements and the possibility to use the result for verification automatically via BIM.

### 2.5.1 Scientific and Societal Relevance

The relevance of this topic regarding how it will benefit science and society of today and tomorrow can be described according to two main subjects that are linked to each other. Relating to the science relevance the lack of research shows the importance of conducting research and contribute to the expanding need of coping with information in BIM of building projects, especially for abstract requirements. The society relevance of coping with abstract requirements is that clients will be better understand and a more efficient design and verification process will arise which benefits all future building projects and there related stakeholders, such as all involved parties during the realization and use of a building and its environment. The possibilities of collecting knowledge of experts and use it on a sharing base and implementation will benefit definitely the construction sector, but also the entire society.

### 2.5.2 Conceptual Framework

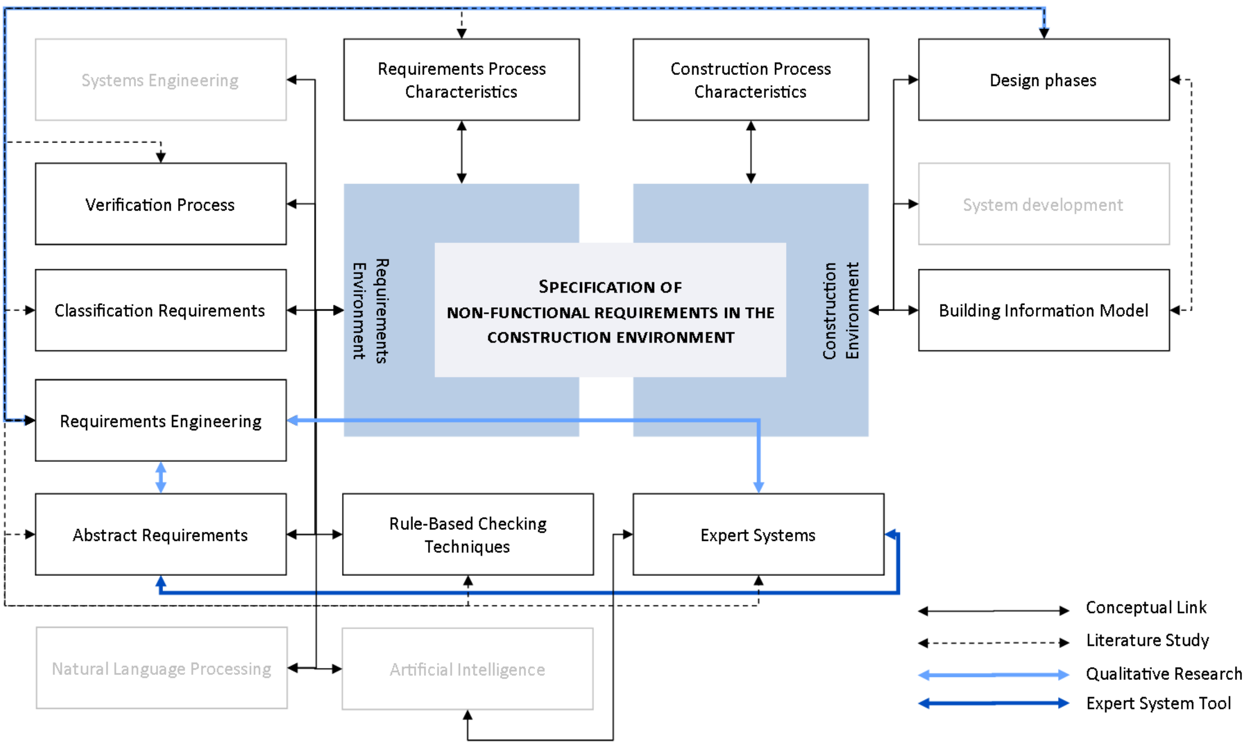



*Figure 2. Conceptual Framework.*

During this research different methods are used to connect the different concepts relating to this research topic. In *Figure 2* the total cohesion is visualized in what can be called the conceptual framework of this thesis (Maxwell, 2013).

Regarding to the research process different methods can be distinguished for connecting different concepts. Starting at a broad base of knowledge and information about several concepts that connecting the requirements environment and construction environment. The literature study will discuss the main concepts. Relating the research problem with the literature the qualitative research will ensure an explorative and in-depth next step focus on the RE process and the abstract requirements. Which will be the base for the development and the corresponding PoC.

## 2.6 Research Questions

Out of the problem definition and the scope the research questions are derived. The main research question is:

**Is it possible to translate abstract formulated requirements into specific defined and computable requirements that are usable for automated verification in building design processes via Building Information Modelling using Knowledge-Based Systems?**

This main research question is divided in seven sub-questions. In this research there are two main parts. These parts are the requirements definition stage and the use of Expert Systems (ES). In the part of requirements definition the questions will be evaluated and answered according to the literature review and conducted interviews. The part of ES's will be evaluated and answered by the background information from the literature review, the interviews, the developed model and a pilot case. The first four sub-questions are evaluating the design process, abstract requirements, specification processes and current approach in relation to requirements definition. The other sub-questions are beside these methods evaluated by the developed tools and pilot case. In chapter 7 the answers on these questions can be found.

_Requirements definition_ *(literature study)*

1. Which different requirements describe a building project in its design phase and which are not usable for direct automated verification yet?

*(literature study + interview)*

2. How are the abstract requirements influencing the dealing with requirements in the design phase?

*(literature study + interview)*

3. What are the characteristics and difficulties of abstract requirements concerning the specification process of these requirements?

*(Interview)*

4. What is the current process of dealing with abstract requirements during the design processes?

_Expert Systems and automated verification_ *(literature study + interview + tool )*

5. How can Knowledge-Based Systems, in the form of Expert Systems, be a support to the requirement definition process?

*(literature study + tool)*

6. What is needed to translate abstract requirements into specific defined and computable requirements?

*(pilot case)*

7. How can automated verification be improved using the translation of abstract requirements and with that optimizing the design process?

## 2.7 Research Design

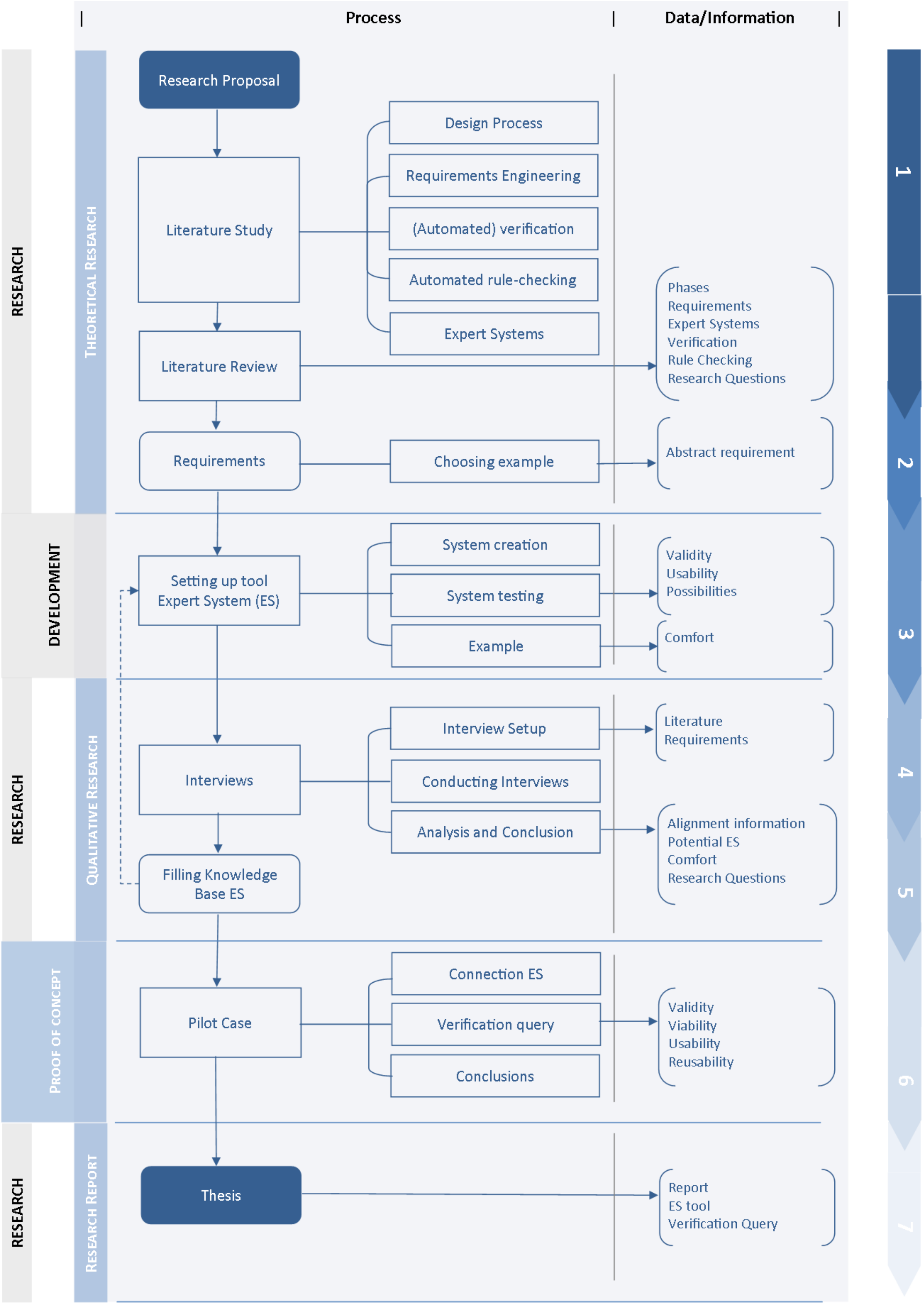



*Figure 3. Research Model.*

This research consists of three main parts; theoretical research, qualitative research and a proof of concept. The concepts mentioned in the conceptual framework and related research questions will be discussed during the research steps. In addition to this research model the connection between the different steps of which this research consists can be separated in four different blocks.

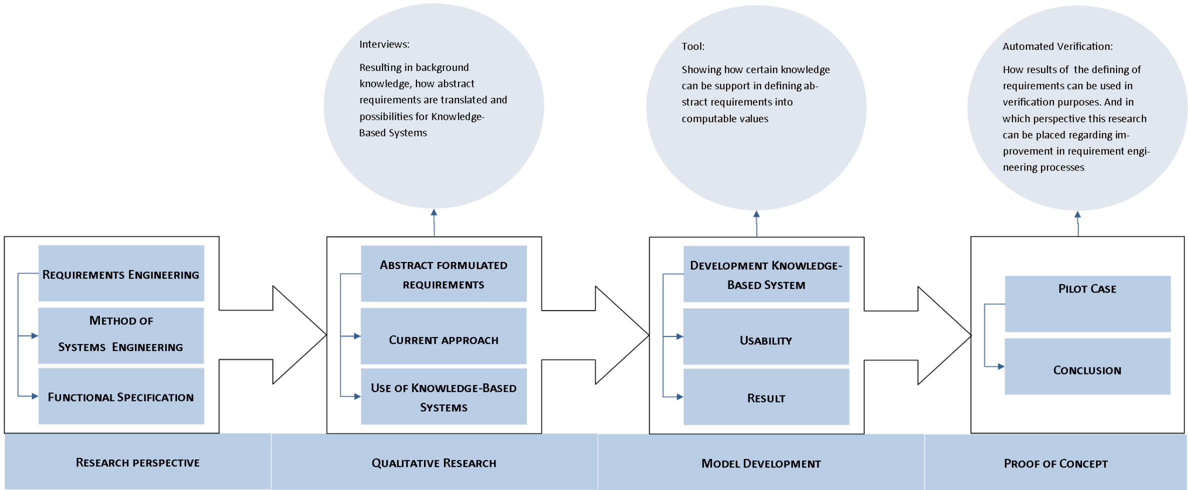



*Figure 4. Research steps.*

As can be seen in *Figure 4* the directive of this research is to conduct interviews to gather enough knowledge about which particular function the developed tool must fulfill. On the one hand it is needed to investigate how mental processes of defining abstract requirements work, but on the other hand it is also needed to show how to deal with certain knowledge in order to translate these requirements and make these usable for verification later on. Based on the interviews the precise directive of the tool will be determined. In any case the tool will be used to show how captured knowledge can be used in the definition process. Following on the result the tool produces, the possibilities of verification will be investigated. The use of Knowledge-Based Systems (KBS) is derived out of the literature review and the idea and belief in capturing knowledge from experts in order to develop an efficient definition process of abstract requirements. The different used methods for conducting this research are described by the research approach.

## 2.8 Research approach

### 2.8.1 Literature study

The literature study is conducted to get a deeper understanding about the different subjects and processes. The sources for this literature study consist of books, scientific articles, conference papers and a selection of different master and PhD theses. To frame this literature study, a selection of subjects has been made to impose the focus. This selection is based on the reason to frame the context and concepts to the earlier mentioned research scope and corresponding main research question.

- Design process
- Requirements Engineering (RE)
- (Automated) verification

- Automated rule-checking
- Expert Systems (ES)

Not every answer will be supported by scientific evidence and current scientific literature, which results in an own research regarding the possibilities of ES and automated verification of abstract requirements that have a lack of a measurable component.

### 2.8.2 Interviews

Because the RE process of abstract requirements is a relative new field in the building design process field, qualitative research will be conducted. It aims to get an in-depth description of the current situation and related processes and possibilities to optimize it. Interviews are useful for implementation, evaluation and problem investigation because they can provide information about real-world phenomena (Wieringa, 2014).

For this research it is important to use a more open interview structure. Therefore a semi-structured interview will be used as a method of research for conducting these interviews. This form of interviewing allows new ideas to be brought upon during the interview and starts from a framework of themes that needs to be explored. It gives the interviewee more space to give extra explanation or details about the questions, which can be used in this thesis.

The focus group of the interviews will be experts in the domain of design-, requirements definition and verification processes. To frame the abstract requirements into a pilot case for the PoC, comfort will be used as an example during this interview and subsequent case. This because specification of abstract requirements partly happens in someone's head during a design process and can be best shown with the use of a concrete example.

### 2.8.3 Expert system tool development

The interviews will gather knowledge, experiences and expertise of several experts who are dealing with these abstract formulated requirements during design processes. The information and results of the interview can be used to answer some research questions and as base for the development of the ES tool. This ES is a specialization of a KBS where the experts input is one of the major factors to support decision-making processes. It is of this reason the focus will be on the so called 'Expert Systems' where the knowledge representation of experts is the leading business. A pilot case can prove if the result of an ES based on an abstract formulated requirement can be used for automated verification. In this research the development of the ES tool is described and explained regarding to its use for translating abstract requirements into specific defined and computable requirements.

### 2.8.4 Proof of Concept

During the ES development a tool is developed and a pilot case will be performed to show the usability of it regarding automated verification. This PoC must result in a conclusion which handling the usability, viability and reusability of the described method and development in this report. This corresponds to an observational case study, where the research tries to influence the case as little as possible. Measurement may influence the measured phenomena, but as in all forms of research, the researcher tries to restrict this to a minimum to strengthen the PoC.

## 2.9 Expected Results

The expectation is an elaboration on the topic of abstract requirements and on related concepts to define these often and differently used terminology. In conjunction with the defining also characteristics and difficulties are evaluated. Out of this elaboration the qualitative research and development of an ES tool, will be used to work towards answering the main research question. In addition to this the PoC will be presented to use the verification query to show the possible implementation of this research concerning automated verification.

The interviews will be held to evaluate what the current approach is and possible added value when using an ES tool for the requirement definition process.

Eventually this research must result in an evaluation of the possibilities for translating abstract requirements into specific defined and computable requirements using KBS's. Which, when a positive answers is given, can be rolled out towards more abstract requirements specification processes and larger building spaces or total buildings and improving this process. Which gives rise to more research and extensive development.

# 3.0 Literature review

## 3.1 Motivation

This traditional literature review narrows down the current knowledge on the field of building design requirements and especially on abstract requirements, automated verification through BIM via rule-based checking and the possibilities of Knowledge-Based Systems within the requirements' specification process. The goal of this literature review is to summarize the literature about the topics mentioned in the conceptual framework. The literature is researched from the relevant databases and has the purpose to give a comprehensive overview of the topics and to highlight significant areas of research and at the same time find gaps in the research area to fit in this research.

The first part of this literature review is about the design process. Next the requirement specification process at the design stage of buildings is evaluated, which is the core of this literature review. Within this part a lot of attention will be given to abstract requirements and the related characteristics and verification possibilities. The topic about Knowledge-Based Systems and several variants, especially ES's, is evaluated after it. The last part of this literature review is focusing on the current knowledge and practice of (automated) verification through BIM via rule-based checking techniques.

## 3.2 Design Process

### Design phases

In the Dutch construction industry the Dutch standardization institute and "The New Rules", created by NLIngenieurs and the Branchevereniging voor Nederlandse Architectenbureaus (BNA), has defined the design phases in their Dutch Standards (NEN2574) and the De Nieuwe Regeling – StandaardTaakBeschrijving (DNR-STB). They define a total of ten phases of a construction project, which are presented in a clear overview in *Figure 5*.

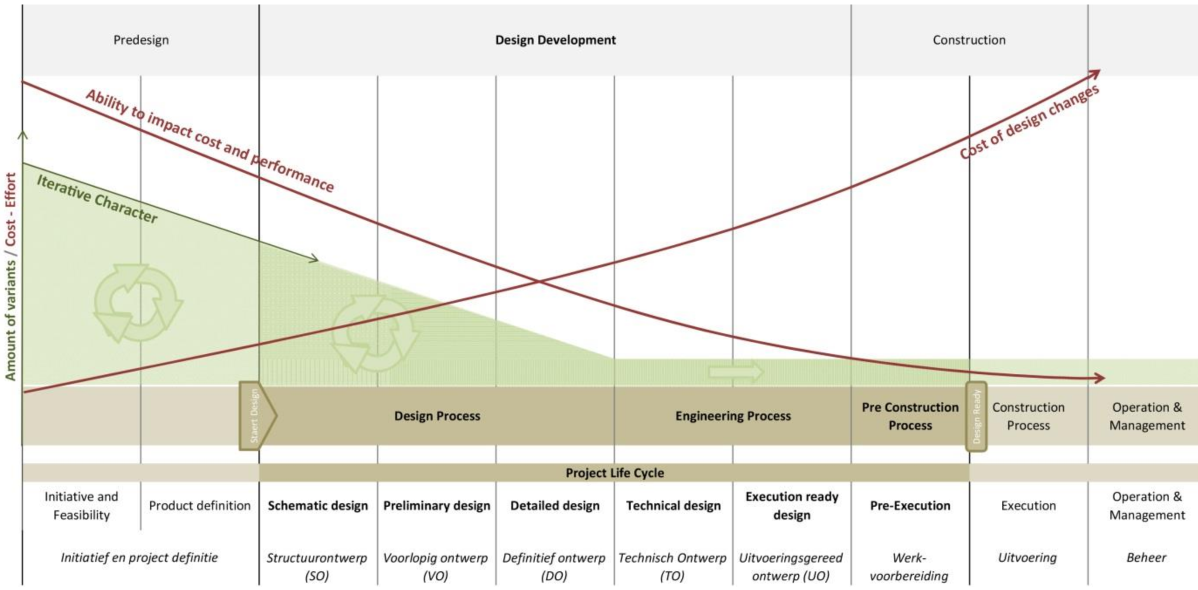



*Figure 5. Design phases according to the Dutch construction sector (Moonen, 2016), own drawing.*

Most of the problems in buildings can related to the requirements processing in the design phases. Most of the problems are recurrent because of the complex and iterative nature and to the great number of stakeholders involved in this type of projects (Pegoraro & Paula, 2017).

During the Project Life Cycle the development of information can be seen as two information flows. On the one hand there is input information of requirements which is provided by clients and on the other hand there is information which is created during the design phases (design development). Between these flows there is a continuously exchange of information, which remarks the iterative character. At the start of the predesign the amount of variants is high, and this should decrease over time as well as the ability to impact cost and performance. During the progress of a project the cost of changes will increase when changes of the design are made in a later phase. This effect is called the MacLeany Curve and is represented in *Figure 5* by the red crossing lines.

This information rich process defines the system and building elements, which is a crucial interaction for relating information to each other (Moonen, 2016). The right comparison between the definition and the performance is essential. This interaction between the whole set of information is a process between function, building element, performance and requirement (Schaap, Bouwman, & Willems, 2008). In the *figure* below this is visualized in a scheme.

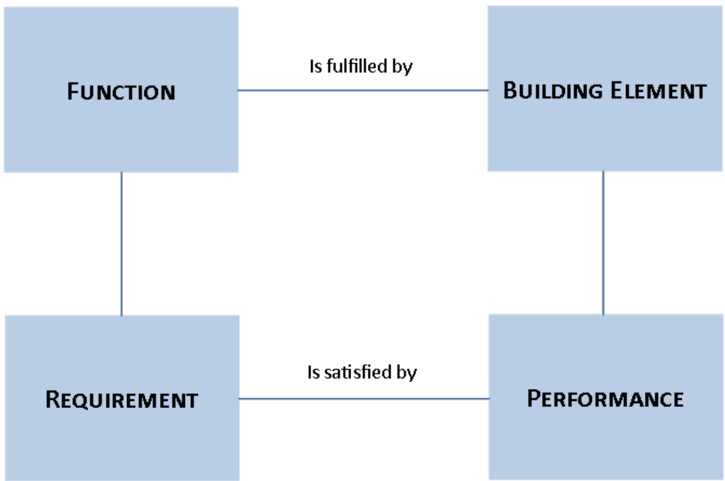



*Figure 6. Interaction of information in design processes (Schaap et al., 2008), own drawing.*

The progress of the design process and development of the design happens iteratively. Design decisions contribute to the development of the design. A system will be described by the demand specification of that system where after variants are made to evaluate which design is satisfying the best solution for the corresponding requirements. This is iteratively done and ensures the evolving from conceptual design to a more detailed design and corresponding Level of Detail (LOD). This LOD must comply with the LOD of the corresponding requirements. The corresponding LOD relates to the developed method of SE and the top-down reasoning for specifying requirements which is elaborated on in the phrases about the translation procedures of requirements. According to the SE approach, a functional specification is developed as part of the demand specification that focuses on functional thinking in contrast to the traditional process where the market had less freedom for interpretation.

## Information exchange

The continuously exchange of information delivers an interaction in requirements on the one hand and design solutions on the other hand. The related processes of RE and creating design solutions is an important interaction to make a right comparison, because otherwise there is an increasing risk the information is not right and will not be useful. The possibilities and importance of managing this information have grown with the use of BIM. BIM models represent objects in three dimensional ways and include graphical and computable data. Something which makes a BIM very useful for usage during the total Project Life Cycle.

## BIM

BIM, as defined by Eastman (Solihin & Eastman, 2015), is a modelling technology and associated set of processes to produce, communicate and analyse building models. These building models are visualized in three dimensions by means of graphical CAD models that are structured from objects which contain both graphical and computable information and data. The models are enriched with information and data during the total lifecycle of a system. Within BIM, information can be captured for numerous purposes as a function of a total process (Moonen, 2016) (Delghandi, 2018).

BIM stores building designs as a collection of objects with associated properties. Building projects become larger and more complex which results in more data and information. This increasing complexity (Wood & Gidado, 2008) is mainly due to its fragmented character, with projects divided into parts that are subcontracted to individual companies. The construction industry itself is an interwoven network of high complexity and a great dynamic. Next to that is it also a working place for humans and a place for cooperation and social interaction which, because of the temporary character, forms a highly transient human system.

The use of BIM is a very promising development within the AEC industry where numerous researchers are currently investigating upon. Campbell (Campbell, 2006) defines a BIM as an intelligent simulation of architecture that exhibits six key characteristics; (1) Digital, (2) Spatial/3D, (3) measurable in the meaning of quantifiable, dimension-able, and query-able, (4) comprehensive, (5) accessible, (6) durable for using through all phases of a facility's life. The focus in this research on requirements that are not defined as quantifiable and thus not computable yet are in this context not ready to use for BIM. The building model itself exist of building components that are enriched with information and data that describe how they behave and are consistent and non-redundant data. These components are modeled as objects that have a digital representation and data about what they are. Precisely this digital representation is the fact that it is not yet possible to verify abstract requirements via BIM. Simply because these requirements are not defined by enough information and including measurable, in other words computable, components. Which is because of the relating design stage, but also due to a lack of information about this abstract statement.

## Shift towards a more integrated design process

Due to the development of BIM which entails more data, an international shift from the traditional building process to a more integrated way of working takes place. The traditional construction process resulted in many ambiguities and inconsistent information transfers. The traditional process is characterized by a succession of different involved parties. In later phases than desired, errors and modifications emerge which at an earlier stage could have been determined and solved. The shift towards a more integrated way results in a more explicit way of working, where market participants give further consideration to the produced requirement specification of the client. This requirement specification results in the eventually created demand specification for the market. The requirement specification is described in a certain way, which is called the functional specification (de Haan, Degenkamp, Schotanus, & Mulder, 2017).

Using explicit information, defined as clearly stated and thus no room for confusion or questions, is of course important. Before introducing it to the market participants, the validation process must ensure that is mentioned what the client  means. This is important to reduce any discussion about it in a later stage (Hoeber, 2012). In the traditional construction process these requirements are included in the Statement of Requirements (SOR) which is used as input to the design. These requirements are often just written down and not formulated in a consistent and unambiguous way, but more often as inference requirements. The interpretation and implementation of these requirements is based on the combination of the available information and reasoning. In the traditional construction process this SOR must be finished before starting developing the design. This process is largely changed by introducing a more integrated design and building process with SE as support for among others functional requirements specification at the construction sector. The focus is lying much more on a good representation of the requirements that describe a building. This is emphasized by the believe that requirements processing is both critical to the success of a construction project and problematical in its effectiveness according to Shen (Shen, Li, Chung, & Hui, 2004).

### 3.3 Requirements Engineering

#### Requirement specification process

A definition for the requirement specification process, also called Requirements Engineering, is given by Ian K. Bray and is described as "*investigating and describing the problem domain and requirements, and designing and documenting the characteristics for a solution system that will meet those requirements*" (Bray, 2002). The industry average investment in the requirement process for a typical system is 2-3% of the total project costs (Young, 2004). This amount is inadequate and in fact the root cause of the failure of many projects. Data has shown that projects that expended the industry average of 2-3% of total project costs on the requirement process experienced a 80-200% cost overrun, while when investing 8-14% of total project costs in the requirement process this has been reduced to 0-50% cost overrun.

A correct system development is therefore important and depends on a precise, correct, and complete system description or specification. RE focusses on the task of obtaining the requirement statements and produce a correct and complete specification of a system. According to Zhi Jin (Jin, 2018) the requirement statements can come from stakeholders, the domain knowledge and regulations. The process of obtaining these requirements of a system is intertwined with the design phases of a construction project.

Before any system can be developed it is essential to establish the need for the system. The need may initially be expressed in fairly vague terms. In a generic development process, according to (Hull, Jackson, & Dick, 2011), the development of a system follows the next steps (*Figure 7*) and can be related to the Dutch construction sector design phases (*Figure 5*). Where the problem domain represents the pre-design phase and solution domain the design development.

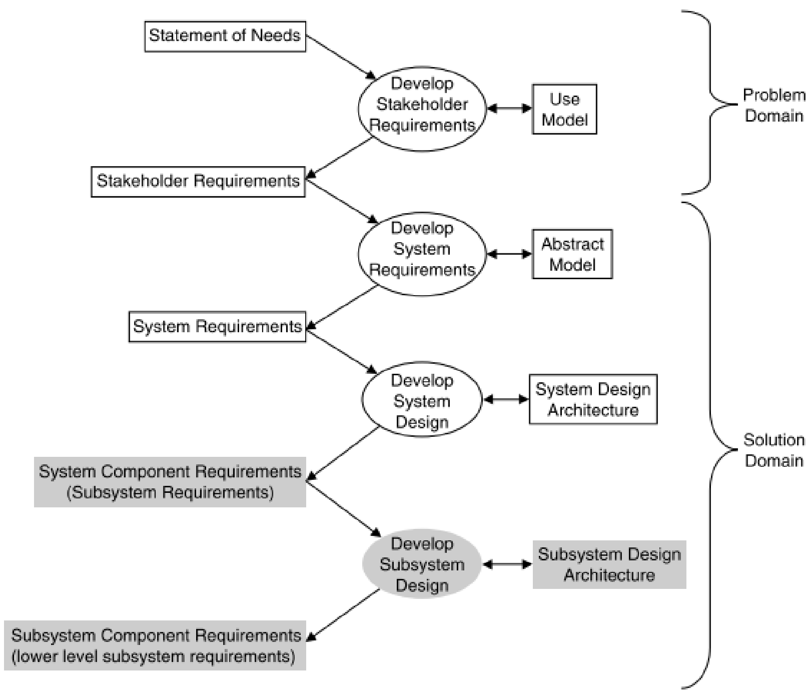



*Figure 7. System development process (Hull et al., 2011).*

The process of RE is divided into defining, documenting and maintaining requirement statements. There is a major importance of RE in early lifecycle phases of development of a system (Daclin, Daclin, Chapurlat, & Vallespir, 2016). Requirements must be checked throughout the systems lifecycle from design phase to execution phase. In order to prove, expectations have been satisfied and avoid problems such as drifting from expected objectives or cancellation. Some requirements only can be verified during operations phase, but still the majority must be verified during the lifecycle of a design, and therefore plays an important role in the success or failure of a project, yet it is often neglected. The process of RE continues to be considered as time- and resource-consuming and without clear added value. However, one should always keep in mind that as more errors are carried to the upstream engineering phases, the remedial costs in downstream phases will increase. RE belongs to the field of the definition of the problem which makes it beneficial to spend time defining clearly what is expected in order to avoid, as much as possible, problems in the later phases of the development and construction phases.

The tasks that are related to RE can be divided into two subtasks (Jin, 2018), requirements modeling and analysis and requirements evolvement and management. Requirement modeling and analysis refers to eliciting and determining needs or conditions to meet for a new or altered product including gathering requirements by requirements models so that the correctness of the requirements can be guaranteed. The other subtask is referred to managing all of the artifacts that have been produced during requirement modeling and analysis and managing the changes to these artifacts. This research mainly focusses on the part of requirement modeling and analysis and can be subdivided into four activities that are in an interleaved, incremental and iterative process.

- *Requirements elicitation:* identifying and collecting requirements(-related) statements from stakeholder or other sources.
- *Requirements analysis:* processing the information to understand it, classify it into various categories, and relating the stakeholder needs to possible system requirements
- *Requirements specification:* structuring this information and derived requirements as written documents and model diagrams
- *Requirements validation:* validating the documents and/or models to confirm that the specified requirements are accurate, complete and correct

Requirements Management (RM) includes the documentation, storage, communication, tracking and traceability in such a way to allow an easy and reliable requirements change management (Pegoraro & Paula, 2017). During the lifecycle of a requirement, tracking the status of the requirements and change activity and tracing requirements to the various phases and product of the development effort can be seen as the core of RM (Young, 2004).

In this research, RE and RM are not dissociated and we will therefore call this whole process as requirements processing. In the AEC sector also referred to as “briefing”. In some research, the techniques of enabling the project team to identify the best values and derive suitable solutions to fulfill the clients requirements are used for considering requirements in construction projects (Pegoraro & Paula, 2017). This has all to do with processing practices to help stakeholders to form adequate relations and make valuable decisions for good requirements.

Among the problem to perform requirements processing activities there are several often cited difficulties (Pegoraro & Paula, 2017). One of the difficulties is accommodating requirements of all involved stakeholders and the lack of open, effective and formal communication. Due to the high number of stakeholders and the poor definition of the project's objectives and priorities. The communication is often hindered by the lack of channels and common language between stakeholders. Another one is the unclear information and the lack of inclusion of end users throughout the design development. This last one makes it difficult to evaluate to what extent the design solutions represent their needs and wishes (Jensen, 2011). This has all to do with interpretation and perception of their importance (Soetanto, Dainty, Glass, & Price, 2006).

### Requirements

According to Ad Sparrius (Sparrius, 2014) a requirement can be fully defined by two definitions. One from the International Organization of Standardization (ISO) that has define a requirement as a statement that expresses a need and its associated constraints and conditions. And another institute, the Institute of Electrical and Electronics Engineers (IEEE) defines it as a statement that identifies a product or process operational, functional or design characteristic or constraint, which is unambiguous, testable or measurable and necessary for product or process acceptability (ISO/IEC/IEEE 15288, 2015). A set of requirements is contained in a specification. This last definition defines more precise the conditions of a requirement.

A requirement that has not been elicited (see *requirements elicitation*) is not a requirement. A thought in someone's mind may be in time converted to a requirement, but is primary not a requirement. If these mere thoughts were requirements, it would make the concept of a requirement meaningless.  A requirement does have also *requirements*, but no literature describe these attributes as an exhaustible list taking into account the different requirement types. In general a requirement must be, according to the aforementioned definition of the *IEEE,* at least unambiguous, testable or measurable.

Requirements are necessary attributes in a system, a statement that identifies a capability, characteristic or quality factor of a system in order to have value and utility to a customer or user (Young, 2004). Requirements are important because they provide the basis for all the development work that follows. Too often, people want to start with developing 'the real work' too quickly, but the industry confirms that a better approach is to invest more time in requirements gathering, analysis and management activities. The reason for this is that additional time is needed to identify the 'real' requirement. According to Young (Young, 2004) there is a significant difference between the mentioned 'real' requirement and a 'stated' requirement. Stated requirements are those provided by a customer at the beginning of a system development effort. Real requirements are those that reflect the verified needs of users for a particular system. These real requirements are the ones that reflect right what the customer wants, and is defined and interpreted by others in one way. These 'real' requirements are subject to this research, in such a way that abstract stated requirements mark the beginning of a process of translation towards these 'real' requirements.

According to different literature about the attributes and characteristics of a 'good', and thus 'real' requirement, *Table 1* on *page 19* summarizes the most used statements of these attributes. Only a small number of relevant articles are discussed in here, but they show corresponding characteristics a requirement needs to have. The concepts that are mentioned more than once are presented and defined related to the different author’s.

| Attributes | Defined by *Young, 2004* | *Leffingwell & Widrig, 1999* | | *Mannion & Keepence, 1995* | | *Van Lamsweerde, 2000* | | Mentioned |
|---|---|---|---|---|---|---|---|---|
| Correct | Fact related to requirement are accurate and it is technically and legally possible | Correct | Only if every requirement stated therein represents something required of the system to built | Correct | *not defined* | | | 67% |
| Unambiguous | Can be interpreted in only one way | Unambiguous | If and only if it can be subject to only one interpretation | | | Unambiguous | May not have multiple interpretations of interest making it true | 67% |
| Complete | All conditions under which the requirement applies are stated incuding expressing a whole idea or comment | Complete | if and only if it describes all significant requirements of concern to the user, including requirements associated with functionality, performance, design constraints, attributes, or external interfaces | Complete | *not defined* | Complete | collection of properties specified must be sufficient to establish the latter | 100% |
| Consistent | Not in conflict with other requirements | Consistent | if and only if no subset of individual requirements described within it are in conflict with one another | Consistent | *not defined* | Consistent | must have a meaningful semantic interpretation that makes true all specified properties taken together | 100% |
| Verifiable | Implementation of the requirement in the system can be proved | Verifiable | if and only if each of the component requirements contained within it is verifiable. And the requirements can be deemed verifiable if and only if there exists a finite, cost-effective process with which a person or a machine can determine that the developed software system does indeed meet the requirement | Verifiable | *not defined* | | | 67% |
| | | Modifiable | if and only if its structure and style are such that any changes to the requirements can be made easily, completely, and consistently, while retaining the existing structure and style of the set | Modifiable | *not defined* | | | 50% |
| Traceable | Source of the requirement can be traced, and can be tracked throughout the system | Traceable | if and only if the origin of each of its component requirements is clear, and if there is a mechanism that makes it feasible to refer to that requirement in future development efforts | Traceable | *not defined* | | | 67% |

*Table 1. Most mentioned attributes according to 'requirements' (Young, 2004); (Leffingwell & Widrig, 1999); (Mannion & Keepence, 1995); (van Lamsweerde, 2000)*

### Taxonomy of requirements

Requirements can be of different levels and the relationships between them can be explained by several perspectives. On the one hand requirements must specify what should be developed and the system designs specify how the system should be developed. An often made distinction is between problem definition and solution definition (Jin, 2018). On the other hand there is a distinction between functional and Non-Functional Requirements (NFR's) where, according to (Glinz, 2005), the first one describes again what the system should do, while all other requirements are considered to be non-functional. According to Glinz's article there is no consensus what is meant with the NFR's. The difference between them is related to the way it is interpreted and written down, but also the fuzziness that arises from the mix of concepts of the traditional classification. In here the classification is made according to kind, satisfaction, role and representation. Kind refers to whether a requirement concerns a function, performance need or constraint. Satisfaction to a scale from weakly satisfied to fully satisfied, where role refers to the role the requirement can play and representation whether a requirement is represented operationally, quantitatively or qualitatively.

This representation facet, according to (Glinz, 2005), goes hand in hand with the way how a requirement can be verified and is divided in quantitatively and qualitatively. Performance requirements must be specified in a quantitative form if it is needed that these are precise, unambiguous and verifiable. Quantitatively specified requirements are verified by measuring. In more abstract levels it is also desirable to state requirements in a qualitative form, which cannot be verified directly. These can only be verified after deployment of the system and subjectively judge whether or not a qualitative requirement is satisfied. Which corresponds the thoughts of functional thinking concerning functional specifications.

(Schneider & Berenbach, 2013) describe a distinction between product requirements or non-technical requirements. Where product requirements are subdivided into physical, functional or NFR's . These product requirements can have only a true or false value, according to Boolean attributes. In Figure 8 these different perspectives are visualized in one schema.

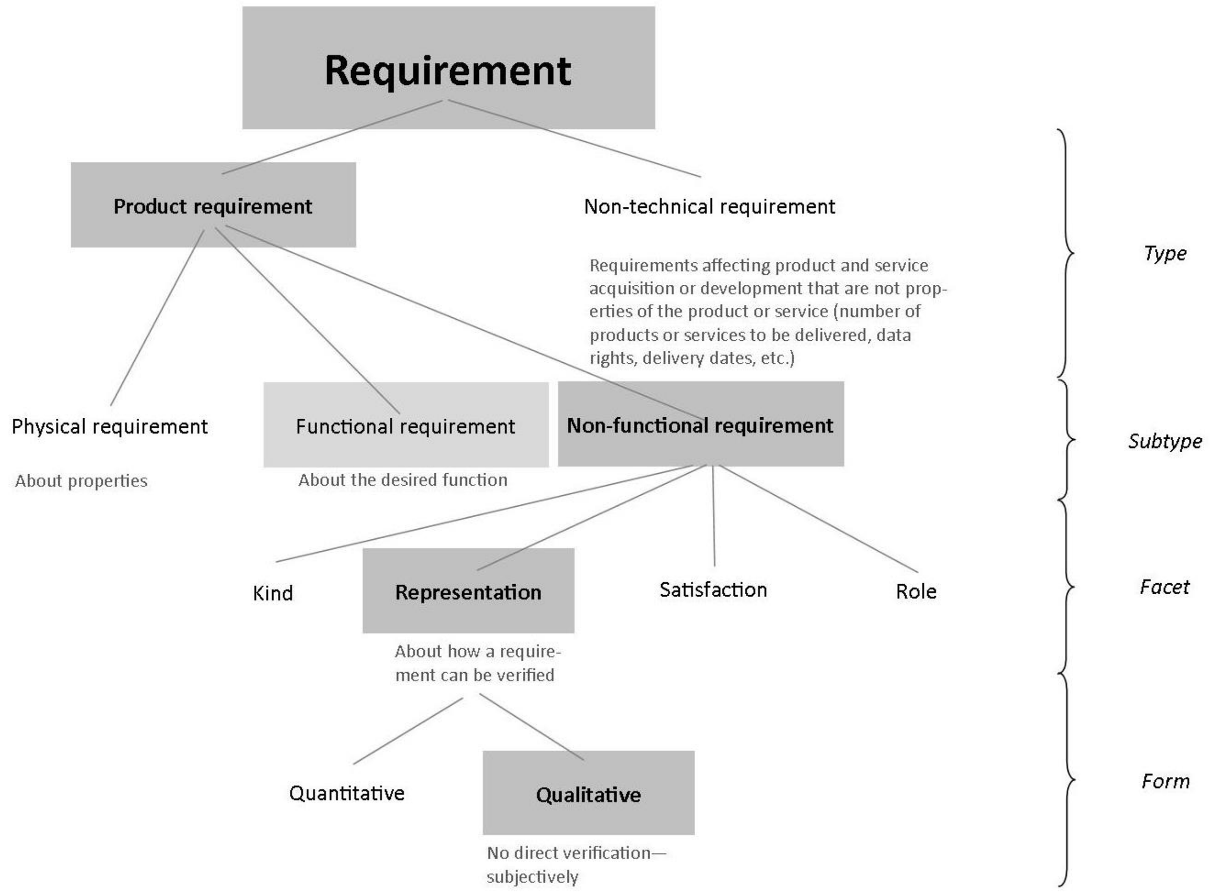



*Figure 8. Simplified requirements taxonomy according to different literature (Glinz, 2005) (Schneider & Berenbach, 2013), own drawing.*

Referring to *Figure 8* the grey blocks represent the framework of the requirements that are not directly verifiable. In this sense also the non-technical requirements can comply with this, but relating to building projects in this research the focus is on design requirements which is consistent with product requirements. It can be described as an interwoven patchwork of different labeled requirements without a distinction between verifiable and non-verifiable requirements.

The different mentioned concepts relating to the *requirements* of requirements will be discussed below according to the different definitions out of the literature of Young (Young, 2004), Leffingwell and Widrig (Leffingwell & Widrig, 1999), Mannion and Keepence (Mannion & Keepence, 1995) and van Lamsweerde (van Lamsweerde, 2000).

*Correct:* Correctness of a requirement describes the correspondence of that requirement with a real need of the intended users (Zowghi & Gervasi, 2003).

*Unambiguous:* Unambiguous means that it does not have multiple interpretations. One of the major requirements according to validation, because the client and contractor have to understand one and the same from a requirement. This ensures that eventually the right system is created.

*Complete:* Completeness can be seen as an important component of correctness. According to Boehm (Boehm, 1984) completeness must exhibit three fundamental characteristics. First, no information must be left unstated or 'to be determined'. Second, the information does not contain any undefined objects or entities and thirdly no information is missing from the requirements.

*Consistent:* Consistency requires that no two or more requirements contradict each other. And also regarded as the case where words and terms have the same meaning in all the requirements stated. This implies that exclusive statements and clashes in terminology should be avoided (Zowghi & Gervasi, 2003).

*Modifiable:* Modification of requirements must be possible in such a way that it can be done easily, completely, and with consistency, while retaining the existing requirements that are not changed. Otherwise this modification may cause inconsistency (Matsumoto, Shirai, & Ohnishi, 2017).

*Traceable:* The concept of tracing a requirement is specified as (1) the origin of each of the requirements are clear and (2) it facilitates the referencing of each requirement in future development or enhanced documentation (Pinheiro, 2004).

*Verifiable:* A requirement must be verifiable. The process of verification is proving that the designed and built system meets its requirements, it addresses the question "*Did we build it right?"* Requirements are not verified, but the system is verified that it meets its requirements. To describe verifiability as a requirement of requirements it must be seen as an overall requirement of the above mentioned concepts. Before verifying a requirement it must be *correct, unambiguous, complete, consistent, modifiable and traceable.*

The list of characteristics is not complete. The most named concepts are not that difficult to understood but their application proves difficult. This has to do with the lack of examples or guidelines to illustrate good or bad practice (Mannion & Keepence, 1995). Next to that, on the other side the sheer number of characteristics can be overwhelming. It is of that reason that the acronym SMART is developed to assist people in setting down good objectives and requirements. This acronym is not to prove that the requirements specification is correct, but for checking the document if every requirement can be verified as correct in terms of expression. The acronym therefore can help in formulating requirements but gives no guarantee. It is therefore important to focus on a description of the requirements that represent the abstract requirements used in this research.

### Requirements classification

The following three types of requirements can be defined according to the following distribution; first are the value (numerical) requirements, secondly the relational requirements and thirdly the textual requirements. These textual requirements are often difficult to handle since they are sensitive for misinterpretations or are judged differently due to their subjective character. This kind of requirements are often marked as complex because these are not tangible or measurable (Delghandi, 2018) and thus too abstract to use for (automated) verification yet. Nevertheless, these abstract requirements still need their own *requirements* to which they are framed to.

### Non-functional

Recent work points out that early-phase RE should address the NFR's, while later-phase focus on completeness, consistency and verification of all requirements (Pinheiro, 2004). Also several reports showing that not properly dealing with NFR's have led to considerable delays in the project and consequently to significant increase in final cost.

NFR's are describing how well specific parts of a project function are just as important as other requirements (see physical and functional requirements in *Figure 8*). They are ensuring that the system operates with the qualities that are needed and stated by the client or user. In short, these kind specifies how functions are to be performed, often referred to in SE as the "ilities" (Young, 2004). Examples are performance, reliability, security or maintainability standards. Because they are important, it does not mean it is easy to handle these kind of requirements (Matsugu, 2015). Often abstract requirements are difficult to model because they vary so much that they lack a consistent method or representation. Without this consistency, the requirements are modelled for each project specific. The requirements are also casually stated out of a nonchalant statement by a user or other stakeholder. And at last abstract requirements are hard to measure, having the capacity to be indefinite which makes them difficult to evaluate.

In addition to this, NFR's are often abstract, undefined and non-computable and represent most of the abstract requirements in this research, which not says that in other structure parts of *Figure 8* no abstract statements can arise.

### Measurability

Measurability is of importance regarding the description of abstract requirements. In addition to the described concepts many other concepts are related to verifiability and 'good' formulated requirements, but this research focusses on the way abstract requirements can be specified in order to use them for verification.

Assuming that a requirement is specific defined, non-measurable requirements fall into two categories: those which cannot be instrumented or those which are specific but for which there is no yardstick available (Mannion & Keepence, 1995). In the context of RE, the meaning of measurable is that it is possible to verify that the requirements have been met. Regarding the two categories, the first one can occur when detailed timing or performance information is required. In a general way the test of such a requirement can influence and change the characteristics of the system and therefore the operation of it. It is believed this same situation can occur with the expected abstract requirements within building designs. To test a quality on certain values within the building these values need to be met within the building and therefore must be present at the building or at least at the involved elements of the building.

The second category is subtler. It is not possible to measure this kind without having a test case (Mannion & Keepence, 1995). To be measurable the requirement must specify a fixed performance against a predefined set of test cases for which certain values are known. Requirements which are in this category need to be made more specific in order to be measurable. In several case studies with NFR's there is shown a high degree of measurable NFR's, but there are still some of them that are not possible to formulate in a quantifiable manner (Doerr, Kerkow, Koenig, Olsson, & Suzuki, 2005). But having more than 90% of the NFR's measurable is a significant improvement over current industry practice and previous methods, according to the several case studies in (Doerr et al., 2005).

### Verifiability

The objective of verification is to demonstrate that a set of selected requirements is satisfied (Mallek, Daclin, & Chapurlat, 2012). A requirement can be seen as verifiable in the aggregate *if and only if each of the component requirements contained within it is verifiable. And the requirements can be deemed verifiable if and only if there exists a finite, cost-effective process with which a person or a machine can determine that the developed system does indeed meet the requirement* (Leffingwell & Widrig, 1999). It is of importance, as a practical matter, that it is necessary to define requirements so that it is possible to test them in a later phase and determine whether they were achieved. In this matter computability of abstract requirements is necessary in order to automatically verify these requirements in for example a BIM model. Of course all possible characteristics of requirements that are mentioned in literature play an important role in defining and specifying 'good' requirements. However, this research focusses on possibilities to verify the abstract requirements that are not yet specified. And therefore is chosen to focus on translating these requirements towards statements that can be used for automated verification. In line with this verification, this research is connected to the concepts of measurability, computability and ambiguity.

There is also a distinction between *non-verifiable* and *non-measurable* requirements. In the sense of verification of requirements, measurability is not mentioned as a condition (*Figure 8*). But relating to the concept of rule-based checking of building design, the condition of measurability must be added to the requirements. This measurability in this research is equal to the computability of a requirement. The computability in here refers to making the requirement usable for (automated) verification in relation to an used software tooling or program. What can consist of quantifiable components or other information which can be used for connecting a requirement to a certain database. In this sense computability refer to a broader view of information of requirements than measurability entails.

In that case non-measurable requirements lacks information to have a computable component, but can still be verified using other methods that not need computability. Measurability can be, out of the formulated context, connected to a quantifiable requirement. If this component of computability is missing in the requirement it is impossible to use it for automated verification. The client often specifies their requirements in words which making it a task of the specialist to transform these into quantifiable requirements. But this does not mean that every requirement a client mentions can be directly transformed into a quantifiable requirement. This makes these requirements not (yet) usable for testing. These requirements are non-verifiable due to the fact these are non-computable, but non-verifiability entails more attributes then only computability.

### Computability

The translation from abstract requirements into specific defined and computable requirements is the objective of this research and entails a process from a client description into computable values. These computable values can help in a later stage to verify this same requirement as described by the client. These requirements describe often subjective matter where a client also does not know what to do with. During discussions between an expert and client about a certain topic the expert must try to find out what the client really means and wants. This can lead to a more specific description or understanding of what a client thought is perhaps not what he or she was mentioning.

It may be clear that not all components of these abstract requirements can be translated into specific values. But a lot of these requirements will be translated into values that are stated as computable and useful for information management purposes a computer can handle. And it is this information that is needed for verification via a BIM model because an automated constraint checking system will only be able to check constraints that be computed (Niemeijer, 2011). Constraints in the building industry fall in one of two categories in terms of computability: either they are simple guidelines or rules of thumb that can be quickly calculated, or they require a computationally intensive numerical simulation, as is the case in for example building physics constraints. It is interesting to look how KBS's can play an supportive role in the computationally translation of human expertise that is difficult to model using conventional techniques.

### Natural language

Natural language (NL) is the most used representation for stating requirements on computer-based systems in industry. However, NL is inherently ambiguous. In RE, NL is the most frequently used representation in which to state requirements that must be met by information technology products or services (Kamsties, Berry, & Paech, 2001). Ambiguity in requirements is a serious problem because often stakeholders are not even aware that there is ambiguity. So there is a major importance in describing a requirements and the components of a requirement in an unambiguous way.

Representing requirements in NL is ideal for human communication and definition, but the correctness of it is not guaranteed due to the inherent ambiguity of NL. Despite of this, NL is still preferred by many as a communication facilitator (Tjong, 2008). Based on this fact there are several approaches for writing good requirement specifications. Avoiding problems regarding ambiguity are based on approaches that define linguistic rules and analytical keywords, guideline rules or language patterns. In the study of (Tjong, 2008) these approaches are explained and insight is given how to deal with ambiguity. This is of importance during this research to translate and formulate usable requirements.

### Ambiguity

In NL, ambiguity is a major problem. People who use NL can usually discern the intended meanings of otherwise ambiguous words and phrases by using various sources of knowledge (Tjong, 2008). Ambiguity is noticed when a statement has more than one distinct meaning. By contrast, a vague statement can have only one meaning, but the distinction between circumstances under which it is true and the circumstances under which it is false is not clear-cut (Nissanke, 1999). Ambiguity arises from among others a conceptual problem such as conceptual vagueness or a lack of correspondence, which is most applicable in this research since the communication between contractor and client must lead to unambiguity. Regarding the study of ambiguity by Tjong (Tjong, 2008), the most applicable form of ambiguity is that of pragmatic ambiguity. It concerns the relationship between the meaning of a sentence and the context in which the sentence occurs. It occurs sometimes when human common sense knowledge and knowledge about context is uncertain. When a client indicates what is desired it is often not immediately clear and unambiguous to understand. In what context it occurs, what really is meant with that sentence and if the sentence really suggests what the client wants are just a few examples in order to get more grip and understanding of the client specific requirements. Through discussing and other forms the contractor tries to get the right understanding.

Also the research of Niemeijer (Niemeijer, 2011) discusses the difficulty of ambiguity in interpreting NL. The exact meaning of words can depend on the context, unlike programming languages. Several types of ambiguity can be mentioned (Hutchins & Somers, 1992).

- *Category ambiguity*: regarding the grammatical category, which can be explained by words that can be used in multiple ways.
- *Homography:* two words with the same spelling, but with different meanings.
- *Transfer ambiguity:* same words which having a different meaning in different languages.
- *Structural ambiguity:* one sentence that can have multiple different interpretations.

In this research, and for most of the requirements specification in the Dutch construction industry structural ambiguity is the most important to avoid. Category ambiguity and homography are often subordinate to structural ambiguity. When a sentence just has one interpretation, than other forms of ambiguity will seldom appear.

### Translation procedure of requirements

The translation procedure for the translation of requirements into product specifications is not researched upon that greatly by researchers from the AEC industry (Delghandi, 2018). In particular, not for abstract requirements. Translation procedures during design phases are often specified through contracted meetings between client and contractor. Within integrated contracts several meetings are planned for specifying needs and wishes of the client. These meetings must provide clarity in order to formulate the right requirements. Then it is important to translate this description into defined, computable and unambiguous statements to use it for verification. Core of this research is to investigate if there are possibilities to improve this process for abstract requirements and how this can be established.

### Systems engineering

One of the current used approaches is the approach of SE which is interesting as this is becoming a more standardized way of working in the construction industry. Used to manage the complexity of construction projects. According to the International Council on Systems Engineering (INCOSE) SE is defined as; "*an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem. […] Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs" (INCOSE, 2018).*

The use of SE within the AEC sector can be understood when elaborate on the characteristics (Moonen, 2016)(Delghandi, 2018). Identifying the steps will give an overview on how the information of requirements interacts with the design concerning this approach.

A system can be seen as a whole of interacting parts that work together for a stated purpose. This system is created by people to provide a certain need within a certain defined environment. In the construction industry these parts are most often objects. System thinking is used to understand the total project or process in a better way and act as the basis of SE. A decomposition of the total system into sub systems is used for dealing with the complexity of certain projects. The ultimate purpose of a system is the main functionality which is subdivided in other functions. Thinking from these functions is called functional thinking and ensures that people are thinking from large to small scale, which aligns with the top-down method of reasoning. By using this decomposition, the eventual tree structure of a system can be created and more insight on a higher level can be given by subtracting lower level parts.

During the system development (*Figure 7*) the need of a client is monitored continuously and is the main directive to a proper design. The focus on requirements during the whole lifecycle is essential for the implementation of SE. From 'good' requirements a solution is derived which suits all the needs of the client in the ideal world. The exploration of these requirements is thus an essential part of the SE process. SE approaches the development in the total life cycle, from initiation until retirement.

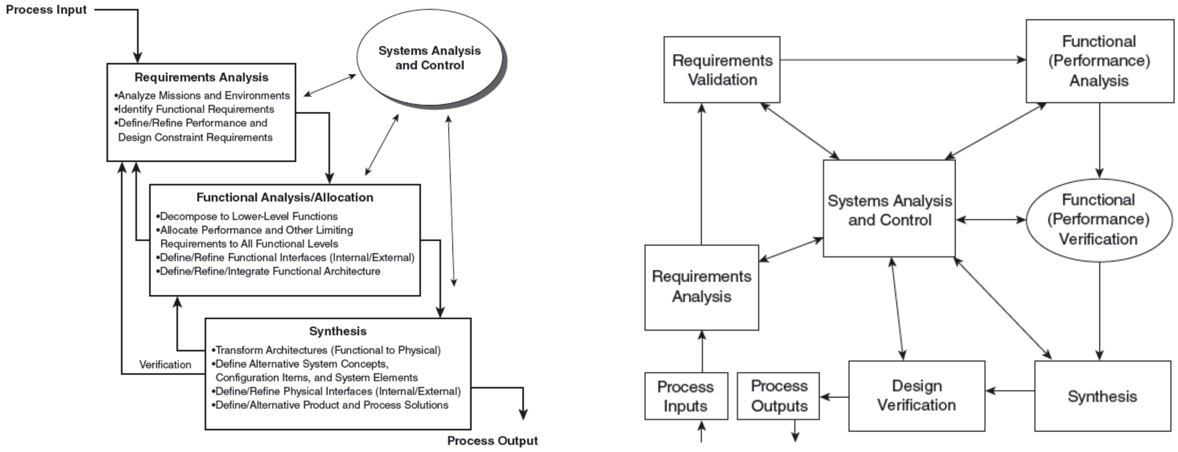



*Figure 9. Left: Systems Engineering methods. Right: Systems Engineering process. (Kossiakoff, Sweet, Seymour, & Biemer, 2011).*

The main elements in the representation of *Figure 9* can be found in the interaction between requirements, functions and design elements. The functionality of the eventual system is determined by the relation between these three elements. To illustrate the top-down methods in the design loop the V-model is often used. The decomposition of the initial system is realized to give more insight in the total system. The whole lifecycle can be seen in *Figure 10* with the different phases. Most important are the stakeholder analysis and requirement analysis. In the stakeholder analysis the key players in a project are identified. To realize a proper requirements analysis in the next step it is important to understand the stakeholders needs. Weak related requirements to stakeholders are a major reason that incites project failure.

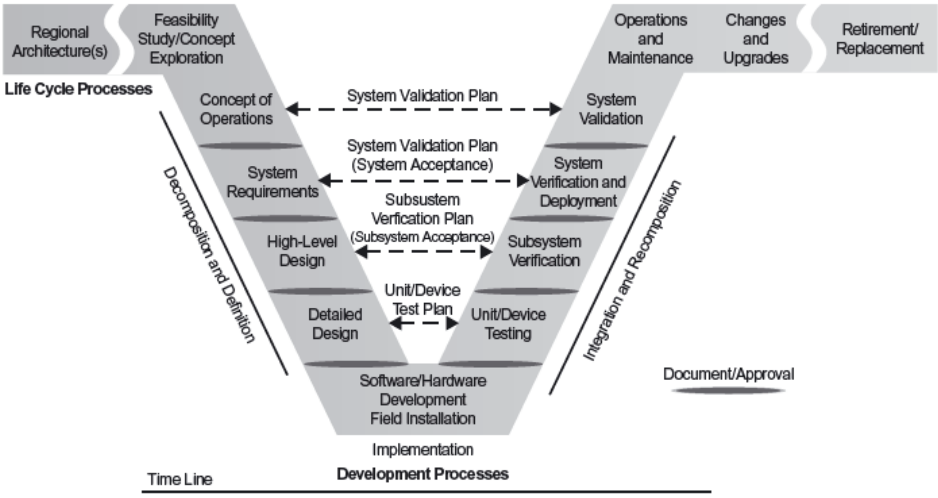



*Figure 10. One of the Systems Engineering approaches (Kossiakoff et al., 2011).*

The requirements analysis is defined as one of the most essential parts of the SE process as the understanding of the requirements defines the design constraints (Moonen, 2016). In this process the understanding of the requirements results in a better understanding of the whole problem. A proper validation with the client is needed to ensure that the interpretation is right and to reduce discussion about the interpretation in further stages. In particular, the reducing of discussion is subject to this research in order to decrease time-consuming meetings and errors in later stages of the design due to misunderstandings.

## 3.4 Verification

In the context of human realization of any system, human thoughts are susceptible to errors. This is also the case with engineering activities. In any activity or resulting outcome of an activity the search for potential errors should not be neglected, regardless of whether or not one thinks they will happen or that they should not happen. The consequences of errors can cause extremely significant failures or threats. The meaning of verification in the context of this research is the verification of a system, in this case an utility building, to check it against design requirements.

Verification is in most of the cases still done through manual inspection of documents. The method for verification of most of the design requirements is by testing, a technique performed on the submitted element by which functional, measurable characteristics, operability, supportability, or performance capability is quantitatively verified when subjected to controlled conditions that are real or simulated. Testing often uses special test equipment or instrumentation to obtain accurate quantitative data to be analyzed. In this context the BIM model and a corresponding specification of requirements can be seen as the equipment of automated verification. This '*testing*' can be done with the methodology of *rule-based checking*.

Current verification methods have largely concentrated on detecting structural anomalies among rules and incompleteness. Several rule-checking techniques can be used to develop a validation and verification system. One of the possibilities is a Rule-Based System (RBS) that can possibly support the specification of abstract requirements what can result in better measurable, computable and unambiguous requirements (Anantaram, Nagaraja, & Nori, 1998).

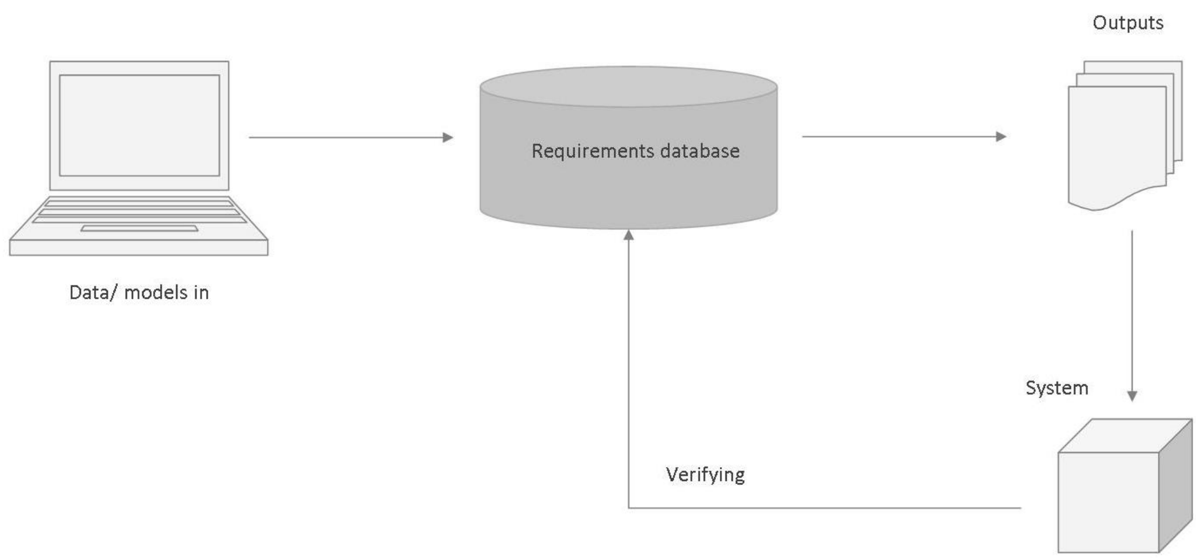



*Figure 11. Verification process components, own drawing.*

### Verification process

The basic objectives in validation and verification of requirements is to identify and resolve problems and high-risk issues early in the life cycle. Defining these two terms can via the following questions (Boehm, 1984). Validation is answering the question "am I building the right product?", where verification is answering the question about "am I building the product right?".

Requirements are the single element that ties all the product development lifecycle processes together. Defective requirements have a direct and significant impact on the project's cost and schedule at the end of the project. Due to this it is important to iterate the validation and verification process during the whole lifecycle of a project. Starting at the beginning of a project it is required to reduce cost to fix requirement defects (*Figure 12*). Requirement defects cause re-work and therefore increased cost that increases exponentially as the product lifecycle progresses (Wheatcraft, 2012). Validation and verification are major cost and schedule drivers for any project.

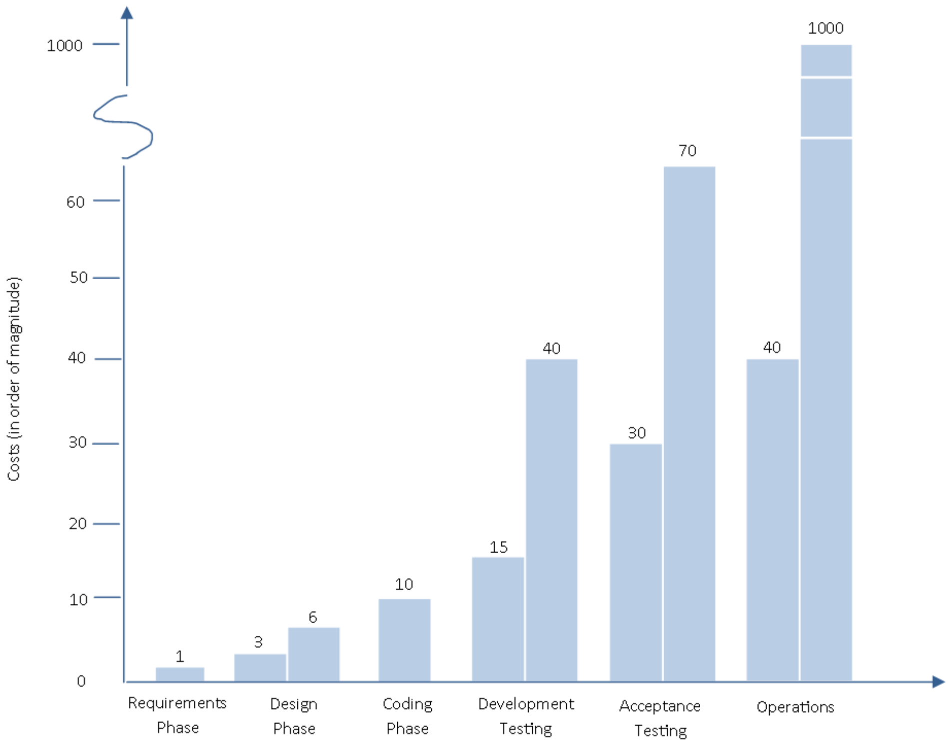



*Figure 12. Cost to fix requirement defects based on (Wheatcraft, 2012), own drawing.*

The requirements specification can be seen as the document that specifies, in a precise, complete and verifiable manner, the requirements of the object together with the procedures of verifying these. This verification process ensures that the solution is tested against the derived requirements.

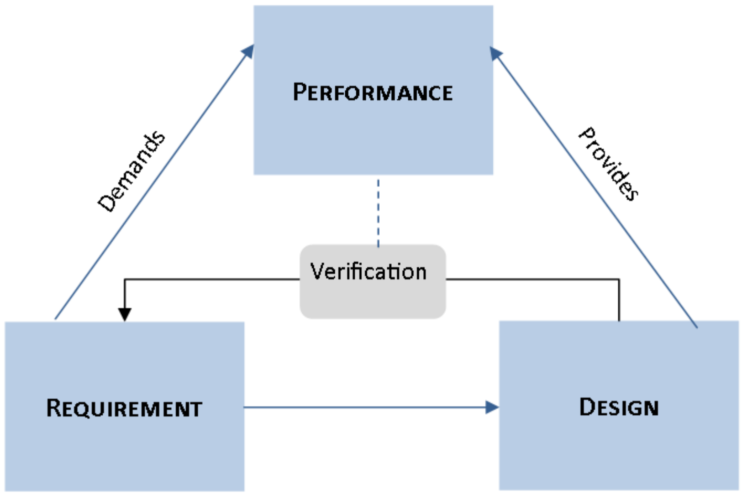



*Figure 13. Essence of the verification process based on (Moonen, 2016), own drawing.*

The verification process is consistent with the leading developments of so called Rule-Based Systems (RBS) (Eastman, Lee, Jeong, & Lee, 2009), which began exploration for building models in the late 1980s. Rule checking, which can be seen as similitude of verification, can be structured into four stages. The first stage consists of rule interpretation and logical structuring of these rules for their application. Next the building model must be prepared including the necessary information required for checking. The third stage is the rule execution phase, which is the actual checking. A last stage consists of the reporting of the checking results. Some research into automated verification of client specific requirements has been done by L. Moonen (Moonen, 2016), and focused on the physical requirements of a building design. To actually make automated verification usable for other requirements than physical requirements, in other words abstract requirements as mentioned before, it is important that this research starts with investigating how abstract requirements must be described in order to have usable statements for automated verification. During the process of developing a building, the requirements will be changed or new ones will be added. Verification of the requirements can therefore best be investigated during the final design of a case. This process is only worth as much as the data which is put in the process (Moonen, 2016). To give a verification value, validation of a requirement should be done to ensure that the verification can be done correctly. The validity of a requirement remains a difficult endeavor. To ensure this validity a requirement must meet the stated conditions of a requirement. As earlier mentioned in this research the verification process benefits from automated applications, which is not yet possible for every building requirement. Nevertheless, a lot can be learned from the application of automated verification for requirements that are ready to use for this purpose. Within the concept of rule-checking the use of computable requirements and conditions under which this will be possible are discussed.

## 3.5 Automated rule-checking

The benefit of automated rule-checking of construction projects is expected to reduce time, cost and errors of the checking process. However, state-of-the-art checking systems cannot achieve fully automation because of relying on the use of hard-coded, proprietary rules for representing requirements, which requires major manual effort in extracting information from textual documents and coding these information into a rule format (Lan & Shih, 2014)(Zhang & El-Gohary, 2017).

Automated checking systems will only be able to check requirements that can be computed, which makes it needed that abstract requirements can be computed (Niemeijer, 2011). Automated rule-checking is already often implemented in the context of applying licenses and permits as well as to asses compliance with building codes, which is all computable data. But also other domains are already subject for research, such as the impact of design options on the performance of buildings that can be assessed automatically using BIM tools (Martins, Rangel, & Abrantes, 2016). In order to use automated rule-checking routines as contribution to the development of building designs by verifying design options, checking the impact of it, the rule-checking software can be used as a KBS that supports the decisions of clients and assessing the impact of design options.

Automated rule-checking, has been a field of research within BIM since the early years of this technology (Martins et al., 2016). According to Eastman (Solihin & Eastman, 2015) an important part of rule-checking are the rule definitions. Today, the rules are typically written in human languages that require significant domain knowledge in order to interpret them into a machine interpretable manner. This '*machine interpretable manner*' is related to generating consistent, precise and quantifiable conditions for each rule. The NFR's, stated in an abstract way, are most of the time not computable yet and thus not ready to use for automated rule-checking.

### Rule-checking technique

Research development of RBS's for building models began approximately three decades ago (Garrett & Fenves, 1987). During this time a lot of development has taken place, but really effective rule checking systems are just now beginning to become available. The technology is therefore still young and quickly evolving. Based on Eastman et al. (Eastman, Lee, Jeong, & Lee, 2009) there are four software platforms that have been developed to support implementation aspects of rule checking systems, which all applying rules to Industry Foundation Classes (IFC) building model data.

Several software tools are available to help a designer manage a building design project and the associated requirements. BIM can be named as one of the most notable efforts in recent years regarding information management in the construction industry (Pauwels et al., 2011).

Rules and regulations are written by people and for a long time, were only read and applied by people. As a result, they were sometimes incomplete (particular conditions were not covered) or contradictory. Their structure was often arbitrarily complex. Improving the logical structure of regulatory codes was an area of early research. Rules for building designs are first defined by people and represented in human language formats, typically written in text, tables and possibly equations. The interpretation of these rules is subject to miscommunication, ambiguity and error prone.

On the one side, the building model must be prepared for using rule-checking automatically, and on the other side the rules themselves must be formatted in such a way these can be used for rule-checking which entails a computable statement.

A rule checking process is separated into four phases considering a rule interpretation phase, building model preparation phase, rule execution phase and a rule check reporting phase (Pauwels et al., 2011). Last year many approaches have been researched upon for rule checking in the construction industry. In several approaches a conversion phase is required for converting the building model viewed in IFC into a building model containing the information needed by the rule checking environment.

According to Eastman (Eastman et al., 2009) checking of requirements is often a costly bottleneck and automated code reviews have the potential to save significant time and cost. Eastman also expects that the application of rule checking systems will move beyond code checking of building codes and accessibility criteria and become a standard tool used throughout the building lifecycle. A rule-based assessment tool can be implemented for various platforms; first it can be an application closely tied to a design tool, such as a plug-in, allowing checking whenever the designer wishes. Second, a stand-alone application parallel to a design generating tool, or third a web-based application that can accept design from a variety of sources. In most of the cases the developed applications are based on the third concept.

### Rule checkers

The purpose of a rule checker is to assess designs by checking models to their configuration towards the requested performance (Eastman et al., 2009)(Hjelseth & Nisbet, 2010). The answer to this kind of check is a yes or a no on the given rules (Pauwels et al., 2011). According to Hjelseth & Nisbet, there are four intentions for model checking; validating systems, guiding systems, adaptive systems and content based checking. This research focusses on the examining character and therefore on validating systems checks. According to Eastman (Eastman et al., 2009), the different categories he discusses as part of the technique of rule checking can be used for a total of seven different categories. This research will focus on the category of abstract formulated client requirements. These are the requirements which are realized by the client to achieve certain qualities they need for a suitable environment according to the needs and functions of the building.

The eventual data (IFC) that will be checked can be asked of four different classes of rules (Solihin & Eastman, 2015).

- Class 1: Rules that require a single of small number of explicit data. The data which is checked in this rule class is accessible directly in the model and does not require extensive preparation. The access can be done with basic queries to evaluate availability of data or relations.
- Class 2: Rules that require simple derived attribute values. Based upon single values or small data sets. New data does not have to be generated, but multiple actions could be needed to execute these rules. Implicit relationships within the requirements should be identified in this class of rules.

- Class 3: Rules require extended data structures. Data must be generated to execute the check. The execution of these rules relies on a geometry engine that can evaluate the model on its geometry and relations with the use of algorithms and calculations.
- Class 4: Rules that require a 'proof of solution'. These rules will give a pass, fail or error. This requires describing how a model passes the rules instead of just complying with the prescribed rules. The complexity of these rules does not have to rise in comparison with the other rule classes formulated, but have a different focus.

In this research is being worked towards verification based on rule class 2.

### BIM-based model checking

One of the available software tools is BIM-based Model Checking (BMC) (Hjelseth, 2015). BMC is software which processes the content of information in BIM-files according to rules specified as pre-defined procedures. There are three components a BMC consists of; software, rule-sets and BIM-files. Model checking explores the content of information in BIM-files in a transparent way. The software is based on the principle that a logic-rule connected to the information from a BIM-file can have the outcome of Pass, Fail or Not checked. This logic of a rule is basically a simple question answered with "yes", "no" or "not checked" due to missing information. These principles of BMC are leading and often the components are the base for all rule-checking software.

In accordance to BMC the specified rules as result from the definition of abstract requirements must be translated via a certain way. The belief is that this translation process is very difficult to capture, but Knowledge-Based Systems (KBS) can support this in the next way. Knowledge of experts provide the base on which decisions are made that define a certain abstract requirement a client has. This knowledge, including expertise, references and experience, contains what is needed for the translation. When capturing this knowledge and then from more than just one expert, this knowledge can lead to a more consistent base on which decisions are found. At the same time an interaction between a client and this information can improve the process of defining what an abstract requirement means to a certain person. This interaction can deliver benefits regarding time, costs and errors in the design process. This is the main idea for using KBS in the definition process. Expert Systems (ES) are one of the possibilities that are concerned with decision making purposes. Concerning the interaction that is believed as a good function of such ES's, a typical Question-Answer System (QAS) will be the concept on which the tool will be based.

### 3.6 Expert Systems

#### Use of expert systems in the construction industry

As mentioned before the rule-checking software can be used as a KBS in the form of an Expert Systems (ES) that supports the decisions of the clients and assessing the impact of design options. In this case the data about requirements can be the base for developing an ES and can be connected to the verification of abstract requirements.

ES's in the area of construction management is not new. Research on ES's has been a topic for more than twenty years (Yang, Li, & Skitmore, 1996). However, as with any other emerging technology, the expectations surrounding ES's have been slow to come to fruition. As a result, there has been a considerable drop in the number of new ES applications in recent years. In the late eighties there has been a hype of ES development and was followed by a 'cool-off' period in the nineties. During the hype several system areas have been used for attempts in construction management. These can be classified into four categories; selection systems, advisory systems, monitoring and control systems, analysis and evaluation systems. None of these categories uses the RE process as base for their ES development.

Every ES is concerned exclusively with decision making (Golik & Golik, 2000). Construction management, which is concerned with both decision making and implementation, is a very likely area for ES applications. Despite this fact, the 'cool-off' period has made an end to some impact on existing construction practices. The expectations were slow to be realized as a result of the limited capacity of development tools, inadequate methodologies and absence of construction knowledge in problem domains (Yang et al., 1996). The potential of ES lies within the ability to capture, incorporate and automate judgements, intuition, rules of thumb and other forms of human expertise that are difficult to model using conventional techniques. ES's can fulfil a specific role in an area where knowledge, experience and qualitative approaches still dominate. Something that is just the issue with abstract requirements during the design of a construction project.

In the construction industry it is difficult to bring industrial experience and knowledge together (Laptali & Bouchlaghem, 1995). Due to the characteristics of the industry it differs from other industries by the physical nature of the product, structure of the industry and organization of the construction process which affects this bridge between industrial experience and knowledge. The building, which is made specially to the requirements of each individual customer and other regulations, determines these characteristics.

#### Knowledge management

Due to the changes from an industrial driven to a more and more knowledge driven society the aspect of knowledge is the foremost important resource a company can have. This is due to the assumption that a lot of administrative benefits might be achievable on an organizational level by the right interpretation of the available information and data. Together with this trend the AEC industry is induced to work more effective and efficient due to the complex demands that arise from the client's need within this domain (Delghandi, 2018).

Knowledge management (KM) has being recognized as a vehicle through which the industry can address its need for innovation and improved business performance (Kamara et al., 2002). The failure to capture and transfer project knowledge, especially within the context of temporary virtual organizations, leads to increased risk of 'reinventing the wheel', wasted activity and impaired project performance. One of the topics within KM are knowledge-based ES's to codify knowledge through the use of Information and Communication Technology (ICT) tools. The need of KM in the AEC sector is fueled by the need for innovation, improved business performance and client satisfaction. While the construction industry operates within a dynamic and changing environment, clients become more sophisticated, insisting on better value for money and demanding more of construction for less expenditure. The interest in capturing knowledge has been expressed in the idea of development of Knowledge-Based ES's.

### Capturing requirements as knowledge

It has been found from past experiences (Laptali & Bouchlaghem, 1995) that the use of ES's in construction projects have several benefits which contains the sharing of expertise within the firm, improvement of expertise, speeding up the early decision process in combination with improvement in quality, an increased consistency in decision making and a formalization of expertise. In the literature of the use of ES's in the construction industry several limitations have been discussed according to the acceptance of ES's. The scope of ES is confined to narrow problems only for individual activities, the systems that were build vary in form and structure which means interchangeability and communication between systems was very limited and the users were not incorporated into the system without reprogramming by the developer.

As stated in earlier phrases of this literature review ambiguity will play a role within capturing requirements as knowledge and translate these into specific and computable requirements that possibly can be used within automated verification. In the development of an ES it is therefore highly important to take into account as much as possible of this aspect. Excluding ambiguity out of the development in this research will not be completely possible, but taking into account the best as possible is nevertheless important.

### Development of Expert Systems

Nowadays there are powerful tools for the generation of ES's. However, creating an ES on the basis of these tools becomes a very difficult task for users without specific training (Ruiz-Mezcua, Garcia-Crespo, Lopez-Cuadrado, & Gonzalez-Carrasco, 2011).

At this moment the construction industry has little knowledge about advanced techniques of artificial intelligence (AI) in the modern time, due to the decreased hype of ES's in the nineties (Yang et al., 1996). ES's, as a form of Knowledge-Based Systems (KBS), can be very difficult to develop. Creating a tool that is easy to use but still has enough power to solve problems and can be used by the domain expert makes the technology of ES's accessible in all types of companies and domains (Ruiz-Mezcua et al., 2011). Including ready for prototyping and further developments and implementations.

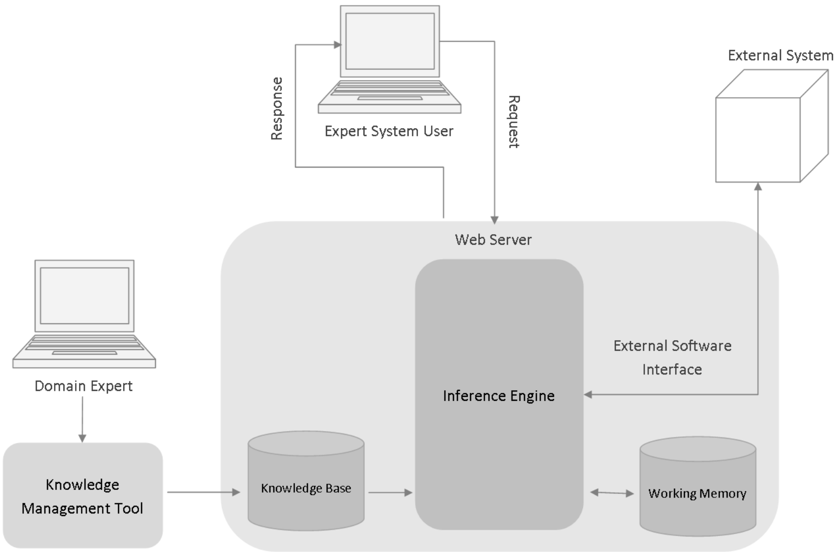



*Figure 14. Framework Architecture of Expert System for users with no experience based on (Ruiz-Mezcua et al., 2011), own drawing.*

In AI, knowledge representation (KR) is an essential area that is concerned with how knowledge is stored and processed and that attempt to understand intelligence (Tanwar, Prasad, & Datta, 2014). KR can be used for representing the knowledge required for Question Answering Systems (QAS). This system can be grouped under Knowledge-based ES's and is the best to support capturing the relation between descriptive text and computable values.

In the eighties people assumed that computers will take over the role of people. However, ES have not realized the expectations. In order to this an ES is downgraded to a supportive tool to the human experts. One of the major problems within this kind of systems is the fact that they assume that experts do not learn more than the knowledge they have on a certain moment in time. But in reality experts keep learning. This issue is not part of this research and development. It is focused on the employability and possibilities of ES in the construction industry. One of the things needed is of course knowledge, but is it possible to use this kind of systems within the requirement specification process? To find this out a QAS is developed and used. This knowledge is represented through rules, for which the specific terminology of Rule-based Expert Systems (RES) is used (Dohn, K. Guminski, A. Matusek, M. Zolenski, 2013). It is interesting to look how experts knowledge can be used in translating abstract requirements into specific defined and computable stated rules. To fit within current emerging trends such as automated verification, this research investigates where and how the concept of ES can be deployed in the requirement specification process.

### 3.7 Conclusion

This literature review has been conducted to allocate the research problem and objectives. And to give an answer to the different research questions. It has been important to describe and evaluate the current issues regarding the design processes, especially the requirement specification process, verification process and the current possibilities of automation. Also the connection with KBS is evaluated. As conclusion of the literature review there can be given some direction to the answer of the main research question.

Abstract requirements are difficult to use within design processes, but will always existing in building projects. Many properties of these requirements are discussed in the literature which makes a defined description of the dealing with these very difficult. In this research is focused on the computability of abstract requirements in order to verify them using BIM. Following the developments of automated verification it is this kind of requirements that are still specified and verified manually by always different experts. Paired with different clients there is plenty of room for misunderstandings which can lead to errors and their consequences. For these kind of requirements there is always specific knowledge needed to specify. From this, the idea arises to collect this knowledge in order to reduce errors and relating time and costs issues and to ensure more satisfaction between client and contractor in the end. One of the best ways to collect certain knowledge is by representing human experts knowledge according to certain rules in a computer based system. This resulted in the concept of QAS as part of Knowledge-based ES's.

Abstract requirements can be captured in three ways;

1. Via a standard – this is just something which must be met
2. To which experts attach a value – a connection between a description and possible values
3. To which experts cannot attach any value to – the concept of Knowledge-Based expert Systems cannot help with this either

Especially the second way is interesting for this research to see which role and value ES's can perform and add during the RE processes in the design phases.

The emerging trend of knowledge capturing in combination with BIM makes it possible to improve several design processes in the construction industry. Since abstract formulated client requirements still cause a considerably amount of errors and cost the use of Knowledge-Based Systems could be part of the solution. Now that is known what abstract requirements are about and made them difficult and different to dealing with, it is important to investigate the possibilities to improve this and not to linger in traditional process methods. But in order to take a step towards this improvement, more research into dealing with abstract requirements and the current practice is needed to prove the idea.

### 3.8 Research Gap

In the conducted literature review the most relevant topics and concepts are discussed. These subjects match the concepts as mentioned in the conceptual framework in chapter 2. Also some of the concepts which are not playing a big role in this research are described in a less extended way. This review gives a comprehensive overview about current and relevant knowledge for this research.

However, there is a lack of research into the dealing with and specification of the abstract requirements that are missing a measurable and computable component. Via interviews this must give a better view about possible improvements and shortcomings in this process. Related to this and during the interviews also the feedback and possibilities of KBS's will be evaluated. The expectation exists that these kind of KBS can support and improve this process regarding these kind of requirements. Development in the latter stage of this research must proof the possibilities. Even though automated verification via, for example, BIM is already broadly discussed and researched upon it is really relevant and interesting to research upon the connection between the specification of abstract requirements and verification via BIM. In this sense this research is focusing partly on subjects that are barely research on in context of already conducted and existing research articles and results. There is a lot of knowledge and research about requirement verification, requirements and their characteristics and KBS's. But the deployment of these kind of systems within the requirement processes to better translate the abstract requirements for, in particular the lack of measurability and computability aspect of it, is undervalued in the studies.

# 4.0 Qualitative Research

## 4.1 Motivation

In order to work towards the objective of this research, an overview and clear understanding is needed about the requirements specification phase of abstract requirements. Next to the conducted literature review an overview of the current practice in the requirements definition phase of abstract requirements is needed to evaluate the possibilities for Expert Systems (ES) and automated verification. But also for gathering information, knowledge, experiences and expertise about the abstract requirement of comfort which will serve as example for the expert system. The method to gather in-depth information about abstract requirements and the requirement specification process in practice is by interviews (Maxwell, 2013) with experts in the field of requirements definition processes. The possibilities of ES's, which are based on experts in a particular field, can be best based on interviews with these experts. The interviews are therefore used to investigate how experts think about the concept of ES's. As well to get knowledge about the current dealing with abstract requirements to better fit the development in current issues of specification.

The expected outcomes of the held interviews can be divided in twofold. On the one side the interviews must result in understanding which steps are taken in the requirements definition phase during design processes. And on the other side the results of translating the abstract requirement of comfort into specific values must be gathered during these interviews as base for the example for the ES. In addition to these results the ES tool will be developed.

## 4.2 Interviews

The objective of the interviews is to get knowledge about how is dealing with abstract requirements in practice. And also to get a founded base for the input of the expert system to develop. The developed method in this research is based on one requirement what should apply to other requirements of the same definition for these abstract requirements. The abstract requirement of comfort is therefore chosen as starting point and scenario during the interviews. comfort has been a research subject for several researchers and often has a twofold concerning concreteness and subjective statements.

### 4.2.1 Interview purpose

The aforementioned twofold as outcomes of these interviews can also be turned into the main purpose of these interviews. For filling the expert system with knowledge, experiences and expertise of the experts a chosen scenario during interviews and the test case makes this research more practical. This more practical approach is in fact the using of comfort as abstract requirement and considering a working office as building space where this requirement is a constraint. This is narrowing down the scope and use of this research, but at the same time this ensures a more usable Proof of Concept (PoC). Also during interviews this can ensure a more in-depth content and deployable result.

Out of the results of the interviews one of the most interesting breakdowns of comfort will be used as base to develop the ES on. This represents the experts knowledge gathering as input for the ES.

Beside this goal, more information about how experts dealing with these requirements in their daily base is valuable for this research and the future research into this topic.

### 4.2.2 Interview setup

#### Motivation

The use of interviews as method is very useful to represent the current situation, and get in-depth information about the process and its strength and weaknesses. And also the use for interviewing several experts about their experience with abstract requirements is very desirable.

These interviews aim to:

- Get insight in the current RE process of abstract, undefined and non-computable requirements.
- And how is dealing with these requirements during the design process.
- Define the difficulty of specifying the requirements parameters and assign measurable values to these requirements.
- How is dealt with specifying an abstract requirement, as example in this research the requirement comfort for workspaces.
- Evaluate how Expert Systems can contribute to facilitate this process in an easier, more efficient and solution-oriented way.

To perform this interview from the perspective of qualitative research in total 11 interviews were held. The focus group of these interviews are people who are dealing with abstract requirements in their work. Thought is about experts who are part of requirement specification processes, design processes or verification processes in the design phase regarding utility buildings. By focusing on the requirement comfort and a specific space of an utility building, it is specific enough to held an interview about and to extract specific results for this research. On the other hand it is also possible to use the interview results and developed ES again on a broader basis for other abstract requirements relating to utility buildings.

#### Interview questions

The literature review delivers the background information on which this research is based and from which the research questions partly can be answered. The interviews are used to lay down more focus on abstract requirements and the requirements definition process. For the design of the interviews the research questions are broken down into sub-questions to address the needed information and accumulate this in order to reach the formulated goals. The following research questions are addressed:

1. *How are the abstract requirements influencing the dealing with requirements in the design phase?*
2. *What are the characteristics and difficulties of abstract requirements concerning the specification process of these requirements?*

3. *What is the current process of dealing with abstract requirements during the design processes?*
4. *How can Knowledge-Based Systems, in the form of Expert Systems, be a support to the requirement definition process?*

In the next paragraphs the sub-questions and goals per research question will be discussed.

1. **How are the abstract requirements influencing the dealing with requirements in the design phase?**

This question aims to gather knowledge about the RE process in relation to the design process. Because there are different requirements within a building project, the influence of abstract requirements can be different then dealing with other requirements. Dealing with these requirements, specifies different sub-questions for the interview related to this RE process:

a. **In which phase of the design process (based on the ten phase scale) was dealt with these requirements?**
b. **Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?**

Goals:

- Identifying the consequences of dealing with abstract requirements in design processes.
- Investigating how the design process is shaped regarding the presence of abstract requirements.
- Evaluate influence of the requirements definition process on design processes.

2. **What are the characteristics and difficulties of abstract requirements concerning the specification process of these requirements?**

Out of the literature review (chapter 3) many characteristics of abstract requirements are discussed. As a consequence of these the difficulty of dealing with abstract requirements became to the utmost extent clear. In practice expert will face many of these difficulties too, but also others will come across. Therefore is it interesting and clarifying to ask experts about these factors of abstract requirements. The following sub-questions are stated:

a. **What are the differences between abstract and specific formulated requirements?**
b. **And what are the characteristics of the abstract requirements?**
c. **How are you specific dealing with the abstract requirement *comfort* in design processes?**
d. **Which expressions and relating rules are you assign to this specific abstract requirement?**
e. **What are the difficulties with these kind of requirements from your point of view?**

Goals:

- Identifying the process of relating expressions and rules to specific requirements.
- Investigating characteristics and difficulties of abstract requirements in addition to the literature review.
- Defining how experts dealing with abstract requirements.

3. **What is the current process of dealing with abstract requirements during the design processes?**

Something only can be clarified by interviews is the current practice of dealing with abstract requirements. Regarding the use of KBS's it is valuable to know what the current process is and where a possible improvement can be made. After the discussion about the design process and abstract requirements this question must fulfill an in-depth study towards the current elements of the specification process. The next sub-questions are therefore asked:

   a. **What is the currently used main approach when you are facing such requirements during design processes?**
   b. **Based on which method or technique have you specified this abstract requirement?**
   c. **How can you justify the choices in a design, when dealing with this abstract requirement?**
   d. **In which way verification of these requirements takes place towards the client?**

Goals:

- Identifying the process of dealing with abstract requirements during design processes.
- Defining how experts dealing with the process of translating abstract requirements into specific values and units.
- Evaluating how verification of these requirements is currently done.

4. **How can Knowledge-Based Systems, in the form of Expert Systems, be a support to the requirement definition process?**

The application of ES's in RE can gather the knowledge and related information of experts into a system to define abstract requirements. The process of translating an abstract requirement into a specific requirement can be optimized according to the use of such ES's. Which defines the following sub-questions:

   a. **Which method are you using to specify an abstract requirement?**
   b. **In which way can Expert Systems contribute to this process?**

Goals:

- The possible added value of using ES's.
- Evaluating how verification of these requirements can be optimized in an automated way.

These different sub-questions that resulted from the research question have led to a questionnaire (Appendix 9.1) which contain all interview questions to address all the formulated goals. In the next part the interview results will be discussed and give an overview about the gathered information and knowledge.

#### 4.2.2.1 Setup per Subject

The formulated goals can be translated back to three subthemes. These subthemes correspond to the main themes that were discussed in the literature review. The subthemes that are discussed here are design process, RE and ES's.

##### Design process

In this part the purpose is to get more insight in the current practice regarding RE during the design process. It must give insight where the difficulties in terms of dealing with the abstract requirements are occurring and what this phase contributes to these difficulties or whether it allows for better dealing of these requirements. This is very relevant to see if the abstract requirements is influencing the structure of the design process and vice versa, i.e. if the design process is affecting the dealing with these abstract requirements.

##### Requirements Engineering

The most comprehensive subtheme is the part about RE. Out of the literature review this process and relevant concepts are already discussed, but for the interviews two in-depth subjects are relevant. The process of RE includes both the specific and well-defined requirements as well as the abstract requirements. In this research it is most relevant to focus on the abstract requirements. For this part the purpose is to evaluate the differences between these two kinds of requirements and the way how is dealing with these during the design process. According to the differences it is interesting to get insight into the characteristics of these requirements and an overview of difficulties regarding abstract requirements. Also to see what the main approach is for dealing with the abstract requirements now and what the possible opportunities are to improve.

For the development of an ES the next in-depth subject is about one of the possible abstract requirements someone can face during the design process of an office building. To use a clear and specific example, comfort at work spaces is used. This example simplifies the specification and can be used as input for an ES. This input has been defined as the knowledge base of the system. This abstract requirement is chosen because of its multi-character. Comfort is both technical and can be specified according different guidelines, and has elements that are for everyone different on the psychological side. Most of the times experts can enter their knowledge and suchlike about a subject via a knowledge management tool. But since this research focus on a PoC for a method of specifying abstract requirements, development of such a tool is out of scope for which these interviews are used as base. Regarding the example of comfort the knowledge about this requirement and their used sources form the input of the ES development. Its subjective character makes it interesting to find out how is dealt with the fact that experts fill in these abstract requirements for their client, while these are different for everyone which often leads to miscommunication and other interpretation issues.

### Expert Systems

The last theme is about the use of ES's. In the building industry its use it not new, but for the specification process it is not (yet) used. Therefore this research will answer in particular the question if and how ES's can support this abstract requirements definition process related to automated verification. During the interviews it is also good to examine the thoughts about the deployment and added value of the use of ES's in the abstract requirements definition process, whereby the support out of the construction sector can be evaluated. A last goal is to see if the ES's only possibly can contribute within this process or if it also can contribute in the possibilities of automatically verifying abstract requirements.

### 4.2.3 Interview results

In total 11 interviews have been held among different companies and different backgrounds. A variety of people have been interviewed, but all with expertise in the domain of RE and/or specification and dealing with abstract requirements. Often these people perform a function out of the advising role during these processes, which is a very common function for involvement in design processes. Since this research is focusing on utility buildings all the interviews that were held have been done with people working in the utility building sector. This is really important since these abstract requirements have a completely different meaning in sectors like infrastructure.

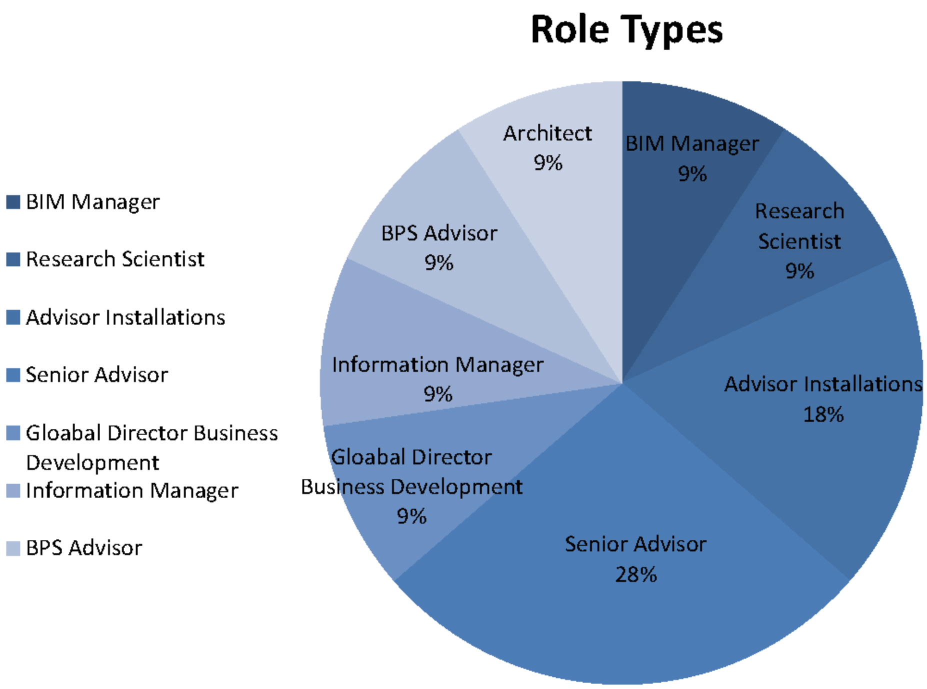



*Figure 15. Different role types of the held interviews.*

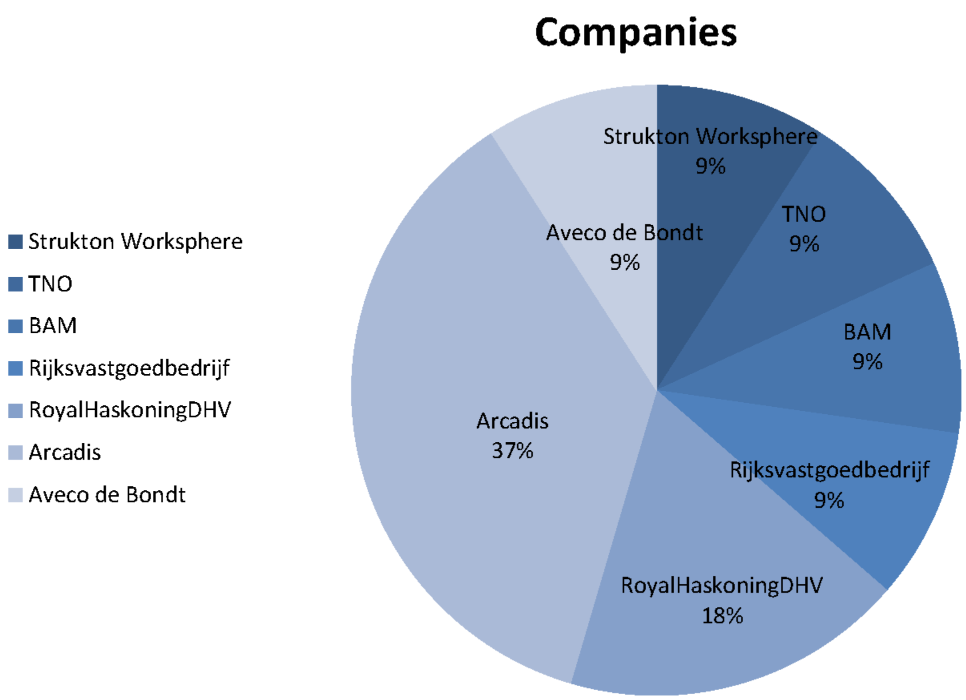



*Figure 16. Division in companies of the held interviews.*

| Nr. | Date | Place | Company | Role | Duration |
|---|---|---|---|---|---|
| 1 | 12-11 | Son | Strukton Worksphere | BIM Managaer | 1 hour |
| 2 | 16-11 | Den Haag | Rijksvastgoedbedrijf | Senior Advisor | 1,5 hour |
| 3 | 16-11 | Rotterdam | RoyalHaskoningDHV | Global Director Business Developement | 1 hour |
| 4 | 19-11 | Amersfoort | Arcadis | Information Manager | 1 hour |
| 5 | 27-11 | Maastricht | Arcadis | Senior Advisor | 1,25 hour |
| 6 | 29-11 | Den Bosch | Arcadis | Senior Advisor Real Estate | 1,25 hour |
| 7 | 29-11 | Breda | TNO | Research Scientist | 1 hour |
| 8 | 4-12 | Bunnik | BAM | BPS Advisor | 1,5 hour |
| 9 | 7-12 | Eindhoven | AvecoDeBondt | BPS Advisor | 1,5 hour |
| 10 | 10-12 | Eindhoven | RoyalHaskoningDHV | BPS Advisor | 1 hour |
| 11 | 11-12 | - | Arcadis | Architect | - |

*Table 2. Overview of held interviews.*

The interviews were held in a semi-structured way. The semi-structured is described as a more open character of the interviews. A predetermined structure of questions is followed during the interviews, but there is space during the interview to deviate from the formulated questions to get more in-depth information and examples or to discuss new ideas. Regarding to this topic this way of interviewing is chosen due to the fact the goal is to get in-depth information and not only to get confirmation or really specific answers about the subjects. It is mainly about discussing and get insight in the currently used approach and dealing with abstract requirements, in which an as large as possible collection of information and knowledge is preferable and needed.

According to *Table 2* interview 7 and interview 11 are marked. This has to do with the fact that these interview cannot be fully used in this research. The interviewee of interview 7 only answered the questions about comfort and barely about the requirements specification process. This because of her role as research scientist she was handling processes about testing and verifying specific knowledge about comfort in existing buildings. Nevertheless, it was an interesting conversation with a lot of information about the specific requirement of comfort, but this gave no input regarding the current handling of such requirements during design processes. Interview 11 was conducted in answering the questions via a mail conversation. This because she was not able to make an appointment for a meeting due to lack of time. The answers on the questions  are real short and give not the intended in-depth information the interviews must deliver. The complete answers on the interview questions per interviewee can be found in Appendix 9.2.

#### 4.2.3.1 Design Process

The two topics that are discussed in this section are the influence on the design process and difficulties regarding dealing with abstract requirements. It is interesting to know what experts are arguing about these topics that are also discussed and described in the literature review.

##### Effect on design phase

In an ideal world the design phase is classified as defined in the literature review for the Dutch construction sector (chapter 3, *Figure 5*). And every phase can be followed successively. Nothing is less true. In the real world a building project is often very complex with lots of processes influencing each other and little clarity at the beginning of it. You would prefer that every requirement is stated defined, specific and unambiguously and no changes occurring during the design. But the truth is that always an attempt is made at the beginning to translate the abstract requirements into specific requirements, but the client is not always able to do this. Several methods and techniques are used to obviate this translation as early as possible, with the focus on investigating and understanding the client.

Other experts says that in-depth translations of abstract requirements already have been made before the design brief is written. During the several moments with the client, that are often contracted, the exact specification of the requirements will be made. This is one of the methods to get some clarity at the beginning. But honestly, this does raise the idea experts are putting themselves on the spot of the client. They are filling in the specification with their own ideas. Of course they have expertise, experience and knowledge about these topics, but as stated before no human being thinks the same and attaches the same value to concepts and requirements.

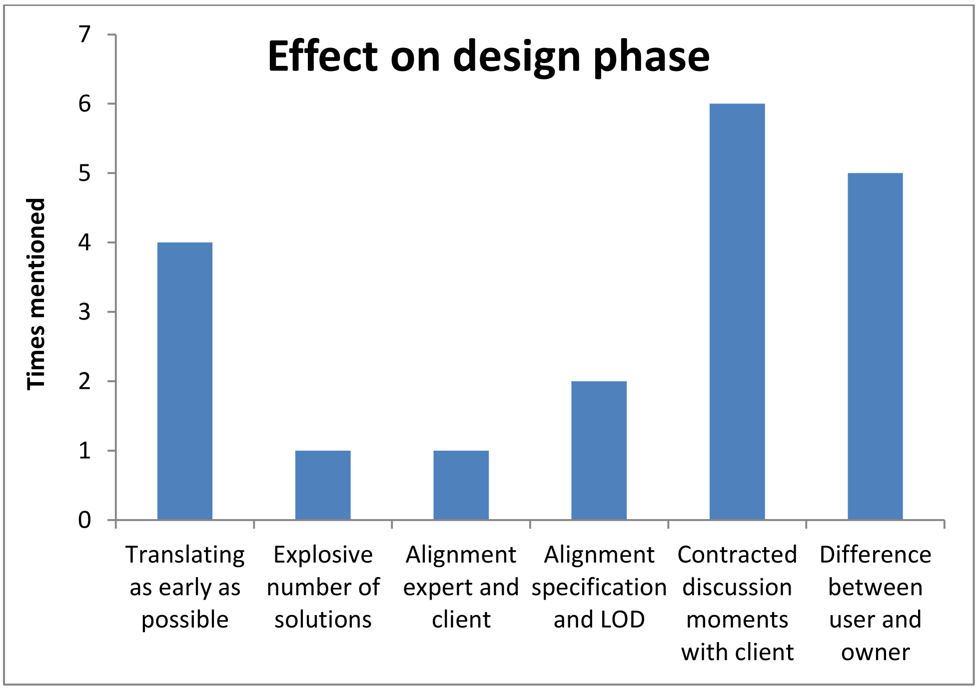



*Figure 17. Effect of abstract formulated requirements on the design phase.*

As can be seen in *Figure 17*, translating as early as possible is one of the topics that is mentioned more often than average. For the continuation of the process this is of course needed, but in the current requirement specification process not always possible. Certainly not when there are just a few discussion meeting with the client contracted. The client, often the one who is the owner of the (future) building, is definitely not the same as the building users. In this research is also focused on the role of the client and that of building users during the specification process.

Another discrepancy is between the level of specification and the level of development in the design process. In the begin of the process the BIM contains to less information to see what these requirements really mean regarding to the model. But changes are not always implemented in a later stage because of the costs these changes cause (chapter 3, *Figure 5*). During the discussion meetings between expert and client they must find each other. The experts possesses the knowledge and expertise and the client does not always know what he wants or cannot explain it. Translate the specified requirement back to abstract requirements could help to investigate if the client really had that in mind.

The abstract requirements are often concepts with a broad meaning and possible interpretation. This raises the freedom of specifying the requirement, but at the same time does it have a big influence on the process regarding time, budget and discussions. During the process of designing after every design phase is looked back at the design brief for conflicting requirements, changes or other modifications. This is based on the fact neither the design and nor the abstract requirements are fully known and specified. In real projects, based on the experts, this does not happen because also in a later phase still not all is known.

### Difficulties

In the graph below all the different answers regarding difficulties of abstract requirements are presented. Together these answers cover the view of the experts about what the difficulties in their daily job are regarding the dealing with abstract requirements.

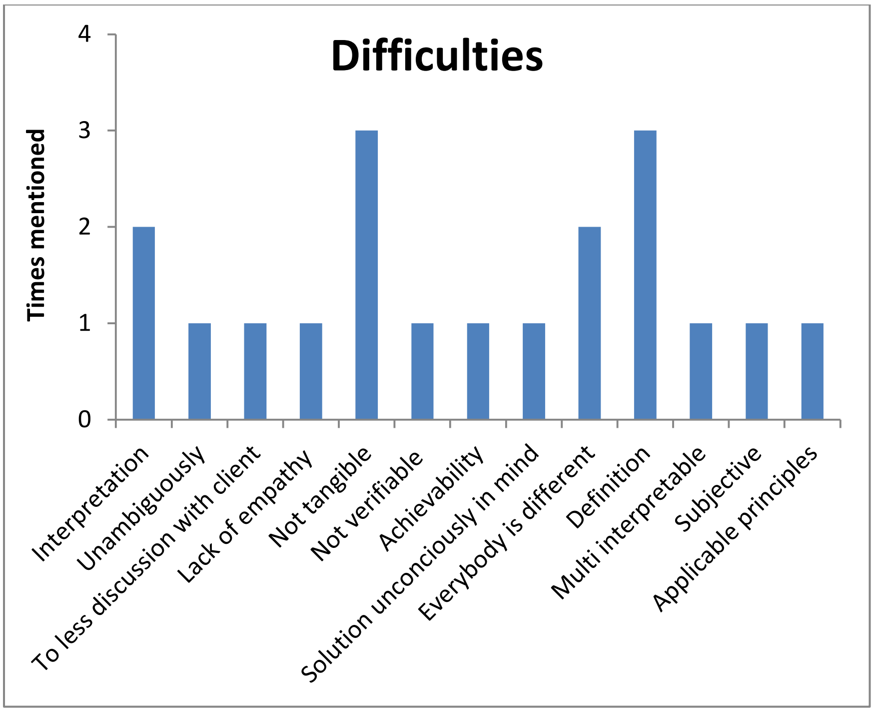



*Figure 18. Difficulties relating to abstract formulated requirements.*

In total thirteen different answers are given and just two of them are slightly stand out. The first one is the reason the requirements are not tangible. Abstract requirements are discussing often concepts that are not tangible within the building and are therefore difficult to dealing with and create struggles in verifying. The definition of abstract requirements corresponds with the core of this research in the way it focuses on translating these requirements into specific, and thus defined, requirements. Other reasons are also very relevant for this research and most of them are also discussed in the literature review. Which shows an almost equal view regarding abstract requirements.

At the start of a project the client and contractor must discuss about the needs and wishes in order to formulate a clear requirement specification that specifies the design constraints. During these discussions the interpretation is always a topic. Even a colleague with whom someone will work reads and thinks different about these same requirements, which makes it multi interpretable. There is also observed that technically trained people seem to have trouble with the connection between the requirement and the specific value it has for a client. The value for the client is one of the elements more focus is needed. Not a specific value of a requirement, but why a certain value of a requirement is important for the client. The real reason in here is the lack of empathy during the discussions with the client.

Something which also has been made clear during the literature review is the difficulty in making these requirements unambiguously. One of the possibilities in avoiding ambiguity in requirements, not for the full hundred percent, is taking enough time to discuss the requirements with the client to make it clear. When there is not enough time for the discussion it is more difficult to make the requirements specific. However, technical solutions as software to standardize can partly obviate this.

During the discussion it is important to work towards achievable requirements. The expert in here is the person who has a lot of knowledge and experience to know what is realizable and what not. Clients often name needs and wishes that look well-thought, but are not always factual and real. About this discussion, experts also mentioned the fact technicians working with these requirements are rather quickly inclined to think in terms of numbers and solutions. This while a requirement in the design brief is often formulated with a certain solution in mind. Actually the best process is to work in a reverse way. The solution in terms of the end result should serve as starting point from which the different needs and wishes can be formulated and specified in a top-down reasoning that corresponds with the earlier described SE approach. The subjects that are named within these requirements are during specification always element of subjectivity, because every human being is different. And in everybody's mind the reasoning pattern works different. This is maybe the most difficult element in the specification process of abstract requirements where a consistent and well-thought procedure is needed for.

#### 4.2.3.2 Requirements

The difficulties of the abstract requirements within the design process says a lot about the characteristics these requirements have and why the RE process is difficult to perform. The characteristics describe where the value comes from, while the difficulties are related why dealing with these during design is so hard.

##### Abstract requirements characteristics

In the literature review a lot of characteristics are discussed. In this paragraph the differences between abstract requirements and specific defined requirements is discussed according to what abstract requirements describe.

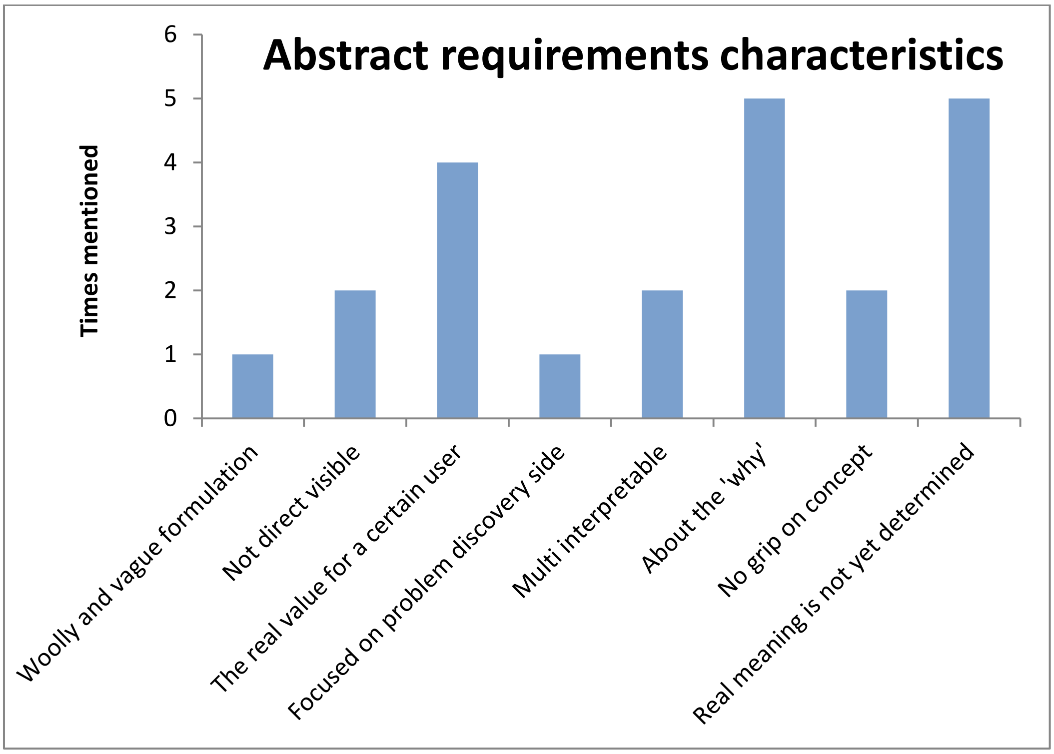



*Figure 19. Characteristics of the abstract formulated requirements in building projects.*

Abstract requirements distinguish themselves by being formulated in vague and woolly terms. Often about concepts that describe qualities of a space or building and are not immediately visible. These kind of requirements require the need to track down the real value of a certain element or user, which is not needed with specific formulated requirements because these are stated clear and unambiguously.

Something that is mentioned which is also interesting is the fact abstract requirements are connected to the *problem discovery* side. Which is about why a client wants something. Something that often occurs is that the client does not know what his own actual problem is. This corresponds to the main challenge regarding abstract requirements. Most of the time, till the end of specification, the requirements stay multi interpretable and getting grip of it is very difficult. The translation is often done with a specific solution in mind. Rather to think with that solution in mind, one should be think much more about the value for a client within the context of the performed activities. This value is the result of the discovered problem, which is evaluated during the next paragraph of the currently main approach.

Not everything of these requirements can be recorded at the beginning. Because of this, several alternatives are looked upon, but decisions can only be made in a later stage of the design. This is an obvious difference between the specific and abstract requirements in terms of dealing with requirements. This also corresponds with the lack of determination of the real meaning of these requirements. It takes time and effort to specify the requirement.

### Currently main approach for definition

In order to find out where and how Knowledge-based ES's can contribute to the requirement specification process the currently main approach for this process is looked upon.

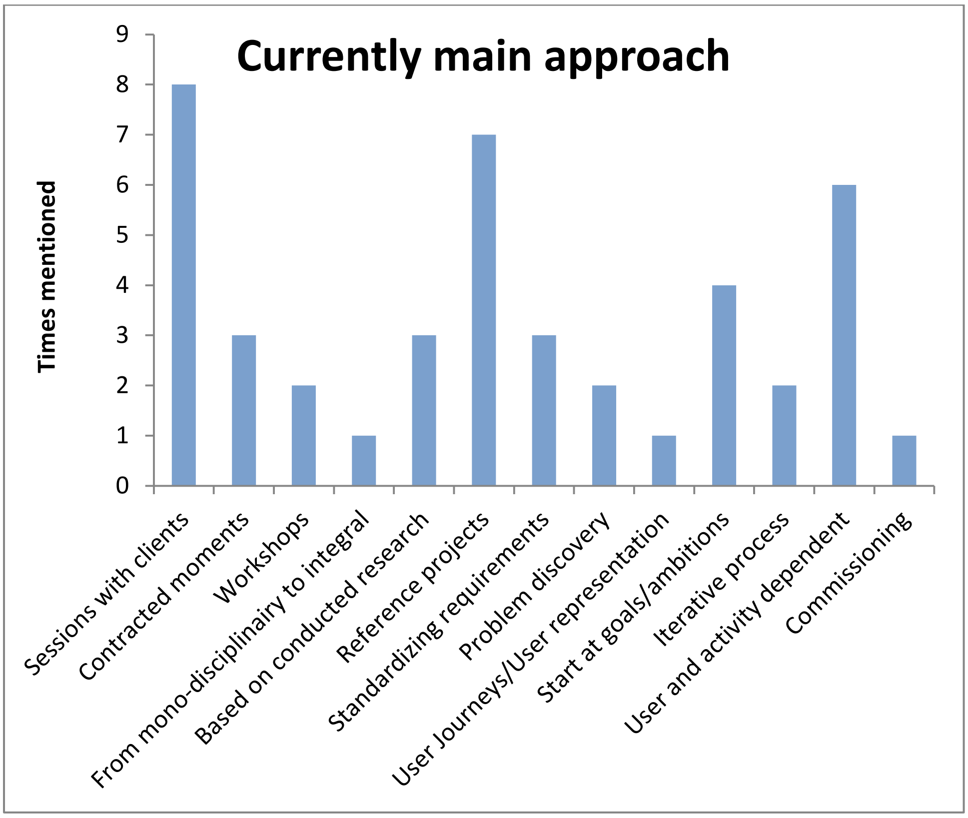



*Figure 20. Currently main approach in specifying abstract formulated requirements.*

As can be seen in *Figure 20*, almost every expert indicates that the abstract requirements are mainly specified during sessions with clients. These meetings with clients are contracted, otherwise new demands keep coming during the design process. The expert who is present at this meeting is often the one who has specific expertise in a specific domain.

A lot of these abstract requirements are already specified before the in-depth meetings. Often via mono-disciplinary sessions and subsequently integral sessions are organized to fully specify the abstract requirement, as fully as possible at a certain moment. Of course the earlier mentioned difficulties and characteristics play an important role in the main approach and result of these meetings.

Due to the lack of empathy, usable knowledge and understanding of the client the use of reference projects is also often a method that is used. In order to make discussed content visible. In the end it has all to do with the fact that for the continuality of the design process the contractor wants the requirements as early as possible defined and specified.

The approach is also dependent on the difference between professional and non-professional clients. Most of the time professional clients can be much more specific about the abstract requirements than non-professional clients. These latter one need several sessions, reference projects and the expertise of experts to make the requirements more specific. Some of the abstract requirements are standardized and then applied for all the projects. They just coping earlier used requirements. It configure mistakes, due to the fact that not every requirement can be copied to another project with another client belonging to it. And also not every requirement applies for different projects.

In order to find out the real value of an abstract requirement the concept of problem discovery can be used. This also focusses more on the real user of a space or building in contrast to the client which is often the owner and not always the user. User journeys and activities that play a role in using the building are more important to specify requirements than standardized values. This idea touches the method of SE in the way that is started at goals and ambitions and via a structured path towards the solution. It is of importance during specification that there should be much more discussion about this problem discovery as basis to formulate the requirements.

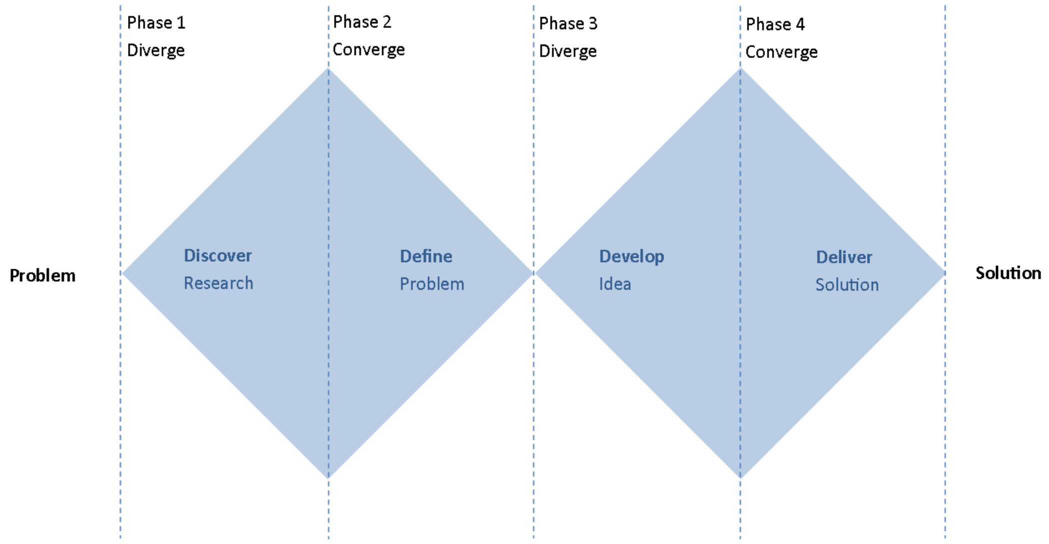



*Figure 21. Problem discovery based on the method of double diamonds, own drawing.*

The client, the one with money, is in the end the one who is making the decisions and choices. And that while also is argued that these requirements are dependent of the user and activity. This is in contradiction with how this is done now. The user must get more influence in this process since they are the subject of this topic. It is often due to money and time that not all (possible) users are heard.

In order to find out in what way ES's can support this user involvement to specify abstract requirements, the example of comfort at workplaces is chosen to focus on to show if it is possible to use this kind of system in the specification process. It is not the intention to develop a fully working ES for abstract requirements or to define comfort at workplaces, but just to find out if the concept of ES's can have a role in the requirement specification process and how.

#### 4.2.3.2.1 Comfort at workplaces

Requirements such as comfort and durability are reasonable easy in contrast to other subjective, abstract requirements. With abstract requirements experts must really discover what kind of idea the client has for himself. In this process the users of the buildings are never asked about the specification of such requirements, while they will work there.

Comfort is chosen because it is an important aspect of working in utility buildings, very depending on users and activities and existing of components that are highly subjective and others that can be encountered via experts and resources. The components together define what comfort entails. In appendix 9.4 a full overview of the various components that can be distinguished by comfort are presented. Due to the exponential increase in information only comfort can produce as input for the knowledge base, just one component is taken based on what the experts mentioned that is most important at a workplace. This single component used in the tool is representative for the other components that are distinguished within comfort. In a broader perspective the use of such a tool can be performed regarding other abstract formulated requirements. Together a certain part of abstract requirements can be defined for clients which sharpens the Statement of Requirements (SOR).

Comfort itself is an abstract, undefined and not-computable requirement, in other words an abstract stated requirement which says something about a quality in a certain environment. The idea behind this is to use the gathered information about comfort as base for the knowledge base. In this case the knowledge management tool, which is used for the build-up of the ES, is based on the results of these interviews.

This way of organizing the research is chosen because of the limited time and resources during the research to get a fair amount of knowledge of experts, whereby also time must be spend in developing a knowledge management tool and belonging knowledge base is not possible. Since the objective is to work towards a PoC it is not necessary to develop a full ES with belonging database.

The gathered elements and specifications that are associated with comfort, act as knowledge base for the ES. In this way an overview of what several experts mention about an abstract requirement is the result out of the held interviews. In *Figure 22* a compact overview of the mentioned components are shown.

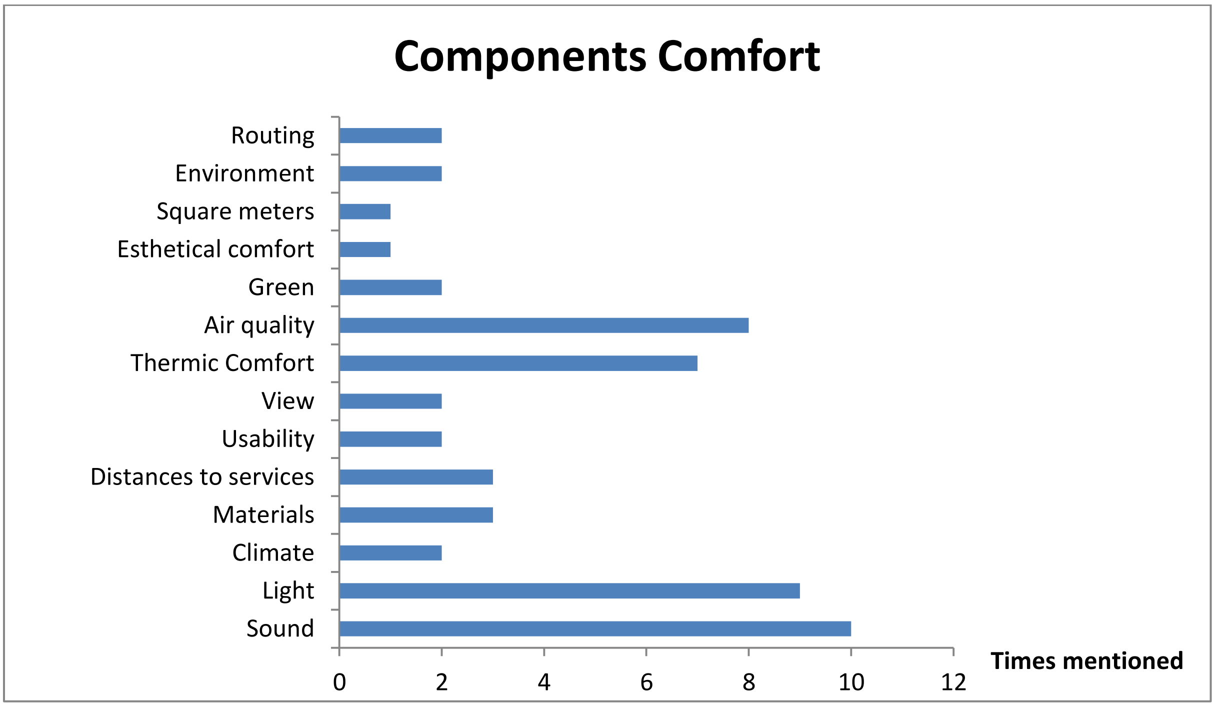



*Figure 22. The different components of comfort mentioned during the interviews.*

According to the method of SE and specific elements belonging to these components the interviewees were asked to break down the components and relate several subjects that were relevant to it. As can be seen in *Figure 22*, sound is argued as the most important component that experts take into account during specifying comfort at workplaces. This importance of sound is supported by the research of Kim & de Dear (Kim & de Dear, 2013) where sound privacy is issued as the most important parameter which results in dissatisfaction. Several questions during the client meetings will be based on sound and the relevant topics. One of these relevant topics is the sound between different adjacent rooms. Because this element contains several descriptions, depending on the applicability of a client's need, on which the eventually value will be based it is a good example during developing a Knowledge-based ES. The knowledge that experts possess to translate a client's need of comfort regarding sound, is asked about during the interviews. Based on what experts said about sound between the workplace and other rooms, and suggested sources are the input for the ES.

As can be concluded from the interviews, specification of abstract requirements takes place during contracted meetings between contractor and client. Often experts in a certain domain supporting this specification process by their expertise, experience and knowledge about certain requirements. The client has a lack of knowledge and decision making abilities in order to specify such requirements. The expert has a certain knowledge base on which the choices he makes and the support he gives to the client are based. Every expert is different and so their thinking pattern towards the specification will differ. By knowing what experts together have as a knowledge base in their head on which the possible values will be based that they are using for defining and making choices based on the client's needs and wishes, makes developing an ES interesting. Off course it is the client who is the leading factor for defining what he or she wants, but the expert is the one with knowledge about what is possible and how defining takes usually place. During the part of the interview where the component of comfort is discussed this process of subtracting their knowledge base is conducted. Of course on a much smaller and less professional scale, but that knowledge about the requirement of comfort is the base for the developed ES in chapter 5. Next to this, a lot of information about the abstract requirements and current process is obtained.

#### Remarks of data

Because the held interviews were set up in a semi-structured way, the answers have a lot of variety. This is not a problem in itself, but to make a good overview of the answers it is needed to group them based on meaning and matching definitions. For example some experts call acoustics as a concept belonging to comfort and others call this sound. Based on the definition, acoustics can be seen as a derivative of the concept sound. In this way the overview of elements is divided in elements and belonging breakdowns. All the given answers are extensive presented in different graphs and tables, which can be found in Appendix 9.3.

### 4.2.3.3 Expert Systems

The main question is often what a building really does for the use of its users. This must be evaluated, something the medical sector does, but the building sector does not. Due to this there is a lack of information in the building sector which harms the development and input of an ES. Also, an ES can collecting a lot of data about choices and decision-making processes regarding these requirements. This makes it important to discuss the use of ES's in requirement specification processes in the building industry.

#### Familiarity in the construction sector

Less people are familiar with these systems and possibilities, beside this not everyone sees the potential. Maybe due to the culture of the construction sector, it is hard to see the innovative character of the concept while it is already a research topic for more than twenty years (Yang, Li, & Skitmore, 1996). Despite this familiarity issue, several experts have their minds open for innovation and know what is needed in the future within these specification processes and see there a role for KBS developments.

#### Added value

A lot of different subjects are named where and how ES can add value. This is due to the relative new concept and idea about the use of ES in these processes. In a broader sense these factors should perhaps be seen more as gaps in which an ES could support.

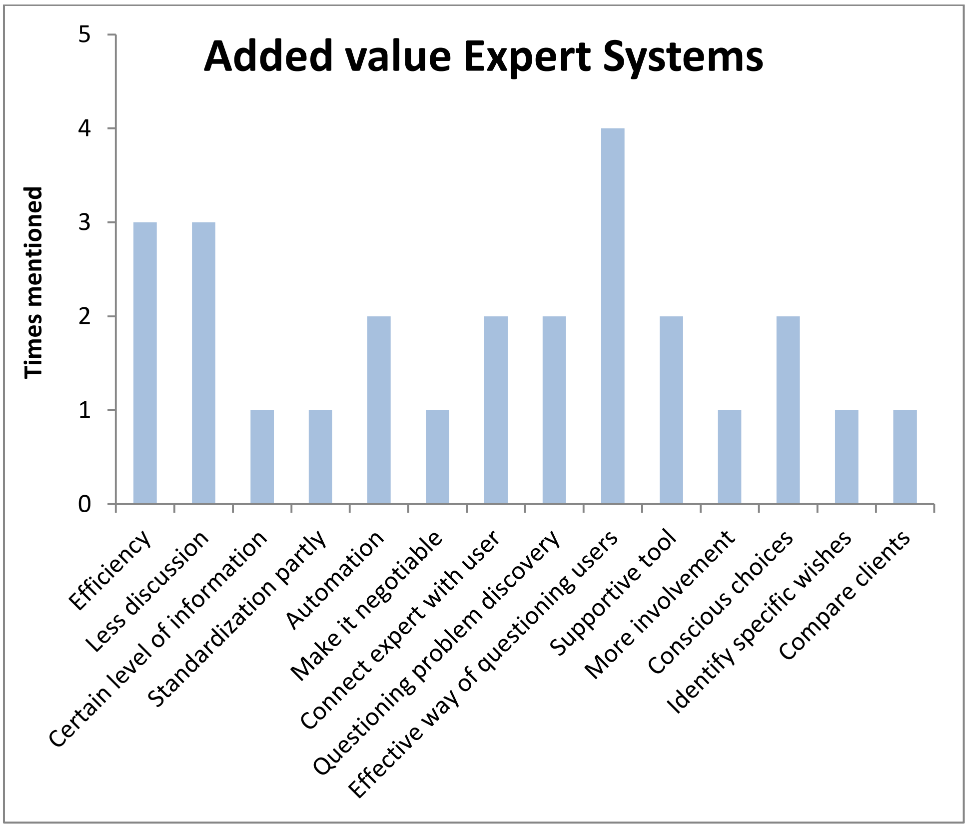



*Figure 23. Mentioned factors regarding the added value of ES within the construction industry.*

In general, ES's are concerned with decision making. During the specification process of requirements there have to be made constantly decisions in order to define what an exact need or wish means for the client. An ES can, assisted by the results of the interviews, improve these processes in means of efficiency. This efficiency can be accomplished due to automate and standardize parts of this process.

When having captured all the knowledge of experts there is always a certain level of information available in these systems. And when using these systems in the processes less time and less discussion is needed for defining abstract requirements. It can also be seen as a way to connect the client or user with the expert. And help in the discussion regarding what elements belong to the different components of an abstract requirement.

The earlier mentioned problem discovery strengthens the use of ES's for questioning about what a client means and wants. As the ES will be developed more and more it can contribute in an effective way of questioning users and have more involvement of the real users of an building, instead of only the client as owner. The client, and in particular the user, will make choices more conscious. And in the end the results of different clients, users and buildings can be compared and lead to new insights when this data is analyzed.

### Potential challenges

An innovative concept which does not yet have an application is susceptible to criticism and potential challenges. Next to discussing the added value an ES can have in requirement definition several components of the current approach are not yet involved. Current approaches work often with references and simulations to show what some choices may cause. And also the human interaction is missing. Next to this, the systems will be very complex, or even impossible to develop because they must base their results on too many parameters and variables. Another thing is the creativity in designs that will always be done by humans, no system can capture creativity. In order to this last challenge and seeing systems as replacement for the discussion meetings, it is argued to better see ES as supportive in these processes.

Other challenges are not captured in this development but have potential to be captured when further developing the ES. Such as the personal wishes every client has. Other challenges as consistency between different requirements and components and connections between choices can be conquered when developing the system. An expert or designer must always look if it is also realizable and possible. And since the specification of requirements are dependent of activities and users, the use of a requirement is limited. Which results in a limited deployment of the ES to a certain requirement, activity, user, room, or a combination of those. Another challenge, which is applicable for the whole generation of ES in the technical industry, is the fact that within the use of an ES the perception of the requirements is missing. The systems is lacking the capability to make it visual which makes it difficult to really experience a certain choice.

The data can be used as background for future requirement specifications. This is part of a broader challenge the building industry is facing. Almost no evaluation of choices during requirements specifications are held. The result of certain choices and the related values in practice are not evaluated or recorded. This can help future designs in developing more specific and customer-dependent specifications. One element an ES is not able to collect and can be relevant in this development is the reason why clients and users make certain choices with respect to certain components of abstract formulated requirements.

#### 4.2.3.4 Use of results

In this research the example of sound between a working place and other rooms is used to work towards the PoC. One of the ideas of the conducted interviews is to gather information and knowledge about how experts are dealing with abstract requirements. This 'dealing with' refers to the process of specifying this abstract requirement down to specific defined and computable values. In this way the computable values can be used within information flows of BIM models and also for verification processes.

The connection between the abstract terminology and specific values that are usable for verification processes are based on experts own interpretation, handling of and filling in concrete values. The objective of gathering this information is to capture the results that can be used for the ES. These are the possible outcomes of the system. The knowledge about how they come to these results is part of the input for the system. And on which the questions of the ES are based. In this way the process of experts handling this requirement is translated to the ES's and this process can be influenced by the response a client gives on the questions.

The use of an example for the ES development makes it easier to answer the research question about the possible use of ES's in the specification process. A lot of components that were mentioned during the interviews are refined in several guidelines with corresponding values and units. Because of the scope of the research about the investigation if and how ES's can be used in the specification process of abstract requirements during the design phase, it is interesting to have a look in the aforementioned twofold of the requirement of comfort. Concerning the components of comfort refined by guidelines the most important part is the description in combination with the question the ES will ask.

### ES use

Regarding the ES user it is important to make a distinction between an user and owner of an utility building. As earlier mentioned these two have a different conception and perspective regarding the specification of abstract requirements. In general an owner of utility buildings is more focused on certain quality classes and ensure that these buildings are rented or sold. An user of the same building is often a different person or group than the owner. Because of this it matters that a distinction is made between these two.

### User-based and Owner-based

It is important that during the translation process is worked towards specific requirements in the sense that these are above all computable for automated verification purposes. Several examples can be presented regarding the use of ES for owners and users of an utility building.

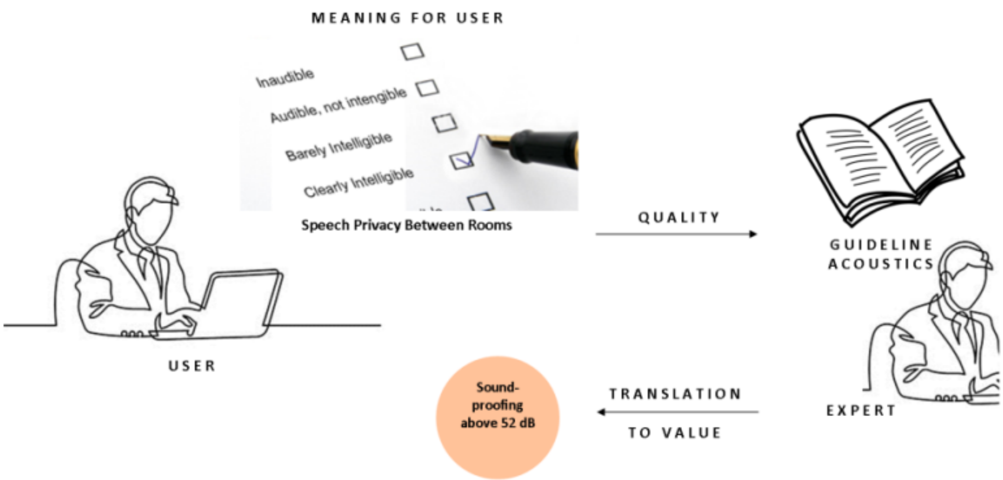



*Figure 24. Example of user-based working of ES, own drawing.*

For users it is important to have a description that is recognizable and clear and that fit within their workplace. An user does not know what certain concepts or values really mean. Therefore it is needed to have a connection between descriptions and values. Descriptions to track down what an user wants, values as result of the translation which make them ready for verification. In **Error! eference source not found.** the next example called as user-based is visualized; *an user wants to be clearly intelligible during his or her work in an open space. No idea what this really means for the sound insulation, but this is his or her need. An expert has sources, knowledge, expertise or experience that have the capability to translate this into the right value for sound insulation that satisfies the need of the user.*

For an owner of the building, who has other assumptions and objectives he wants to achieve, other questions must be answered to have a solid translation into computable values for verification. Often an owner wants to achieve a certain level for a certain quality of a building. For sustainability or durability this is often in the form of a certificate, but for comfort it is classified according to different classes. Class A represents the best level of comfort and results in a 'better' building than one with comfort class B. For an owner of that building who wants to sell or rent that building a better class of comfort will result in a higher price he can asked. The owner wants the best Return of Investment. In that case he will make choices based on this and not on that of an user. This situation is visualized in *Figure 25* and named as owner-based.

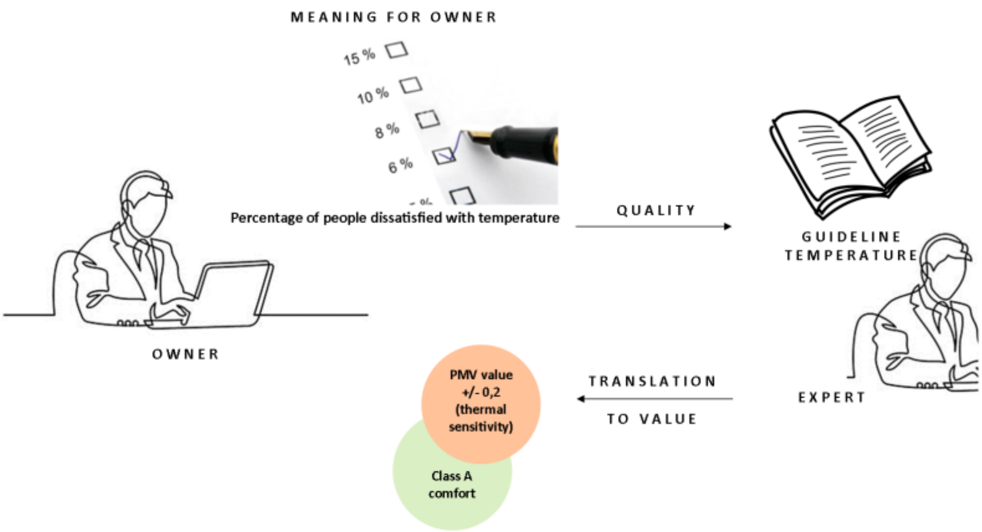



*Figure 25. Example of owner-based working of ES, own drawing.*

### Results regarding the subjective character

When the ES will be used for aspects with a much more subjective character the ES will get a more supportive role within the specification process. In that case the answers of the user regarding questions that are based on the knowledge base are collected and help by translating the abstract requirement. The translation of an abstract requirement is then based on the need of an user and the expertise of an expert towards a possible decision. This decision is of course the choice to agree with a certain value that stands for what an user has answered to the ES and in return the expert uses for the specification.

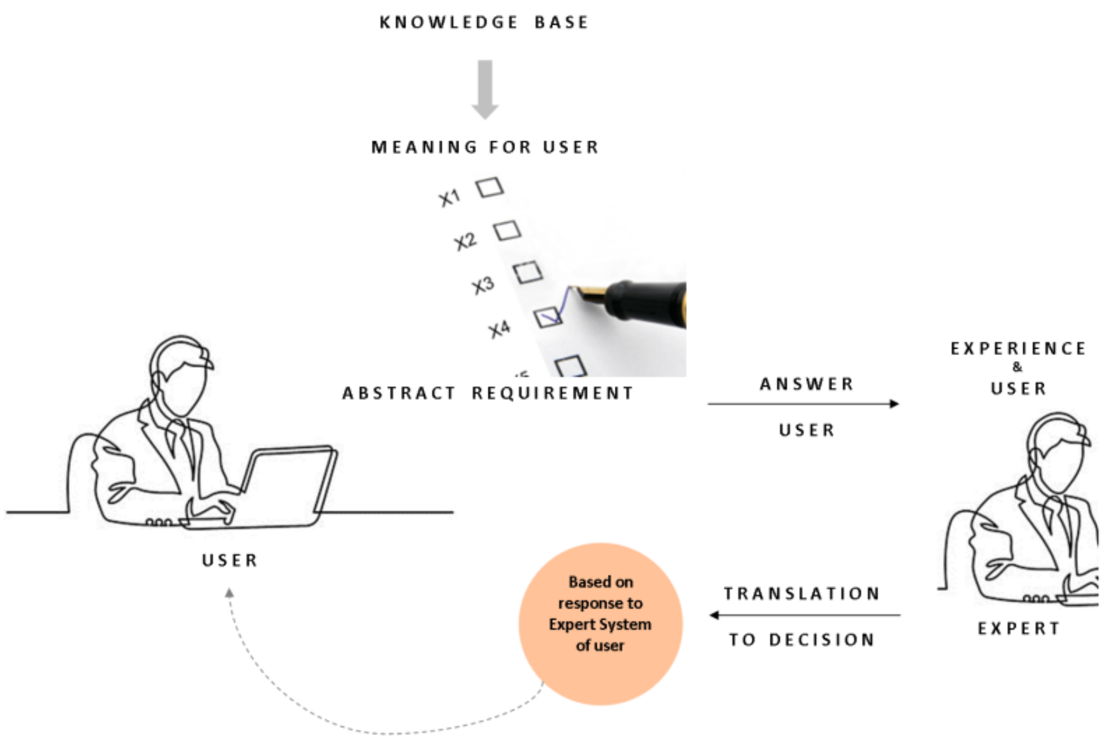



*Figure 26. Example of working of ES regarding the subjective character, own drawing.*

The developed ES is focusing on the example sound between the workplace and other rooms and are withdrawn from guidelines. The part of subjective elements of an abstract requirement is captured from experts as input without guidelines, where the ES can ask questions about that knowledge to the ES user. This will have added value in this process of requirements specification, but is much harder to collect than other elements related to abstract requirements. The difficulty in here lies in the fact this decision-making process for subjective aspects often happens in someone's head and differs between people. This cognitivism of someone, in this case the experts, is hard to capture.

### 4.2.4 Input Expert System

Normally the knowledge collected from experts is gathered in a knowledge base. In this research the knowledge base is not filled automatically with the experts knowledge. It is replaced with an example and corresponding knowledge out of what experts have said during interviews of what they use and other sources.

In the question about comfort, sound comes forward as the most important subject related to comfort at workplaces. Out of this outcome the subject of sound between rooms is chosen as input for the example model of an ES. The way experts normally translate this aspect of an abstract requirement determines on what information the ES will be based and developed. The knowledge base, and therefore the information as input to the ES, is the combination between descriptions related to a requirement and computable quantities.

The information used for this example at workplaces, is presented in *Table 3*. This *table* is part of the tree structure of comfort, which is based on several sources next to the interviews (NVBV, 2018) (SBR, n.d.).

This example as base to develop an ES is in addition very useful in order that it represents clearly in which way the information related to abstract requirements is used. The connection between a clearly stated sentence that the abstract requirement describes and computable values.

The information in *Table 3* is used in the next way: *for example, a sound insulation value of 52 dB is needed when the background noise is 30 dB and someone wants to have a good speech privacy. This speech privacy, as well as the voice volume and audibility are factors to which a client or user can answer. These answers together are needed for the translation towards a computable value for verification.*

| **Speech privacy tussen kamers** | | | | |
|---|---|---|---|---|
| **Stemvolume** | Hoorbaarheid gesprekken | | | |
| **- Luid** | Duidelijk verstaanbaar | Duidelijk verstaanbaar | Met moeite verstaanbaar | Hoorbaar, niet verstaanbaar |
| **- Verhoogd** | Duidelijk verstaanbaar | Met moeite verstaanbaar | Hoorbaar, niet verstaanbaar | Onhoorbaar |
| **- Normaal** | Met moeite verstaanbaar | Hoorbaar, niet verstaanbaar | Onhoorbaar | Onhoorbaar |
| **Speech Privacy** | Marginaal | Redelijk | Goed | Zeer goed |
| **Geluidsisolatie tussen kamers [dB]** | | | | |
| **52** | | a.g. 25 dB (A) | a.g. 30 dB (A) | a.g. 35 dB (A) |
| **47** | a.g. 25 dB (A) | a.g. 30 dB (A) | a.g. 35 dB (A) | a.g. 40 dB (A) |
| **42** | a.g. 30 dB (A) | a.g. 35 dB (A) | a.g. 40 dB (A) | a.g. 45 dB (A) |
| **37** | a.g. 35 dB (A) | a.g. 40 dB (A) | a.g. 45 dB (A) | |
| **32** | a.g. 40 dB (A) | a.g. 45 dB (A) | | |

*Table 3. Used input results for developing an expert system (NVBV, 2018). In Dutch.*

### 4.2.5 Conclusion

The conducted interviews have resulted in a comprehensive view in the current practice regarding requirement specification, and in particular for abstract requirements. This shows several snags in the current process, which partly can be filled in and optimized with the use of KBS's. This is also one of the main reasons for seeing how and at what point KBS's can be used. In that sense is chosen for qualitative research in order to have a better understanding of the improvement these systems can make.

Firstly there is a very important difference between user and owner, which is not tackled during this process. The reason behind this is the fact that the owner is the one who is paying and making decisions. But the user is the one that will work at a certain place and has certain needs and wishes, and is often not heard during specification processes. This tension can be reduced when ES's are used to show an overview about what these user's needs are and the corresponding results for the later realized building. In this way you will better meet the user's needs. This resulted in too much focus on solutions and earning money, instead of creating and realizing 'good' buildings that meet the user's needs. Much more important is what these user's want, what the problem is they want to solve and why?

Secondly, it remains very difficult in most cases, to translate abstract descriptions or concepts into specific, defined and computable values that can be used for automated verification via BIM. That will not differ, but the use of these KBS's have several advantages that can help with this process. One of these is the difficulty that lies in the clarity between expert and client. This clarity is reached through meetings where the requirements are discussed and where the expert tries to show what several choices cause. For clients it is often hard to explain themselves and try to formulate what they search for, not knowing what is possible. But it is important that they are heard and not an expert just fills in what he thinks is the best for that client, because everyone is different and has different needs. In here the knowledge of experts used for the translation is very valuable and can help these clients to choose and relate these client specific requirements with verifiable values for building elements. Collecting this knowledge and use if for improving this process is the most valuable what can be done with the knowledge of the experts. Experts have specific knowledge that can be used in more processes than that these experts involved themselves. But this knowledge can be of great importance during specification of these requirements.

Thirdly, the interviews have shown that the building industry does not yet embrace the application of ES's according to specification processes. And that while KBS's eminently are concerned with decision-making processes. In line with this, the PoC will extend the usability of these systems.

Regarding abstract requirements themselves, there has to be made a distinction in parts where someone needs to make a decision by the client specific requirements and parts that will always apply for a certain space. KBS's can be used more effectively for parts of an abstract requirement where someone needs to make a decision.

So, KBS's, for example ES's, can definitely help in getting a better overview and understanding of the user of a building. And thereby meet the most difficult topic within requirements definition, namely make the translation of abstract formulated client requirements even better and reduce discussion time and costs to fix in later stages (Wheatcraft, 2012)(Young, 2004) as well as improving and expanding the possibilities for automated verification. Which has all to do with the fact the contractor wants the requirements as early as possible defined and specified in as little time as possible. However, due to the nature of abstract requirements and human expert, that will be needed during requirement definition processes, an ES shall never fully take over the role of an human expert.

# 5.0 Model

## 5.1 Introduction

During design processes the experts are selecting several values for abstract requirements based on their knowledge, expertise and experiences. But also based on reference projects and discussions with other experts during planned sessions. This knowledge and information about the abstract requirement can be used as input for a so called Expert Systems (ES). These systems are applications that can help by taking decisions and support advice. The knowledge is present and the results can be derived via this ES. The application of the ES must enable the user to answer some simple questions and based on these leading them to the related result. The use of such a QAS works in two directions. On the one side knowledge is gathered and can be used during specification processes, and on the other side experts can benefit from the given input of the users of the QAS. Normally the experts with their knowledge are responsible for the establishment and management of such an ES. They are the best persons to determine which knowledge is needed and which criteria must be observed to come to the right results. In this research the establishment and management is based on the knowledge that is derived out of the interviews with several experts and sources in the field of abstract requirements. For the model and Proof of Concept (PoC) the example of sound as derivative of comfort is taken as base of the ES development. The development in here is of a very simplified view of how this in reality will work. This is due to the complexity of capturing mental processes and the exponential increase of information when using more than one component of an abstract requirement. Also there is chosen for this because the purpose is to show how an ES can contribute to the process and not showing how comfort must be defined.

The development, testing and evaluation of how the use of ES's can improving the design process regarding the specification of abstract requirements, can be seen as PoC and is explained after the developed model.

## 5.2 Motivation

The common denominator of ES's is the use in organizations where only a limited number of people have specific knowledge about something. The use of ES's makes it possible to extent the knowledge use within an organization and can give support by realizing several goals. These systems can be used for reducing costs, increase efficiency, improve services and to ensure that is complied with laws and regulations. Typical applications of ES's are *calculations, compliance checks, advising tools, risk analyses and document generators.* Advantages of the use of an ES are efficiency, uniformity, knowledge assurance, easy maintenance and management, quick use and easily build.

The practical aim of developing ES's is to support key human decisions by drawing conclusions from the expert's knowledge that is introduced to the system. In some cases there is argued that ES's aimed to replace the human thinking process by a machine's reasoning algorithm according to (Golik & Golik, 2000). Out of the literature the background information of ES's is described and analyzed. The term of ES's may also include self-learning capabilities. These self-learning capabilities are the ones that acquiring new knowledge, presenting it in their structure and applying to the task set. Regarding this machine learning it can be simple defined as *computer software that is capable of improving the quality of its functioning on the basis of its past experience.* In this research there is focused on ES's that don't have any self-learning capabilities and only possess a fixed knowledge base that does not develop in the course of their work. In this research there is focused on ES's that can improve human decision making based on the human experts fixed knowledge. Out of this dichotomy between self-learning systems and not self-learning systems it can generally be stated that an ES is the computer software that aims at supporting or replacing a human being in making difficult decisions on the basis of various prerequisites with the use of a particular reasoning algorithm. Even though it seems it is pretty simple to capture knowledge and develop an ES to make difficult decisions, it is very complex to capture human mental processes and develop such a system. Therefore just an easy and little example is chosen to only prove how ES's can contribute to the requirement definition process.

These mentioned prerequisites will be the result of the ES user who response to the questions the system will provide. This QAS, will be based on one of the earlier mentioned Knowledge Representation (KR) techniques. This can be used to work towards a list of outcomes as result of the given answers on the questions that are based on the experts knowledge which is captured in the related knowledge base. Knowledge representation is therefore focused on procedural knowledge because the system must focus on tasks, as certain decisions, that must be performed by the ES user to reach the particular result.

## 5.3 Importance

Research upon the possibilities of using ES's in the specification process of abstract requirements is of great importance as these abstract requirements are as important as the other requirements during design processes. These abstract requirements are ensuring the quality of a building and the possibilities of automation. It can optimize the RE process for abstract requirements and create a better focus on the user-value that is connected with these kind of requirements.

## 5.4 Previous work

In the past various efforts have been made to improve requirements validation and verification based on rule-checking. The development of IFC have led to early research for using this building model schema for automated rule-checking (Eastman, Lee, Jeong, & Lee, 2009). Regarding the classification of rules needed for translation the study of Solihin & Eastman (Solihin & Eastman, 2015) have shown different classes. This classification of various rules for application in building models are related to computational complexity and requirements imposed on the rule execution environment. In this research there is being worked towards checking rules that require a single or small number of explicit data. As will be seen in the developed ES later on, the explicit data needed for these rules must be the result of the ES. This first class is relatively straightforward to implement (Solihin & Eastman, 2015) and yet it has enormous potential to help improve building model quality and predictability, especially if there is a consistent and easy way for users to define the rules to check basic information.

Since translating written content into a computer-executable format for the purpose of rule checking is important, it is needed to focus on how this translation could work for abstract formulated content concerning requirements. For other requirements than abstract ones studies have tried this same process, such as for building legislation (Lee, Lee, Park, & Kim, 2016). But no studies conduct this translation process in combination with KBS's. Therefore this first class of rules is chosen because of the increasing complexity of the rule classes. This research is namely focusing on the translation possibility of KBS's and the combination with the process of rules to check properties using explicit relationships from each of the entities being processed.

## 5.5 Method

One of the results of research in the area of ES's has been the development of techniques which allow the modeling of information at higher levels of abstraction. These mentioned techniques are embodied in languages and tools that allows programs to be built that closely resemble human logic in their implementation and are therefore easier to develop and maintain. These programs, which emulate human expertise in well-defined problem domains, are called ES's (Gary Riley, 2008).

The development of an ES for abstract requirements can be done with many tools and languages for programming this. The used possibility in this research is C Language Integrated Production System (CLIPS) (Golik & Golik, 2000). CLIPS constitutes a complete environment for developing, testing and running ES's and includes both an inference engine and a rule editor. It is an extended tool set which is allowed for rule-based, object-oriented and procedural representation of knowledge. Besides this it has extensive documentation what makes it a good choice to use for setting up an ES without specific knowledge about ES. In this way it corresponds with the aforementioned objective and conditions for developing the ES for abstract requirements.

The knowledge base of an Expert System developed in CLIPS can only be entered by rules of an if-then scheme. Important is to investigate how usable these rules are according to use it for the translation process of the abstract formulated requirements. The inference engine, when it knows the rules in the knowledge base and knows that the condition is true, enters into the fact base a new fact that informs about the truthfulness of the conclusion. Thus, CLIPS applies forward chaining on the basis of presented facts where it constructs a new logical statement.

The choice of CLIPS for developing an ES is based on the usability as rule-based and/or object-based ES's. Rules-based programming is one of the most commonly used techniques for developing ES's and based on this it is often called as RES. In here rules are used to represent heuristics, or "rules of thumb", which specify a set of actions to be performed for a given situation (Gary Riley, 2008). In CLIPS, the if part of a rule specifies the facts or data which cause the rule to be applicable. The ES provides a mechanism which is called the inference engine, which automatically matches facts against patterns and determines which rules are applicable. The if part of a rule can actually be thought of as the whenever part of a rule. The then part of a rule is the set of actions to be executed when the rule is applicable. These actions of applicable rules are executed when the inference engine is instructed to begin execution. The inference engine selects a rule and the actions of the selected rule are executed. This process continues until no applicable rules remain. *If* a person is choosing an answer, certain rules will be applicable and *then* the corresponding set of actions is executed. This set of actions will deduce a list of possible results to the most appropriate outcome. To make the use of such an ES even more tangible it can be thought of as *if* a person selects a certain description that is applicable for him or her, one that is part of the parent abstract requirement. *Then* a certain result in the form of a value and unit will be the outcome due to the inference engine, which is processing the applicable rules. If this will be done for one element of an abstract requirement someone can also do this from his own memory and experience or knowledge. But one can imagine that when this will be a long list of elements that are part of an abstract requirement and all the combinations that can be influence the outcome, it is impossible to make a well-balanced decision. In addition, projects in the building sector are based on multiple abstract requirements. The underlying idea of such an ES is that it can capture immense databases of applicable facts that are based on this knowledge, experience and expertise to make well-balanced decisions. The gathering and collecting of this knowledge is out of scope for this research, and filled in by a qualitative research in the form of interviews. Based on this expanding knowledge in an ES all the underlying facts can be changed and adjusted, which causes a quick expansion of data.

It is of these characteristics that the use of CLIPS may be applicable for translating abstract requirements into specific values based on rules and facts. Rules as representation of experts knowledge and facts of chosen results as computable values.

The questions that are generated, based on the knowledge and answers that are needed to select and execute the applicable rules, must be answered by someone. In the current processes of translating and specify abstract requirements this is done during client sessions and discussions between client and contractor. Due to lack of time to do this with every involved (potential) end-user most of the time the client, who is investing, decides and his or her chosen applicable descriptions are based on the answers that are given. With the use of ES's this can be expanded to more involved (potential) end-users and a more applicable outcome, and at the same time an overview of what a client really wants. In this way there can be deployed a shift towards a better foundation of what end-users want. This can have influence in the way the questions must be asked based on the difference between owner-based and user-based (chapter 4.2.3.4).

## 5.6 Development

The input as discussed in chapter 4 is replacing a knowledge management tool (Ruiz-Mezcua, Garcia-Crespo, Lopez-Cuadrado, & Gonzalez-Carrasco, 2011) where experts are able to represent their knowledge. This representation of knowledge is often via a tool which supports interactive editing of knowledge bases, conversion and deduction expert spread sheets, tools for graphical representation of knowledge and visualization of reasoning paths, as well as a module for simulation of complex logical functions (Dohn, K. Guminski, A. Matusek, M. Zolenski, 2013).

This knowledge management tool is just one of the elements that are needed to develop an ES. The different elements create together the RES. These elements are divided in the next way:

1. Fact-list: memory for data
2. Knowledge Base: contains all the rules
3. Inference engine: controls execution of rules

### Knowledge Management Tool and Knowledge Base

Knowledge Management Tools are systems that are used for sharing information internally and externally by organizations. Examples of such tools are customer relationship systems (CRS), learning management systems, but also a Frequently Asked Questions (FAQ) system.

When the information of experts is extracted, the knowledge base can be deployed. As mentioned before the knowledge base of the used environment of CLIPS can only exist of an if – then scheme. On the basis of presented facts it constructs a new logical statement. These if – then rules as part of the Rule-based Expert System (RES) are helping in the process of finding a solution when the path leading to that solution is uncertain. This uncertain path is comparable to the way abstract requirements are translated into specific values due to their characteristics. It is the role of the advisor to create a clear view about this and translate the descriptions into applicable values for the project. Conventional Rule-based Expert Systems (RES), use human expert knowledge to solve real-world problems that normally would require human intelligence. This experts knowledge is often represented in the form of rules or data within the computer. The application of RES can be found in strategic goal-setting, planning, design, scheduling, fault monitoring, diagnosis and so on (Abraham, 2005). The most important advantages for the use of ES's can be found in the ability to capture and preserve irreplaceable human experience and developing a system that is more consistent than human experts. But also minimize the needed human expertise and developing faster solutions than human experts.

In the developed knowledge base all relevant information, data, rules, cases and relationships are stored that will be used by the ES. In here knowledge of multiple human domain experts can be combined and captured. The stated rules in here links given conditions to actions or outcomes. Knowledge is the theoretical of practical understanding of a subject or a domain. Those who possess the knowledge are called experts. Everyone can be considered as a domain expert, if he or she has deep knowledge. This knowledge is about both facts and rules. So the decision-making part and the belonging possible outcomes.

### Inference Engine

The inference engine sends the external user forms where the user inputs the information necessary for making decisions. Once this form is completed, the user sends the inference engine the values or statements so that it can continue with the decision-making process. This decision-making process which is captured in the knowledge base consist of three main parameters (Golik & Golik, 2000). These exist of a context, situations and decisions. Out of these parameters in the knowledge base the fact base can be built up. Into the knowledge base only rules of *if (premise) – then (conclusion)* scheme can be entered. In this way the truthfulness of the conclusion results from the truthfulness of the premise. The inference engine is the system that works between the knowledge base and the fact base. When the inference engine knows the rules in the knowledge base, based on the input of experts, and knows the rule is true than a new fact is entered that informs about the truthfulness of the conclusion. This process is also referred to as forward chaining. The aforementioned elements are part of the RES, of which in the next phrases every element is elaborated on.

Relating to a problem, an abstract requirement can be seen as a component that must be solved. This requirement depends on the context, the situation that occurred within this context and multiple decisions must be made to fill in this requirement. To fill in an abstract requirement it is often divided into sub requirements that eventually must be translated to a specific value that matches the clients wish and represents this wish in a value that can be verified. In the way this is true for a certain problem, but can be represent also the translation of abstract requirements. The decisions that must be made are subject to what the clients thinks and means regarding to an abstract stated requirement.

An ES based on rules consist thus of if-then rules, facts and an interpreter. The inference engine executes the forward chaining system based on the rules. This forward chaining is searching in the rules if a rule is true and can be fired. A requirement that is generated by rules already stored the context in it. The ES works in a way that is representing constrained elements and is linked to the attributes.

The used rules can have multiple antecedents joined by the keywords *and, or,* but also a combination of both.

```
IF    <antecedent 1>
AND   <antecedent 2>
          .
          .
          .
AND   <antecedent n>
THEN<consequent>
```

```
IF    <antecedent 1>
OR    <antecedent 2>
          .
          .
          .
OR    <antecedent n>
THEN<consequent>
```

*Figure 27. Structure of rules (Negnevitsky, 2002).*

### CLIPS Expert System development

CLIPS constructs a set of conflicts that includes rules whose conditions are met. The solution of the set of conflicts is conducted by selecting a given rule from the set of conflicts and then firing it. The firing of the rule consists in substituting the facts from the fact base to the rule and checking whether the premise of the rule (which depends on the fact or the facts) is true.

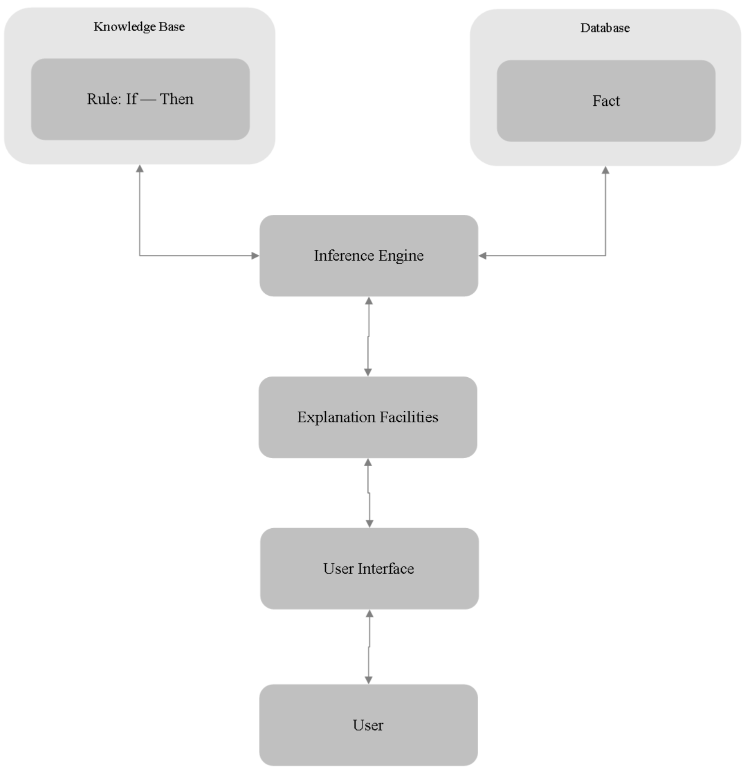



*Figure 28. Basic structure of a rule-based expert system. Based on (Negnevitsky, 2002)*

The user of a RES is limited in their answer options. The main problem in using a Rule-Based System (RBS) lies in the fact that the reality only can be represented with the use of two-value logic. Which involves that only certain knowledge can be represented.

A RES written in CLIPS is a data-driven program where the facts, and objects if desired, are the data that stimulate execution via the inference engine. Because abstract requirements are based on data the rules can only executed based on this data in the fact-list.

### Start and End CLIPS

For this research CLIPS version 6.3 is used. The following statements can be entered to start and end using CLIPS. After entering the appropriate run command for your system, the next prompt will be appear in the screen.

```
CLIPS>
```

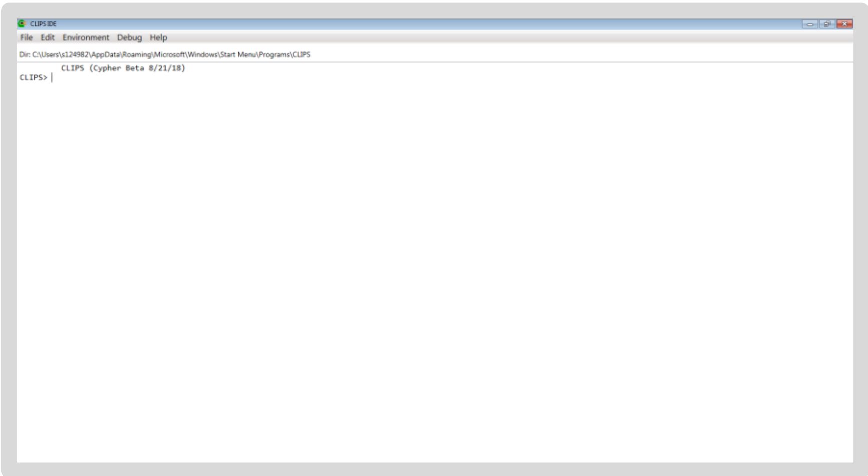



*Figure 29. The opening screen of CLIPS IDE (Integrated Development Environment).*

From this point on it is possible to entering commands into CLIPS. The different used commands that are used for the model in this research are elaborated on in the next paragraph. A more comprehensive overview of all the possible commands can be found in the extensive documentation (Giarratano, 2000). When entering the next statement and pressing the return key CLIPS can be leaved.

```
(exit)
```

### The built up of the a KBS

The different components that are needed to run an ES in CLIPS are explained in a simple example. When performing a certain activity at your workplace the light comfort depends among others on the luminance strength. A certain luminance strength can be assigned to a certain activity according to experts and guidelines. This will logically not apply for everybody as resulted from the literature review and interviews. However as example, here the choice of a certain activity will lead to a corresponding luminance strength. The first part of the script starts with the 'deffunction' . In general a deffunction is comprised of five elements (Riley, 2017). These elements are a name, an optional comment, a list of required parameters, an optional wildcard parameter and a sequence of actions. In the script below only a name, parameters and a sequence of actions are mentioned. This script ensures that when CLIPS is started and this script is the input, that CLIPS shows the questions, read the answers and bind these with the facts based on an *if – then* scheme. Of course this works only when there is data mentioned further on in the script in the form of rules, questions and facts.

```
(deffunction MAIN::ask-question (?question ?allowed-values)
   (printout t ?question)
   (bind ?answer (read))
   (if (lexemep ?answer) then (bind ?answer (lowcase ?answer)))
   (while (not (member$ ?answer ?allowed-values)) do
      (printout t ?question)
      (bind ?answer (read))
      (if (lexemep ?answer) then (bind ?answer (lowcase ?answer))))
   ?answer)
```

*Listing 1. Deffunction in CLIPS.*

The 'deftemplate' states the list of named fields, which are called slots. The keyword of deftemplate stand for define template. It defines a group of related fields in a pattern. In a deftemplate the attribute values are specified. The field values are allowed to be of any primitive type such as a string, integer, float and so on. The use of a deftemplate greatly simplifies accessing a specific field in a pattern because the desired field can be identified by its slot name. In the script below an attribute must have a name, value and a certainty regarding the defined deftemplate.

```
(deftemplate MAIN::attribute
   (slot name)
   (slot value)
   (slot certainty (default 100.0)))

(defrule MAIN::start
  (declare (salience 10000))
  =>
  (set-fact-duplication TRUE)
  (focus QUESTIONS CHOOSE-COMPONENTS REQUIREMENT PRINT-RESULTS))
;;(focus QUESTIONS CHOOSE-COMPONENTS REQUIREMENT PRINT-RESULTS-
again))

(defrule MAIN::combine-certainties ""
  (declare (salience 100)
           (auto-focus TRUE))
  ?rem1 <- (attribute (name ?rel) (value ?val) (certainty ?per1))
  ?rem2 <- (attribute (name ?rel) (value ?val) (certainty ?per2))
  (test (neq ?rem1 ?rem2))
  =>
  (retract ?rem1)
```

```
  (modify ?rem2 (certainty (/ (- (* 100 (+ ?per1 ?per2)) (* ?per1
?per2)) 100.00))))
```

*Listing 2. Deftemplate and Defrule in CLIPS.*

One of the primary methods of representing knowledge in CLIPS is via a rule. This rule, as described before, is a collection of conditions and the actions to be taken if the conditions are met. The execution is based on the existence or nonexistence of facts. CLIPS provides the mechanism, called as the inference engine, which attempts to match the rules to a current state of the system and applies the actions. The state of the system here is represented by the factlist. Rules are defined using the 'defrule' construct. Which is the keyword for define rule. Everything according to the developed ES will be captured in several defrules.

```
(defrule  "This is an example of a simple rule"
LHS
  =>
RHS)
```

*Figure 30. General format for Defrules.*

This general format is also used in the above shown script in CLIPS. The Left-Hand Side (LHS) is made up of a series of conditional elements (CE's) which typically consist of pattern conditional elements to be matched against pattern entities. The Right-Hand Side (RHS) contains a list of actions to be performed when the LHS of the rule is satisfied.

In this format the deftemplates and defrules are instantiated for the initial state, questions, rules module, choosing components based on rules and print the results. The total script of this example can be found in Appendix 9.5.

### Facts

With the 'deffacts' construct, a list of facts can be defined which are automatically asserted whenever the reset command is performed. As can be seen in the script phrase below the attribute values determine the construct of the facts. Out of the rules the facts will be looked up upon to control for which component (fact) the rules are true, or not.

```
CLIPS> (facts)
f-11    (attribute (name best-activity) (value any) (certainty
100.0))
```

```
  (component (name 300lux) (activity archiving copying reception))
  (component (name 500lux) (activity writing typing reading
conference))
  (component (name 750lux) (activity technical))
  (component (name 200lux) (activity general)))
```

*Listing 3. Deffacts scheme and filled in form in CLIPS.*

### Deffacts rules

The rules on which the facts will be mirrored are the rules which are based on the answers of the questions, which is presented as defining facts and abbreviated to 'Deffacts'. These rules and questions are also part of deffacts templates. This is the known data on which the outcomes will be based. But because not everything is known from the beginning for choosing the best possible fact the answers of an external user are needed.

The content of these rules and questions are based on what domain experts are saying about a certain subject in their expertise. Experts use sources and their own experience, knowledge and expertise to work towards the best possible result. This is specific knowledge an expert has and an external user does not have, that is why he or she cannot answer the related question immediately by a certain computable value but needs a description about that subject. The power of the ES is to translate the given answers based on this description into a specific value that is useful for verification later on in the design process.

| soort ruimte, taak of activiteit | verlichtingssterkte op werkplek [lux] | Ra [-] |
|---|---|---|
| archiveren, kopiëren, e.d. | 300 | 80 |
| schrijven, typen, lezen, gegevensverwerking e.d.<br>werken met een cad systeem<br>conferentie- en vergaderzaal | 500 | 80 |
| technisch tekenen | 750 | 80 |
| receptiebalie | 300 | 80 |
| Archieven (algemene verlichting) | 200 | 80 |

*Table 4. Performance artificial light according to (NVBV, 2018). In Dutch.*

In the *table* can be seen which illuminance is the best for which activity. By asking someone which activity he or she performs the needed computable value of the illuminance can be subtracted from this source. In this way an ES can do the same. Of course, this is a very simple example and an expert can do this themselves without any help. But when it comes to all the elements belonging to an abstract requirement or total requirement specification it will be an efficient way to gather all the decision making considerations and possible outcomes in the inference engine. When combining this with the prompted questions every possible end-user can answer and a good overview can be created of which values is applicable for whom in a certain situation. Because of the huge amount of information that is involved with formulating abstract requirements an expert cannot have all possible considerations and knowledge prepared and ready to use. An ES can offer a solution in here.

### Results

The information in the *table* can be formulated in facts and corresponding rules to define which one satisfies.

```
  (rule (if activity is archiving)
        (then best-activity is archiving with certainty 100))
  (rule (if activity is copying)
        (then best-activity is copying with certainty 100))
  (rule (if activity is writing)
        (then best-activity is writing with certainty 100))
  (rule (if activity is typing)
        (then best-activity is typing with certainty 100))
  (rule (if activity is reading)
        (then best-activity is reading with certainty 100))
  (rule (if activity is conference)
        (then best-activity is conference with certainty 100))
  (rule (if activity is technical)
        (then best-activity is technical with certainty 100))
  (rule (if activity is reception)
        (then best-activity is reception with certainty 100))
  (rule (if activity is general)
        (then best-activity is general with certainty 100))
)
```

*Listing 4. Deffacts rules in CLIPS.*

These rules are based on the first column of the *Table 4*. The possible outcomes, that are based on which answer is given on the question(s), are in the second column and stated as facts.

```
(component (name 300lux) (activity archiving copying reception))
  (component (name 500lux) (activity writing typing reading
conference))
  (component (name 750lux) (activity technical))
  (component (name 200lux) (activity general)))
```

*Listing 5. Deffacts component list in CLIPS.*

The total script of the ES can be saved as text document (.clp) and load into the CLIPS IDE. This has the advantage of adjusting different things in a text editor and load it again into CLIPS IDE without rewriting the whole script over and over again.

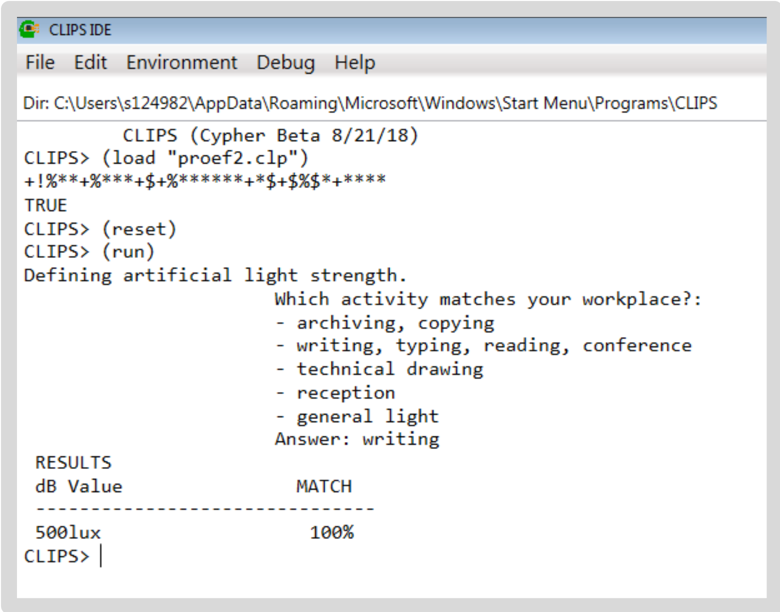



*Figure 31. Example of CLIPS expert system operation.*

In the *figure* can be seen the command to *load* the earlier explained script. The *reset* command to activate the defrules and the *run* command to start. When enter run, the ES starts asking and an external user can give in the answer that applies to him or her. After finishing all the questions the results are presented. As can be seen when enter *writing* as activity the corresponding value of the artificial light is *500 lux*. In this way an abstract requirement can be peeled off towards a specific and computable value, which otherwise was a possible outcome during a discussion with a human expert. In the next part a more extensive script for an ES will be presented and discussed.

### Data alignment

It is of importance to talk about the same things in requirements, the design and in the verification of a design (Moonen, 2016). This is essential to proving performance and to give insight in what the quality of the design is. When there is no alignment between the information it will be difficult to talk about the same thing. Since the variation in projects, clients but also experts influences the information structure of requirements alignment can be difficult. To let the total process work the interpretation of requirements must be done correctly. Another thing is the allocation to provide a process that works properly and providing a system that talks about the same things. The main reason in here is to prevent comparing different concepts with each other.

Regarding this last reason of allocation, it is important to have a connection of the same concepts between the requirement, ES and the verification. An ES will be focus on a particular requirement and this same concept must be subject of verification. Same use and descriptions of the concepts during the whole process is therefore needed. Important is the fact of correctly interpretation, what is related to the ambiguity of requirements. Even though ambiguity will always play a role in written content, alignment in concepts during the whole process can reduce this. And when knowledge of experts is combined in the knowledge base the variety between experts decreases since their separate knowledge is merged into the knowledge base and for every case this same knowledge is applicable. Therefore the questions about a certain requirement must be evaluated on the concept of ambiguity. This results in formulating these in a short, well-defined and without possible misinterpretations between the client and what an expert or a contractor wants to know.

### Expert System User(s)

There is a difference between the human expert, knowledge engineer and the user of the ES. The human expert is in this research represented by information about a certain subject that these experts themselves uses for specification. The knowledge engineer uses the knowledge to develop the ES. This ES is the representation of the knowledge based system according to human expert knowledge. In the end the user can rely on this ES by answering the rule-based engine inside the ES. In this research the possible use of an ES is researched upon. Therefore these separated roles are completed in a different way. The role of knowledge engineer is conducted by developing an example ES to show how it can work and the user will be represented by a pilot case.

### Usability of CLIPS

It has to be pointed out that the user is limited in his or her answer options. The system can only conclude about the truthfulness of particular facts or express its lack of knowledge about them. Thus, linguistic possibilities to represent the knowledge are poor and do not model the reality in a complete way, which may seem to the users of such ES's both useless and unfriendly in solving their problems. Despite these obvious drawbacks, the rule-based knowledge representation has its advantages – the intuitiveness and clarity of rules. Regarding all methods of knowledge representation that are used in machine learning, there is no other method as close to the methods of recording knowledge by humans as the rule-based method. A rule, consisting of the if- part and the then- part, provides a decision adequate to the situation where conditions are met; it can be noted as: *if* conditions, *then* decision. That makes the rule-based method so popular; it is considered as the clearest to humans and is applied in inductive learning related to decision trees.

## 5.7 Test case – model

From the example in the previous part an expansion of the model will be developed to make use of it in the pilot case. Based on the interviews with experts *acoustics as component of sound* resulted in one of the most important subjects according to comfort at workplaces. One of the major things during someone's work is their speech privacy in relation with the audibility (NVBV, 2018). The next context of this requirement formulation is applicable:

*With a loud voice it can be the requirement of someone that this should not be heard in an adjacent space. This results in the requirement of not audible. This is an users need, a choice he will make if questioning him. Consequences and what this specific will mean for the design and applied products is based on sources and the experts knowledge. Collecting this knowledge from experts gives more knowledge than an individual can possess and consult during this process. The result can be seen as condition that must be met by the design, as a specific translation of an abstract formulated requirement.*

This element of acoustics is really fitting this research because of its combination between computable values and description where people can find themselves. Most of the time no client can answer the question how much *sound insulation* is needed between rooms. But most of the client can give an answer to questions about their voice volume, audibility they prefer and which speech privacy they wish. Based on this the rule-based engine can translate these descriptions into computable values based on the answer that someone is giving. The full script can be found in Appendix 9.6. Based on the *Table 5* the knowledge base has been built up.

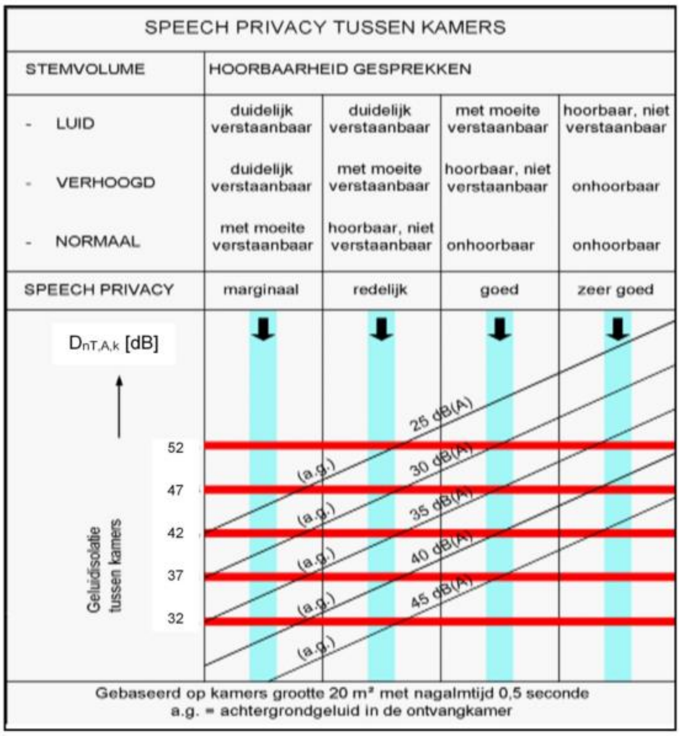



*Table 5. Speech privacy between rooms according to the official source (NVBV, 2018) in Dutch.*

The script has the same elements as the example script, only now with more questions, rules and possible outcomes. When even more subjects of an abstract formulated requirements are used in the ES these three components will be expand quickly. Also more subjects means more connections between these subjects. And as shown in the *table* above also within a subject of comfort there are several connections between different parts of this subject. All these different parts must be used within the questions asked to the user. Together these will result in one of the computable values needed for verification.

### Script

All the elements of an ES are used for developing the system and to show the translation. In the next steps this will be explained, first the development and thereafter the use of the ES. The script is based on other ES's scripts that are developed and published on (GitHub, 2019).

Together with more possible outcomes, more questions are needed to be answered to filter the results. That is also the reason why not all the components belonging to comfort are used to develop an ES, this is too extensive in this research in order to show the PoC. Where in the example just one question was stated to search for the best outcome, are for this script more different questions needed. Not in every situation all the questions need to be answered, this is depending on the given answers at previous questions. The answers on the questions must filter the clients wishes in such a way that in the end the best result, computable value, as outcome must remain. The next questions must be asked in order to sound insulation between a workplace and other rooms.

`Which activity matches your workplace?`

Depending on the activity someone performs another speech privacy will be preferred. The answer on this question will result in other applicable rules. In this example there are 4 different activities; *consultation, conversation by phone, computer work or a combination of all above.*

`At which level would you scale your own voice?`

This will always a bit hard to scale for yourself, but most of the time everyone can scale themselves into one of the scales. The scales are divided into *loud, raised or normal.*

`What category of speech privacy do you prefer?`

Depending or your activity chosen and your own wish a particular speech privacy is preferred. Do you want for example *marginally, reasonable, good or very good* speech privacy during your activity. This question is asked after every choice of activity, but the applicable rules depends on that answer.

`Which audibility do you prefer regarding your work?`

With audibility is meant how others, in spaces next to your workplace, can hear what you say. For some people everybody else may hear you, but for others it is preferred to only hear some noise or hear nothing at all. Regarding the *Table 5* there is a distinction between *clearly understandable, barely understandable, audible and not understandable, inaudible*. Logically when chosen inaudible the sound insulation for your workplace must be much higher than when clearly understandable is chosen.

`Which layout matches your workplace?`

Also the needed sound insulation depends on the layout of your workplace. Are you working in an open space or is it a completely closed and private place.

`Which background noise is applicable [in dB] ?`

This last question goes into the concept of background noise. Which background noise is applicable depends on what someone prefers, but also on the noise from machines. Depending on this and the combination with your voice volume, speech privacy and audibility the best match for sound insulation can be determined.

This developed ES matches the best possible outcome and shows the percentage of the match. The outcome with the highest percentage will be used as computable value for the specification of the requirement. This percentage is included to choose the best applicable result, otherwise it would be possible to have no answer that can be used for verification. This last step is discussed in chapter 6.

After the questions are answered the rule engine, the deffacts rules, determines which of the components, deffacts components, is the best match. It is possible that someone gives all the answers corresponding one of the possible outcomes, but vice versa is also possible. In both ways a best match must be determined.

In the full script (Appendix 9.6) all the rules are stated. Because of the long list here are the different structures of the rules presented.

```
;     47-35dB

  (rule (if background-noise is increased and
                workplace-consultation is good and
                workplace-volume is loud)
          (then best-audibility is barely with certainty 70))

  (rule (if background-noise is increased and
                workplace-consultation is good and
                workplace-volume is raised)
          (then best-audibility is audible with certainty 65))

  (rule (if background-noise is increased and
                workplace-consultation is good and
                workplace-volume is normal)
          (then best-audibility is inaudible with certainty 60))
```

*Listing 6. Structure of the used rules.*

As can be seen these rules are applicable on sound insulation between rooms of 47 dB when the background noise is 35 dB. In *Table 5* can be found that this is possible when the audibility is barely understandable, audible and not understandable or inaudible. In combination with all the different voice volumes, but only for a good speech privacy. When someone is answering with these different aspects than this component will score high. When someone is answering the question with other answers than this possible outcome will score definitely lower. Since the sound insulation can be determined of all the different factors out of the table in this research is chosen to base the outcomes on the audibility someone prefers and the applicable background noise.

```
  (component (name 47-35dB) (audibility barely audible inaudible)
(background-noise increased))
```

Besides the above mentioned rules there are also a lot of rules to determine the connections between the aspects that rule the sound insulation. When someone is choosing a particular answer for voice volume, than a particular description for audibility results in a better match for the speech privacy and thus sound insulation. Of course it is in this example only possible to have a result consisting of 32, 37, 42, 47 or 52 dB, but this does not hamper the way this system works. In the real world all the in between lying values are of course also possibilities.

```
  (rule (if workplace-layout is open and
                background-noise is increased and
                audibility is inaudible)
        (then best-audibility is clearly with certainty 40 and
           then best-audibility is barely with certainty 20))
```

This rule is an example of how different aspects are influencing each other and the eventually outcome. If someone is mentioning that his or her workplace has an open layout, with an increased background noise (scaled as 35 dB), and prefers an audibility of inaudible, then best possibility based on the table are clearly understandable and barely understandable.

The different possible outcomes in this developed ES are as follows;

```
(deffacts REQUIREMENT::the-component-list
  (component (name 52-35dB) (audibility inaudible audible)
(background-noise increased))
  (component (name 47-40dB) (audibility inaudible audible)
(background-noise high))
  (component (name 42-45dB) (audibility inaudible audible)
(background-noise dominant))
  (component (name 52-30dB) (audibility barely audible inaudible)
(background-noise slightly))
  (component (name 47-35dB) (audibility barely audible inaudible)
(background-noise increased))
  (component (name 42-40dB) (audibility barely audible inaudible)
(background-noise high))
  (component (name 37-45dB) (audibility barely audible inaudible)
(background-noise dominant))
  (component (name 52-25dB) (audibility clearly barely audible)
(background-noise low))
  (component (name 47-30dB) (audibility clearly barely audible)
(background-noise slightly))
  (component (name 42-35dB) (audibility clearly barely audible)
(background-noise increased))
  (component (name 37-40dB) (audibility clearly barely audible)
(background-noise high))
  (component (name 32-45dB) (audibility clearly barely audible)
(background-noise dominant))
  (component (name 47-25dB) (audibility clearly barely) (background-
noise low))
  (component (name 42-30dB) (audibility clearly barely) (background-
noise slightly))
  (component (name 37-35dB) (audibility clearly barely) (background-
noise increased))
  (component (name 32-40dB) (audibility clearly barely) (background-
noise high)))
```

*Listing 7. Component list in CLIPS.*

In this way for every possible answer and outcome rules can be determined. Which quickly grows into a long list of rules. This shows on the one hand how much information is needed, what an expert impossible directly can have ready to use, and on the other hand the reason why just a small example with limited information is used in this research. In the following screenshots the working of the ES as raw version is shown.

```
Defining sound insulation between rooms.
                       Which activity matches your workplace?:
                       - consultation
                       - conversation by phone
                       - computer work
                       - combination of all above
                       Answer: consultation
What category of speech privacy do you prefer for consultation?:
                       - marginally
                       - reasonable
                       - good
                       - very good
                       Answer: good
At which level would you scale your own voice?:
                       - loud voice
                       - raised voice
                       - normal voice
                       Answer: normal
Which audibility do you prefer regarding your work?:
                       - clearly understandable
                       - barely understandable
                       - audible and not understandable
                       - inaudible
                       Answer: audible
Which layout matches your workplace?:
                       - private workplace
                       - open workplace
                       - clustered workplace
                       Answer: private
Which background noise is applicable [in dB] ?:
                       - Low
                       - Slightly increased
                       - Increased
                       - High
                       - Dominant
                       Answer: slightly
```

*Figure 32. Questions and one possibility of answers of the developed expert system in CLIPS.*

These possible answers that someone can give result in the following results as example (*Figure 32*). Based on the rules the system determines the best match.

```
RESULTS
dB Value                 MATCH
-------------------------------
52-30dB                   99%
47-30dB                   94%
47-35dB                   87%
42-40dB                   78%
52-35dB                   75%
42-35dB                   75%
37-45dB                   65%
47-40dB                   64%
37-40dB                   64%
42-30dB                   55%
42-45dB                   51%
32-45dB                   51%
37-35dB                   50%
32-40dB                   40%
```

*Figure 33. Possible outcomes and percentages of matches in CLIPS.*

As can be seen the ES gives all the possible outcomes with a corresponding percentage. This shows the best applicable results based on the answers. The best match in here shows 99%. The resulted value will be used for the verification process by BIM. Within the building industry the requirements a building must meet are stated in structured formats. These formats not only contain a description of the requirements but also other information. By applying a specific command in the script the result of the best value will be displayed in such a format. By using `(dribble-on "…")` before running the script of the ES and after the results are determined `(dribble-off)` the value is saved into an Excel file. The next result shows the best value of *Figure 33* in the structured format.

| REQUIREMENT | | | | | | | VERIFICATION METHOD | | | VERIFICATION RESULT | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | *<Requirement>* | Parent req. | Underlying req. | Req. initiator | Source | Appendices? | Verification level | Type | Method | Satifies? | Value |
| *<1.0>* | *Defining sound insulation between rooms.* | *<PARENT>* | *<CHILD>* | *<USER>* | *<…>* | *<YES/NO>* | *<X>* | *<X>* | *<X>* | *<X>* | ***52-30dB 99%*** |

*Table 6. Formatted information about the stated requirement.*

In the example here one element of an abstract requirement is taken to develop a KBS in the form of an ES. Every answer has influence on the outcome, since the factors influencing this element are influencing each other. What applies here on a small scale is even more applicable on a bigger scale. All the different components of an abstract requirements are influencing each other. And so on, all the abstract requirements, but in fact every requirement in a design is influencing each other. This is one of the characteristics of the complexity of the building industry. Nevertheless, this is something to be reckoned with when developing an ES on broader scale in for example several abstract requirements that influencing each other. These connections between elements and requirements themselves must be captured within the rule engine. As example for this perspective the combination of sound insulation and ventilation can be given. Background noise influences the applied sound insulation for the speech privacy. This background noise is a combination of a lot of noise from installations, environment in- and outside, other people, etcetera. Ventilation is created by an air installation and when more ventilation in a certain room is needed, this installation will cause more background noise. When also a very low level of background noise is preferred, these two requirements are influencing and conflicting. A certain balance must be sought in order to fulfil both needs. Rules must have to take this effect into account too.

## 5.8 Results

The result out of the ES is a value that describes a measurable, defined and above all a computable value. This is needed for verifying the value with a design model in BIM. In the beginning several outcomes are possible, which are filtered by answering on questions that show which is the most applicable for that specific user. The connection between descriptions and computable values can be established by sources and experts knowledge. Doing this with multiple abstract requirements it becomes possible to take a step in-depth for this abstract requirements in the Statement of Requirements (SOR) and make these usable for automated verification either.

## 5.9 Conclusion and Discussion

Simple (components of) requirements will have less benefit by this translation method. But as you can imagine, thousands of requirements that are stated for designs with different disciplines and experts involved can have benefit by an efficient, more consistent and consequent requirement definition process. And certainly when it is based on the collected knowledge, expertise and experience all the experts have to offer. This stored knowledge is subject to improve the quality of designs. As already elaborated within the chapter of the conducted interviews (chapter 4.2.3.3), this idea still has many challenges and issues that influence the use of it that also this research cannot solve and take out. However, it shows the possibilities and how the implementation can be done for using it.

This development has proven the use of KBS regarding requirement specifications for abstract requirements. By using an QAS based on experts knowledge it is possible to create a more efficient and consistent system that at least can support the decision-making process between human experts and clients. This support is proven here in the form of the developed ES and a simple example of just a component of an abstract requirement. As mentioned earlier in this research there is a gap between the simplified view that is presented in here and the reality in dealing with abstract requirements. Nevertheless, it is believed that the use of KBS in requirement definition processes can improve the process and this can be seen as a first step in using it.

For example, the different influences relative to each other components of requirements have are not considered in here and make the development of an ES even more difficult. These contradictions that requirements can have are a main purpose that must been solved in this model before real implementation can take place. However, the result of this model shows that also this can be captured in ES's in the future. Together with the collection of human mental processes a tool can be developed that represents more the reality than the model represented in this research does. In addition to representing more the reality than just the concept the applicability can be increased when more abstract requirements are used in the tool and the aforementioned issues are also processed.

Human mental processes are always internal and too complex to be represented as an algorithm. Despite this, most experts are capable of expressing their knowledge in the form of rules for making decisions. These rules have the form of *if – then* expressions. In this way the use of ES is a well-considered choice for developing towards a real prototype. Regarding this development of a prototype an interface can be developed. This interface is left out for this research, since the relevance is not big enough for now.

# 6.0 Proof of Concept

## 6.1 Introduction

In this chapter the Proof of Concept (PoC) is presented to test the idea and assumption of this research (Singaram & Jain, 2018). The main purpose of developing a PoC is to demonstrate the functionality and to verify a certain concept or theory that can be achieved in development. It can be seen as a small exercise to verify a single or a set of concepts to be unified into other systems. Often the usability of it in the real world is not even taken into consideration when creating a PoC because integration with technologies is not only time-consuming, but also might weaken the ability to determine if the principle concept is viable. During the PoC the developer conducts research and begins to develop the feature with the goal of proving that it is feasible. Once this is proven the PoC is extended to develop an integrated working model to provide a snippet of the final product. After this, it can be presented to the client or product team to sell the idea for an upcoming project or to be used internally to share knowledge and stimulate innovation.

This is in contrast to the concept of prototyping where a first attempt is made at making a working model that might be real-world usable. Prototyping is a quick and effective way of bringing a client's idea to life and serves as a sample for the potential users to evaluate, test and share their feedback to make improvements.

The goal of this research regarding translating abstract requirements using a KBS as concept to see how and where this can improve this specification processes, the method of PoC is chosen to validate the working of it. In this chapter the PoC will complete this research by showing how automated verification can take place of a computable value as result of the Expert System (ES). It can be seen as a demonstration to prove the idea of translation abstract requirement and verify these against a BIM model. In accordance to the study of Solihin & Eastman (Solihin & Eastman, 2015), it shows an example diagram for typical application implementing the class 1 rules as discussed in the previous chapter. The implementation of verification in this research corresponds to that diagram.

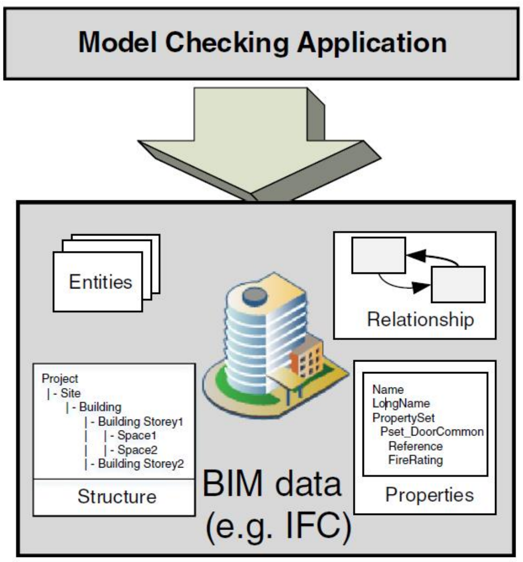



*Figure 34. Diagram for typical application implementation of class 1 rules (Solihin & Eastman, 2015).*

## 6.2 Concepts

In order to frame this PoC, the relevant concepts are discussed. In the *figure* below the relationships for this PoC between the concepts are presented.

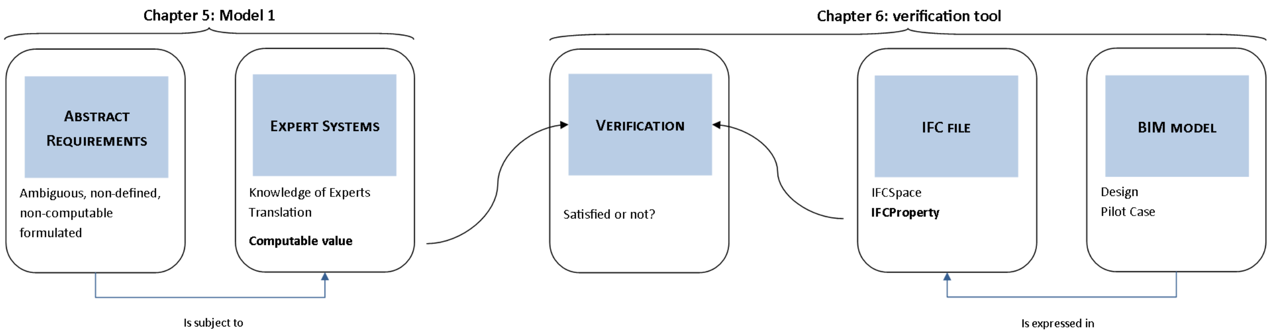



*Figure 35. Cohesion of relevant concepts in the Proof of Concept, own drawing.*

In the previous chapter the ES tool is presented from which a computable value can be resulted. Since the value of doing this lies within the possibility to verify these results with a BIM model this chapter presents how this verification may take place. The open standard of IFC is widely recognized as the interoperability solution for different software applications. The format establishes international standards to import and export building objects and their properties. The next paragraphs will discuss how to establish a working connection between the computable and the related IFCPROPERTY of the BIM model. This connection will be known as verification of this value and the corresponding requirement.

## 6.3 Validation

The contribution of using an ES in translating abstract requirements, regarding the result, is in threefold and indicates the importance regarding the developed idea in this research.

First of all, the Statement of Requirements (SOR) can take a step further into the degree of specification. In other words, requirements that are normally not yet defined can now be made more specific based on input of experts and clients in a more consistent and valuable way. It is depending on the collected knowledge and the kind of requirement how completely this can be done for the applicable requirements in a project.

Secondly, the client and user is now concerned with the specification process in a more effective way. Not discussing with several experts, but a system that can improve this process concerning the translation of abstract requirements. Helping the contractor and at the same time the client and user, or at least support the decision-making process. In addition to this the role of the user can be better involved in the process due to time efficiency. As mentioned during the interviews, the user is the one who seldom was involved in the decision-making process, because of money issues and no knowledge about the concepts. But the use of ES can change this, because no one needs to have that knowledge since the knowledge of experts is collected and serves as input to the system.

Next to this, the value is much more specific and now usable for verification via BIM. Because of the computability it is possible to connect the value resulted from the ES with the IFC file from BIM. This connection mentioned here relates to the subject of automated verification.

### 6.3.1 Experts input

One of the components of ES is the knowledge base. This contains the fact that describes the problem area and knowledge representation technique. It means that it contains a really high-quality and extraordinary knowledge in that particular domain. In this research it means knowledge about specific domains within the building industry. The whole success of an ES depends upon the knowledge, that is why it is considered as one of the important components of an ES.

### 6.3.2 Use cases

An use case can be used to represent the user's interaction with a system. In the figures below the relationship between the user and the different use cases in which the different users are involved is shown. Since there are two different systems developed in this research for translating an abstract requirement and verify the resulted value, an use case is shown for the ES (*Figure 36*) and for the verification (*Figure 37*).

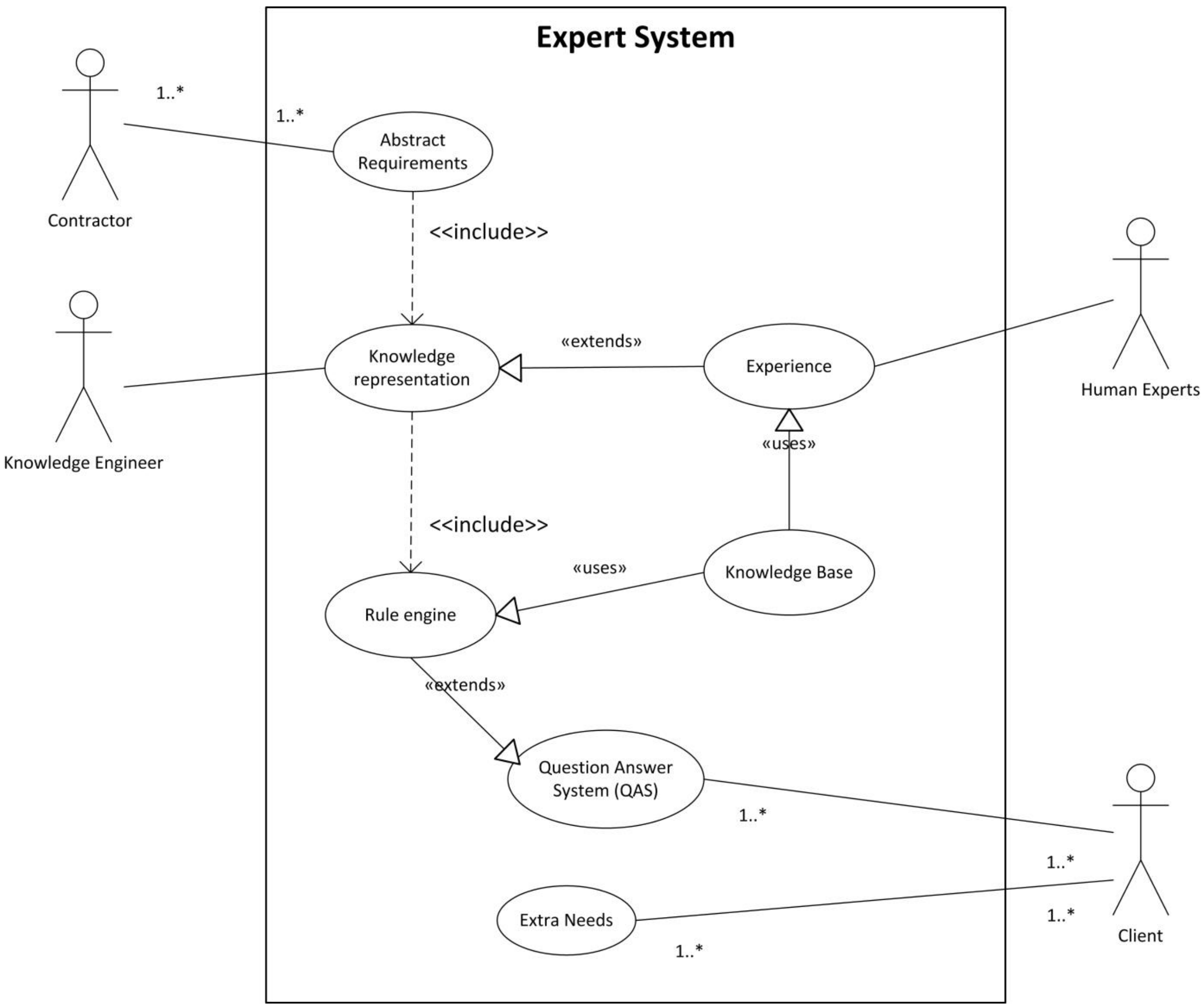



*Figure 36. Use case of developed Expert System.*

In the developed use case diagrams the scenarios are presented. It contains different steps by which the interaction between systems and users is visualized. In the use case of the ES there are several different users which have interaction with the ES. The contractor is the one who is starting with formulating requirements out of the ambition and needs that a client describes. As part of the requirements a substantial part are abstract formulated. Normally human experts will be involved in this process to define these requirements, but with the use of ES they 'share' their experience and knowledge with the knowledge engineer to collect this in the knowledge base. The knowledge engineer uses the knowledge representation to develop a rule engine. This rule engine is used for an ES in the form of a QAS. This QAS is used to let the client involve in the process of defining the abstract requirements. In this way the client can be part of the decision-making process without having to know specific knowledge about a certain domain. The computable value as results of the used QAS will be used as input for the automated verification. This verification is visualized in a next use case diagram.

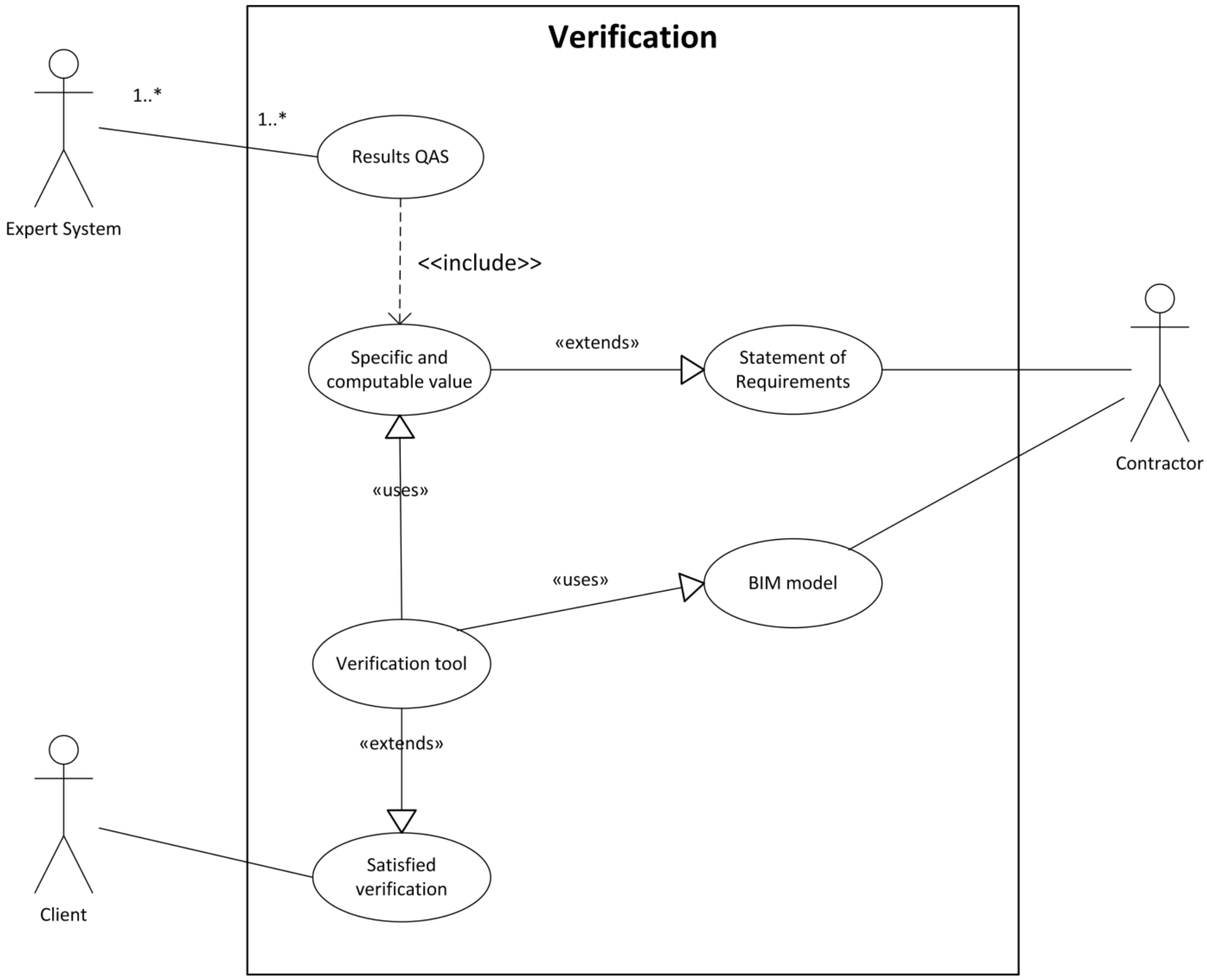



*Figure 37. Use case of verification tool.*

The ES is one of the actors in the system of verification. The results contain a specific and computable value, which means a value that can be checked in a BIM model. This value specifies the SOR and is used by the verification tool. Next to this the verification tool uses the BIM model to check. The contractor is the one who has responsibility for the SOR and the corresponding BIM model. Since these have to cooperate with each other it is important to talk about the same subject as in chapter

5 (5.6 'data alignment'). The result of the verification tool will be a check between the result of the ES and the BIM model to proof if the requirement is satisfied. This results can be shared with the client to see that his need, his answers to a certain abstract subject, are processed and satisfied in the model.

Regarding the use of this method, if a result or combination of answers is not realizable this will be of course subject to discussion between contractor and client. Which will start a change procedure. In no circumstance this system will act on his own or will be directive.

### 6.3.3 Comparison

Concerning the described improved process the steps between the normal procedure as indicated by the experts during the conducted interviews and the use of KBS's can be presented as shown in *Figure 38*.

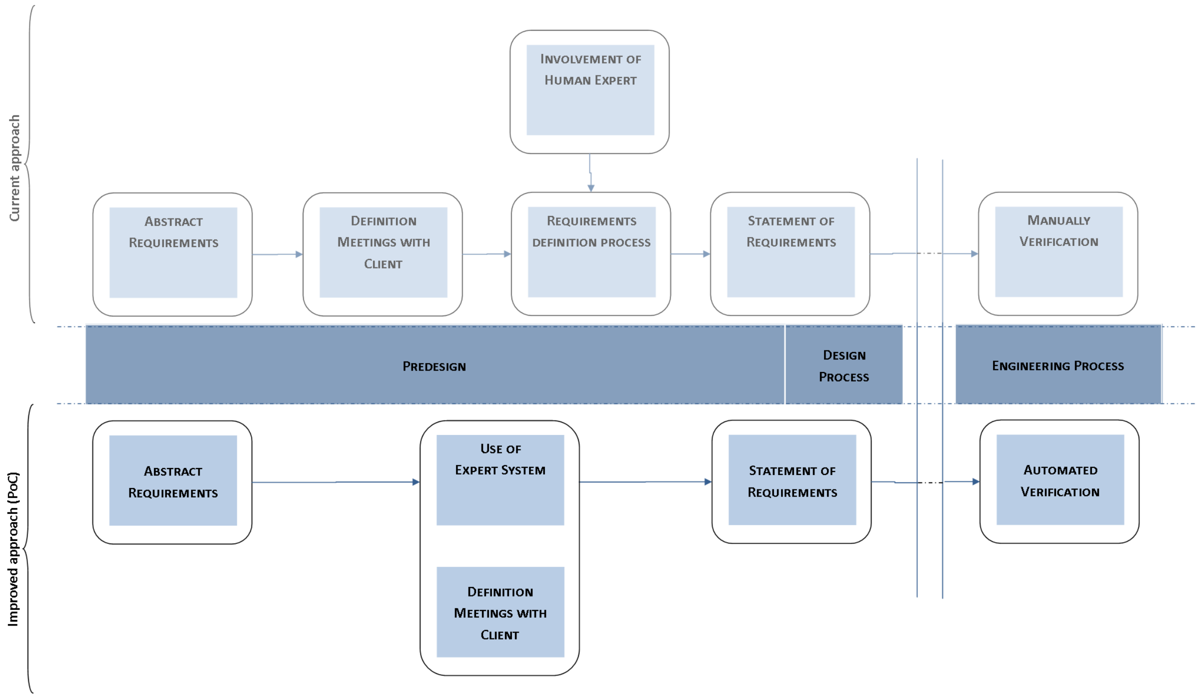



*Figure 38. Improved approach, own drawing.*

The meetings with the client are differently managed. Not based on discussion towards the requirements definition process, but the use of ES's. Also the verification of abstract requirements can now take place based on automation rather than on manually base in most of the cases, because there is worked towards computable values.

## 6.4 Automated verification

Despite the core of this research is about the translation process via a KBS, it is nevertheless interesting to look into the way this value can be verified via a BIM model in order to extend this research in a practical way.

This verification will be conducted with the use of a pilot case. This pilot case can be defined as a small scale study to evaluate feasibility and is shown prior to the performance of a full-scale project. It is therefore not the intention to develop a full verification tool for automated verification, but showing the possibility to automatically verify the value out of an ES. Regarding automated verification of specific values, several studies are conducted including developed verification tooling. Such as the study of (Lee, Lee, Park, & Kim, 2016) which focuses on translating legislation into a computer-executable format for evaluating building permit requirements. In here the authors describe a translation approach and define rules, which can be used for checking a BIM model and resulted in a report. In the study of Eastman (Eastman, Lee, Jeong, & Lee, 2009) is focused on a Solibri-based spatial checking regarding space planning and facility management. Using performance requirements to evaluate and check certain object properties if there is insufficient information. Another study (Martins, Rangel, & Abrantes, 2016) focuses on the impact of design options in early design phase stages and possibilities to develop automatically software as a parametric approach towards design, where formal rules and relationships between entities influence the development of complex building models.

None of the found studies examined client specific design requirements that are formulated in an abstract way. With this in mind this research suits the trend of automated rule-checking in combination with BIM. However, it distinguishes itself in the subject of translating abstract requirements in combination with using automated verification possibilities.

### 6.4.1 Pilot Case

The case is a design for an integral child center. The phase of this BIM model is the final design. Next to this BIM model also the functional as technical requirements of this building are obtained.

The reason a pilot case is used is the fact that these kind of case studies are often performed before implementing a large scale investigation. The basic function of these studies is to help identify questions and select types of measurement prior to the main investigation. For this research this type of case study will help to show how usable the use of ES's are in the specification process of abstract requirements, but also where more investigation is needed to scale up this PoC. One of the most important things to keep in mind about this type of study is that initial findings may seem convincing enough in here to be released as conclusions, but this is not always correct. Which means that not simply can be assumed that if it works, it will always work.

This BIM model consist of some workplace spaces which are the subject of the used abstract requirement. In the technical specifications of the case some measurable values are specified. These values are based on experts and other standards and guidelines. By using the tool for defining a more specific and computable value this technical specification can be extended and more precise formulated. However, this does not mean that this tool resulted in computable values that immediately can be used for verification in a BIM model. This research shows how it ideally can work, but in reality it will be needed to use certain calculation models in some cases before a computable value is stated that indeed can be used within the BIM model. This shows the importance of alignment between the result of the tool and the possibilities for verification via a BIM model and that not every BIM model contains the information and computable value the translation resulted in.

2.5.2. *Installatiegeluid*

Maximaal geluidsniveau in niet onderwijsruimten ten gevolge van installaties:

- Kantoren/spreekruimte: 35 dB(A).
- Aula/plein: 40 dB(A).
- Verkeersruimte: 40 dB(A). *Indien als onderwijsruimte in gebruik geldt die eis.*
- Zelfstandige werkruimte 35 dB(A)
- Sanitair: 45 dB(A).
- Technische ruimte: 70 dB(A).

| 2.5.3. *Eisen luchtgeluidsisolatie $D_{nT;A}$ [dB]* | *resp.* | *met -* | *zonder deur* |
|---|---|---|---|
| – Van groepsruimte naar verblijfsruimte: | 31 | 41 | |
| – Van groepsruimte naar verkeersruimte: | | 25 | 25 |
| – Van slaapruimte naar verblijfsruimte: | 31 | 41 | |
| – Van slaapruimte naar verkeersruimte: | 31 | 31 | |
| – Van speellokaal naar verblijfsruimte: | 31 | 51 | |
| – Van speellokaal naar verkeersruimte: | 31 | 31 | |
| – Van kantoor naar verblijfsruimte: | | 31 | 37 |
| – Van kantoor naar verkeersruimte: | | 25 | 25 |
| – Van spreek/directiekamer naar verblijfsruimte: | 31 | 41 | |
| – Van spreek/directiekamer naar verkeersruimte: | 31 | 31 | |

| 2.5.4. *Eisen contactgeluidsisolatie $L_{nT;A}$ [dB]* | | |
|---|---|---|
| – Van groepsruimte naar verblijfsruimte: | 59 | |
| – Van groepsruimte naar verkeersruimte: | | 69 |
| – Van slaapruimte naar verblijfsruimte: | 59 | |
| – Van slaapruimte naar verkeersruimte: | 59 | |
| – Van speellokaal naar verblijfsruimte: | 49 | |
| – Van speellokaal naar verkeersruimte: | 59 | |
| – Van kantoor naar verblijfsruimte: | | 59 |

*Figure 39. Technical specification of 'sound' in pilot case (Technisch Programma van Eisen - Onderwijshuisvesting De Basis Fluvius, 2017) in Dutch.*

In the floor plan of the first floor some workplaces can be detected for which the requirement of sound insulation is applicable. In the corresponding BIM model this workplace need to be a space, and the adjacent building elements must contain information about the specific aspect of the requirement. In a BIM model it is always possible to add information to certain building elements. To verify the computable value it is also needed that the information at which will be checked is available and accessible. BIM models contain more and more information but despite this not all information needed is present in this model. That is why in the BIM model of this case information about sound insulation is added as property.

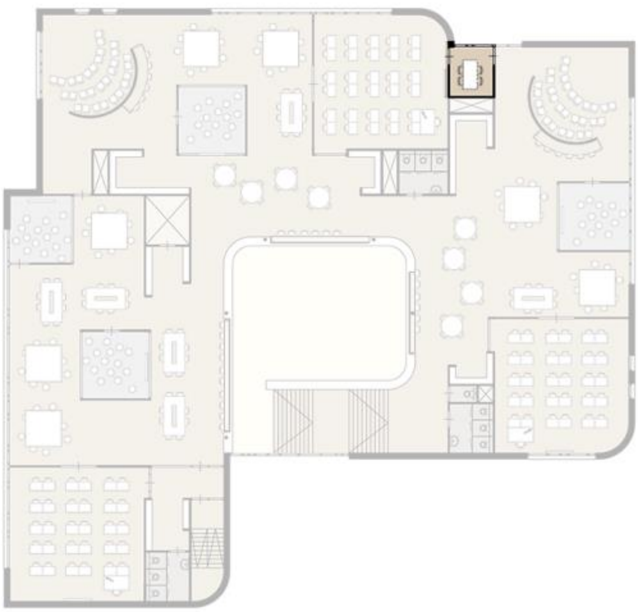
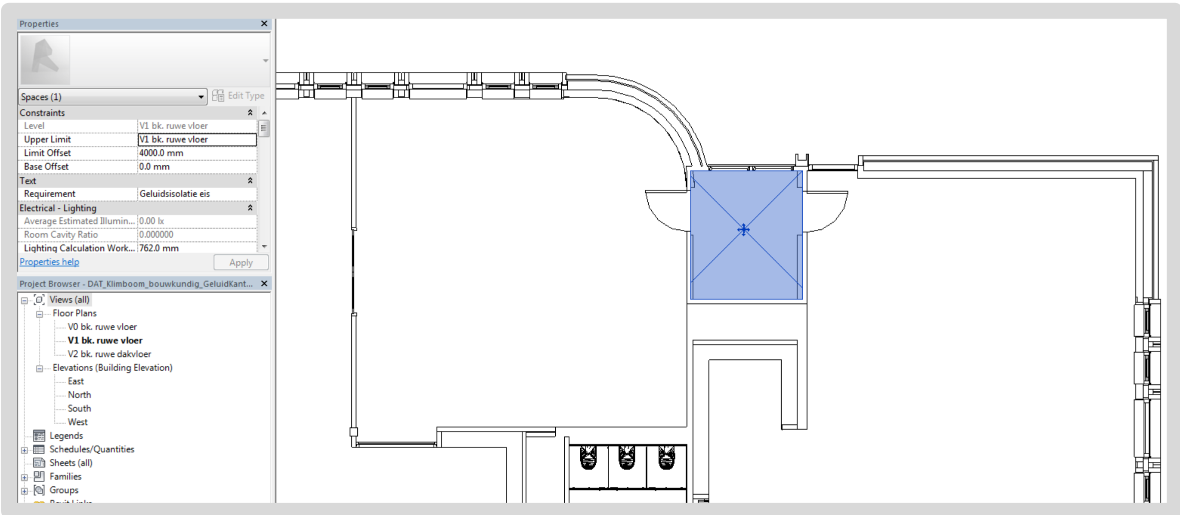


*Figure 40. Floor plan of the first floor of the pilot case regarding the workplace.*

### 6.4.1.1 IFC/BIM model

In the BIM model the workplace is defined as space. This is because the IFC file defines in the file a IFCSPACE to which several properties can be awarded. In this way it is reasonably easy to check this property during verification. Since the sound insulation value is not awarded to every single object in the BIM model this value cannot be found in the IFC file. This is the reason the BIM model and the exported information in the IFC file is adjusted by adding the sound insulation value to the adjacent objects of the workplace in the case.

To the created IFCSPACE the sound insulation requirement is allocated. This space must meet the sound insulation requirements. The value of this sound insulation in this particular space is determined by de different adjacent objects. The values for the floor, walls, windows, doors and ceiling create together the value of the sound insulation for the working place. To verify this value every object must have a property which specifies the sound insulation value of it. In the BIM model these values are allocated to these different objects by adding this as a new property. Hereafter these properties can be found in the exported IFC file under IFCPROPERTYSINGLEVALUE.




```
#143= IFCPRODUCTDEFINITIONSHAPE($,$,(#140));
#147= IFCSPACE('0666CFtfr1Xf1db2ZGslbu',#41,'1',$,$,#130,#143,'Space',.ELEMENT.,.INTERNAL.,$);
#152= IFCPROPERTYSINGLEVALUE('Reference',$,IFCIDENTIFIER('Space 1'),$);
```

*Figure 41. Created space in the BIM model and corresponding statement in IFC file.*

The added property and corresponding value of the different adjacent objects are now recorded in the BIM model and IFC file. These values will of course differ depending on the used products and for example the sound insulation requirement. Therefore, in this research the values for the sound insulation of the objects are not based on the used products in the case. It is just to show what the BIM model and IFC file must contain to verify the result of the translated requirement.

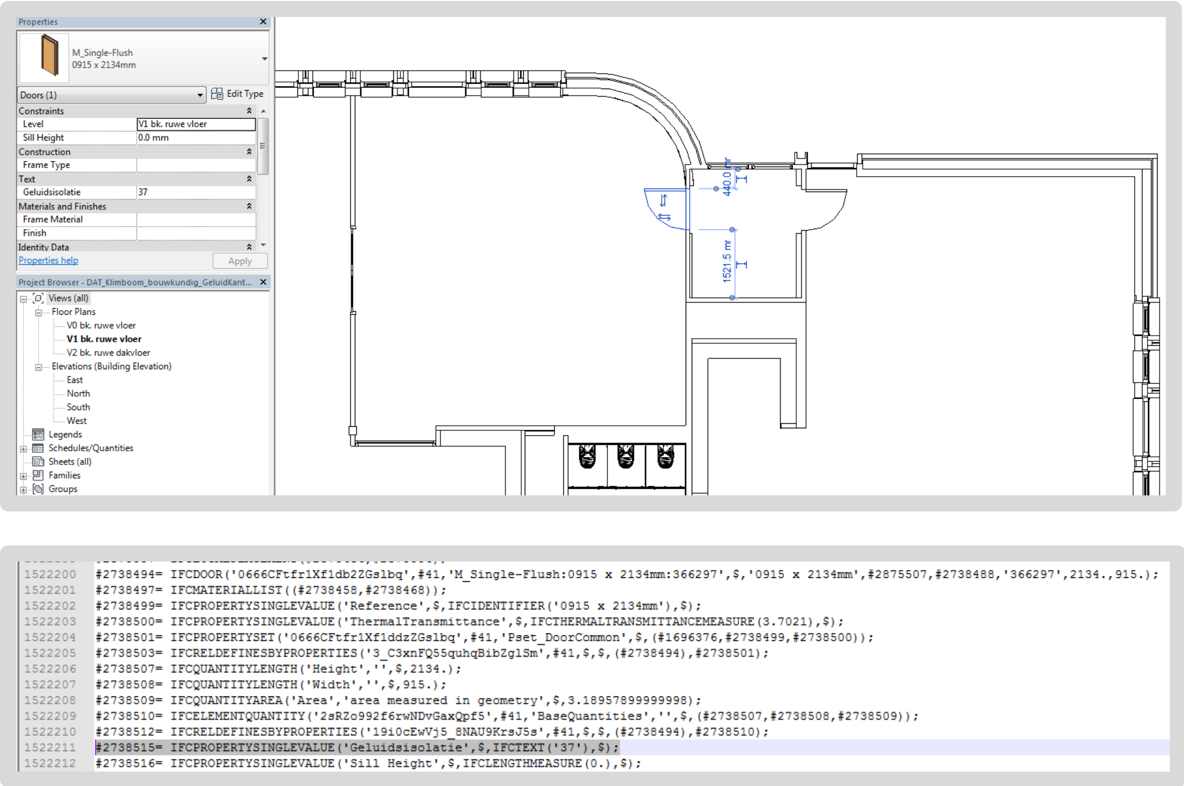



*Figure 42. Added property of sound insulation in the BIM model and corresponding statement in IFC file.*

For every object related to the space a different value is allocated. In *Figure 42* it can be seen that to the door a sound insulation value of 37 dB is given. This same property and corresponding value can be found in the IFC file and when the building is viewed in 3D (*Figure 43*).

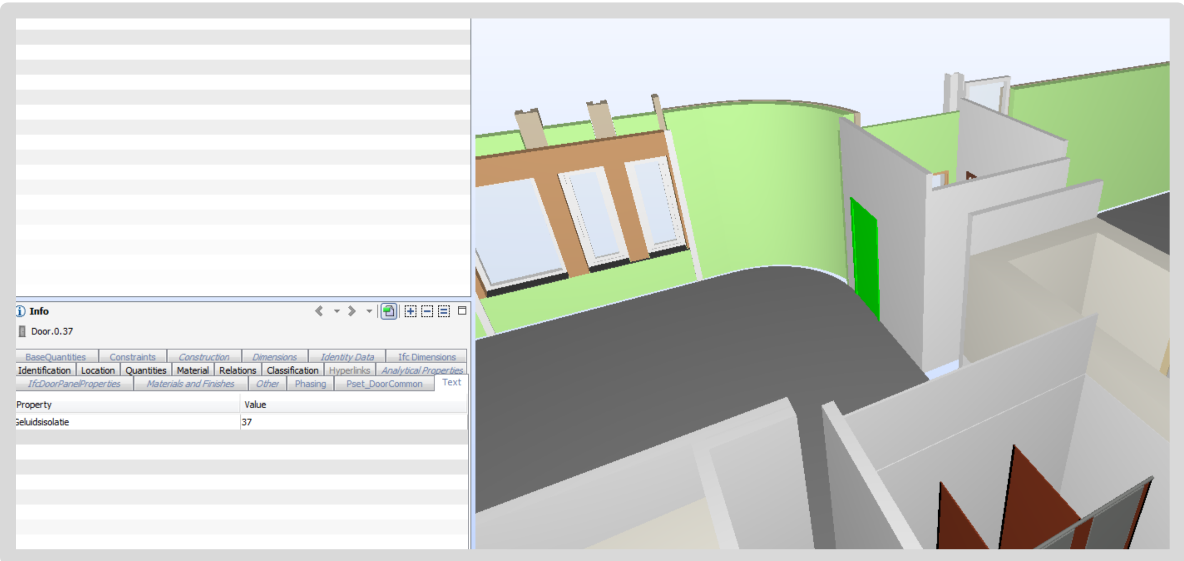



*Figure 43. 3D model viewer shows the added property of sound insulation to the door.*

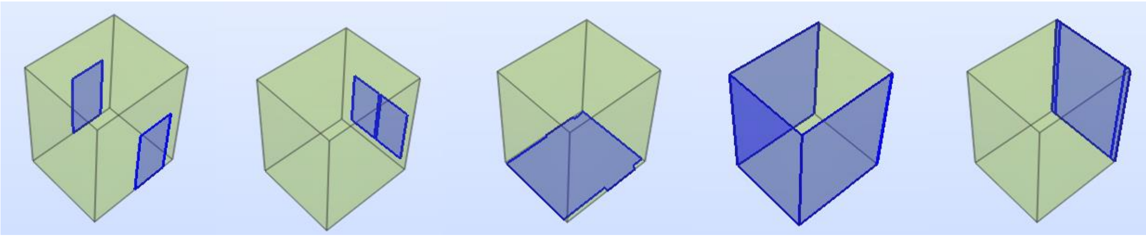


*Figure 44. The different Building Elements that surround the space of a workplace in the case, own drawing.*

The more knowledge, expertise and experience is added to the ES the better the specification and the more valuable the verification will be. Which makes it important to improve the information related to building elements in BIM.

#### 6.4.1.2 Verification query

The script below shows how to subtract the sound insulation requirement and belonging values of the objects from the spaces created in the BIM model of the pilot case. In here an example of 40 dB is set as value to verify.

```
def get_property(product, pset_name = '', p_name = '', p_value =
''):
for IfcRelDefines in product.IsDefinedBy:
if IfcRelDefines.is_a() == "IfcRelDefinesByProperties":
IfcPropertySetDefinition = IfcRelDefines.RelatingPropertyDefinition
Pset_name = IfcPropertySetDefinition.Name
if Pset_name == pset_name:
for IfcProperty in IfcPropertySetDefinition.HasProperties:
Property_name = IfcProperty.Name
Property_value = int(filter(str.isdigit,
str(IfcProperty.NominalValue)))
if Property_name == p_name:
#print "\t\t", Property_name
#print "\t\t", Property_value
return {'Object name':product.Name, 'Object guid': product.GlobalId,
'property name':Property_name, 'property value' :Property_value}

for space in model.by_type("IfcSpace"):
print ("space with global id: "+str(space.GlobalId))
print space, '\n'
for IfcRelSpaceBoundary in space.BoundedBy:
#print "\t", IfcRelSpaceBoundary
RelatedBuildingElement = IfcRelSpaceBoundary.RelatedBuildingElement
result = get_property(RelatedBuildingElement, 'Text',
'Geluidsisolatie')
for k,v in result.iteritems():
print '\t', k, '\t', v
if k == 'property value':
if v < 40:
print '\t\t\t>> geluidsisolatie is te laag'
print '\t\t\t>>', RelatedBuildingElement.Name
viewer.set_color(RelatedBuildingElement, 0.8,0,0)
if v > 45:
```

```
print '\t\t\t>> geluidsisolatie is te ruim'
print '\t\t\t>>', RelatedBuildingElement.Name
viewer.set_color(RelatedBuildingElement, 0.9,1,0)
if v == 40:
print '\t\t\t>> geluidsisolatie is goed'
print '\t\t\t>>', RelatedBuildingElement.Name
viewer.set_color(RelatedBuildingElement, 0.0,1,0)
print '\t-----------------------------------------------'
```

*Listing 8. Verification query used in TUEviewer.*

In the first part the script focusses on filtering the properties and print the property name and value of a particular object. Also the object GUID must be printed, the code for which the objects easily can be traced back in the IFC file. The second part describes that this only must be done for the defined spaces in the IFC file. Thus, only the objects adjacent to the defined spaces. These objects are defined by the *IfcRelSpaceBoundary*. Subsequent the property including corresponding value must be subtracted from the IFC file for the adjacent objects. When this is finished these values must be checked towards the value as result of the use of the ES, that are formulated in the script. If the values of the objects meet the requirement stated these building elements are colored green, when these not meet the requirement they are colored red. In this way it is visually obvious which parts of the building does not meet a particular requirement and which parts does. Values that meet the requirement can be subject to over-dimensioning because a sound insulation of 50 dB satisfies the requirement, but with a lower sound insulation this requirement would be also satisfied. These building elements are colored yellow. Reason can be that other requirements are causing this over-dimensioning and in that case this is just for your information.

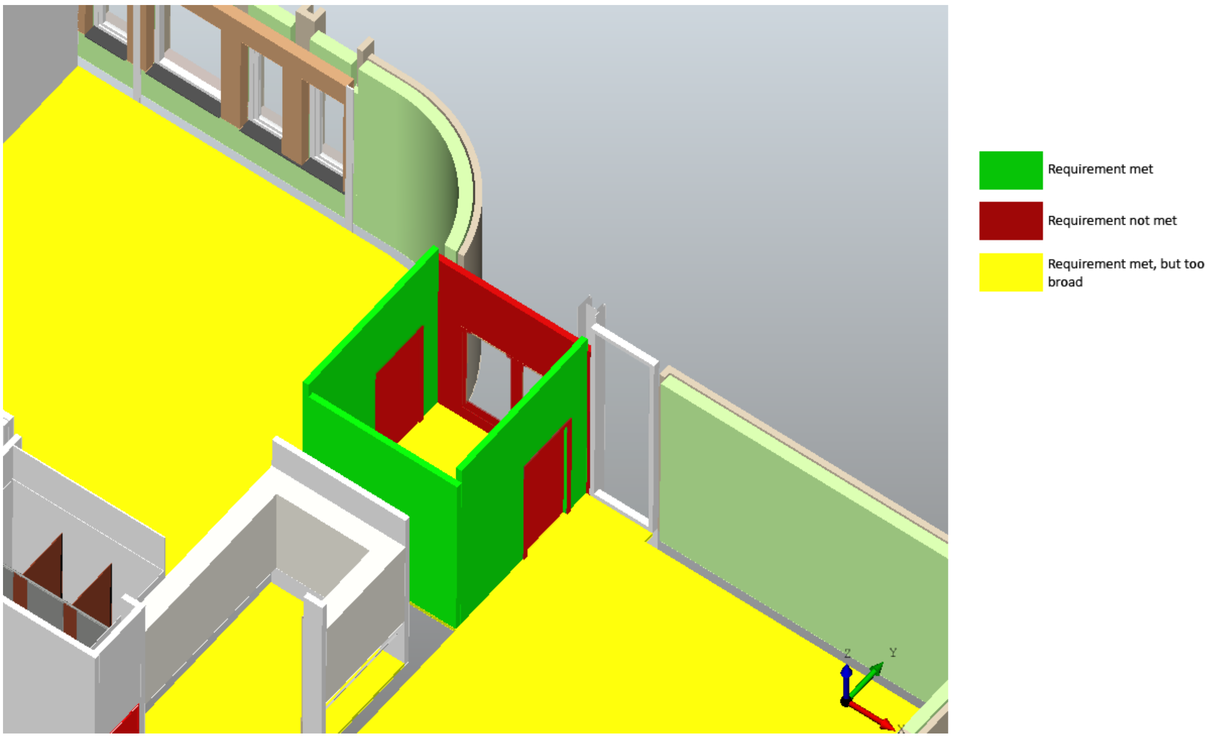



*Figure 45. Visual representation of verification between ES result and BIM in IfcOpenShell.*

```
-----------------------------------------------
Object name           Basic Wall:300_22_binnenwand_100:4598433
property name         Geluidsisolatie
property value        40
                      >> geluidsisolatie is goed
                      >> Basic Wall:300_22_binnenwand_100:4598433
Object guid           3YSSiVGqj6qv0WoAOq7oDa
-----------------------------------------------
Object name           Floor:Floor:300_43_cementdekvloer_70:343800
property name         Geluidsisolatie
property value        48
                      >> geluidsisolatie is te ruim
                      >> Floor:Floor:300_43_cementdekvloer_70:343800
Object guid           1E2WW5iFn57Pn2$LDWHGsa
-----------------------------------------------
```

*Figure 46. Textual representation of verification between ES result and BIM in IfcOpenShell, in Dutch.*

The *figures* above shows that the inner walls meet the requirement of sound insulation and are colored green. The floor does meet the requirement of sound insulation for that particular space, based on the result of the ES, but is overvalued and is colored yellow because of this. For the outer wall the color is turned red, and will be marked as "sound insulation too low". The full list of result from the verification can be found in Appendix 9.8.

## 6.5 Conclusion

The PoC is conducted using presenting verification of the value as an result of a KBS. The translation in combination with verification has proven the use of KBS's regarding the translation of abstract requirements and to connect this to automated verification possibilities. Just as the demonstration about how KBS's in the form of ES's by using questions and answers of clients are a simplified view, this also applies for the automated verification via a BIM model. A result of an ES that can immediately be checked in a BIM model sounds as the ideal world of working with abstract requirements. This research has presented this in order to show how the use of ES's for translation and working towards verification can be improved. Before real implementation and use can take place more research and development is needed, but this Proof of Concept (PoC) indeed shows that there is a certain contribution these concepts can establish. With this in mind the idea of using ES's in requirement definition processes to specify abstract requirements with the use of human expert knowledge is definitely supported. It works in the defined way, but is not optimal for use yet. Opportunities to improve this method are based on the concepts of viability, a good chance to be subject for improvement and usability, the ease with which an application can be used.

### 6.5.1 Viability

This research fills in the trend of automated rule-checking. Currently rule-checking tools can relatively easy check specified values, but not yet non-measurable and abstract stated elements. These elements always have certain characteristics to deal with before appeal any verification possibilities. In this research the characteristics of ambiguity, computability and measurability are caught and have resulted in the development of a KBS. In the first instance this shows a good viability to scale up the possibilities of translation via a KBS to more abstract requirements. Frequent mentioned abstract requirements in building projects form the best chance to scale up and create a tool which specify these kind of requirements depending on users, activities and space characteristics.

### 6.5.2 Usability

Of course this research shows just a small part of an abstract requirement. But it also shows how can be dealt with these kind of requirements that are not yet computable and ready for automated verification. Not every abstract requirement will be that good usable to break down into parts that can be used for ES's, but a lot does. Think about security, sustainability and other sub requirements belonging to comfort that are often stated in an abstract way. Whereby the automated verification of that value via BIM, extends the already conducted researches. Since abstract requirements are hard to measure because of their multi-interpretability and subjective character, this way of specifying can help the contractor and end-user to translate their wishes more efficient and consistent. Certainly in the current time where functional specification is more and more used, with as core to tender clearly formulated and described requirements to which you must comply.

# 7.0 Conclusion and Discussion

## 7.1 Conclusion

The goal of this research is investigating if it is possible to translate abstract requirements of buildings by using Knowledge-Based Systems in the specifically form of QAS, based on Expert Systems (ES). In order to develop a translation procedure towards automated verification using Open BIM for abstract requirements. The relevance lies in the emerging developments of information flows and BIM concerning abstract requirements on which is not researched upon that much. The different methods used in this research are based to work in a structured way towards the Proof of Concept (PoC). In chapter 3 the theoretical background have evaluated the topic of this research and related topics. Which showed the difficulties of dealing with abstract requirements in design processes. Together with the emerging knowledge based possibilities and increasing information what is involved during this process resulted in the research gap where this research gives substance to. Therefore a starting point in research using this ensures a good base of information and knowledge which can be drawn from. This research have resulted in a continuous progress by minimizing the scope in interacting with an increasing in-depth knowledge development (*Figure 47*). In order to combine this background with in-depth information about current practices and issues in specifying abstract requirements, the next step resulted in conducting interviews with field experts. As well to find missing information regarding the research gap as well to get a grounded base to select how to develop an ES that could be used and show future possibilities in requirements definition. The conducting of interviews has certainly added value in creating more grip on this topic, but seen in retrospect these interviews could be better used for subtracting real data about decision-making that could be used for the tool development. The disadvantage in here is that they disturb the phenomena investigated, since answering a question is disturbing and takes time, and only a few aspects of the phenomena can be investigated. Nevertheless questioning out of a semi-structured design has resulted in a lot of extra information about this topic in practice.

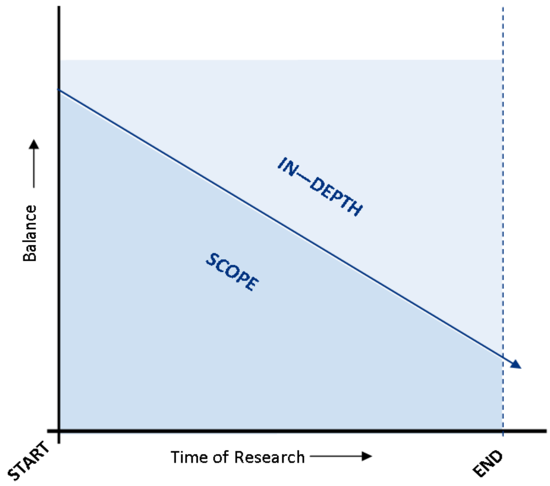



*Figure 47. Balance between scope and in-depth knowledge during the progress of the research, own drawing.*

This two used methods has evaluated the main assumptions that must be taken into account during the development and can be summarized as follows:

- Influence of specifying abstract requirements on the design process.
- Difficulties and characteristics of abstract requirements that form the principles on which the ES is based.
- The development and use of an ES in order to work towards automated verification.
- The added value of a KBS to improve the requirement specification processes.

The development of an ES tool and the connection with a pilot case to complete a PoC describes the continuation and results of this research. Since this is a new concept regarding the use during requirement translation the choice of realizing a PoC is a good starting point for any further research into this topic. Beside this, it was possible to move along the full process of abstract statements towards verification which has contributed to the choice of developing just an example tool and no prototype. This tool showed several contributions to improve the translation process and resulted in a computable and usable value. To conclude with showing the possibility of automated verification the described use of KBS becomes more practical and shows the begin of successful implementation. Despite a successful implementation the development of an ES in order to translate meaningless concepts into measurable statements is more difficult then assumed in the beginning of this research. The used methods and the corresponding results are definitely showing the possibilities this can have, but still a gap exists before any prototype can be developed and implemented. Nevertheless, this research has shown a procedure than can improve the specification process of a abstract requirements and where more research and development is needed. The directives of this research are recorded in the earlier formulated research questions.

***Which different requirements describe a building project in its design phase and which are not usable for direct automated verification yet?***

Concerning the different types of requirements not all characteristics are of importance for working towards automated verification. Until now most of the time requirements that are defined and formulated specific have a computable element and were used for automated verification. Abstract requirements differ from this and needed extra effort and time to formulate that specific and in a computable way. The characteristics and difficulties regarding this need are described in chapter 3 and 4. Where in the literature review an extensive overview is given of all the related requirements and the connection with the concept of abstract is evaluated.

***How are the abstract requirements influencing the dealing with requirements in the design phase?***

Next to the characteristics and difficulties that dealing with abstract requirements have, they cause critical factors in the design process. Due to the constant interaction between time, budget and quality several key factors are defined which are important in dealing with abstract requirements. These key factors are the reason the development of an ES can contribute in the improvement. The outcome can be found in the interview results and the part of RE within the literature review.

***What are the characteristics and difficulties of abstract requirements concerning the specification process of these requirements?***

Abstract requirements, as the word says, are requirements in which someone tries to capture his or her need regarding an object. But it has resulted in a vague term that is susceptible to multi-interpretability, ambiguity and non-computability regarding verification. A non-exhaustive list of characteristics is filtered by literature which made it impossible to take all these into account during this research and no complete answer can be given according to all characteristics and difficulties. A grounded selection has been made, based on the literature and interview results, to filter ambiguity, measurability, computability and verifiability as leading characteristics in this research and not all options are elaborated on. These characteristics are, according to this research, the major factors that describe abstract requirements concerning the (automated) verification process. On this basis the developed ES tries to reduce the difficulties and come up with an improved method for capturing knowledge and translate abstract requirements.

***What is the current process of dealing with abstract requirements during the design processes?***

The conducted interviews as well as the literature specifies three main parts of the current process in specifying requirements.

1. Session with client; most of the time several meetings between experts and clients are contracted for specifying the requirements.
2. Reference projects are used; for seeking what the client really means and wants. This option can be best placed according to the third way where experts also cannot use their knowledge and expertise in connecting a description with a certain value.
3. User and activity dependent; it was also indicated that the requirements were specified according to the characteristics of an user and the corresponding activity this user conducts in the building. The involvement and importance of the user must be valued a lot more in order to translate, which is also important for the use of an ES.

This selection justify the most mentioned approaches, but it also shows that no leading procedure or method regarding translation abstract requirements is not available yet. This research is an attempt to work towards such a procedure in order to improve.

***How can Knowledge-Based Systems, in the form of Expert Systems, be a support to the requirement definition process?***

The steps that are needed for specification is the translation. This translation is researched and the use of ES is used to investigated to find out the possibilities. In order to the use of ES there is needed knowledge from experts or sources as input, the knowledge base. Also questions that can be answered by users and that are based on the knowledge is needed. As a result an model is developed to show how this KBS can be used in this translation process. In addition to this development it can be concluded that a KBS can contribute to this process, but a well-defined balance and commitment of such systems has to be determined. Only then the full improvement and reducing of time can be used in the right place of the process.

***What is needed to translate abstract requirements into specific defined and computable requirements?***

This developed KBS needs to have several critical components to get a right computable value. The knowledge base is one of the most important things which is filled with the collected knowledge of experts and form the base on which the translation can be created. The connection with descriptive text every kind of user can understand will form the questions on which the results are filtered.

Unless when this is all present, not every abstract requirement can be translated using these components. As stated, this procedure will be best for requirements that are subject to which experts attach a value – a connection between a description and possible computable values.

***How can automated verification be improved using the translation of abstract requirements and with that optimizing the design process?***

The verification of computable values within a BIM model is not that hard to accomplish. In the form of automated rule-checking, it can improve the design process by reducing time, costs and errors. This improvement can resulted in less errors, due to the fact that the requirement process is a critical and problematic process where the problems in buildings often can be traced back to. A good investment in the requirement process is needed as base. Often a too less percentage of the total project costs is invested in this process, which makes it important to not decrease the investment in this process. More investment will result in a more precise, correct and complete system specification and reducing costs of the total project.

***Is it possible to translate non-functional requirements, that are abstract, undefined and non-computable described, into specific, objectively measured and usable requirements for automated verification in building design processes?***

The main possibility lies within the use of knowledge-bases systems, which generate the capturing of a lot of knowledge that can be used in specifying abstract requirements. Not every requirement that cannot be specified is immediately usable in these systems. But a significant part of abstract requirements can be translated in this way and improving the design process. It is only possible when having an amount of knowledge as input and the translation process based on rules. Together it is possible to translate an abstract requirement into a computable value and improving this process, but it is still not a complete procedure of translating and verification yet.

The added value of this research and the results can be formulated in the next way. By using such a systems and procedure as described and developed in here for translation abstract requirements, you are able to collect a big amount of knowledge from experts and use it for the translation. With this more consistent outcomes of the requirements can be established and more value is represented in here due to a more efficient process and broadening the response to the real users. In contrast to have just a couple of meetings that must result in a good understanding and on which decisions are based, the most important advantages for the use of expert systems can be found in the ability to capture and preserve irreplaceable human experiences and developing a system that is more consistent than human experts. But also minimize the needed human expertise and developing a faster solution than human experts. An ES can perhaps not replace the human expert , but a human expert can never have more knowledge, expertise and experience than that of himself. In this way the client can be part of the decision-making process without having to know specific knowledge about a certain domain.

## 7.2 Discussion

### 7.2.1 Relevance

This research towards abstract requirements and the possibility of translating and verifying these has revealed a possible improvement of the design process. Where in the current practice only measurable requirements were subject to automated verification, has this research shown the possibility of extend this with also abstract requirements. The possible developments can be summarized as follows:

- Difficulties of abstract requirements and possibilities how to deal with these
- Improving the current approaches of specifying abstract requirements
- The use of Expert Systems in requirement specification processes
- Proof of Concept of the use of ES in the requirement specification process
- Connection to already developed automated verification via BIM

As discussed within the literature are these the main developments to overcome the research gap and is an addition to the already conducted research of requirement definition processes and automated verification. This revealed improvements this research has focused on shows a simplified view about how to cope with abstract requirements and make the translation usable for automated verification. This simplified view proves the possible use of this concepts in the building industry, but the main objective to capture mental processes is undervalued in here. Next to this the process presented in this research from an abstract requirement to the eventual automated verification is in reality not that simple, because several calculation models will be needed and just one component of an abstract requirement is used. But as demonstrated there are possibilities to broaden the applicability.

### 7.2.2 Limitations

The limitations of this research relate to the requirement specification process on the one side and the verification process on the other side.

Requirement specification process (ES):

- Only one component of one abstract requirements is used in this research to develop and prove the concept of the use of an Expert System. And only for a workplace in a pilot case.
- The input of knowledge, out of the conducted interviews, is just from a limited number of people and not a full knowledge base is developed.
- The answers of a client to the system will always be subject to other processes, inspirations and ideas. Also the possible answers are now always viable, which will not the case in the real-world. A possibility can be a more advanced system with possibilities of no applicable answers or other requirement connections that are used.
- Assumptions are made towards the choice of an abstract requirement, possible answers of the questions and for which place the requirement was applicable.
- It is a Proof of Concept, which always need some development to come to a working prototype.

Verification process (automated verification):

- Only one pilot case is used.
- Information needed for verification was not present in the pilot case. And is manually added to show the possible working.
- Only checked one BIM on one value.
- Only computable values are used and useful for this automated verification.

## 7.3 Recommendations for implementations

The recommendations can be divided into recommendations for implementation and for future research.

- The Expert System, in a more advanced form, can be used during discussions and meetings with the client and user of an utility building. To make the requirement more clear and work towards a more founded decision, based on the user. In addition this result can be subject for discussion, but it will reduce errors in interpretation next to time and costs reductions. It also reduces the risk of work in later stages.
- This Expert System can be part of a chain of translations up to and including verification of these values. There is a growing interest in verifying more and more requirement components, including abstract formulated parts. More and more data and information are linked to each other, which makes it possible to verify certain parts of abstract requirements.
- It will be possible to expand the deployment to other requirements that are abstract formulated, such as sustainability, security, maintainability, etcetera.

## 7.4 Recommendations for future research

- The mental processes that must be captured in order to know how experts translate and define abstract requirements are difficult to capture. Nonetheless, it is important that information about these mental processes are combined in any translating tool, such as presented in this research, to use how the decision making capabilities and belonging mental processes are working and can be used as input for such translating tools.
- The mental processes together with the presented tools in this research are just a first step to improve requirement definition processes using a certain Expert System. More research and development are needed to broaden the implementation and to work towards a real prototype or even real use in the building industry.
- The knowledge base, needed as base for the ES, is in this research filled with not much information. To effective gather information, knowledge and experiences from experts a tool can be developed to collect this. In this research the possible way it can contribute and work is presented, but to enroll this into a more advanced tool and prototype the knowledge base must be filled, which is manually not feasible.
- In the medical world the results of diagnoses are evaluated and learned from. In the building industry decisions that are made in connection with users and activities are not evaluated. This can help to learn from and use for future project designs. It is worth investigating how this can be done.

## 8.0 Bibliography


Abraham, A. (2005). *Rule-based Expert Systems*. (P. Sydenham & R. Thorn, Eds.), *Handbook for Measurement Systems Design*. London: John Wiley and Sons Ltd.

Anantaram, C., Nagaraja, G., & Nori, K. V. (1998). Verification of accuracy of rules in a rule based system. *Data and Knowledge Engineering*, *27*(2), 115–138. https://doi.org/10.1016/S0169-023X(98)00009-3

Boehm, B. W. (1984). Verifying and Validating Software Requirements and Design Specifications. *IEEE Software*, *1*(1), 75–88. https://doi.org/10.1109/MS.1984.233702

Bray, I. K. (2002). *An Introduction to Requirements Engineering*. Pearson Education.

Campbell, D. A. (2006). Modeling Rules. Retrieved from http://www.architectureweek.com/2006/1011/tools_1-1.html

Daclin, N., Daclin, S. M., Chapurlat, V., & Vallespir, B. (2016). Writing and verifying interoperability requirements: Application to collaborative processes. *Computers in Industry*, *82*, 1–18. https://doi.org/10.1016/j.compind.2016.04.001

de Haan, B., Degenkamp, A., Schotanus, F., & Mulder, I. (2017). *Zachte én harde factoren bepalend voor toepassing functioneel specificeren*. Den Haag.

Delghandi, M. (2018). *The translation of ambiguous client requirements into product specifications*. Eindhoven.

Doerr, J., Kerkow, D., Koenig, T., Olsson, T., & Suzuki, T. (2005). Non-functional requirements in industry - three case studies adopting an experience-based NFR method. *13th IEEE International Conference on Requirements Engineering (RE'05)*, 373–382. https://doi.org/10.1109/RE.2005.47

Dohn, K. Guminski, A. Matusek, M. Zolenski, W. (2013). Implementation of expert system in knowledge management in mechanical engineering enterprises k. *Information Systems in Management*, *2*(4), 253–262.

Eastman, C., Lee, J. min, Jeong, Y. suk, & Lee, J. kook. (2009). Automatic rule-based checking of building designs. *Automation in Construction*, *18*(8), 1011–1033. https://doi.org/10.1016/j.autcon.2009.07.002

Garrett, J. H., & Fenves, S. J. (1987). A knowledge-based standards processor for structural component design. *Engineering with Computers*, *2*(4), 219–238. https://doi.org/10.1007/BF01276414

Giarratano, J. C. (2000). *CLIPS User's Guide*. *CLIPS Reference Manual*. https://doi.org/10.13140/RG.2.1.3966.1200

GitHub. (2019). Built for developers. Retrieved January 2, 2019, from https://github.com

Glinz, M. (2005). Rethinking the Notion of Non-Functional Requirements. *Proceedings of the Third World Congress for Software Quality*, (September), 55–64.

Golik, A., & Golik, M. (2000). *Development and Functioning of Expert Systems as Illustrated by selected examples*.

Hjelseth, E. (2015). BIM-based Model Checking (BMC). *Building Information Modeling*, (May 2015), 33–61. https://doi.org/10.1061/9780784413982.ch02

Hjelseth, E., & Nisbet, N. (2010). Overview of concepts for model checking. *Proceedings of the CIB W78 2010: 27th International Conference –Cairo, Egypt*, (September), 16–18.

Hoeber, H. (2012). *De meerwaarde van BIM in het specificatie en ontwerpproces volgens SE*. Arnhem.

Hull, E., Jackson, K., & Dick, J. (2011). A Generic Process for Requirements Engineering. *Requirements Engineering*, 25–45. https://doi.org/10.1007/978-1-84996-405-0

Hutchins, W. J., & Somers, H. L. (1992). *An introduction to machine translation. Chapter 5: Analysis*. London: Academis Press.

Jensen, P. A. (2011). Inclusive briefing and user involvement: Case study of a media centre in denmark. *Architectural Engineering and Design Management*, *7*(1), 38–49. https://doi.org/10.3763/aedm.2010.0124

Jin, Z. (2018). Requirements and Requirements Engineering - Chapter 1. *Environment Modeling-Based Requirements Engineering for Software Intensive Systems*, 3–11. https://doi.org/10.1016/B978-0-12-801954-2.00001-7

Kamara, J. M., Augenbroe, G., Anumba, C. J., Carrillo, P. M., Kamara, J. M., Augenbroe, G., … Management, A. (2002). Knowledge management in the architecture, engineering and construction industry. *Construction Innovation*, *2*, 53–67.

Kamsties, E., Berry, D. M., & Paech, B. (2001). Detecting Ambiguities in Requirements Documents Using Inspections, 1–12. Retrieved from http://publica.fraunhofer.de/documents/N-6838.html\nhttp://www.cas.mcmaster.ca/wise/wise01/KamstiesBerryPaech.pdf\nhttp://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.145.6497&rep=rep1&type=pdf

Kim, J., & de Dear, R. (2013). Workspace satisfaction: The privacy-communication trade-off inopen plan offices. *Journal of Environmental Psychology*, *36*, 18–26. https://doi.org/10.1016/j.jenvp.2013.06.007

Kossiakoff, A., Sweet, W. N., Seymour, S. J., & Biemer, S. M. (2011). *Systems Engineering Principles and Practice: Chapter 4*. https://doi.org/10.1002/0471723630

Lan, K., & Shih, W. (2014). Expert Systems with Applications An intelligent driver location system for smart parking. *Expert Systems With Applications*, *41*(5), 2443–2456. https://doi.org/10.1016/j.eswa.2013.09.044

Laptali, E., & Bouchlaghem, N. M. (1995). Expert systems within the construction industry in the UK. *Automation in Construction*, *3*(4), 321–325. https://doi.org/10.1016/0926-5805(94)00037-N

Lee, H., Lee, J. K., Park, S., & Kim, I. (2016). Translating building legislation into a computer-executable format for evaluating building permit requirements. *Automation in Construction*, *71*, 49–61. https://doi.org/10.1016/j.autcon.2016.04.008

Leffingwell, D., & Widrig, D. (1999). *Managing Software Requirements* (Vol. 10). https://doi.org/10.1002/spip.383

Mallek, S., Daclin, N., & Chapurlat, V. (2012). The application of interoperability requirement specification and verification to collaborative processes in industry. *Computers in Industry*, *63*(7), 643–658. https://doi.org/10.1016/j.compind.2012.03.002

Mannion, M., & Keepence, B. (1995). SMART requirements. *ACM SIGSOFT Software Engineering Notes*, *20*(2), 42–47. https://doi.org/10.1145/224155.224157

Martins, J. P., Rangel, B., & Abrantes, V. (2016). Automated rule-checking – a tool for design development. *41st IAHS World Congress on Housing - Sustainability and Innovation for the Future*. Retrieved from http://hdl.handle.net/10216/85536.

Matsugu, B. (2015). All Requirements Are Not the Same – Using Non-Functional Requirements for Long-Term Success. Retrieved from https://www.blueprintsys.com/blog/non-functional-requirements/

Matsumoto, Y., Shirai, S., & Ohnishi, A. (2017). A Method for Verifying Non-Functional Requirements. *Procedia Computer Science*, *112*, 157–166. https://doi.org/10.1016/j.procs.2017.08.006

Maxwell, J. A. (2013). Conceptual Framework: What do you think is going on? *Qualitative Research Design: An Interactive Approach*, 39–72. https://doi.org/10.1007/978-3-8349-6169-3_3

Moonen, L. T. (2016). *Improving the design process : the implications of automated verification of client specific requirements using semantic web standards and rule checking techniques*. Retrieved from https://pure.tue.nl/ws/files/51208947/Moonen_2016.pdf

Negnevitsky. (2002). Lecture 2 Rule-based expert systems. *Knowledge Creation Diffusion Utilization*, 1–47. https://doi.org/10.1145/1329125.1329326

Niemeijer, R. A. (2011). *Constraint specification in architecture: a user-oriented approach for mass customization*. Eindhoven. https://doi.org/10.6100/IR715226

Nissanke, N. (1999). *Formal Specification*.

NVBV. (2018). *Handboek Bouwfysische Kwaliteit Gebouwen*.

OpenUniversiteit. (n.d.). Requirements engineering and quality attributes. In *Software architecture* (pp. 20–32). Open Universiteit.

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van De Walle, R., & Van Campenhout, J. (2011). A semantic rule checking environment for building performance checking. *Automation in Construction*, *20*(5), 506–518. https://doi.org/10.1016/j.autcon.2010.11.017

Pegoraro, C., & Paula, I. C. de. (2017). Requirements processing for building design: a systematic review. *Production*, *27*(0), 1–18. https://doi.org/10.1590/0103-6513.212116

Pinheiro, F. A. C. (2004). *Requirements traceability*. *Kluwer International Series in Engineering and Computer Science*.

Riley, G. (2008). CLIPS. Retrieved December 12, 2018, from http://clipsrules.sourceforge.net/WhatIsCLIPS.html

Riley, G. (2017). Basic Programming Guide. *CLIPS Reference Manual*, *I*.

Ruiz-Mezcua, B., Garcia-Crespo, A., Lopez-Cuadrado, J. L., & Gonzalez-Carrasco, I. (2011). An expert system development tool for non AI experts. *Expert Systems with Applications*, *38*(1), 597–609. https://doi.org/10.1016/j.eswa.2010.07.009

SBR. (n.d.). *Voldoet dit gebouw? Het bepalen van de functionele kwaliteit*. Rotterdam.

Schaap, H., Bouwman, J., & Willems, P. (2008). COINS-referentiekader voor functioneel specificeren. Retrieved from www.coinsweb.nl

Schneider, F., & Berenbach, B. (2013). A literature survey on international standards for systems requirements engineering. *Procedia Computer Science*, *16*(October 2014), 796–805. https://doi.org/10.1016/j.procs.2013.01.083

Shen, Q., Li, H., Chung, J., & Hui, P. Y. (2004). A framework for identification and representation of client requirements in the briefing process. *Construction Management and Economics*, *22*(2), 213–221. https://doi.org/10.1080/0144619042000201411

Singaram, M., & Jain, P. (2018). What is the difference between Proof of Concept and Prototype? Retrieved January 3, 2019, from https://www.entrepreneur.com/article/307454

Soetanto, R., Dainty, A. R. J., Glass, J., & Price, A. D. F. (2006). Towards an explicit design decision process: The case of the structural frame. *Construction Management and Economics*, *24*(6), 603–614. https://doi.org/10.1080/01446190600568173

Solihin, W., & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, *53*, 69–82. https://doi.org/10.1016/j.autcon.2015.03.003

Sommervile, I. (2009). *Software Engineering* (Ninth Edit). https://doi.org/10.1111/j.1365-2362.2005.01463.x

Sparrius, A. (2014). *The Life Cycle of a Requirement*. Pretoria. Retrieved from https://www.thoughtco.com/life-cycle-of-a-jellyfish-4112280

Tanwar, P., Prasad, T. V., & Datta, K. (2014). An Effective Reasoning Algorithm for Question Answering System. *International Journal of Advanced Computer Science and Applications*, 52–57. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.674.3325&rep=rep1&type=pdf#page=56

*Technisch Programma van Eisen - Onderwijshuisvesting De Basis Fluvius*. (2017).

Tjong, S. F. (2008). Avoiding ambiguitiy in requirements specifications. *Faculty of Engineering & Computer Science*, *PhD*(February), 416.

van Lamsweerde, A. (2000). Formal specification: a roadmap. *Proceedings of the Conference on The Future of Software Engineering*, 147–159. https://doi.org/10.1145/336512.336546

Wheatcraft, L. S. (2012). Thinking Ahead to Verification and Validation, 1–25.

Wieringa, R. J. (2014). *Design science methodology: For information systems and software engineering*. Enschede: Springer Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-662-43839-8

Wood, H., & Gidado, K. (2008). *Project Complexity in Construction*. Brighton.

Yang, J., Li, H., & Skitmore, M. (1996). Expert Systems in Construction Management : Is the Hype Over ? *Proceedings International Construction Information Technology Conference*, 131–136.

Young, R. (2004). *The requirements engineering handbook*. Boston / London: Artech House. Retrieved from http://medcontent.metapress.com/index/A65RM03P4874243N.pdf\n http://196.29.172.66:8080/jspui/handle/123456789/2642

Zhang, J., & El-Gohary, N. M. (2017). Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. *Automation in Construction*, *73*, 45–57. https://doi.org/10.1016/j.autcon.2016.08.027

Zowghi, D., & Gervasi, V. (2003). On the interplay between consistency, completeness, and correctness in requirements evolution. *Information and Software Technology*, *45*(14), 993–1009. https://doi.org/10.1016/S0950-5849(03)00100-9

# 9.0 Appendices

## 9.1 Interview questionnaire

1) Introduction (10 minutes)
   a. What is your function within <Company>?
   b. In which specific area do you have an expertise?
   c. How does this expertise relate to requirements engineering?
   d. Have you ever dealt with what we called "non-functional requirements" (described as abstract, undefined and non-computable requirements) in projects?

2) Design process (10-15 minutes)
   a. What are the difficulties with these kind of requirements from your point of view?
   b. In which phase of the design process (based on the ten phase scale) was dealt with these requirements?
   c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?
      - Yes, how?
      - No, why not?

3) Requirements (40 minutes)

   ***Part 1 – Abstract requirements*** *(15 minutes)*

   a. How is dealt with this specific type of requirements?
      i. What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements?
      ii. Which different abstract requirements are you facing during the design process?
      iii. And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common?
      iv. What is the currently used main approach when you are facing such requirements during design processes?

   ***Part 2 – Comfort at work spaces*** *(20 minutes)*

   b. What steps do you perform to deal with comfort?
      i. Which expressions and relating rules do you assign to this specific abstract requirement, comfort? (think of temperature, light and sound/acoustics)
      ii. Which parameters and relating values, including units, follow from this? (if Vabi Elements is used can you be specific about how you use these values from the parameter during the use of this program?)
      iii. Which method are you using to perform the aforementioned expressions, rules, values and units?
      iv. How are the choices you made justified towards the company and towards the client?

   ***Part 3 – Expert Systems*** *(5 minutes)*

   c. How can Expert Systems support this abstract requirements definition process?
      i. Are you familiar with the concept of Expert System?
         - Yes, do you use it in design processes?
         - No, (explanation)
      ii. Do you think Expert Systems can contribute to the requirements engineering process, mainly for abstract requirements, during design phases of construction projects?
      iii. How are you currently dealing with verification of these requirements now and what could be the changes when using an ES?

## 9.2 Interviews

**1. BIM Manager Strukton**

1) Introduction
   a. What is your function within <Company>?
      *Strukton (Worksphere) richt zich op civiele projecten, grootste onderdeel is gericht op rail, vanuit prorail en een deel worksphere welke projecten aanpakt met betrekking tot werkplekken ed. in gebouwen en dan de hele lifecycle tot en met beheer en onderhoud. Dit laatste werk ik bij.*
   b. In which specific area do you have an expertise?
      *Ooit begonnen als installatieadviseur. Daarna 2 jaar bij BAM BIM centre gewerkt en vervolgens 10 jaar bij RHDHV als BIM Manager. Nu bij strukton vooral bezig met digitale transformatie om eigenlijk de lifecycle data van werkplekken ed. te koppelen aan BIM. Vooral bezig om de visie/missie en strategie die gevormd wordt te bewaken en aan te passen.*
   c. How does this expertise relate to requirements engineering?
      *Vanuit de projecten wordt vooral gezien dat PvE met geschreven tekst veel fouten en vragen oproept en dat er een vertaling nodig is om deze eisen smart te formuleren. Hierin zit een uitdaging bij ieder project.*
   d. Have you ever dealt with what we called "non-functional requirements" (described as abstract, undefined and non-computable requirements) in projects?
      *Zeker, deze eisen komen vaak terug en in 99,9% van de gevallen zijn eisen nog niet specifiek genoeg geformuleerd.*

2) Design process
   a. What are the difficulties with these kind of requirements from your point of view?
      *Moeilijkheden die warden genoemd zijn met name de interpretatie hiervan, omdat zelfs een collega die naast mij zit alweer dingen anders ziet en leest. Heel lastig om hier mee om te gaan en vervolgens eenduidig te maken. Een ander punt is dat er te weinig discussie is met de OG om een abstracte eis specifiek te krijgen. Dit wordt geprobeerd om te ondervangen door middel van technische oplossingen zoals software (relatics) om abstracte eisen ed. op te bouwen adhv van decompositiebomen en zo deze eisen te kunnen afleiden.*
   b. In which phase of the design process (based on the ten phase scale) was dealt with these requirements?
      *Vaak wordt aan het begin van een project al geprobeerd om deze eisen specifieker te krijgen, maar een OG (vaak niet de gebruiker, maar wel formeel degene die zegt wat hij/zij wilt) kan dit niet altijd. Gedurende het ontwerpproces zijn er een aantal momenten contractueel vastgelegd om hierover te praten, maar na DO kan er niks meer aangepast worden, dus aan de voorkant moet al zoveel mogelijk gespecificeerd worden en dat lukt niet bij dit soort eisen.*
   c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?
      - Yes, how?
        *Ja, op het moment dat je aan een project begint is het nog heel abstract net als de eisen, maar bij het specifieker maken van het Ontwerp loopt de eis specifieker maken niet op het zelfde niveau mee. Maar wijzigen later gebeurt niet vanwege kosten ed.*

3) Requirements
   ***Part 1 – Abstract requirements***
   a. How is dealt with this specific type of requirements?
      i. What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements?

*Een voorbeeld; een beeldscherm goed kunnen zien tijdens het werken (wollig geformuleerd = abstract) en het beeldscherm moet licht van 500 lux uitstralen tijdens het werken (specifiek geformuleerd)*

ii. Which different abstract requirements are you facing during the design process?
*Verschillende soorten zoals echt wensen, maar ook comfort, duurzaamheid, voldoen aan BREAAM etc.*

iii. And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common?
*Zie moeilijkheden eerder genoemd.*

iv. What is the currently used main approach when you are facing such requirements during design processes?
*De manier waarop Strukton werkt is als volgt: dialogen met OG/gebruiker zijn contractueel vastgelegd anders krijgen ze steeds extra wensen (echter zijn er maar een paar toetsingsmomenten voor dit soort eisen).*
*Gedurende het ontwerpproces wordt er eerst mono-disciplinair gekeken naar de eisen en vervolgens via integrale sessies de volledige invulling gegeven aan een dergelijke eis. Door het smart maken van eisen en referenties van projecten te gebruiken (zoals eerdere ontworpen ziekenhuizen en labaratoria ed.) kunnen deze weer gebruikt worden en aangepast worden. Zo wordt omgegaan om abstracte eisen toch al zo specifiek mogelijk te maken aan de voorkant van het proces. Echter is het wel zo dat een professionele OG al aan de voorkant specifiek over dit soort eisen kan zijn en de minder of niet professionele OG kan dit niet, dan vullen wij dit adhv referenties en al opgeschreven invullingen van deze eis uit eigen inzichten ed. in (vanuit dus andere projecten). Deze zijn dus ingevuld door de sessies.*

***Part 2 – Comfort at work spaces***

b. What steps do you perform to deal with comfort?

i. Which expressions and relating rules do you assign to this specific abstract requirement, comfort? (think of temperature, light and sound/acoustics)

*De hoofdelementen die hieraan worden gehangen zijn; geluid, licht, klimaat, materialisatie en verschillende raakvlakken. (raakvlak is bijvoorbeeld de afstand tot verschillende zaken zoals uitgang, parkeren, koffie, etc.). vaak ook specifiek gelet op B&O.*

ii. Which parameters and relating values, including units, follow from this? (if Vabi Elements is used can you be specific about how you use these values from the parameter during the use of this program?)

*Gebruik van Vabi inderdaad, daar proberen ze bij de bepaling van waarden al wel rekening mee te houden zodat ze voor berekeningen inderdaad de juiste waarden al uit het specificeren krijgen. De volgende lijst geeft een overzicht:*

- *Licht: Beeldscherm licht, niet zitten tegen daglicht in, sterkte van het licht op bureaublad, regelbaar daglicht (hiermee invloed op sterkte), ook kleuren RGB, LEDS regelbaar kleur vaak, koel is blauw licht voor in de ochtend, rood is warm licht voor in de middag na de pauze*
- *Geluid: nagalmtijd, positie ventilatierooster, geluid rooster dus (ook invloed op temperatuur), buitengeluid, dempende maatregelen*
- *Klimaat: temperatuur, co2 (ppm). Luchttemp, vochtigheid, luchtverandering, mate van beinvloedbaarheid, microklimaat. (waardes ed. die hier voor gelden vaak vanuit TNO achtige instellingen)*
    - *Temperatuur: straling van buiten (koude) ondervang je niet met warme lucht*
- *Materialisatie: gevoel, onderhoud*

- *Overig: individueel te regelen, of voor 1 kantoortuin. Individueel zou dan weer lucht ed. zoals in autos per beeldscherm oid.*
- *Hoogte van bureau instelbaar (arbo ed.)*
- *Contactdozen onder tafel dus bukken, of in bureau/USB ed.*
  *Dan is het ook nog zo dat dit dan op aantal personen /ruimte wordt bepaald, maar als er meer mensen in gaan zitten oid. Dan wordt het dus misbruikt door het gebruik.*

iii. Which method are you using to perform the aforementioned expressions, rules, values and units?

*Dit is dus gedurende de mono disciplinaire sessies van verschillende invalshoeken waarna deze samen komen in integrale sessies. De waardes worden vaak aangegeven mbv een grenswaarde (temperatuur max zo hoog bij buitentemperatuur van X, ventilatie minimaal zoveel) = grenswaardes, niet gemiddeldes.*

iv. How are the choices you made justified towards the company and towards the client?

*Vanuit de sessies, maar ook vanuit de expertise van de adviseurs ed. Klant wordt dus op aantal toetsmomenten meegenomen, maar verder zoveel mogelijk aan de voorkant zeggen anders eigenlijk pech.* ***(hier kan dus ES bij helpen?).*** *Het doel is wel om een installatie (ruimte) te ontwerpen die past bij de eis, maar binnen de kosten. (vaak OG toch vanuit financiele invalshoek).*

***Part 3 – Expert Systems***

c. How can Expert Systems support this abstract requirements definition process?

i. Are you familiar with the concept of Expert System?

- No, (explanation)
  *Niet mee bekend, maar na uitleg wel duidelijk.*

ii. Do you think Expert Systems can contribute to the requirements engineering process, mainly for abstract requirements, during design phases of construction projects?

*Dit gebeurt nu ook al, maar dan door de sessies die worden gehouden en hebben de mensen samen (de experts dus) deze kennis bij zich. Kosten zal wel altijd belangrijk blijven bij het maken van keuzes. Een stuk standaardisatie is nuttig en nodig (vandaar referenties gebruiken en eigen invulling abstracte eisen).* ***Sommige stukjes zul je moeten aantonen voor iemand die een onderdeel van die eis zo wil, kan niet altijd beantwoord worden met vragen. Bijvoorbeeld als alle keuzes te simuleren zijn met bijvoorbeeld VR dan kan alles wat gekozen is/kan worden aangetoond worden. Dit is nog toekomst nu.***

iii. How are you currently dealing with verification of these requirements now and what could be the changes when using an ES?

*Toegevoegde waarde zit in: efficientie, minder discussie en dat er zo altijd een bepaald informatie niveau beschikbaar zal zijn wat gebruikt kan worden.*

**2. Senior Advisor Rijksvastgoedbedrijf**

1) <u>Introduction</u>

a. What is your function within <Company>?

*Vanuit de bouwfysica gestart en nu aan de kant van OG gegaan. Voor de overheid en dan met name vanuit de rol tussen Architectuur en Techniek. Bezig met transitie projecten en nieuwe projecten als overheidsgebouwen. Hiervoor hebben ze vaak standaard opzetten met betrekking tot eisen. Vaak op financieel gebied, beheer en onderhoud en strategie van overheidsgebouwen.*

*Opstellen van het PvE op de eisen duurzaamheid/comfort, maar ook veiligheid en beeldkwaliteit. Daarbij wordt dan een marktpartij gezocht die de functionele specificatie uit moet voeren.*

*Prijs/kwaliteit verhouding zeer belangrijk want deze projecten worden betaald van het geld van de belastingbetalers. Credo is 'sober en doelmatig'.*

b. In which specific area do you have an expertise?
*met name in het specificeren van comfort/duurzaamheids eisen.*

c. How does this expertise relate to requirements engineering?
*Specificities zorgen voor de uitvraag van deze eisen.*

d. Have you ever dealt with what we called "non-functional requirements" (described as abstract, undefined and non-computable requirements) in projects?
*Expertise volgt vanuit bouwfysica op het gebied van comfort en duurzaamheid, welke meteen al omschreven kunnen worden als dit soort eisen waarop gefocust wordt.*

2) <u>Design process</u>

a. What are the difficulties with these kind of requirements from your point of view?
*Eigenlijk zijn er geen moeilijkheden hoor (eerste reactie). Moeilijkheden bij deze eisen zit vaak in het feit dat ze niet zichtbaar zijn. Een afmeting is te zien, maar comfort zie je niet direct terug. Om toch dit tastbaarder te maken wordt dit vaak gesimuleerd in rekenmodellen ed. en vervolgens wordt gekeken naar hoe de markt de eisen invult of er inderdaad rekening gehouden wordt met de gestelde eisen (bijvoorbeeld geluid → en dan materiaal wat inderdaad zorgt voor een bepaalde isolatie hiervoor, dat de markt dit ziet en doorheeft en dus naar deze eisen handelt)*

b. In which phase of the design process (based on the ten phase scale) was dealt with these requirements?
*In het pve kom je niet snel een eis als comfort tegen, maar al gespecificeerder. En dan wordt de exacte invulling hiervan ingevuld door middel van gesprekken met de gebruiker voor de vertaling.*

c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?

- Yes, how?

*Ja omdat er dus niet heel gemakkelijk al invulling kan worden gegeven, en met name omdat expert (technische invulling) en gebruiker (weinig technische kennis, maar wel wat ie wil) elkaar moeten vinden. Dit moet dan gebeuren via een aantal gesprekken. Wat ook nog telt is dat gebruiker en OG anders (kunnen) zijn, en dan wordt er dus invulling aan dit soort (subjectieve) onderdelen van eisen gegeven door een expert en OG en de gebruiker daar eigenlijk niet tussenkomt terwijl diegene daar straks wel zit en gebruik maakt van de ruimte.*

3) <u>Requirements</u>

***Part 1 – Abstract requirements***

a. How is dealt with this specific type of requirements?

i. What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements?
*Vooral dat abstracte eisen dus eigenlijk vaak over dingen gaan die niet meteen zichtbaar zijn.*

ii. Which different abstract requirements are you facing during the design process?
*Met name comfort, duurzaamheid, veiligheid en beeldkwaliteit.*

iii. And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common?
*Zie punt i.*

iv. What is the currently used main approach when you are facing such requirements during design processes?
*In de projecten bij de overheid gaan ze er als volgt mee om: deze abstracte eisen worden (met name comfort) gespecificeerd via sessies en wat er vanuit onderzoek over gezegd is. Dit staat allemaal in Relatics, en ieder project wordt deze opbouw erbij*

*gepakt en geldt dit voor alle projecten al voor 99%. Een aantal dingen zullen wel afwijken en die worden dan "uit" gezet. Wel wordt er soms wat vergeten en dan moet dit nog gewijzigd worden in het totaalpakket aan eisen. Maar deze eisen worden dus altijd op een zelfde manier gebruikt en toegepast voor de verschillende projecten en eigenlijk wordt er vanuit een gebruiker nooit iets gewijzigd, aan toegevoegd of weggehaald. Het is dus echt kopieren van deze eisen.*

***Part 2 – Comfort at work spaces***

b. What steps do you perform to deal with comfort?

i. Which expressions and relating rules do you assign to this specific abstract requirement, comfort? (think of temperature, light and sound/acoustics)
*Geluid, thermisch comfort (temperatuur, verloop binnen,buiten), luchtkwaliteit (ventilatie), daglicht*

ii. Which parameters and relating values, including units, follow from this? (if Vabi Elements is used can you be specific about how you use these values from the parameter during the use of this program?)
*Altijd worden waardes vanuit onderzoek gebruikt end it geldt dan voor 90%. Nooit zal 100% van de mensen tevreden zijn met welke eis je stelt en als daaraan wordt voldaan. Het document met waardes is eigenlijk voor comfort leidend (hebben ze zelf ook aan geschreven). Comfort bevat op die manier +/- 2000 eisen. Je zult altijd klagers hebben, en bij bijvoorbeeld comfort klasse A zijn dit 5% en bij klasse B 10%, etc.*

*1. Isso 74 voor behaaglijkheid*

iii. Which method are you using to perform the aforementioned expressions, rules, values and units?
*Via gesprekken en vanuit onderzoeken. Eigenlijk mist het hier dus vooral aan individuele inspraak voor de gebruikers van een ruimte. Want waardes zouden eigenlijk ruimte afhankelijk en eisafhankelijk moeten zijn,* ***maar het is ook afhankelijk van de gebruiker!***

iv. How are the choices you made justified towards the company and towards the client?
*Experts worden aangehangen en dus als iemand er zo veel onderzoek naar gedaan heeft dan betekend het dat dat voor de meeste wel zal gelden en wordt dit overgenomen en verwezen naar deze documenten.*

***Part 3 – Expert Systems***

c. How can Expert Systems support this abstract requirements definition process?

i. Are you familiar with the concept of Expert System?

- Yes, do you use it in design processes?
  *In zeker zin wel, maar niet geheel in de context voor deze processen. Na uitleg het volgende.*
- No, (explanation)

ii. Do you think Expert Systems can contribute to the requirements engineering process, mainly for abstract requirements, during design phases of construction projects?
*De toegevoegde waarde zal met name zitten in het feit dat het eisen bespreekbaar kan maken. Omdat aan de ene kant de Expert met veel kennis erover zit en aan de andere kant de OG/Gebruiker die er niet concreet en specifiek weet wat het betekent. Via zo'n dergelijk systeem kan je dit aan elkaar koppelen. Het kan ook als ondersteuning zijn voor expertsessies doordat zo subjectieve onderdelen van deze eisen naar voren komen en ingevuld kunnen worden door de gebruikers. Wat minder goed gebeurt en gaat tijdens een sessie met OG oid. daarnaast kan het ook zorgen*

*voor een stuk automatiseren van de sessies en voor uitleg van dit soort eisen aan klanten.* ***Voorbeeld van akoestiek: iemand als klant zegt ik wil verstaanbaar zijn in een open ruimte, geen idee wat dat voor dB betekent, maar als dat zijn antwoord is dan hoort er vanuit expert deze waarde bij (uit dus onderzoeksdocument met waardes).***
***en ook interessant dan hoe je deze simpele vragen moet stellen? Welke factoren (mbt de vragen stellen) zijn van invloed voor het goed functioneren van een ES voor abstracte eisen?***
***En met name zorgen dat experts op zelfde niveau kunnen komen met de klant/gebruiker!***

iii. How are you currently dealing with verification of these requirements now and what could be the changes when using an ES?
*Verificatie gaat nu doordat steeds dezelfde eisen dus gehaald moeten worden. Bewezen door simulaties en later door metingen, maar dan gaat het wel steeds om dezelfde eis en eigenlijk achteraf gekeken of dat ook inderdaad hetgene is wat iemand bedoelde en of dit dus klopt. Maakt het proces dus niet volledig goed vanuit de V&V gezien.*

### 3. Global Business Development Director RoyalHaskoningDHV

1) <u>Introduction</u>
   a. What is your function within <Company>?
   *Huisvestingsadvies ; wat wil de gebruiker? Echt de PvE's opstellen vanuit de gebruiker gezien.*
   b. In which specific area do you have an expertise?
   *In het kijken vanuit de gebruiker wat diegene wilt, en vanuit hier dus echt invulling geven aan de eisen. Technisch veel minder echt vanuit gevoel wat die gebruiker nu wilt en wat die eis dan dus voor invulling krijgt tov de gebruiker.*
   c. How does this expertise relate to requirements engineering?
   *Gedachte is veel meer dat er user-centric gekeken dient te worden. Wie is die gebruiker en wat wilt die dan dus? Hoe hiermee omgegaan wordt is terug te brengen naar het omgaan met eisen, en dan dus met name met de subjective die nog niet gedefinieerd kunnen worden.*
   d. Have you ever dealt with what we called "non-functional requirements" (described as abstract, undefined and non-computable requirements) in projects?
   *Daar ligt de uitdaging om dit soort eisen te specificeren, heel veel verschillende eisen kom je tegen.*

2) <u>Design process</u>
   a. What are the difficulties with these kind of requirements from your point of view?
   *De moeilijkheid zit vooral in dat technische mensen moeite lijken te hebben wat een bepaalde waarde of eis doet voor de gebruiker. Er zit dus een verbinding tussen prestatie (eis) en de waarde voor een klant en dit maakt het moeilijk om deze eis in te vullen.*
   b. In which phase of the design process (based on the ten phase scale) was dealt with these requirements?
   *Deze eisen worden in hoofdlijnen beschreven aan het begin van het proces, maar nog niet ingevuld tot meer dan wat er nodig is om te weten. Je wilt van deze gebruikerseisen weten op hetzelfde niveau als waar je met ontwerpen van het gebouw zit. Tenzij er dingen later in het proces zullen zijn die bij wijziging te veel geld gaan kosten. (als er door een eis een andere vloer nodig is dan wil je dat dus wel al aan het begin weten). Essentie van het gebruik wil je aan begin weten.*

c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?

- Yes, how?

  *Ja, want juist deze essentie van het gebruik is belangrijk om te weten in het begin maar dat kan de klant niet altijd vertellen aan je. Hiervoor worden verschillende methoden en technieken voor ingezet om deze specificatie toch zover te krijgen en de gebruiker te onderzoeken en doorgronden.*

3) <u>Requirements</u>

***Part 1 – Abstract requirements***

a. How is dealt with this specific type of requirements?

i. What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements?

*Bij de abstracte eisen wil je echt de waarde achterhalen die het voor de gebruiker zal zijn, dit hoeft bij specifieke eisen (zoals afmetingen) niet te doen, dit is zo en daar moet het aan voldoen en is voor iedereen duidelijk of het zo is of niet.*

ii. Which different abstract requirements are you facing during the design process?

*Allerlei soorten die je in projecten tegenkomt. Maakt verder niet uit welke het is, maar de methoden om de waarde voor de klant te bepalen is veel belangrijker, dat kan via verschillende methoden.*

iii. And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common?

*Bij dit soort eisen zit echt een belangrijk punt bij de problem discovery, waarom wil die klant dit zo. Daar moet je zien achter te komen, want misschien weet ie niet wat ie wil en het probleem is, maar als je daarachter komt kan je voor een oplossing gaan. En misschien wel een betere en andere oplossing dan de klant aandraagt. Veel mensen denken in oplossingen meteen, maar zonder het echte probleem te weten en dat is de uitdaging bij dit soort eisen.*

iv. What is the currently used main approach when you are facing such requirements during design processes?

*Er worden verschillende methoden ingezet voor het specificeren.*

- *User journey's: obv typen gebruikers, waar zit de waarde voor die gebruiker in het gebouw, methodieken vanuit ID*
- *Goede gebruikersvertegenwoordiging; verschillende soorten gebruikers*
- *Begint bij welke doelstellingen/ambities ze willen, vanuit daar komen dan de topeisen.*
  - *ID methodieken gebruiker voor het vragen in ES*
  - *Double diamonds over het waarom (probleem discovery)*

***Part 2 – Comfort at work spaces***

b. What steps do you perform to deal with comfort?

i. Which expressions and relating rules do you assign to this specific abstract requirement, comfort? (think of temperature, light and sound/acoustics)

*Licht, lucht (temperatuur, daglicht), geluid (akoestiek), groen, esthetisch comfort (identiteit/beeldverwachting)*

ii. Which parameters and relating values, including units, follow from this? (if Vabi Elements is used can you be specific about how you use these values from the parameter during the use of this program?)

*Bij de uitkomsten rekening houden met de diversiteit van de gebruikers(types). Comfort komt echt uit een handboek.* ***Maar dit is wel een goede voorbeeldeis voor dit onderzoek, want bevat al de dingen van een abstracte eis; het is grijpbaar met een technische component en een psychologische component. En sluit heel erg bij een user-centric gedachte aan.***

iii. Which method are you using to perform the aforementioned expressions, rules, values and units?
*Zie de main approaches geldend voor abstracte eisen.*

iv. How are the choices you made justified towards the company and towards the client?
*Eigen invulling: doordat vanuit die gebruiker uitgegaan word en hiermee de eisen worden ingevuld kan ook echt hieraan vast worden gehouden dat dit is wat de klant wil en bedoelde.*

***Part 3 – Expert Systems***

c. How can Expert Systems support this abstract requirements definition process?

i. Are you familiar with the concept of Expert System?
- Yes, do you use it in design processes?
*Ja, maar wordt niet gebruikt nog.*

ii. Do you think Expert Systems can contribute to the requirements engineering process, mainly for abstract requirements, during design phases of construction projects?
***ES tool kan helpen bij het echt doorvragen aan de kant van problem discovery, wat wil die gebruiker, waar zit de waarde voor hem in?*** *Nu is er vaak maar weinig energie voor dit doorvragen bij een gebruiker, ES kan hier veel meer energie stoppen in het doorvragen. Uitkomsten hieruit kunnen dan mee worden genomen in de oplossingen.*
*En voor een effectievere manier: gebruikersgroepen in beeld brengen is echt nodig om een goed beeld te creeeren voor wat ze nu willen, daar kan en moet ES voor gaan zorgen*
*Wel zal ES niet alles kunnen vervangen, maar wel kunnen ondersteunen en een meer effectieve manier verzorgen. Bijvoorbeeld bij duurzaamheid; zit ook een stukje creativieit in bij invulling. Dus een ES zal hier wat minder inzetbaar zijn. Wil dit wel werken dan moeten Experts echt in concepten kunnen gaan denken zodat dit als input kan dienen voor een ES.*
***Hindernis is: wat doen gebouwen voor de mensen in het gebruik ervan ? (dit wordt in de medische wereld wel echt geevalueerd en meegenomen en daardoor kan ES ed. ook doorontwikkelen en gebruikt worden) in de bouw gebeurt dit niet op deze manier en als je dit dus niet doet ontbreekt deze info en kan een ES veel minder doorontwikkelen.***

iii. How are you currently dealing with verification of these requirements now and what could be the changes when using an ES?

## 4. Information Manager Arcadis

1) <u>Introduction</u>

a. What is your function within <Company>?
*Met name bezig vanuit de informatiekant om hetgeen achter de eisenspecificatie te configureren. Achtergrond vanuit BPS, maar begonnen in recessie en toen begonnen met werken in de software ontwikkeling. Vanuit hier doorgegaan en nu uiteindelijk op het gebied van informatie management.*

b. In which specific area do you have an expertise?

*Afgestudeerd op BPS, comfort eis goed mee bekend, en vooral hoe processen in software in te richten die door anderen worden ingevuld. Configuratie hiervan.*

c. How does this expertise relate to requirements engineering?
*Het zit niet op het vlak van abstract tot specifiek vertalen, maar meer hetgeen ingevuld moet worden voor deze eisen configureren in bijvoorbeeld Relatics. <X> is iemand die eisen vertalen wel doet en daar standaard eisen uit probeert te halen die bruikbaar zijn en aangevuld worden met projectspecifieke eisen (of aanpassingen).*

d. Have you ever dealt with what we called "non-functional requirements" (described as abstract, undefined and non-computable requirements) in projects?
*Eisen als comfort wel mee bekend, andere abstracte eisen minder.*

2) <u>Design process</u>

a. What are the difficulties with these kind of requirements from your point of view?
*Bij abstracte eisen heb je veel vrijheid, maar ze zijn ongelooflijk lastig te verifieren. Comfort is dan eentje die dit wel in zich heeft, maar andere abstracte eisen niet of nauwelijks. Dit komt met name doordat iedereen als de eis abstract geformuleerd is anders interpreteert.* ***(maar de gebruiker dient dit eigenlijk in te vullen deze moet er werken en de expert kan daar de waardes aan verbinden!?)***

b. In which phase of the design process (based on the ten phase scale) was dealt with these requirements?
*De abstracte eisen staan altijd aan het begin van het project. Vaak worden specifiek geformuleerde eisen ook nog een stap terug gezet om ze abstract te maken en dan wordt er gekeken waarom een klant dit wil om zo niet meteen een oplossing te hebben, want misschien is een andere oplossing wel beter.*

c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?

- Yes, how?
*Ja in het begin is er nog weinig duidelijk en is het ontwerpen onzeker en zit er in het model nog te weinig informatie.*

3) <u>Requirements</u>

***Part 1 – Abstract requirements***

a. How is dealt with this specific type of requirements?

i. What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements?
*Dat zijn opnieuw de moeilijkheden; specifieke eisen zeggen vaak al iets terwijl abstracte eisen nog multi interpreteerbaar zijn. Desalniettemin worden ook specifieke eisen een stap teruggezet om ze als abstract te formuleren om het echte waarom erachter te vinden vanuit de klant.*

ii. Which different abstract requirements are you facing during the design process?
*De bekende abstracte eisen. Hoewel comfort dus wel een ander soort omschrijft omdat deze al wel helemaal uitgeplozen is en niet hetzelfde moeilijkheidsniveau heeft als beeldverwachting. Overigens wordt deze nooit als eis in een project gesteld…*

iii. And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common?
*Ze zijn allemaal anders interpreteerbaar, en worden ook heel erg vanuit kostenplaatje bekeken om zo goedkoop mogelijk te voldoen aan een standaard. Het waarom erachter blijft wel het hoofddoel tijdens eisenprocessen voor specificatie.*

***Ook abstracte eisen echt afhankelijk van type gebouw en type OG, maar een bepaalde standaard wordt altijd al wel gebruikt en dan projectspecifiek aangevuld. (niet weggestreept zoals bij RVB).***

iv. What is the currently used main approach when you are facing such requirements during design processes?

*Het omgaan met abstracte eisen begint bij doelen. Vanuit deze doelen worden functies benoemd en daarna objecten. Deze doelen met het echte waarom omschrijven. Ook de specifieke eisen worden dus eerst abstracte eisen. Klant eisen kunnen of abstract zijn of specifiek maar worden dan teruggebracht tot abstracte eisen. Hierna komen de projecteisen en worden deze gevalideerd met de klanteisen. Dit is een iteratief proces welke herhaalt blijft worden gedurende het ontwerpen. Via workshops met OG wordt invulling proberen te geven aan deze eisen, dit zijn slechts een paar milestones gedurende het proces. Om er dus vooral achter te komen* ***dat wat een klant (OG) zegt ook echt is wat diegene nodig heeft?***

***Part 2 – Comfort at work spaces***

b. What steps do you perform to deal with comfort?

i. Which expressions and relating rules do you assign to this specific abstract requirement, comfort? (think of temperature, light and sound/acoustics)

*de dingen die eraan hangen zijn: m2 opp, ventilatie, daglicht – glare (schittering), uitzicht (is echt anders dan daglicht)-> koudeval (veel uitzicht, heeft veel koudeval), verlichting, temperatuur (man en vrouw anders, is bewezen), omgeving (kleur en groen), akoestiek – ruis, materiaal (tapijt – geluid dempen), routing*

ii. Which parameters and relating values, including units, follow from this? (if Vabi Elements is used can you be specific about how you use these values from the parameter during the use of this program?)

*Meeste waardes van comfort komen toch echt uit richtlijnen en beslaan altijd een bepaalde bandbreedte. In het verifieren hiervan wordt vaak een 3d model via aparte software gemaakt om deze te verifieren. Want informatie voor de specifieke waardes is in zon model niet aanwezig. Denk aan een bureau wat niet in model staat, kan je lichtsterkte daarin niet verifieren, maar is aparte software voor, zo voor licht en geluid etc. ontwerpen blijft onzeker dus je gaat niet door tot laatste info, maar je stopt op een gegeven moment en dan gebruik je die programma's. om rekenmodellen op te zetten voor die specifieke onderdelen die er dan iets over zeggen. En vanuit daar kom je dan op specifieke materialen ed. die worden toegepast.*

iii. Which method are you using to perform the aforementioned expressions, rules, values and units?

*Een eis zoals comfort wordt vaak redelijk 1 op 1 overgenomen voor kantoren omdat het redelijk meetbaar is en echt opgehangen kan worden aan richtlijnen (dit is eigenlijk een uitzondering omdat comfort al uitgeplozen is, voor andere abstracte eisen geldt dit niet.)*

iv. How are the choices you made justified towards the company and towards the client?

*Verificatie comfort wordt met name dan via richtlijnen ed. gedaan. Andere abstracte eisen wordt dan toch geverifieerd via expertise.* ***Dit sluit wel weer aan om deze kennis te vatten en als input voor ES??***

***Part 3 – Expert Systems***

c. How can Expert Systems support this abstract requirements definition process?

i. Are you familiar with the concept of Expert System?

- No, (explanation)

  *Nee niet bekend mee, idee niet helemaal duidelijk, en inzet ziet hij niet helemaal voor deze processen.*

ii. Do you think Expert Systems can contribute to the requirements engineering process, mainly for abstract requirements, during design phases of construction projects?

*Nee zie ik niet zitten: denkt dat ik er niet uit ga komen zo. Comfort is niet altijd anders voor mensen dus dit is geen goede om te pakken hiervoor. Uiteindelijk zullen bij abstracte eisen teveel parameters en variabelen komen kijken bij locatie specifieke projecten. Moeilijkheid zit ook in dat OG abstracte eisen stelt en de vraag is meer of die goed zijn? Via workshops met een OG kom je tot invulling, maar echt de eisen boeit een OG minder hij kijkt naar geld en zo goedkoop mogelijk een gebouw dat wel voldoet aan de eisen. Gebruikers weten richtlijnen niet en zeggen iets, maar is maar de vraag of dat het is, vaak niet en dan ga je daar mee aan de slag maar is het toch niet wat ze willen (ik vind 26 graden aangenaam, maar uiteindelijk is dat het niet en willen ze dat toch niet en kan je beter richtlijnen hebben,* ***hoe dit voor andere abstracte eisen?). OG en gebruiker hebben niet hetzelfde doel. Binnen Arcadis is het proces zo goed, dus zal dit niet perse gebruikt kunnen worden voor deze processen.*** *Misschien is dit idee wel een te ideale wereld…*

iii. How are you currently dealing with verification of these requirements now and what could be the changes when using an ES?

*Echt inzet van dit systeem zie ik niet zitten, moeite met aan te tonen door LOD. Nu dus vooral via richtlijnen en expertise (****maar dit is dus weer vanuit expert bekeken en wordt overlegd met OG, en niet gebruiker).***

- *Ergens lijkt het erop dat er misschien een beweging moet komen voor gebruikers op het gebied van bepaalde eisen, deze lijken te worden achtergesteld omdat geld uiteindelijk de doorslaggevende factor is binnen het ontwikkelen en ontwerpen. Eisen worden gesteld, maar voldoen aan de minimale eisen wordt als richtlijn gebruikt. OG en gebruiker totaal vanuit ander doel, maar OG gaat in gesprek met expert over invulling van eisen? Wie is nu de gebruiker? Die moet toch zijn zegje kunnen doen over wat het gebouw moet bieden voor waarde? (Ellis ten Dam).*
- *En let op want je onderzoek gaat met name over methode aantonen dus pas op dat je niet teveel op inhoud van comfort oid in gaat.*

**5. Senior Advisor Arcadis**

1) Introduction

a. What is your function within <Company>?

*Senior adviseur voor de gebouwde omgeving bij Arcadis. Projecten mee bezig vaak kantoorgebouwen en nu met groot project van Philips in Best.*

b. In which specific area do you have an expertise?

*In de processen van het formuleren van eisen en vanuit een achtergrond in de bouwfysica met name. Bouwfysica wat zich richt op onder andere comfort, met daarnaast ook akoestiek en brandveiligheid.*

c. How does this expertise relate to requirements engineering?

*In projecten constant bezig met het formuleren van deze eisen wat leidt tot PvE's. deze komen vanuit de wens van een OG die een gebouw wilt bouwen. Vanuit het PvE moeten de eisen vertaald worden naar een ontwerp. Bij deze vertaalslag en uitwerking in een ontwerp (oplossing) komen verschillende personen kijken die zich hier mee bemoeien, een er van is de adviseur (ik dus). Daarnaast architecten, constructeur en vaak als tussenpersoon tussen OG en deze adviseurs de projectmanager die voor beide partijen aanspreekpunt is.*

d. Have you ever dealt with what we called “non-functional requirements” (described as abstract, undefined and non-computable requirements) in projects?
*Zeker, deze komen natuurlijk overal terug. Dit soort eisen kunnen gesteld worden zoals je wilt, maar belangrijk is vooral dat de eisen haalbaar moeten zijn vanuit met name financieel en praktisch oogpunt.*

2) <u>Design process</u>

a. What are the difficulties with these kind of requirements from your point of view?
*Omdat we allemaal techneuten zijn, zijn we nogal snel geneigd om in getallen en oplossingen te denken. Wat vaak gebeurt is dat een eis in het PvE al zo wordt geformuleerd dat voor een bepaalde oplossing is gekozen. Dit gebeurt vaak onbewust, maar hiermee heb je dus een oplossing al en daar wordt de eis op geschreven terwijl dit juist andersom zou moeten (ook vanuit SE gedachte).*

b. In which phase of the design process (based on the ten phase scale) was dealt with these requirements?
*Idealiter zou erna elke fase een moment zijn waarbij het PvE wordt aangepast. Dus van te voren is nog niet alles te vangen en uitgewerkt, maar gedurende het ontwerp (oplossing) komen soms andere dingen naar voren, zoals conflicterende eisen. Dit zorgt natuurlijk voor moeilijkheden en kosten en planningsissues. Deze zou je willen ondervangen door het pve na elke fase na te lopen en te wijzigen, maar dit gebeurt in de praktijk natuurlijk niet.*

c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?

- Yes, how?
*Jazeker, omdat er dus nog te weinig concreet is wat betreft de invulling van deze eisen.*

3) <u>Requirements</u>

***Part 1 – Abstract requirements***

a. How is dealt with this specific type of requirements?

i. What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements?
*Belangrijkste verschil tussen specifiek geformuleerde eisen en de abstract geformuleerde eisen zit in het feit dat bij abstracte eisen niet van te voren alles kan worden vastgelegd. Een deur moet zo breed zijn is eenvoudig te gebruiken vanaf begin van het ontwerp. Maar een eis die noemt dat geluidsisolatie bepaalde waarden moet hebben bij een bepaalde ruimte kan van te voren niet zodanig bepaald worden voor een specifieke ruimte. Want deze indeling qua ruimten is nog niet bekend. Dus dan worden er eisen opgesteld voor alle mogelijkheden en dan bij een latere stap als er meer is uitgewerkt wordt deze eis van toepassing op een bepaalde ruimte. Op deze manier zit er een duidelijk verschil tussen hoe er met het soort eis dient om te worden gegaan. Daarin zit ook het werk en de moeilijkheid om deze eisen beet te pakken en in te vullen.*

ii. Which different abstract requirements are you facing during the design process?
*Veel verschillende natuurlijk. Zoals beeldkwaliteit, comfort, veiligheid ed. Maar ook veel over duurzaamheid wat tot gevolg heeft dat er vaak een bepaalde kwaliteitsklasse wordt gekozen die dan wordt bijgeschaafd (lees dingen worden er af gehaald) om zowel aan de klasse te voldoen maar wel binnen budget te blijven. Daar wordt altijd naar een balans gezocht vanuit de OG*

iii. And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common?

*Waar vaak naar toe wordt gewerkt is dat een OG bepaalde eisen/wensen/ambities heeft en deze wil zien worden ingevuld om te zorgen dat het object wordt verhuurd. Een hoger niveau betekent natuurlijk een hogere investering, maar ook een hogere huur later.* ***De OG wil een bepaald niveau vanuit gedachte/ visie en zit veel minder op de concrete invulling hiervan, als het niveau maar behaald wordt. Dus de invulling wordt niet bepaald vanuit de gebruiker…***

iv. What is the currently used main approach when you are facing such requirements during design processes?

*Zoals bij andere ook al benoemd gaat dit vaak via sessies met de OG om te kijken wat hij graag wilt qua klasse. Dit is bij eisen zoals comfort en duurzaamheid nog wel te doen.* ***Maar bij andere eisen wordt dit natuurlijk moeilijker, daar moet de architect ed. vaak proeven wat de OG dan bedoelt en hier worden dan een aantal alternatieven voor gecreëerd (vanuit bijvoorbeeld deels functionele eisen die er wel zijn, m2 en aantal personen ed.).*** *Het subjectieve karakter blijft dan natuurlijk moeilijk om voor iemand in te vullen. Dit wordt toch vaak gedaan door de OG (klant) te laten zien wat andere projecten uit ervaring hebben gebracht en of dat is wat diegene bedoelt. Bij professionele OG's wordt alleen overlegd bij conflicterende eisen. Contractueel worden wel momenten ed. Vastgelegd die dan zorgen voor het pve en vanuit daar vertaald worden in een ontwerp en later dus worden gerealiseerd.* ***Het stuk waar ik om inzoom is dus niet of de eisen voldoen aan wat het gebouw uiteindelijk is, maar of de eisen inderdaad met degene (OG?) die klant is overeenkomen.*** *Gebruikers komen bij verhuur nooit ertussen om wensen te geven, maar bij koop zijn er wel sessies waar echt naar de gebruiker wordt gekeken.* ***Deze komen wel vaak vanuit een functionele benadering (aantal verdiepingen etc.).*** *en vaak zijn het de technische eisen die toch bepaalde dingen (oplossingen) dicteren.*

***Part 2 – Comfort at work spaces***

b. What steps do you perform to deal with comfort?

i. Which expressions and relating rules do you assign to this specific abstract requirement, comfort? (think of temperature, light and sound/acoustics)

*Comfort speelt voor utiliteit altijd op 2 niveau's. enerzijds het gebouw en anderzijds de inrichting.* ***Vaak is voor kantoorgebouwen wel 1 op 1 grotendeels hetzelfde over te nemen.*** *Stoelen goed, tafel hoog/laag, staan optie. Overleggen zonder overlast → geluidseisen; absorptie, schotten, afstand.* ***Waarbij de getallen uit ervaringen en de normen komen (welke zowel proefondervindelijk als door onderzoek komen). Misschien is ES wel een manier om dit soort eisen proefondervindelijk mee te nemen in wat het voor iemand betekend???***

***Daglicht, geluidsoverlast (akoestiek)*** *van zowel collega's als van ventilatiesysteem. En ook verstaanbaarheid.*

*Materialen gebruikt (GPR)*

*Tocht*

*Kleurindex licht (heel wit vindt bijvoorbeeld bijna niemand fijn).*

***Luchtkwaliteit*** *→ geur, stank (duurzaamheidsvcertificaat WELL) over gezondheid, waar dan voor gekozen wordt blijft een spanningsveld met de OG die een insteek vanuit geld heeft voor bepaalde kwaliteit.*

***Luchtvochtigheid***

*Koudeval*

*Daarnaast is het zo dat er veel dingen onder comfort hangen, maar keuzes die gemaakt worden hangen ook af van andere dingen, bijvoorbeeld als vanuit GPR een*

*bepaald materiaal wordt gekozen dan beinvloed dit ook comfort eisen (met name heeft dit dan bepaalde consequenties voor wat er nodig is ed.)*
*Meubilair (planten (groen), kastruimte)*
*Stopcontacten/usb aansluitingen genoeg*
***Ppd/pmv → fanger over aantal ontevredenen, dit moet ook mee in onderzoek. Is gebaseerd op subjectiviteit.***

ii. Which parameters and relating values, including units, follow from this? (if Vabi Elements is used can you be specific about how you use these values from the parameter during the use of this program?)
*Uiteindelijk neemt iemand een beslissing en dit wordt voorgelegd aan de OG.* ***Gebruikers komen hierin dus niet echt terug!!*** *De waardes die eraan gehangen worden komen dus uit normen en referenties.*
*De bandbreedtes lijken wel vast te liggen in de onderzoeken ed. maar Individuele Beinvloeding is daarbij belangrijk om iedereen tevreden te houden →* ***ES kan inzicht geven hierin??***

iii. Which method are you using to perform the aforementioned expressions, rules, values and units?
*Zie vorige punt en hoe proces verloopt.*

iv. How are the choices you made justified towards the company and towards the client?
*Ja kan nooit iedereen tevreden stellen, er moet altijd een keuze gemaakt worden. Tussen man en vrouw al veel verschillen → voor invulling kies je de grootste gemene deler, welke voor comfort vastgelegd ligt in de normen (bepaalde bandbreedte).*

***Part 3 – Expert Systems***

c. How can Expert Systems support this abstract requirements definition process?

i. Are you familiar with the concept of Expert System?
- Yes, do you use it in design processes?
  *Yes.*

ii. Do you think Expert Systems can contribute to the requirements engineering process, mainly for abstract requirements, during design phases of construction projects?
*Jazeker wel, het kan zorgen voor meer betrokkenheid, op een expliciete manier. Het in beeld brengen wat de gebruiker dus wilt.*
*Maren zijn:*
*Gebruiker is nu bij projecten niet altijd in beeld.*
*En systeem moet wel verbanden kunnen leggen tussen opties (kwaliteitsniveaus). Als een bepaald niveau voor licht wordt gekozen dan moet er uiteindelijk wel een zelfde niveau voor een ander element uitkomen (bijvoorbeeld temp.) ook als dit vanuit de gebruiker niet genoemd wordt.*
*Maar ook moeilijkheid in consistentie tussen de onderdelen.*
***Via AI die een hoop pve's kent kijken of dingen overeenkomen en of het dus goed is. een gebruiker focust altijd op wat slecht is nu, en gaat er van uit dat wat eerder al goed was dat dat altijd zo blijft, dit moet daarom wel ondervangen worden in het systeem (consistentie ed.)***

iii. How are you currently dealing with verification of these requirements now and what could be the changes when using an ES?
*Op dit moment gebeurt de verificatie van dit soort eisen vaak vanuit relatics waarbij eisen en verificatiemethode worden genoemd vanuit de vraagspecificatie. En het blijft toch een lerend beroep waarbij klachten. input geven hoe het de volgende keer anders en dus beter kan.* ***Dit kan bijvoorbeeld de inzet van ES doen tijdens sessies.***

*Er wordt namelijk uiteindelijk toch toegewerkt naar **controleerbaarheid en meetbaarheid van de eisen.***

- *Andere dingen voor het ES;*
    - *Gebruikersafhankelijkheid; ze moeten wel antwoorden vanuit een bepaalde ervaring (zie consistentie met wat ze al hebben en wat ze volgende keer krijgen)*
    - *Activiteitsafhankelijk; meerdere ruimtes betekent meerdere keuzes. Je kan niet het hele gebouw over 1 kam scheren en dus moet je wel bepaalde eisen aan bepaalde ruimten hangen en niet geldend voor het hele gebouw. Als je verstaanbaar wilt telefoneren geldt dit niet voor een ruimte waar je samen zit te werken, maar voor een telefooncel oid.*
        - *Er zijn dus grenzen voor gebruik van de eisen.*
    - *Reverse engineering: eis die specifiek is (waardes) terig schrijven tot bepaalde omschrijving → iets wat ik dus ook doe om via vragen tot waardes die individueel geldend zijn.*
        - *Stukje wat ik uitwerk en iemand invult moet dus met norm oid. overeenkomen.*
        - *En de inzet van ES / bijdrage voor dit stuk proces laten zien.*
            - *Voor zowel dit voorbeeld als de uitrol naar andere eisen en de totale inzet hiervan (validatie PoC).*
    - *Je kan inderdaad op twee manier kijken welke waarden komen eruit, maar dit wordt vooral bekeken nu dus vanuit we hebben deze waarden, welke omschrijving hoort daarbij dus wat moeten we vragen:*
        - *Temperatuuroverschrijdingsuren → hoeveel wil je die, dan betekent dat deze waardes voor installaties ed. als voorbeeld.*
    - *Vooral interessant hoe deze dingen in het ES komen te zitten en hoe dit gedaan dient te worden.*

### 6. Senior Consultant Corporate Real Estate Arcadis

1) Introduction
    a. What is your function within <Company>?
    *Huisvestingsadviseur en daarmee het helpen en begeleiden van de OG bij de vraag. Middel om business te laten verlopen en te onderzoeken.*
    b. In which specific area do you have an expertise?
    *In het begeleiden en helpen van de klant. Welke mogelijke keuzes er te maken zijn en in het gesprek met de klant welke wat er bij past. Er moet altijd balans worden gezocht.*
    c. How does this expertise relate to requirements engineering?
    *Je probeert om via de gesprekken echt harde eisen vast te leggen.*
    d. Have you ever dealt with what we called "non-functional requirements" (described as abstract, undefined and non-computable requirements) in projects?
    *Jazeker, hier gaan vaak de gesprekken over om deze helder te krijgen. Maar het blijft dat 100% tevreden nooit lukt.*

2) Design process
    a. What are the difficulties with these kind of requirements from your point of view?
    *Dat het vaak heel persoonlijk is voor een klant.*
    b. In which phase of the design process (based on the ten phase scale) was dealt with these requirements?
    *Doorlopend moet worden omgegaan met deze eisen om ze invulling te geven.*
    c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?
        - Yes, how?
        *In het begin is nog niks bekend van een dergelijke eis en gedurende het Ontwerp moet je hier achter zien te komen.*

3) <u>Requirements</u>

***Part 1 – Abstract requirements***

a. How is dealt with this specific type of requirements?

   i. What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements?
   *Het gaat over mensen en de eisen zijn vaak heel persoonlijk.*

   ii. Which different abstract requirements are you facing during the design process?
   *Comfort, duurzaamheid, etc.*

   iii. And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common?

   iv. What is the currently used main approach when you are facing such requirements during design processes?
   *Alle opties laten zien adhv referenties (bij andere OG's) en gebruikersgroepen naar gaan kijken. Ook gebaseerd op ervaring. Klachten dienen als leidraad en concurrenten ook. Wat willen ze en waarom?*
   *Vooral de verwachtingen in gesprek met OG en/of gebruiker goed duiden. Wat moet beter? En daarbij is het referentiekader zoals het nu is. Klasse wordt bepaald door middel van wat we nu hebben en wat we straks willen.*
   *Op verschillende aspecten kunnen er verschillende keuzes worden gemaakt. Alle aspecten moet je nalopen.*
   *Wat je gaat doen is ook van invloed voor eisen.*
   *De basis zijn de NEN normen en hierin neem je OG mee en probeer je het tastbaar te maken. En dit alles benoem je in het PvE. En de gestelde uitgangspunten neem je mee om zoveel mogelijk de juiste richting in te gaan.*
   *Gesprekken gaan als volgt:*

   1. *Gesprek met directie (uitgangspunten;tijd;budget en bepaald tot welk niveau je kan gaan).*
   2. *Via dat kader naar gebruikers (levert t echt iets op of is het alleen voor de gemoedstoestand?)*
   3. *Daarna ga je met OG het pve doorlopen → en laten zien waar je voor kiest en waar voor niet. Hangt ook af van wie OG is.*
   ***Input komt van OG en gebruiker, maar OG beslist en de adviseur moet hierin snappen wat ze willen.***

***Part 2 – Comfort at work spaces***

b. What steps do you perform to deal with comfort?

   i. Which expressions and relating rules do you assign to this specific abstract requirement, comfort? (think of temperature, light and sound/acoustics)
   *Je probeert aan de normen te voldoen omdat dit in ieder geval bewijst dat je aan de normen voldoet. Is niet mogelijk om met alle gebruikers te bespreken en het is heel persoonlijk.*

   ii. Which parameters and relating values, including units, follow from this? (if Vabi Elements is used can you be specific about how you use these values from the parameter during the use of this program?)
   *Daglicht*
   *Uitzicht*
   *Faciliteiten (makkelijk bereikbaar, goed eten/drinken)*
   *IT (stabiel, snel, makkelijk)*

*Beleving*
*Geluid*
*Temperatuur*
***Vanuit normen naar waardes ed.***

iii. Which method are you using to perform the aforementioned expressions, rules, values and units?
*Gesprekken, referenties, ervaringen.*

iv. How are the choices you made justified towards the company and towards the client?
*Doordat klant is meegenomen in het specificeren.*

***Part 3 – Expert Systems***

c. How can Expert Systems support this abstract requirements definition process?

i. Are you familiar with the concept of Expert System?
- No, (explanation)
  *No*

ii. Do you think Expert Systems can contribute to the requirements engineering process, mainly for abstract requirements, during design phases of construction projects?
*Technische oplossingen vanuit gebruikers wordt al gedaan. Experts kunnen wel bedenken wat de oplossing moet zijn vanuit de waardes.* ***Maar gebruikers zetten altijd hoog in, hoe kunnen zij heel bewust keuzes maken. Het kan dus bijdragen in het vergemakkelijken van gesprekken. Maar wat willen ze nu echt en wat is daar voor nodig? Dit levert prijsverschillen op.***
*Bepaalde verwachtingen creeeren wat iemand krijgt.* ***Bijvoorbeeld 39-33dB scheelt een ton, wat betekent dit dan. Zijn hele moeilijke discussies wat t betekent. Je moet wel echt een expert zijn om te kunnen vertalen.***
***Wat betekent een bepaalde term nu?***
- ***Veel praten***
- ***Laten zien***
- ***Anderen***

***Veel oplossingen in PvE, alles raken wat er mee te maken heeft.***

iii. How are you currently dealing with verification of these requirements now and what could be the changes when using an ES?
*Persoonlijk, functie, nooit iedereen tevreden, bewuster keuzes maken.*

### 7. Research Scientist TNO

1) Introduction

a. What is your function within <Company>?
*Research scientist; op gebied van energie en klimaat (geluid, lucht en licht).*

b. In which specific area do you have an expertise?
*Het monitoren van of de prestatie wordt gehaald.*

c. How does this expertise relate to requirements engineering?
*Er zit een verschil tussen wat je verwacht te halen vanuit de eis en wat het echt is, dit komt door aannames. Je wilt een eigen controle hebben, niet perse alleen aan een norm hangen (zoals Fanger, uit jaren 70/80 en alleen maar witte jonge mannen gemeten).*

d. Have you ever dealt with what we called "non-functional requirements" (described as abstract, undefined and non-computable requirements) in projects?
*Onderzoek doen naar parameters; parameters worden gekozen op basis van literatuur en onderzoek. Welke moet je pakken? Hoe beoordelen? Hoe berekenen?*

2) Design process

a. What are the difficulties with these kind of requirements from your point of view?

*Gaat over mensen en die zijn allemaal anders (meeste zullen iets wel prettig vinden, maar niet iedereen). Daarom deze mensen er ook bij betrekken wat via tool kan.*

b. In which phase of the design process (based on the ten phase scale) was dealt with these requirements?

*Het beste is om de gespecificeerde waarden al aan de voorkant te weten. En belangrijk is om een goed beeld van gebruikers te hebben omdat iedereen iets anders wilt.*

c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?

- Yes, how?
- No, why not?

3) Requirements

***Part 1 – Abstract requirements***

a. How is dealt with this specific type of requirements?

i. What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements?

ii. Which different abstract requirements are you facing during the design process?

iii. And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common?

iv. What is the currently used main approach when you are facing such requirements during design processes?

*Manco zit in het feit dat de OG gaat investeren en de huurder bepaalde eisen zou willen hebben. Liefste heb je ze allemaal om tafel, maar dit gebeurt zo niet. Huurder en OG hebben allebei een andere invalshoek en zitten niet op hetzelfde vlak.*

***Part 2 – Comfort at work spaces***

b. What steps do you perform to deal with comfort?

i. Which expressions and relating rules do you assign to this specific abstract requirement, comfort? (think of temperature, light and sound/acoustics)

*Geluid, temperatuur en lucht, licht.*

ii. Which parameters and relating values, including units, follow from this? (if Vabi Elements is used can you be specific about how you use these values from the parameter during the use of this program?)

iii. Which method are you using to perform the aforementioned expressions, rules, values and units?

iv. How are the choices you made justified towards the company and towards the client?

***Part 3 – Expert Systems***

c. How can Expert Systems support this abstract requirements definition process?

i. Are you familiar with the concept of Expert System?

- No, (explanation)

  *No*

ii. Do you think Expert Systems can contribute to the requirements engineering process, mainly for abstract requirements, during design phases of construction projects?

*Mensen bewuster maken wat ze willen en kunnen bereiken, wat iets betekent en daarbij specifieke wensen obv activiteiten in kaart brengen kan handig zijn.*

iii. How are you currently dealing with verification of these requirements now and what could be the changes when using an ES?

### 8. Building Physics Advisor RoyalHaskoningDHV

1) Introduction
   a. What is your function within <Company>?
      *Adviseur BPS. Op het gebied van bouwfysica, akoestiek en duurzaamheid (Breeam en Leed waaronder dus ook comfort valt). En onder bouwfysica vallen thermisch comfort, licht, warmte comfort en stedenbouwfyisch comfort (windhinder).*
   b. In which specific area do you have an expertise?
      *Bovenstaand.*
   c. How does this expertise relate to requirements engineering?
      *Constant bezig om deze eisen invulling te geven.*
   d. Have you ever dealt with what we called "non-functional requirements" (described as abstract, undefined and non-computable requirements) in projects? *Ja daar ben je eigenlijk altijd wel mee bezig.*

2) Design process
   a. What are the difficulties with these kind of requirements from your point of view?
      *Niet gedefinieerd wat het voor iemand betekent.*
      *Uitgangspunten waaronder het moet gelden*
      *Heel veel verschillen tussen mensen obv culturen en geografie, iedereen heeft een andere beleving.*
      *Afhankelijk van gebruikers*
   b. In which phase of the design process (based on the ten phase scale) was dealt with these requirements?
      *Alle fases.*
   c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?
      - Yes, how?
        *Yes*

3) Requirements
   ***Part 1 – Abstract requirements***
   a. How is dealt with this specific type of requirements?
      i. What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements?
         *See difficulties. Ongrijpbaar.*
      ii. Which different abstract requirements are you facing during the design process?
         *Energiezuining, energieneutraal, healing environment, stil zijn/niks kunnen horen. Het zijn allemaal containerbegrippen* ***wanneer doe je het goed?***
      iii. And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common?
      iv. What is the currently used main approach when you are facing such requirements during design processes?
         *Gesprek klant (wat nu niet goed → en wat moet anders?)*
         *Wat wil de klant hebben – handboek / norm schrijft een hoog kwaliteitsniveau voor, maar dat lukt niet met het budget van de OG*
         ***Zelf invulling geven gebaseerd op kennis en ervaring bij het afwijken van de normen. (vaker genoemd).***

*Er wordt een technisch PvE geschreven en dat moet de OG goedkeuren, Maar hij snapt de getallen helemaal niet (dus vaak wordt er dan met* ***kleurcodering*** *gewerkt). OG zal altijd voor beste gaan, dus groen, maar dit is wel duur*

- *Of het zijn niet te realiseren eisen*
- *Of conflicterende eisen*

*Ontwerpteam spreekt vaak OG niet maar op basis van documenten worden er varianten ontwikkeld. Lastig om naar de beste oplossing te zoeken. Wat is het uiteindelijke resultaat?* ***Niet alle eisen worden met klanten doorgenomen (alleen afwijkingen ed.) Tijden SO en VO wordt er nog niet gerekend, maar naarmate er steeds meer bekend is en het ontwerp specifieker wordt wordt er ook meer uitgerekend → maar dan komen de afwijkingen en conflicterende eisen. Hoe concreter ontwerp, hoe meer het begint te leven (maar afwijkingen wil je eigenlijk al aan de voorkant vangen zodat je niet hoeft te wijzigen, maar dan moet je de OG wel duidelijk begrijpen).***

***Part 2 – Comfort at work spaces***

b. What steps do you perform to deal with comfort?

i. Which expressions and relating rules do you assign to this specific abstract requirement, comfort? (think of temperature, light and sound/acoustics)
*NEN normen – thermisch comfort, visueel comfort, akoestisch comfort. Begin van ontwerp al rekening houden met klasse en de uitgangspunten. Pve soms aangeleverd en soms zelf geschreven (icm klantsessies). En soms functionele pve vertalen naar technisch pve.*

ii. Which parameters and relating values, including units, follow from this? (if Vabi Elements is used can you be specific about how you use these values from the parameter during the use of this program?)
*Zoninstraling*
*Temp*
*Straling gevel*
*Luchtstraling*
*Daglicht*
*akoestiek*

iii. Which method are you using to perform the aforementioned expressions, rules, values and units?
*Vertaalslag wordt gemaakt obv referenties (nu – straks) en bekijken met de klant, visualisaties en aurilisaties (voorbeelden hoe het is)*
- *Waar komt eis van klant vandaan?*

iv. How are the choices you made justified towards the company and towards the client?
*Klant weet vaak niet beter; waarom komt vanuit een eerdere slechte ervaring.*
***Wat je vaak ziet is dat bij BB eis van geluid bijvoorbeeld 25% niet tevreden is en thermisch comfort bijvoorbeeld 10%.***
***Klant is onervaren en een duidelijk verschil tussen een professionele OG en degene die 1 keer iets doen***
***Soms zorgt een eis dat er meer gerealiseerd wordt dan gevraagd. OG hangt wet aan, maar dan is bewoner niet tevreden. Komt door verschil tussen personen die betalen en degene die afneemt. (OG en gebruiker). Vaak komt ook voor dat een OG eis te duur vindt en dan moet het later alsnog als blijkt dat gebruiker het niet goed vindt. De een geeft hier meer om dan de ander, maar OG oid. is wel degene die financiert. Gebruikers worden vaak helemaal niet meegenomen in proces!!***

***Part 3 – Expert Systems***

c. How can Expert Systems support this abstract requirements definition process?

i. Are you familiar with the concept of Expert System?

- No, (explanation)

  *No*

ii. Do you think Expert Systems can contribute to the requirements engineering process, mainly for abstract requirements, during design phases of construction projects?

*Contributions:*

*Minder discussie*

*Zelfde soort klanten met elkaar indentificeren*

*Terugkoppeling eruit halen*

*Vragen wat ze het belangrijkste vinden; hoewel keuze ook kan komen doordat iets niet prettig is op het moment dat ze het daarom belangrijker vinden.*

*Challenges:*

*Mensen kiezen altijd het beste (**dus vragen dat ze niet weten wat het beste is?)***

***Je mist terugkoppeling (geen klachten dan zal het wel goed zijn, maar is natuurlijk niet altijd zo) → KOMT DOOR GELD DAT DIT NIET IN BOUW GEDAAN WORDT EN OMDAT ELK PROJECT WEER ANDERS IS (interview 8 en 10).***

***Enqueteren om tevredenheid te meten***

***Als er een dingetje niet goed is binnen een eis kan hele eis niet goed worden, terwijl 90% wel goed zal zijn.***

***Hoe alles met elkaar in evenwicht brengen***

***Geld op juiste plek uitgeven.***

***Situaties nooit 1 op 1 over te nemen***

***Missen van evaluatie: was eis goed, te licht (klachten) of te zwaar (te duur). Dan kan advies helpen om overdimensioneren te voorkomen.***

***X aantal gebouwen -> voldoet aan bepaalde eis -> zoveel mensen tevreden dan is deze eis dus goed voor dit type gebouw en gebruiker.***

iii. How are you currently dealing with verification of these requirements now and what could be the changes when using an ES?

*Rekenmodellen/simulaties en waardes die iets over uitkomst zeggen zoals Temperatuuroverschrijdingsuren.*

*Eis is pas gehaald op het moment dat het gebouw is opgeleverd. Model is nooit zelfde als werkelijkheid. Er komt steeds meer verificatie, voor jezelf al kijken in model of dat het straks zal voldoen.*

## 9. Advisor Building Physics BAM

1) <u>Introduction</u>

a. What is your function within <Company>?

*BPS, w-adviseur, steeds meer richting klantwens vertalen voor PvE.*

b. In which specific area do you have an expertise?

*Vooral bevragen en hierna komt de stap naar het Ontwerp.*

c. How does this expertise relate to requirements engineering?

*Om naar een meetbare waarde te gaan.*

d. Have you ever dealt with what we called "non-functional requirements" (described as abstract, undefined and non-computable requirements) in projects?

*Mooi/lelijk, duurzaam, comfortabel, veiligheid. Een bepaalde klasse kan nooit voor de hele eis gelden omdat dit te duur is, dus moet daar nog een slag in worden gemaakt en dus kan dit ook gezien worden als nog steeds een abstracte eis. Veel zit in de tekst die wordt beschreven.*

2) <u>Design process</u>

a. What are the difficulties with these kind of requirements from your point of view?

*Moeilijkheid zit in het feit dat het veel kan bevatten en wat wordt er precies bedoeld. Maar ook dat klant iets anders in gedachten had, doordat hij het niet goed heeft begrepen, onreeele eis ON.*

b. In which phase of the design process (based on the ten phase scale) was dealt with these requirements?

*Advies is handig om tot meetbare waarden te komen, maar aannemer zit veel meer op de verificatie hiervan.*

c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?

- Yes, how?

  *Yes, vaak bij projecten niet met klant aan tafel. Er wordt gereageerd via documenten. Elke fase wordt dit opnieuw gedaan. En wordt aangegeven of de waarde meetbaar is en dus acceptabel is om mee door te gaan. Zo niet dan wordt er overlegd met de klant.* ***Hierin wordt de klant dan begeleid wat het betekent en kostentechnisch wat het inhoudt.***

3) <u>Requirements</u>

***Part 1 – Abstract requirements***

a. How is dealt with this specific type of requirements?

i. What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements?

*Dat zit in het vertaalproces. Je moet duidelijk krijgen welke waarde de OG wil aanhangen. Voor dat je aan de slag gaat moet dit duidelijk zijn. Dit is niet altijd mogelijk. Klant denkt soms dat het voor elkaar is, maar dat is dan niet zo. Bestek en pve komen niet overeen (door adviseur). Omdat eisen dus niet goed genoeg benoemd en gespecificeerd zijn.*

ii. Which different abstract requirements are you facing during the design process?

*Mooi/lelijk, duurzaam, comfortabel, veiligheid.*

iii. And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common?

iv. What is the currently used main approach when you are facing such requirements during design processes?

*Niet iedereen geeft aan dat niet alles voldoet. Dan is er wel een goed gebouw neergezet, maar niet aan alles eisen is voldaan. Op het einde komt het dan niet goed met de verificatie.* ***Vraag is of de eis niet klopt in zo'n geval?***

***Part 2 – Comfort at work spaces***

b. What steps do you perform to deal with comfort?

i. Which expressions and relating rules do you assign to this specific abstract requirement, comfort? (think of temperature, light and sound/acoustics)

*Vanuit normen en ervaring; keuzes zelf maken wat je zelf goed vindt. Verschillende normen en daar kies je er zelf een uit.*

ii. Which parameters and relating values, including units, follow from this? (if Vabi Elements is used can you be specific about how you use these values from the parameter during the use of this program?)

*Luchtsnelheid*

*Stralingsassymmetrie*
*Luminantie verhouding*
*Contrast*
***Niet alles wordt op getoets omdat sommige dingen uit normen worden overgenomen.***
*Temperatuur*
*Geluid*
*Lichtwering*
*IB = wassen neus (per 4/6 personen en komt niemand aan) en is ruimte afhankelijk dus schiet zijn doel voorbij.*
*Flexfactor*
*Koudeval*
***Als het gebruik weer verandert dan kloppen de eisen niet meer, dus flexfactor hoog is oplossing om verschuivingen in gebruik te ondervangen.***

iii. Which method are you using to perform the aforementioned expressions, rules, values and units?
*Vanuit ervaring en normen; maar vooral goed bevragen moet zorgen dat het juiste wordt gekozen.*
*Wat vindt de klant nu echt belangrijk? En waarom? En hier van leren obv gebruikersafhankelijkheid en conceptontwikkeling. Klant laten ervaren wat hij wil.*

iv. How are the choices you made justified towards the company and towards the client?
*Visueel maakt veel goed. Maar hoe leg je uit aan de klant als klasse B ook goed is, maar A beter maar kost meer geld waarvoor hij moet kiezen? En klachten die later komen daar ben je altijd te laat mee om er nog iets aan te doen eigenlijk.*

***Part 3 – Expert Systems***

c. How can Expert Systems support this abstract requirements definition process?

i. Are you familiar with the concept of Expert System?
- No, (explanation)
*No*

ii. Do you think Expert Systems can contribute to the requirements engineering process, mainly for abstract requirements, during design phases of construction projects?
*Systeem kan het proces niet vervangen, maar tussen fasen wordt er steeds opnieuw gevraagd naar de eisen. En hoe deze keuze is gemaakt (vanuit OG) weet je niet, maar kan wel worden vastgelegd. Zodat er van geleerd kan worden. Uitkomsten van de een zijn anders dan bij de ander en waarom zijn bepaalde keuzes gemaakt?* ***Adviseur doet ook zijn best, maar niet altijd de tijd voor om eindig over door te blijven gaan en dan worden soms keuzes gewoon gemaakt.***
*Belangrijk is:*
- ***Manier van vragen***
- ***Vtv al budget weten***

***Contribution: Maakt discussie duidelijker | Minder kosten/wijzigen | Automatiseren en effectiever***

***Challenges: Data aan elkaar koppelen | Tekst bevragen, maar schort aan beleving ervan***

iii. How are you currently dealing with verification of these requirements now and what could be the changes when using an ES?

### 10. Advisor Installations AvecodeBondt

1) Introduction
   a. What is your function within <Company>?
      *Installatie adviseur – om installatie advise model op te zetten. Daarvoor eigen bedrijf in BIM. Sinds bouwenquete is bestek slecht. Bestek klopte nooit en databases BIM klopt ook niet wat betreft de installaties.*
   b. In which specific area do you have an expertise?
      *Commissioning: begin prestatie vaststellen die je aan het einde wilt bereiken op basis van:*
      - *Geld*
      - *Organisatie*
      - *Tijd*
      - *Informatie*
      - *Kwaliteit*
        *>> komt vanuit systematiek ruimtevaart en vliegtuigindustrie. Anders hebben de ontwerpers de uitgangspunten niet meer op hun netvlies.*
   c. How does this expertise relate to requirements engineering?
      *Daar gaat het om, om tot die uiteindelijke prestaties te komen.*
   d. Have you ever dealt with what we called "non-functional requirements" (described as abstract, undefined and non-computable requirements) in projects?

2) Design process
   a. What are the difficulties with these kind of requirements from your point of view?
      *Abstracte onderwerpen wat het moeilijk maakt om te beleven. En specifieke eisen zijn rekentechnisch wel te bewijzen. Wat is de definitie van iets? Hangt af van de vragen die aan de klant worden gesteld. Vragen stellen aan klant om verweven patroon helder te krijgen. Klant weet het beste te omschrijven via voorbeelden.* ***Verwachting van de klant****.*
   b. In which phase of the design process (based on the ten phase scale) was dealt with these requirements?
      *Bijna altijd vanaf de beginkant van het proces.*
   *c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?*
      - Yes, how?
        *Yes, je kan het nog zo goed doen als een partij erna fouten maakt dan gaat t uiteindelijke resultaat alsnog mis.*
      - No, why not?

3) Requirements
   ***Part 1 – Abstract requirements***
   a. How is dealt with this specific type of requirements?
      i. What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements?
         *Abstracte onderwerpen wat het moeilijk maakt om te beleven. En specifieke eisen zijn rekentechnisch wel te bewijzen.*
      ii. Which different abstract requirements are you facing during the design process?
         *Comfort en klimaat vooral mee bezig.*
      iii. And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common?

*Abstracte eisen hebben een softe en harde kant. De softe kant kan door communicatie benaderd worden en de harde kant door te berekenen en te toetsen.* ***De softe kant kan je ook benaderen door enquetes, wat verwacht iemand van een eis oid?***
*En ook definitie, waarom, reden die iemand noemt daarvoor is vaak anders dan wat ie in begin noemde.* ***Heeft ie het geld er voor over.*** *Geld speelt altijd een rol, bij elk besluit.*

iv. What is the currently used main approach when you are facing such requirements during design processes?

1. *Functie van ruimten (spaces maken) daaraan worden de vastgesteld eisen opgehangen en worden er simulaties gemaakt. Deze valideren de waarden aan het latere bim model (=bim validatie model).*
2. *Wat gebeurd daar? (activiteit)*
3. *Eis vaststellen*
4. *Waarvoor wil je die ruimte gebruiken?*

*Hierna eis valideren en verifieren.*
*Je moet je verdiepen in de klant anders kan je nooit de verwachtingen waarmaken.* ***'tevredenheid is perceptie minus verwachting'.***
***Waar wordt de eis door beinvloed? Aan de hand van die factoren ga je bepalen (invloedsfactoren -> waarde aangeven).***
***Je gaat de eis afpellen tot invloedsfactoren, en hiervan kan je sommige berekenen en andere niet (subjectieve). Door de juiste vragen te stellen kom je wel achter wat dit inhoudt voor iemand.***
***Dan bied je mogelijkheden aan (oplossingen voor die waarde/eis) en welke invloedsfactoren ondervang je hiermee? Mogelijkheden komen voort uit marktkennis.***
***Vragen stel je op een manier waarin een eis(onderdeel) specifiek kan worden gemaakt. Als je dat vraagt dan wil je weten of een klant daar waarde aan hecht.***
***Alleen adviseren waar je zelf ook achter staat.***

- ***Wat voor niveau verwacht je?***
- ***Wat is prettig?***
- ***War wil je ervaren?***
  - ***Normen niet mee lastig vallen; waar wil de klant dat er moet worden afgeweken van de norm? Wat verwacht ie er dan van?***

*Een gebruiker heeft altijd de keus om een gebouw van een OG te kiezen, je hebt altijd communicatie met degene die de beslissingen maakt (OG).*

***Part 2 – Comfort at work spaces***

b. What steps do you perform to deal with comfort?

i. Which expressions and relating rules do you assign to this specific abstract requirement, comfort? (think of temperature, light and sound/acoustics)
*Verlichting*
*Temperatuur*
*Luchtkwaliteit*
*Geluid*
*Zitten stoel*
*Beweegbaarheid*
*Werkende systemen*
*ergonomie*

ii. Which parameters and relating values, including units, follow from this? (if Vabi Elements is used can you be specific about how you use these values from the parameter during the use of this program?)
*Afhankelijk van gebruik en activiteit. Normen veranderen wel.*

iii. Which method are you using to perform the aforementioned expressions, rules, values and units?
*Hecht iemand heel veel waarde aan een bepaalde eis dan PMV laag houden zodat er minder mensen ontevreden zijn. 100% lukt nooit.*

iv. How are the choices you made justified towards the company and towards the client?
*Helder te krijgen door te vragen, meenemen naar voorbeelden/beeldmateriaal en meegaan met de klant bij keuzes maken.*

***Part 3 – Expert Systems***

c. How can Expert Systems support this abstract requirements definition process?

i. Are you familiar with the concept of Expert System?
- No, (explanation)
*No*

ii. Do you think Expert Systems can contribute to the requirements engineering process, mainly for abstract requirements, during design phases of construction projects?
*Experts weten vtv al wat het gaat zijn en waar ze op uit gaan komen, dus echt hulpmiddel is niet de oplossing. Wel handig als leerinstrument om te kijken welke vragen er gesteld moeten worden en welk effect dat in een waarde teweeg brengt.*
***In medische wereld zijn ze bezig met probleemanalyse (=problem discovering Ellis ten Dam?). bouw doet niet aan evalueren en leren van vorige projecten en keuzes omdat altijd andere mensen bij projecten zijn en het spel steeds anders is.***
***Challenges:***
***Je mist interactie***
***Om bij klant goede gevoel te zien zie je niet op deze manier***
***Om snel eruit te halen het waarom?***

iii. How are you currently dealing with verification of these requirements now and what could be the changes when using an ES?

**11. Architect Arcadis**

1) <u>Introduction</u>

a. What is your function within <Company>? *ontwerpleider*

*b.* In which specific area do you have an expertise? *Architectuur stations civiel*

c. How does this expertise relate to requirements engineering? *Make requirements and use them.*

*d.* Have you ever dealt with what we called "non-functional requirements" (described as abstract, undefined and non-computable requirements) in projects? *Yes every day*

2) <u>Design process</u>

a. What are the difficulties with these kind of requirements from your point of view? *They are subjective*

*b.* In which phase of the design process (based on the ten phase scale) was dealt with these requirements? *All phases*

c. Is the specific design phase mentioned, affecting the difficulties of dealing with these requirements?

- *Yes, all phases are different and very specific.*

3) <u>Requirements</u>

***Part 1 – Abstract requirements***

a. How is dealt with this specific type of requirements?

   *i.* What are the differences between these and the specific, well-defined, unambiguously and quantifiable formulated requirements, like building code requirements*? You need a method, a proces and a specialist to verify and validate results*

   ii. Which different abstract requirements are you facing during the design process? *Architecture (beauty) / comfort / usability / sustainability / (social) safety*

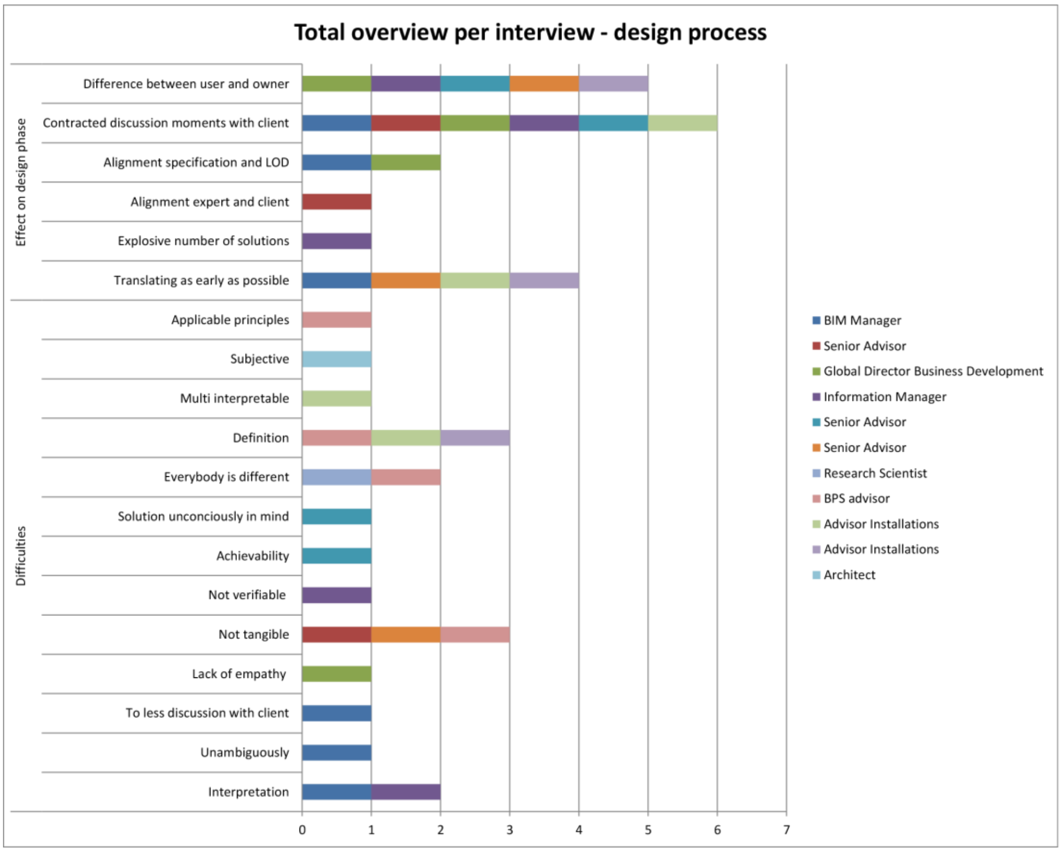
   *iii.* And what are the characteristics of the aforementioned abstract requirements? Do they have characteristics in common*? You need a method, a proces and a specialist to verify and validate results*

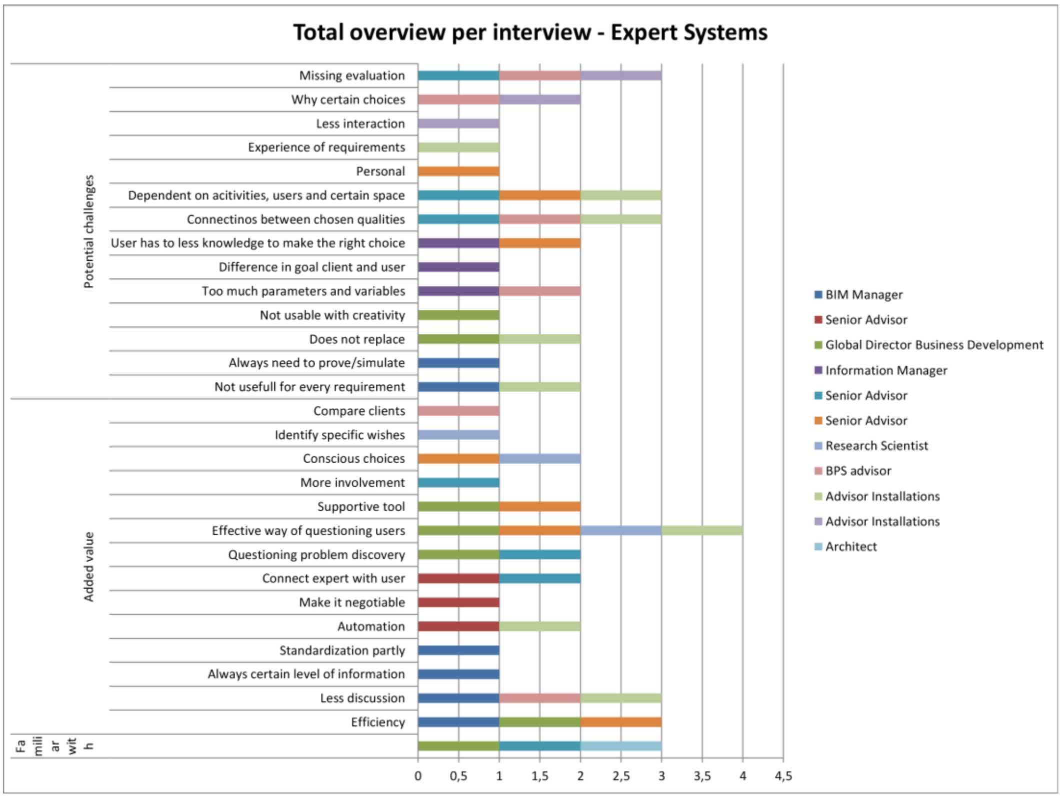
   iv. What is the currently used main approach when you are facing such requirements during design processes? *Using a method, a proces and a specialist to verify and validate results*

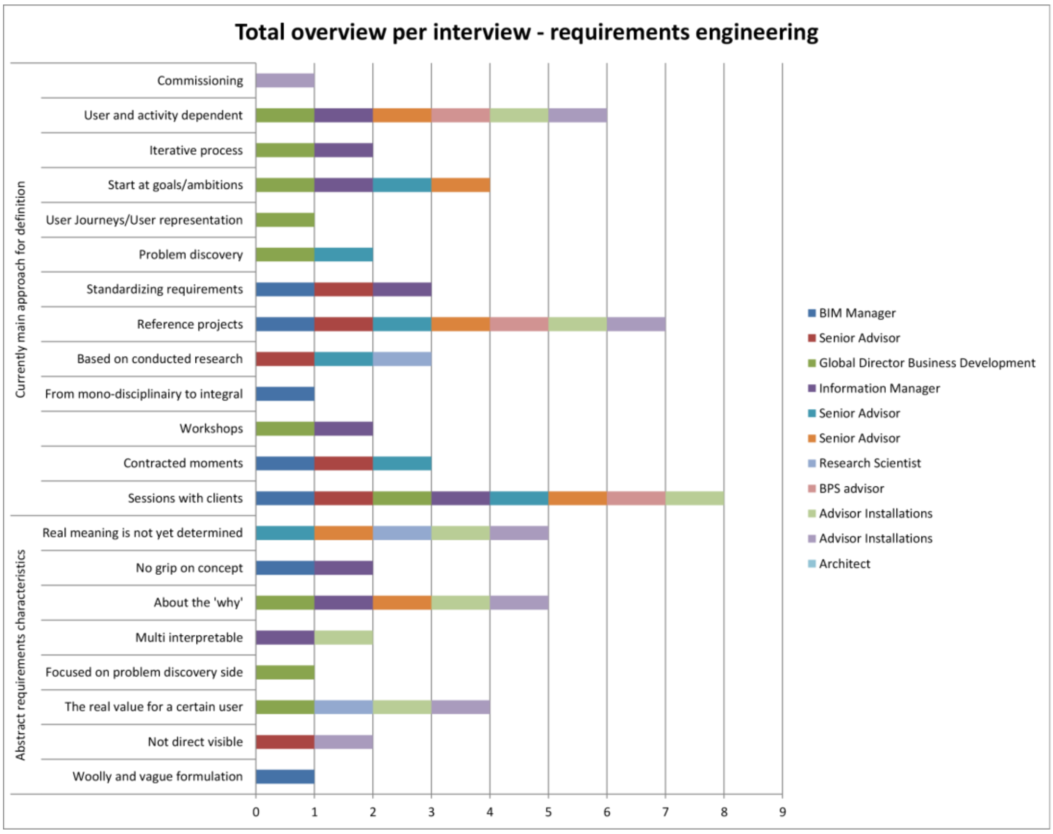
The part 2 of your form about "comfort" is in my eyes not so relevant to fill in because comfort is just an example of dealing with a very specific topic.

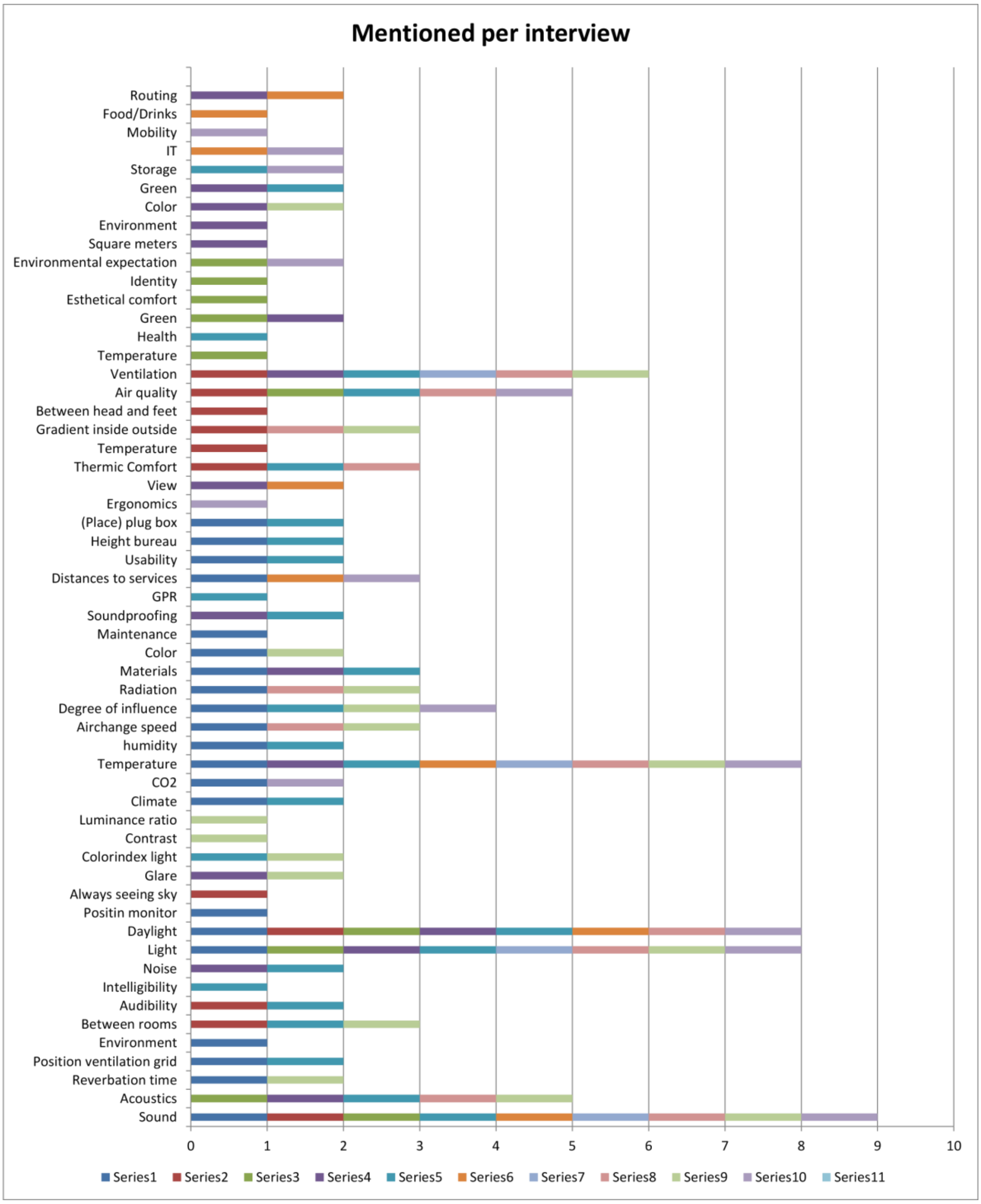
*"Expert systems" sounds to me as the normal way I am used to work with these type of topics; we are using a method, a proces and a specialist to verify and validate results. We use technical requirements and process requirements for contracting these topics.*

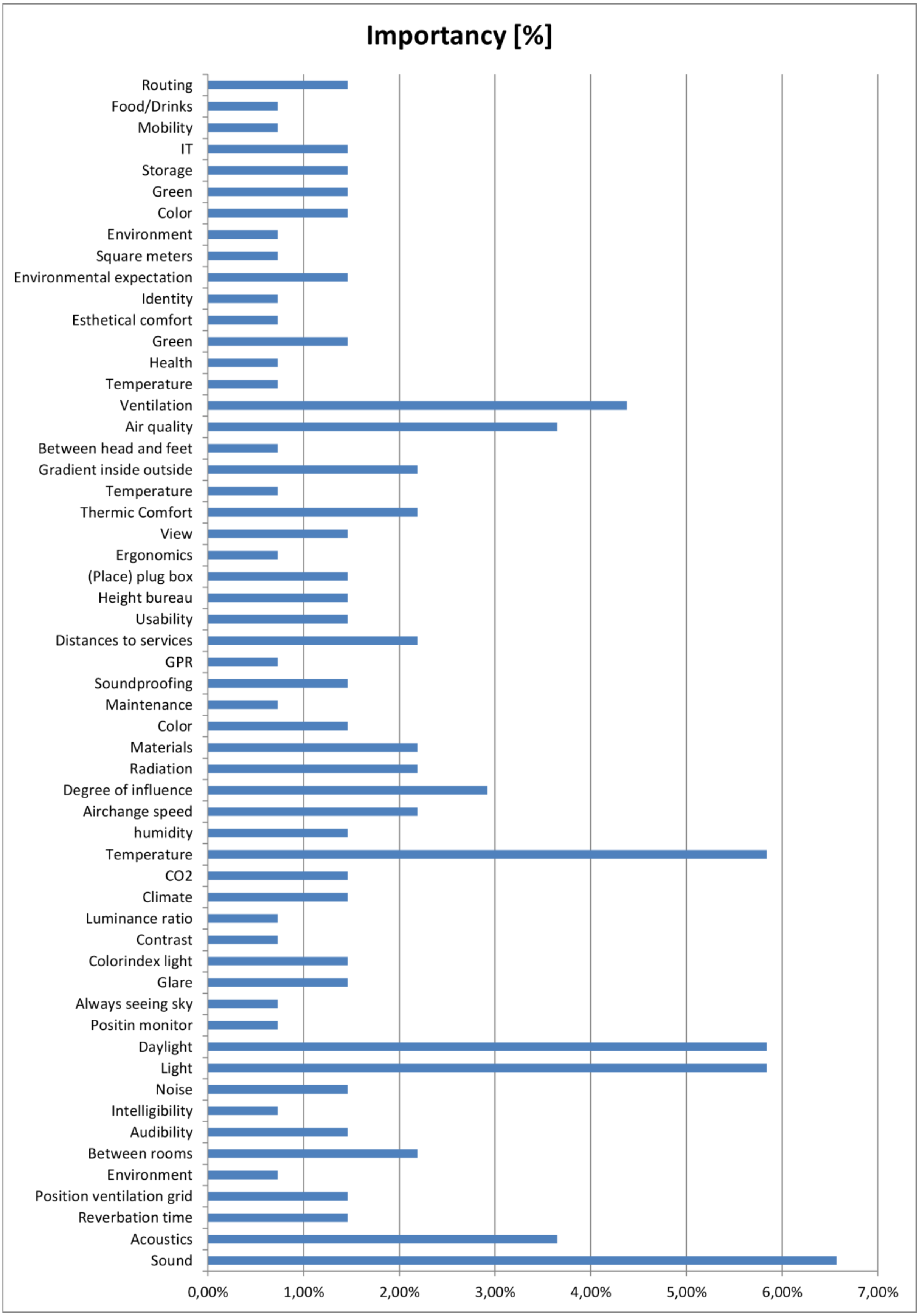
## 9.3 Interview outcomes

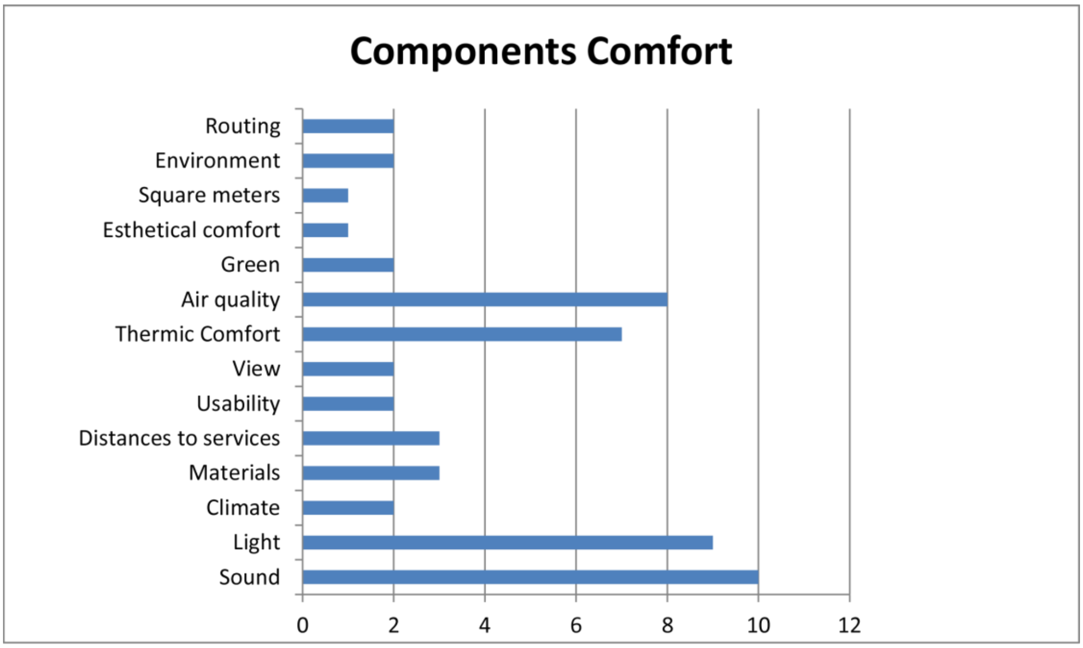


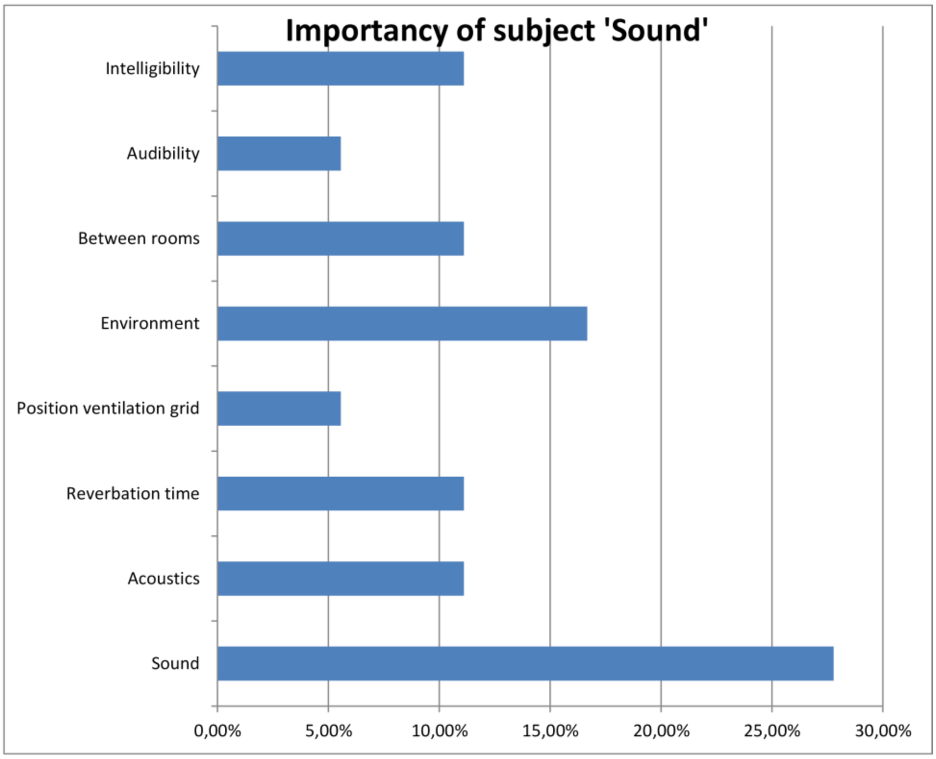




Total overview per interview - requirements engineering
Currently main approach for definition
Commissioning
User and activity dependent
Iterative process
Start at goals/ambitions
User Journeys/User representation
Problem discovery
Standardizing requirements
Reference projects
Based on conducted research
From mono-disciplinairy to integral
Workshops
Contracted moments
Sessions with clients
Abstract requirements characteristics
Real meaning is not yet determined
No grip on concept
About the 'why'
Multi interpretable
Focused on problem discovery side
The real value for a certain user
Not direct visible
Woolly and vague formulation
0
1
2
3
4
5
6
7
8
9
BIM Manager
Senior Advisor
Global Director Business Development
Information Manager
Senior Advisor
Senior Advisor
Research Scientist
BPS advisor
Advisor Installations
Advisor Installations
Architect

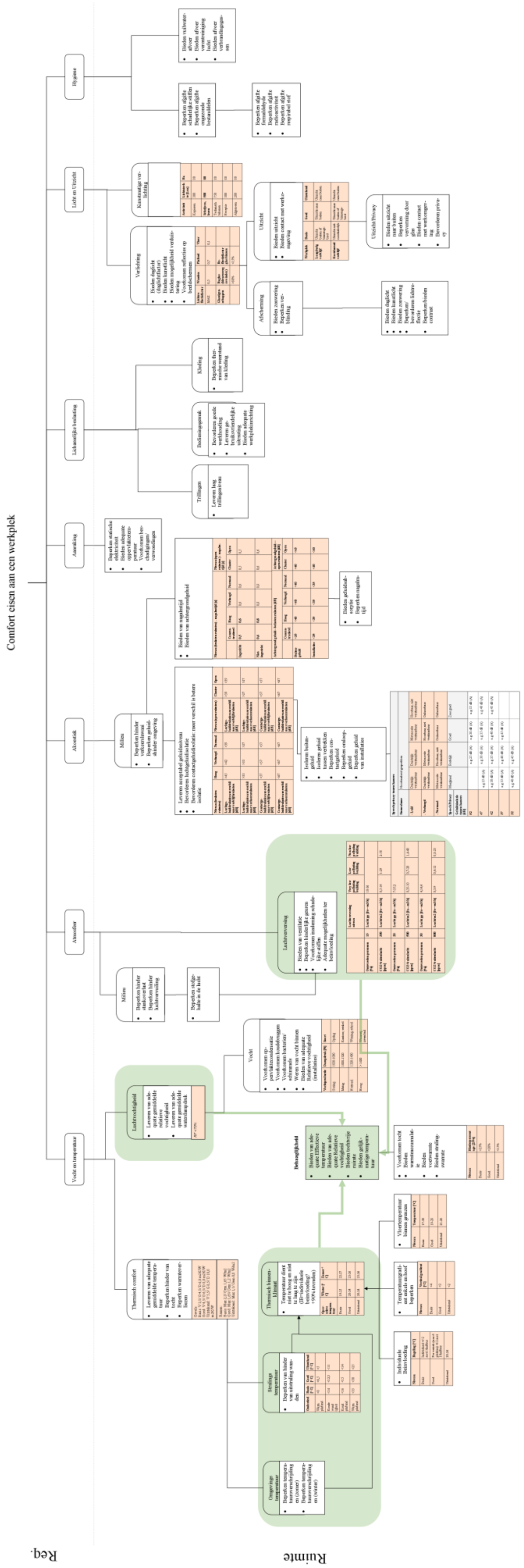

Mentioned per interview
Routing
Food/Drinks
Mobility
IT
Storage
Green
Color
Environment
Square meters
Environmental expectation
Identity
Esthetical comfort
Green
Health
Temperature
Ventilation
Air quality
Between head and feet
Gradient inside outside
Temperature
Thermic Comfort
View
Ergonomics
(Place) plug box
Height bureau
Usability
Distances to services
GPR
Soundproofing
Maintenance
Color
Materials
Radiation
Degree of influence
Airchange speed
humidity
Temperature
CO2
Climate
Luminance ratio
Contrast
Colorindex light
Glare
Always seeing sky
Positin monitor
Daylight
Light
Noise
Intelligibility
Audibility
Between rooms
Environment
Position ventilation grid
Reverbation time
Acoustics
Sound
0
1
2
3
4
5
6
7
8
9
10
Series1
Series2
Series3
Series4
Series5
Series6
Series7
Series8
Series9
Series10
Series11


Importancy [%]
Routing
Food/Drinks
Mobility
IT
Storage
Green
Color
Environment
Square meters
Environmental expectation
Identity
Esthetical comfort
Green
Health
Temperature
Ventilation
Air quality
Between head and feet
Gradient inside outside
Temperature
Thermic Comfort
View
Ergonomics
(Place) plug box
Height bureau
Usability
Distances to services
GPR
Soundproofing
Maintenance
Color
Materials
Radiation
Degree of influence
Airchange speed
humidity
Temperature
CO2
Climate
Luminance ratio
Contrast
Colorindex light
Glare
Always seeing sky
Positin monitor
Daylight
Light
Noise
Intelligibility
Audibility
Between rooms
Environment
Position ventilation grid
Reverbation time
Acoustics
Sound
0,00%
1,00%
2,00%
3,00%
4,00%
5,00%
6,00%
7,00%


Components Comfort
Routing
Environment
Square meters
Esthetical comfort
Green
Air quality
Thermic Comfort
View
Usability
Distances to services
Materials
Climate
Light
Sound
0
2
4
6
8
10
12


Importancy of subject 'Sound'
Intelligibility
Audibility
Between rooms
Environment
Position ventilation grid
Reverbation time
Acoustics
Sound
0,00%
5,00%
10,00%
15,00%
20,00%
25,00%
30,00%

## 9.4 Comfort



| **Speech privacy tussen kamers** | | | | |
|---|---|---|---|---|
| **Stemvolume** | Hoorbaarheid gesprekken | | | |
| **- Luid** | Duidelijk verstaanbaar | Duidelijk verstaanbaar | Met moeite verstaanbaar | Hoorbaar, niet verstaanbaar |
| **- Verhoogd** | Duidelijk verstaanbaar | Met moeite verstaanbaar | Hoorbaar, niet verstaanbaar | Onhoorbaar |
| **- Normaal** | Met moeite verstaanbaar | Hoorbaar, niet verstaanbaar | Onhoorbaar | Onhoorbaar |
| **Speech Privacy** | Marginaal | Redelijk | Goed | Zeer goed |
| **Geluidsisolatie tussen kamers [dB]** | | | | |
| **52** | | a.g 25 dB (A) | a.g 30 dB (A) | a.g 35 dB (A) |
| **47** | a.g 25 dB (A) | a.g 30 dB (A) | a.g 35 dB (A) | a.g 40 dB (A) |
| **42** | a.g 30 dB (A) | a.g 35 dB (A) | a.g 40 dB (A) | a.g 45 dB (A) |
| **37** | a.g 35 dB (A) | a.g 40 dB (A) | a.g 45 dB (A) | |
| **32** | a.g 40 dB (A) | a.g 45 dB (A) | | |

## 9.5 Example Script

```
;;;======================================================
;;;   Requirement Expert Sample Problem
;;;
;;;     Reqex: The Requirement Expert system.
;;;     This example selects an appropriate value
;;;     according to the components of a the abstract Comfort requirement.
;;;
;;;     CLIPS Version 6.0 Example
;;;
;;;     To execute, merely load, reset and run.
;;;======================================================

(defmodule MAIN (export ?ALL))

;;****************
;;* DEFFUNCTIONS *
;;****************

(deffunction MAIN::ask-question (?question ?allowed-values)
   (printout t ?question)
   (bind ?answer (read))
   (if (lexemep ?answer) then (bind ?answer (lowcase ?answer)))
   (while (not (member$ ?answer ?allowed-values)) do
      (printout t ?question)
      (bind ?answer (read))
      (if (lexemep ?answer) then (bind ?answer (lowcase ?answer))))
   ?answer)

  ;;*****************
;;* INITIAL STATE *
;;*****************

(deftemplate MAIN::attribute
   (slot name)
   (slot value)
   (slot certainty (default 100.0)))

(defrule MAIN::start
  (declare (salience 10000))
  =>
  (set-fact-duplication TRUE)
  (focus QUESTIONS CHOOSE-COMPONENTS REQUIREMENT PRINT-RESULTS))
;;(focus QUESTIONS CHOOSE-COMPONENTS REQUIREMENT PRINT-RESULTS-again))

(defrule MAIN::combine-certainties ""
  (declare (salience 100)
           (auto-focus TRUE))
  ?rem1 <- (attribute (name ?rel) (value ?val) (certainty ?per1))
  ?rem2 <- (attribute (name ?rel) (value ?val) (certainty ?per2))
  (test (neq ?rem1 ?rem2))
  =>
  (retract ?rem1)
  (modify ?rem2 (certainty (/ (- (* 100 (+ ?per1 ?per2)) (* ?per1 ?per2))
100.00))))

;;******************
;;* QUESTION RULES *
;;******************
```

```
(defmodule QUESTIONS (import MAIN ?ALL) (export ?ALL))

(deftemplate QUESTIONS::question
   (slot attribute (default ?NONE))
   (slot the-question (default ?NONE))
   (multislot valid-answers (default ?NONE))
   (slot already-asked (default FALSE))
   (multislot precursors (default ?DERIVE)))

(defrule QUESTIONS::ask-a-question
   ?f <- (question (already-asked FALSE)
                   (precursors)
                   (the-question ?the-question)
                   (attribute ?the-attribute)
                   (valid-answers $?valid-answers))
   =>
   (modify ?f (already-asked TRUE))
   (assert (attribute (name ?the-attribute)
                      (value (ask-question ?the-question ?valid-answers)))))

(defrule QUESTIONS::precursor-is-satisfied
   ?f <- (question (already-asked FALSE)
                   (precursors ?name is ?value $?rest))
         (attribute (name ?name) (value ?value))
   =>
   (if (eq (nth$ 1 ?rest) and)
    then (modify ?f (precursors (rest$ ?rest)))
    else (modify ?f (precursors ?rest))))

(defrule QUESTIONS::precursor-is-not-satisfied
   ?f <- (question (already-asked FALSE)
                   (precursors ?name is-not ?value $?rest))
         (attribute (name ?name) (value ~?value))
   =>
   (if (eq (nth$ 1 ?rest) and)
    then (modify ?f (precursors (rest$ ?rest)))
    else (modify ?f (precursors ?rest))))

;;*******************
;;* REQEX QUESTIONS *
;;*******************

(defmodule REQUIREMENT-QUESTIONS (import QUESTIONS ?ALL))

(deffacts REQUIREMENT-QUESTIONS::question-workplaces
  (question (attribute activity)
            (the-question "Defining artificial light strength.
                   Which activity matches your workplace?:
                   - archiving, copying
                   - writing, typing, reading, conference
                   - technical drawing
                   - reception
                   - general light
                   Answer: ")
            (valid-answers archiving copying writing typing reading
conference technical reception general)))

;;******************
;; The RULES module
;;******************
```

```
(defmodule RULES (import MAIN ?ALL) (export ?ALL))
(deftemplate RULES::rule
  (slot certainty (default 100.0))
  (multislot if)
  (multislot then))

(defrule RULES::throw-away-ands-in-antecedent
  ?f <- (rule (if and $?rest))
  =>
  (modify ?f (if ?rest)))

(defrule RULES::throw-away-ands-in-consequent
  ?f <- (rule (then and $?rest))
  =>
  (modify ?f (then ?rest)))

(defrule RULES::remove-is-condition-when-satisfied
  ?f <- (rule (certainty ?c1)
              (if ?attribute is ?value $?rest))
  (attribute (name ?attribute)
             (value ?value)
             (certainty ?c2))
  =>
  (modify ?f (certainty (min ?c1 ?c2)) (if ?rest)))

(defrule RULES::remove-is-not-condition-when-satisfied
  ?f <- (rule (certainty ?c1)
              (if ?attribute is-not ?value $?rest))
  (attribute (name ?attribute) (value ~?value) (certainty ?c2))
  =>
  (modify ?f (certainty (min ?c1 ?c2)) (if ?rest)))

(defrule RULES::perform-rule-consequent-with-certainty
  ?f <- (rule (certainty ?c1)
              (if)
              (then ?attribute is ?value with certainty ?c2 $?rest))
  =>
  (modify ?f (then ?rest))
  (assert (attribute (name ?attribute)
                     (value ?value)
                     (certainty (/ (* ?c1 ?c2) 100)))))

(defrule RULES::perform-rule-consequent-without-certainty
  ?f <- (rule (certainty ?c1)
              (if)
              (then ?attribute is ?value $?rest))
  (test (or (eq (length$ ?rest) 0)
            (neq (nth$ 1 ?rest) with)))
  =>
  (modify ?f (then ?rest))
  (assert (attribute (name ?attribute) (value ?value) (certainty ?c1))))

;;*******************************
;;* CHOOSE REQUIREMENT-COMPONENTS RULES *
;;*******************************

(defmodule CHOOSE-COMPONENTS (import RULES ?ALL)
                             (import QUESTIONS ?ALL)
                             (import MAIN ?ALL))
```

```
(defrule CHOOSE-COMPONENTS::startit => (focus RULES))
(deffacts the-requirement-rules

  ; Rules for picking the best audibility based on background-noise,
workplace-consultation and workplace-volume

  (rule (if activity is archiving)
        (then best-activity is archiving with certainty 100))

  (rule (if activity is copying)
        (then best-activity is copying with certainty 100))

  (rule (if activity is writing)
        (then best-activity is writing with certainty 100))

  (rule (if activity is typing)
        (then best-activity is typing with certainty 100))

  (rule (if activity is reading)
        (then best-activity is reading with certainty 100))

  (rule (if activity is conference)
        (then best-activity is conference with certainty 100))

  (rule (if activity is technical)
        (then best-activity is technical with certainty 100))

  (rule (if activity is reception)
        (then best-activity is reception with certainty 100))

  (rule (if activity is general)
        (then best-activity is general with certainty 100))
)

;;************************
;;* REQUIREMENTS SELECTION RULES *
;;************************

(defmodule REQUIREMENT (import MAIN ?ALL))

(deffacts any-workplaces
  (attribute (name best-activity) (value any)))

(deftemplate REQUIREMENT::component
  (slot name (default ?NONE))
  (multislot activity (default any)))

(deffacts REQUIREMENT::the-component-list
  (component (name 300lux) (activity archiving copying reception))
  (component (name 500lux) (activity writing typing reading conference))
  (component (name 750lux) (activity technical))
  (component (name 200lux) (activity general)))

(defrule REQUIREMENT::generate-requirements
  (component (name ?name)
            (activity $? ?t $?))
  (attribute (name best-activity) (value ?t) (certainty ?certainty-1))

  =>
  (assert (attribute (name component) (value ?name)
                     (certainty (min ?certainty-1)))))
```

```
;;****************************
;;* PRINT SELECTED component RULES *
;;****************************

(defmodule PRINT-RESULTS (import MAIN ?ALL))

(defrule PRINT-RESULTS::header ""
  (declare (salience 10))
   =>
   (printout t )
   (printout t " RESULTS" crlf)
   (printout t " dB Value             MATCH" crlf)
   (printout t " -------------------------------" crlf)
   (assert (phase print-requirements)))
(defrule PRINT-RESULTS::print-requirements ""
  ?rem <- (attribute (name component) (value ?name) (certainty ?per))
  (not (attribute (name component) (certainty ?per1&:(> ?per1 ?per))))
  =>
  (retract ?rem)
  (format t " %-24s %2d%%%n" ?name ?per))

(defrule PRINT-RESULTS::remove-poor-component-choices ""
  ?rem <- (attribute (name component) (certainty ?per&:(< ?per 10)))
  =>
  (retract ?rem))

(defrule PRINT-RESULTS::end-spaces ""
   (not (attribute (name component)))
   =>
   (printout t ))
```

## 9.6 Script model

```
;;;======================================================
;;;     Requirement Expert Sample Problem
;;;
;;;     Reqex: The Requirement Expert system.
;;;     This example selects an appropriate value
;;;     according to the component of soundinsulation of
;;;         the abstract Comfort requirement.
;;;
;;;     CLIPS Version 6.0 Example
;;;
;;;     To execute, merely load, reset and run.
;;;======================================================

(defmodule MAIN (export ?ALL))

;;****************
;;* DEFFUNCTIONS *
;;****************

(deffunction MAIN::ask-question (?question ?allowed-values)
   (printout t ?question)
   (bind ?answer (read))
   (if (lexemep ?answer) then (bind ?answer (lowcase ?answer)))
   (while (not (member$ ?answer ?allowed-values)) do
      (printout t ?question)
      (bind ?answer (read))
      (if (lexemep ?answer) then (bind ?answer (lowcase ?answer))))
   ?answer)

;;*****************
;;* INITIAL STATE *
;;*****************

(deftemplate MAIN::attribute
   (slot name)
   (slot value)
   (slot certainty (default 100.0)))

(defrule MAIN::start
  (declare (salience 10000))
  =>
  (set-fact-duplication TRUE)
  (focus QUESTIONS CHOOSE-COMPONENTS REQUIREMENT PRINT-RESULTS))
;;(focus QUESTIONS CHOOSE-COMPONENTS REQUIREMENT PRINT-RESULTS-again))

(defrule MAIN::combine-certainties ""
  (declare (salience 100)
           (auto-focus TRUE))
  ?rem1 <- (attribute (name ?rel) (value ?val) (certainty ?per1))
  ?rem2 <- (attribute (name ?rel) (value ?val) (certainty ?per2))
  (test (neq ?rem1 ?rem2))
  =>
  (retract ?rem1)
  (modify ?rem2 (certainty (/ (- (* 100 (+ ?per1 ?per2)) (* ?per1 ?per2))
100.00))))

;;******************
;;* QUESTION RULES *
```

```
;;******************

(defmodule QUESTIONS (import MAIN ?ALL) (export ?ALL))

(deftemplate QUESTIONS::question
   (slot attribute (default ?NONE))
   (slot the-question (default ?NONE))
   (multislot valid-answers (default ?NONE))
   (slot already-asked (default FALSE))
   (multislot precursors (default ?DERIVE)))

(defrule QUESTIONS::ask-a-question
   ?f <- (question (already-asked FALSE)
                   (precursors)
                   (the-question ?the-question)
                   (attribute ?the-attribute)
                   (valid-answers $?valid-answers))
   =>
   (modify ?f (already-asked TRUE))
   (assert (attribute (name ?the-attribute)
                      (value (ask-question ?the-question ?valid-answers)))))

(defrule QUESTIONS::precursor-is-satisfied
   ?f <- (question (already-asked FALSE)
                   (precursors ?name is ?value $?rest))
         (attribute (name ?name) (value ?value))
   =>
   (if (eq (nth$ 1 ?rest) and)
    then (modify ?f (precursors (rest$ ?rest)))
    else (modify ?f (precursors ?rest))))

(defrule QUESTIONS::precursor-is-not-satisfied
   ?f <- (question (already-asked FALSE)
                   (precursors ?name is-not ?value $?rest))
         (attribute (name ?name) (value ~?value))
   =>
   (if (eq (nth$ 1 ?rest) and)
    then (modify ?f (precursors (rest$ ?rest)))
    else (modify ?f (precursors ?rest))))

;;******************
;;* REQEX QUESTIONS *
;;******************
;; 25-30-35-40-45 dB

(defmodule REQUIREMENT-QUESTIONS (import QUESTIONS ?ALL))

(deffacts REQUIREMENT-QUESTIONS::question-workplaces
  (question (attribute background-noise)
            (the-question "Which background noise is applicable [in dB] ?:
                   - Low
                   - Slightly increased
                   - Increased
                   - High
                   - Dominant
                   Answer: ")
            (valid-answers low slightly increased high dominant))
  (question (attribute workplace-layout)
            (the-question "Which layout matches your workplace?:
                   - private workplace
                   - open workplace
```

```
                    - clustered workplace
                    Answer: ")
             (valid-answers private open clustered))
   (question (attribute workplace-audibility)
                   (the-question "Which audibility do you prefer regarding your
work?:
                    - clearly understandable
                    - barely understandable
                    - audible and not understandable
                    - inaudible
                    Answer: ")
             (valid-answers clearly barely audible inaudible))
   (question (attribute workplace-combination)
             (precursors workplace-choice is combination)
                   (the-question "What category of speech privacy do you
prefer?:
                    - marginally
                    - reasonable
                    - good
                    - very good
                    Answer: ")
             (valid-answers marginally reasonable good very))
   (question (attribute workplace-computer)
             (precursors workplace-choice is computer)
                   (the-question "What category of speech privacy do you
prefer?:
                    - marginally
                    - reasonable
                    - good
                    - very good
                    Answer: ")
             (valid-answers marginally reasonable good very))
   (question (attribute workplace-conversation)
             (precursors workplace-choice is conversation)
                   (the-question "What category of speech privacy do you
prefer?:
                    - marginally
                    - reasonable
                    - good
                    - very good
                    Answer: ")
             (valid-answers marginally reasonable good very))
   (question (attribute workplace-consultation)
             (precursors workplace-choice is consultation)
                   (the-question "What category of speech privacy do you prefer
for consultation?:
                    - marginally
                    - reasonable
                    - good
                    - very good
                    Answer: ")
             (valid-answers marginally reasonable good very))
   (question (attribute workplace-volume)
                   (the-question "At which level would you scale your own
voice?:
                    - loud voice
                    - raised voice
                    - normal voice
                    Answer: ")
             (valid-answers loud raised normal))
   (question (attribute workplace-choice)
```

```
            (the-question "Defining sound insulation between rooms.
                  Which activity matches your workplace?:
                  - consultation
                  - conversation by phone
                  - computer work
                  - combination of all above
                  Answer: ")
            (valid-answers consultation conversation computer combination)))

;;******************
;; The RULES module
;;******************

(defmodule RULES (import MAIN ?ALL) (export ?ALL))

(deftemplate RULES::rule
  (slot certainty (default 100.0))
  (multislot if)
  (multislot then))

(defrule RULES::throw-away-ands-in-antecedent
  ?f <- (rule (if and $?rest))
  =>
  (modify ?f (if ?rest)))

(defrule RULES::throw-away-ands-in-consequent
  ?f <- (rule (then and $?rest))
  =>
  (modify ?f (then ?rest)))

(defrule RULES::remove-is-condition-when-satisfied
  ?f <- (rule (certainty ?c1)
              (if ?attribute is ?value $?rest))
  (attribute (name ?attribute)
             (value ?value)
             (certainty ?c2))
  =>
  (modify ?f (certainty (min ?c1 ?c2)) (if ?rest)))

(defrule RULES::remove-is-not-condition-when-satisfied
  ?f <- (rule (certainty ?c1)
              (if ?attribute is-not ?value $?rest))
  (attribute (name ?attribute) (value ~?value) (certainty ?c2))
  =>
  (modify ?f (certainty (min ?c1 ?c2)) (if ?rest)))

(defrule RULES::perform-rule-consequent-with-certainty
  ?f <- (rule (certainty ?c1)
              (if)
              (then ?attribute is ?value with certainty ?c2 $?rest))
  =>
  (modify ?f (then ?rest))
  (assert (attribute (name ?attribute)
                     (value ?value)
                     (certainty (/ (* ?c1 ?c2) 100)))))

(defrule RULES::perform-rule-consequent-without-certainty
  ?f <- (rule (certainty ?c1)
              (if)
              (then ?attribute is ?value $?rest))
  (test (or (eq (length$ ?rest) 0)
```

```
            (neq (nth$ 1 ?rest) with)))
  =>
  (modify ?f (then ?rest))
  (assert (attribute (name ?attribute) (value ?value) (certainty ?c1))))

;;******************************
;;* CHOOSE REQUIREMENT-COMPONENTS RULES *
;;******************************

(defmodule CHOOSE-COMPONENTS (import RULES ?ALL)
                             (import QUESTIONS ?ALL)
                             (import MAIN ?ALL))

(defrule CHOOSE-COMPONENTS::startit => (focus RULES))

(deffacts the-requirement-rules

  ; Rules for picking the best audibility based on background-noise, workplace-
consultation and workplace-volume

;     47-35dB

  (rule (if background-noise is increased and
              workplace-consultation is good and
              workplace-volume is loud)
        (then best-audibility is barely with certainty 70))

  (rule (if background-noise is increased and
              workplace-consultation is good and
              workplace-volume is raised)
        (then best-audibility is audible with certainty 65))

  (rule (if background-noise is increased and
              workplace-consultation is good and
              workplace-volume is normal)
        (then best-audibility is inaudible with certainty 60))

;     52-30dB

  (rule (if background-noise is slightly and
              workplace-consultation is good and
              workplace-volume is loud)
        (then best-audibility is barely with certainty 70))

  (rule (if background-noise is slightly and
              workplace-consultation is good and
              workplace-volume is raised)
        (then best-audibility is audible with certainty 65))

  (rule (if background-noise is slightly and
              workplace-consultation is good and
              workplace-volume is normal)
        (then best-audibility is inaudible with certainty 60))

;     42-40dB

  (rule (if background-noise is high and
              workplace-consultation is good and
              workplace-volume is loud)
        (then best-audibility is barely with certainty 70))
```

```
  (rule (if background-noise is high and
                 workplace-consultation is good and
                 workplace-volume is raised)
             (then best-audibility is audible with certainty 65))

  (rule (if background-noise is high and
                 workplace-consultation is good and
                 workplace-volume is normal)
             (then best-audibility is inaudible with certainty 60))

;      37-45dB

  (rule (if background-noise is dominant and
                 workplace-consultation is good and
                 workplace-volume is loud)
             (then best-audibility is barely with certainty 70))

  (rule (if background-noise is dominant and
                 workplace-consultation is good and
                 workplace-volume is raised)
             (then best-audibility is audible with certainty 65))

  (rule (if background-noise is dominant and
                 workplace-consultation is good and
                 workplace-volume is normal)
             (then best-audibility is inaudible with certainty 60.5))

;      52-25dB

  (rule (if background-noise is low and
                 workplace-consultation is reasonable and
                 workplace-volume is loud)
             (then best-audibility is clearly with certainty 70))

  (rule (if background-noise is low and
                 workplace-consultation is reasonable and
                 workplace-volume is raised)
             (then best-audibility is barely with certainty 65))

  (rule (if background-noise is low and
                 workplace-consultation is reasonable and
                 workplace-volume is normal)
             (then best-audibility is audible with certainty 60))

;      47-30dB

  (rule (if background-noise is slightly and
                 workplace-consultation is reasonable and
                 workplace-volume is loud)
             (then best-audibility is clearly with certainty 70))

  (rule (if background-noise is slightly and
                 workplace-consultation is reasonable and
                 workplace-volume is raised)
             (then best-audibility is barely with certainty 65))

  (rule (if background-noise is slightly and
                 workplace-consultation is reasonable and
                 workplace-volume is normal)
             (then best-audibility is audible with certainty 60))
```

```
;     42-35dB

  (rule (if background-noise is increased and
                 workplace-consultation is reasonable and
                 workplace-volume is loud)
            (then best-audibility is clearly with certainty 70))

  (rule (if background-noise is increased and
                 workplace-consultation is reasonable and
                 workplace-volume is raised)
            (then best-audibility is barely with certainty 65))

  (rule (if background-noise is increased and
                 workplace-consultation is reasonable and
                 workplace-volume is normal)
            (then best-audibility is audible with certainty 60))

;     37-40dB

  (rule (if background-noise is high and
                 workplace-consultation is reasonable and
                 workplace-volume is loud)
            (then best-audibility is clearly with certainty 70))

  (rule (if background-noise is high and
                 workplace-consultation is reasonable and
                 workplace-volume is raised)
            (then best-audibility is barely with certainty 65))

  (rule (if background-noise is high and
                 workplace-consultation is reasonable and
                 workplace-volume is normal)
            (then best-audibility is audible with certainty 60))

;     32-45dB

  (rule (if background-noise is dominant and
                 workplace-consultation is reasonable and
                 workplace-volume is loud)
            (then best-audibility is clearly with certainty 70))

  (rule (if background-noise is dominant and
                 workplace-consultation is reasonable and
                 workplace-volume is raised)
            (then best-audibility is barely with certainty 65))

  (rule (if background-noise is dominant and
                 workplace-consultation is reasonable and
                 workplace-volume is normal)
            (then best-audibility is audible with certainty 60))


;     47-25dB

  (rule (if background-noise is low and
                 workplace-consultation is marginally and
                 workplace-volume is loud)
            (then best-audibility is clearly with certainty 70))

  (rule (if background-noise is low and
```

```
                  workplace-consultation is marginally and
                  workplace-volume is raised)
            (then best-audibility is clearly with certainty 65))

   (rule (if background-noise is low and
                  workplace-consultation is marginally and
                  workplace-volume is normal)
            (then best-audibility is barely with certainty 60))

;      42-30dB

   (rule (if background-noise is slightly and
                  workplace-consultation is marginally and
                  workplace-volume is loud)
            (then best-audibility is clearly with certainty 70))

   (rule (if background-noise is slightly and
                  workplace-consultation is marginally and
                  workplace-volume is raised)
            (then best-audibility is clearly with certainty 65))

   (rule (if background-noise is slightly and
                  workplace-consultation is marginally and
                  workplace-volume is normal)
            (then best-audibility is barely with certainty 60))

;      32-40dB

   (rule (if background-noise is high and
                  workplace-consultation is marginally and
                  workplace-volume is loud)
            (then best-audibility is clearly with certainty 70))

   (rule (if background-noise is high and
                  workplace-consultation is marginally and
                  workplace-volume is raised)
            (then best-audibility is clearly with certainty 65))

   (rule (if background-noise is high and
                  workplace-consultation is marginally and
                  workplace-volume is normal)
            (then best-audibility is barely with certainty 60))

;      37-35dB

   (rule (if background-noise is increased and
                  workplace-consultation is marginally and
                  workplace-volume is loud)
            (then best-audibility is clearly with certainty 70))

   (rule (if background-noise is increased and
                  workplace-consultation is marginally and
                  workplace-volume is raised)
            (then best-audibility is clearly with certainty 65))

   (rule (if background-noise is increased and
                  workplace-consultation is marginally and
                  workplace-volume is normal)
            (then best-audibility is barely with certainty 60))

;      52-35dB
```

```
(rule (if background-noise is increased and
             workplace-consultation is very and
             workplace-volume is loud)
         (then best-audibility is audible with certainty 70))

(rule (if background-noise is increased and
             workplace-consultation is very and
             workplace-volume is raised)
         (then best-audibility is inaudible with certainty 65))

(rule (if background-noise is increased and
             workplace-consultation is very and
             workplace-volume is normal)
         (then best-audibility is inaudible with certainty 60))

;     47-40dB

(rule (if background-noise is high and
             workplace-consultation is very and
             workplace-volume is loud)
         (then best-audibility is audible with certainty 70))

(rule (if background-noise is high and
             workplace-consultation is very and
             workplace-volume is raised)
         (then best-audibility is inaudible with certainty 65))

(rule (if background-noise is high and
             workplace-consultation is very and
             workplace-volume is normal)
         (then best-audibility is inaudible with certainty 60))

;     42-45dB

(rule (if background-noise is dominant and
             workplace-consultation is very and
             workplace-volume is loud)
         (then best-audibility is audible with certainty 70))

(rule (if background-noise is dominant and
             workplace-consultation is very and
             workplace-volume is raised)
         (then best-audibility is inaudible with certainty 65))

(rule (if background-noise is dominant and
             workplace-consultation is very and
             workplace-volume is normal)
         (then best-audibility is inaudible with certainty 60))



; Rules for picking the best audibility based on volume and speech privacy

(rule (if workplace-consultation is good and
             workplace-volume is raised)
      (then best-audibility is audible with certainty 70 and
              best-audibility is inaudible with certainty 50 and
              best-audibility is barely with certainty 10))

(rule (if workplace-consultation is good and
```

```
                workplace-volume is loud)
         (then best-audibility is barely with certainty 70
                best-audibility is audible with certainty 50 and
                  best-audibility is inaudible with certainty 10))

(rule (if workplace-consultation is good and
                workplace-volume is normal)
         (then best-audibility is inaudible with certainty 70
                best-audibility is audible with certainty 50 and
                  best-audibility is barely with certainty 10))



(rule (if workplace-consultation is reasonable and
                workplace-volume is raised)
         (then best-audibility is barely with certainty 70 and
                  best-audibility is audible with certainty 50 and
                  best-audibility is clearly with certainty 10))

(rule (if workplace-consultation is reasonable and
                workplace-volume is loud)
         (then best-audibility is clearly with certainty 70
                best-audibility is barely with certainty 50 and
                  best-audibility is audible with certainty 10))

(rule (if workplace-consultation is reasonable and
                workplace-volume is normal)
         (then best-audibility is audible with certainty 70
                best-audibility is barely with certainty 50 and
                  best-audibility is clearly with certainty 10))



(rule (if workplace-consultation is marginally and
                workplace-volume is raised)
         (then best-audibility is clearly with certainty 70 and
                  best-audibility is barely with certainty 25))

(rule (if workplace-consultation is marginally and
                workplace-volume is loud)
         (then best-audibility is clearly with certainty 70
                best-audibility is barely with certainty 25))

(rule (if workplace-consultation is marginally and
                workplace-volume is normal)
         (then best-audibility is barely with certainty 70
                best-audibility is clearly with certainty 25))



(rule (if workplace-consultation is very and
                workplace-volume is raised)
         (then best-audibility is inaudible with certainty 70 and
                  best-audibility is audible with certainty 25))

(rule (if workplace-consultation is very and
                workplace-volume is loud)
         (then best-audibility is audible with certainty 70
                best-audibility is inaudible with certainty 25))

(rule (if workplace-consultation is very and
```

```
                workplace-volume is normal)
        (then best-audibility is inaudible with certainty 70
                best-audibility is audible with certainty 25))



(rule (if workplace-layout is private and
                background-noise is dominant and
                audibility is inaudible)
        (then best-audibility is inaudible with certainty 80 and
             then best-audibility is audible with certainty 60 and
             then best-audibility is barely with certainty 40 and
             then best-audibility is clearly with certainty 20))


(rule (if workplace-layout is private and
                background-noise is high and
                audibility is inaudible)
        (then best-audibility is inaudible with certainty 70 and
             then best-audibility is audible with certainty 50 and
             then best-audibility is barely with certainty 30 and
             then best-audibility is clearly with certainty 10))


(rule (if workplace-layout is private and
                background-noise is increased and
                audibility is inaudible)
        (then best-audibility is inaudible with certainty 60 and
             then best-audibility is audible with certainty 40 and
             then best-audibility is barely with certainty 20))

(rule (if workplace-layout is private and
                background-noise is slightly and
                audibility is inaudible)
        (then best-audibility is inaudible with certainty 50 and
             then best-audibility is audible with certainty 30 and
             then best-audibility is barely with certainty 10))

(rule (if workplace-layout is private and
                background-noise is low and
                audibility is inaudible)
        (then best-audibility is inaudible with certainty 40 and
             then best-audibility is audible with certainty 20))



(rule (if workplace-layout is open and
                background-noise is dominant and
                audibility is inaudible)
        (then best-audibility is clearly with certainty 60 and
             then best-audibility is barely with certainty 40 and
             then best-audibility is audible with certainty 20))


(rule (if workplace-layout is open and
                background-noise is high and
                audibility is inaudible)
        (then best-audibility is clearly with certainty 50 and
             then best-audibility is barely with certainty 30 and
             then best-audibility is audible with certainty 10))
```

```
(rule (if workplace-layout is open and
            background-noise is increased and
            audibility is inaudible)
      (then best-audibility is clearly with certainty 40 and
          then best-audibility is barely with certainty 20))

(rule (if workplace-layout is open and
            background-noise is slightly and
            audibility is inaudible)
      (then best-audibility is clearly with certainty 30 and
          then best-audibility is barely with certainty 10))

(rule (if workplace-layout is open and
            background-noise is low and
            audibility is inaudible)
      (then best-audibility is clearly with certainty 20))


; Rules for picking the best consultation


(rule (if background-noise is dominant)
        (then best-workplace-consultation is very with certainty 70 and
                best-workplace-consultation is good with certainty 50 and
                best-workplace-consultation is reasonable with certainty 30
and
                best-workplace-consultation is marginally with certainty
10))

(rule (if background-noise is high)
        (then best-workplace-consultation is very with certainty 70 and
                best-workplace-consultation is good with certainty 50 and
                best-workplace-consultation is reasonable with certainty 30
and
                best-workplace-consultation is marginally with certainty
10))

(rule (if background-noise is increased)
        (then best-workplace-consultation is very with certainty 70 and
                best-workplace-consultation is good with certainty 50 and
                best-workplace-consultation is reasonable with certainty 30
and
                best-workplace-consultation is marginally with certainty
10))

(rule (if background-noise is slightly)
        (then best-workplace-consultation is very with certainty 70 and
                best-workplace-consultation is good with certainty 50 and
                best-workplace-consultation is reasonable with certainty 30
and
                best-workplace-consultation is marginally with certainty
10))

(rule (if background-noise is low)
        (then best-workplace-consultation is very with certainty 70 and
                best-workplace-consultation is good with certainty 50 and
                best-workplace-consultation is reasonable with certainty 30
and
                best-workplace-consultation is marginally with certainty
10))
```

```
  (rule (if workplace-consultation is very)
           (then best-audibility is inaudible with certainty 70 and
                   best-audibility is audible with certainty 30))

  (rule (if workplace-consultation is good)
           (then best-audibility is inaudible with certainty 70 and
                   best-audibility is audible with certainty 50 and
                   best-audibility is barely with certainty 30))

  (rule (if workplace-consultation is reasonable)
           (then best-audibility is audible with certainty 70 and
                   best-audibility is barely with certainty 50 and
                   best-audibility is clearly with certainty 30))

  (rule (if workplace-consultation is marginally)
           (then best-audibility is barely with certainty 70 and
                   best-audibility is clearly with certainty 30))




  (rule (if background-noise is low)
           (then best-audibility is audible with certainty 70 and
                   best-audibility is barely with certainty 50 and
                   best-audibility is clearly with certainty 30))

  (rule (if background-noise is low)
           (then best-workplace-consultation is reasonable with certainty 70
and
                   best-workplace-consultation is marginally with certainty
40))

  (rule (if background-noise is slightly)
           (then best-audibility is inaudible with certainty 70 and
                   best-audibility is audible with certainty 50 and
                   best-audibility is barely with certainty 30))

  (rule (if background-noise is slightly)
           (then best-workplace-consultation is good with certainty 70 and
                   best-workplace-consultation is reasonable with certainty 40
and
                   best-workplace-consultation is marginally with certainty
20))

  (rule (if background-noise is increased)
           (then best-audibility is inaudible with certainty 70 and
                   best-audibility is audible with certainty 50 and
                   best-audibility is barely with certainty 30))

  (rule (if background-noise is increased)
           (then best-workplace-consultation is very with certainty 80 and
                   best-workplace-consultation is good with certainty 60 and
                   best-workplace-consultation is reasonable with certainty 40
and
                   best-workplace-consultation is marginally with certainty
20))

  (rule (if background-noise is high)
```

```
           (then best-audibility is inaudible with certainty 70 and
                 best-audibility is audible with certainty 50 and
                 best-audibility is barely with certainty 30))

  (rule (if background-noise is high)
           (then best-workplace-consultation is very with certainty 80 and
                 best-workplace-consultation is good with certainty 60 and
                 best-workplace-consultation is reasonable with certainty
40))

  (rule (if background-noise is dominant)
           (then best-audibility is audible with certainty 70 and
                 best-audibility is barely with certainty 50 and
                 best-audibility is clearly with certainty 30))

  (rule (if background-noise is dominant)
           (then best-workplace-consultation is good with certainty 60 and
                 best-workplace-consultation is reasonable with certainty 40
and
                 best-workplace-consultation is marginally with certainty
20))


  (rule (if workplace-choice is consultation)
           (then best-background-noise is low with certainty 70 and
                 best-background-noise is slightly with certainty 60 and
                 best-background-noise is increased with certainty 50 and
                 best-background-noise is high with certainty 40 and
                 best-background-noise is dominant with certainty 30))

  (rule (if workplace-choice is conversation)
           (then best-background-noise is slightly with certainty 70 and
                 best-background-noise is increased with certainty 50 and
                 best-background-noise is high with certainty 30))

  (rule (if workplace-choice is computer)
           (then best-background-noise is low with certainty 60 and
                 best-background-noise is slightly with certainty 30))

  (rule (if workplace-choice is combination)
           (then best-background-noise is increased with certainty 60 and
                 best-background-noise is high with certainty 40 and
                 best-background-noise is dominant with certainty 20))

  (rule (if background-noise is low)
           (then best-background-noise is low with certainty 80))


  (rule (if background-noise is slightly)
           (then best-background-noise is slightly with certainty 80))


  (rule (if background-noise is increased)
           (then best-background-noise is increased with certainty 80))


  (rule (if background-noise is high)
           (then best-background-noise is high with certainty 80))

  (rule (if background-noise is dominant)
```

```
            (then best-background-noise is dominant with certainty 80))

)

;;************************
;;* REQUIREMENTS SELECTION RULES *
;;************************

(defmodule REQUIREMENT (import MAIN ?ALL))

(deffacts any-workplaces
  (attribute (name best-volume) (value any))
  (attribute (name best-audibility) (value any))
  (attribute (name best-workplace-consultation) (value any))
  (attribute (name best-background-noise) (value any)))

(deftemplate REQUIREMENT::component
  (slot name (default ?NONE))
  (multislot volume (default any))
  (multislot audibility (default any))
  (multislot workplace-consultation (default any))
  (multislot background-noise (default any)))

(deffacts REQUIREMENT::the-component-list
  (component (name 52-35dB) (audibility inaudible audible) (background-noise
increased))
  (component (name 47-40dB) (audibility inaudible audible) (background-noise
high))
  (component (name 42-45dB) (audibility inaudible audible) (background-noise
dominant))
  (component (name 52-30dB) (audibility barely audible inaudible) (background-
noise slightly))
  (component (name 47-35dB) (audibility barely audible inaudible) (background-
noise increased))
  (component (name 42-40dB) (audibility barely audible inaudible) (background-
noise high))
  (component (name 37-45dB) (audibility barely audible inaudible) (background-
noise dominant))
  (component (name 52-25dB) (audibility clearly barely audible) (background-
noise low))
  (component (name 47-30dB) (audibility clearly barely audible) (background-
noise slightly))
  (component (name 42-35dB) (audibility clearly barely audible) (background-
noise increased))
  (component (name 37-40dB) (audibility clearly barely audible) (background-
noise high))
  (component (name 32-45dB) (audibility clearly barely audible) (background-
noise dominant))
  (component (name 47-25dB) (audibility clearly barely) (background-noise low))

  (component (name 42-30dB) (audibility clearly barely) (background-noise
slightly))
  (component (name 37-35dB) (audibility clearly barely) (background-noise
increased))
  (component (name 32-40dB) (audibility clearly barely) (background-noise
high)))

(defrule REQUIREMENT::generate-requirements
  (component (name ?name)
            (volume $? ?d $?)
        (audibility $? ?c $?)
```

```
           (workplace-consultation $? ?t $?)
           (background-noise $? ?r $?))
  (attribute (name best-volume) (value ?d) (certainty ?certainty-1))
  (attribute (name best-audibility) (value ?c) (certainty ?certainty-2))
  (attribute (name best-workplace-consultation) (value ?t) (certainty
?certainty-3))
  (attribute (name best-background-noise) (value ?r) (certainty ?certainty-4))

  =>
  (assert (attribute (name component) (value ?name)
                     (certainty (min ?certainty-1 ?certainty-2 ?certainty-3
?certainty-4)))))

;;****************************
;;* PRINT SELECTED component RULES *
;;****************************

(defmodule PRINT-RESULTS (import MAIN ?ALL))

(defrule PRINT-RESULTS::header ""
  (declare (salience 10))
   =>
   (printout t )
   (printout t " RESULTS" crlf)
   (printout t " dB Value             MATCH" crlf)
   (printout t " -------------------------------" crlf)
   (assert (phase print-requirements)))

(defrule PRINT-RESULTS::print-requirements ""
  ?rem <- (attribute (name component) (value ?name) (certainty ?per))
  (not (attribute (name component) (certainty ?per1&:(> ?per1 ?per))))
  =>
  (retract ?rem)
  (format t " %-24s %2d%%%n" ?name ?per))

(defrule PRINT-RESULTS::remove-poor-component-choices ""
  ?rem <- (attribute (name component) (certainty ?per&:(< ?per 90)))
  =>
  (retract ?rem))

(defrule PRINT-RESULTS::end-spaces ""
   (not (attribute (name component)))
   =>
   (printout t ))
```

## 9.7 IFC Pilot Case

```
ISO-10303-21;
HEADER;

/***************************************************************************
****************
* STEP Physical File produced by: The EXPRESS Data Manager Version
5.02.0100.07 : 28 Aug 2013
* Module:                          EDMstepFileFactory/EDMstandAlone
* Creation date:                   Thu Dec 20 14:59:20 2018
* Host:                            S124982
* Database:
C:\Users\s124982\AppData\Local\Temp\{5EA3C6B3-6A22-4C32-A401-
6AF325A7C9C3}\ifc
* Database version:                5507
* Database creation date:          Thu Dec 20 14:57:59 2018
* Schema:                          IFC2X3
* Model:                           DataRepository.ifc
* Model creation date:             Thu Dec 20 14:58:00 2018
* Header model:                    DataRepository.ifc_HeaderModel
* Header model creation date:      Thu Dec 20 14:58:00 2018
* EDMuser:                         sdai-user
* EDMgroup:                        sdai-group
* License ID and type:             5605 : Permanent license. Expiry date:
* EDMstepFileFactory options:      020000
***************************************************************************
***************/
FILE_DESCRIPTION(('ViewDefinition [CoordinationView,
QuantityTakeOffAddOnView]'),'2;1');
FILE_NAME('IKC De Klimboom','2018-12-20T14:59:20',(''),(''),'The EXPRESS
Data Manager Version 5.02.0100.07 : 28 Aug 2013','20150714_1515(x64) -
Exporter 16.0.490.0 - Default UI','');
FILE_SCHEMA(('IFC2X3'));
ENDSEC;

DATA;
#1= IFCORGANIZATION($,'Autodesk Revit 2016 (ENU)',$,$,$);
#5= IFCAPPLICATION(#1,'2016','Autodesk Revit 2016 (ENU)','Revit');
[…]
#147=
IFCSPACE('0666CFtfr1Xf1db2ZGslbu',#41,'1',$,$,#130,#143,'Space',.ELEMENT.,.
INTERNAL.,$);
[…]
#364= IFCPROPERTYSINGLEVALUE('Requirement',$,IFCTEXT('Geluidsisolatie
eis'),$);
[…]
#2711811= IFCWALLSTANDARDCASE('3YSSiVGqj6qv0WoAOq7oDa',#41,'Basic
Wall:300_22_binnenwand_100:4598433',$,'Basic
Wall:300_22_binnenwand_100',#2711791,#2711809,'343208');
#2711814= IFCQUANTITYLENGTH('Height',$,$,3330.);
#2711815= IFCQUANTITYLENGTH('Length',$,$,2581.99999999999);
#2711816= IFCQUANTITYLENGTH('Width',$,$,100.);
#2711817= IFCQUANTITYAREA('GrossSideArea',$,$,8.59805999999996);
#2711818= IFCQUANTITYAREA('GrossFootprintArea',$,$,0.258200000000007);
#2711819= IFCQUANTITYVOLUME('GrossVolume',$,$,0.859805999999987);
#2711820=
IFCELEMENTQUANTITY('3JZbtcP6bEDutFowZHE5E9',#41,'BaseQuantities',$,$,(#2711
814,#2711815,#2711816,#2711817,#2711818,#2711819));
```

```
#2711822=
IFCRELDEFINESBYPROPERTIES('3FEAVVCPH2iepxUxbCSOV1',#41,$,$,(#2711811),#2711
820);
#2711826= IFCMATERIALLAYERSETUSAGE(#2697007,.AXIS2.,.NEGATIVE.,50.);
#2711827= IFCPROPERTYSINGLEVALUE('Geluidsisolatie',$,IFCTEXT('39'),$);
#2711828= IFCPROPERTYSINGLEVALUE('Unconnected
Height',$,IFCLENGTHMEASURE(3330.),$);
#2711829=
IFCPROPERTYSINGLEVALUE('Area',$,IFCAREAMEASURE(8.59805999999996),$);
#2711830=
IFCPROPERTYSINGLEVALUE('Length',$,IFCLENGTHMEASURE(2581.99999999999),$);
#2711831=
IFCPROPERTYSINGLEVALUE('Volume',$,IFCVOLUMEMEASURE(0.859805999999987),$);
[…]
#2711890= IFCWALLSTANDARDCASE('3YSSiVGqj6qv0WoAOq7oAw',#41,'Basic
Wall:300_22_binnenwand_100:4598655',$,'Basic
Wall:300_22_binnenwand_100',#2711870,#2711888,'343211');
#2711893= IFCQUANTITYLENGTH('Height',$,$,3330.);
#2711894= IFCQUANTITYLENGTH('Length',$,$,2318.);
#2711895= IFCQUANTITYLENGTH('Width',$,$,100.);
#2711896= IFCQUANTITYAREA('GrossSideArea',$,$,7.71894000000001);
#2711897= IFCQUANTITYAREA('GrossFootprintArea',$,$,0.231800000000008);
#2711898= IFCQUANTITYVOLUME('GrossVolume',$,$,0.771893999999992);
#2711899=
IFCELEMENTQUANTITY('16xk8iye9AhOyLSWSeODQx',#41,'BaseQuantities',$,$,(#2711
893,#2711894,#2711895,#2711896,#2711897,#2711898));
#2711901=
IFCRELDEFINESBYPROPERTIES('3xhfVuARnF8eOe7ZMCm_a0',#41,$,$,(#2711890),#2711
899);
#2711905= IFCMATERIALLAYERSETUSAGE(#2697007,.AXIS2.,.NEGATIVE.,50.);
#2711906= IFCPROPERTYSINGLEVALUE('Geluidsisolatie',$,IFCTEXT('39'),$);
#2711907= IFCPROPERTYSINGLEVALUE('Unconnected
Height',$,IFCLENGTHMEASURE(3330.),$);
#2711908=
IFCPROPERTYSINGLEVALUE('Area',$,IFCAREAMEASURE(7.71894000000001),$);
#2711909= IFCPROPERTYSINGLEVALUE('Length',$,IFCLENGTHMEASURE(2318.),$);
#2711910=
IFCPROPERTYSINGLEVALUE('Volume',$,IFCVOLUMEMEASURE(0.771893999999992),$);
[…]
#2715754=
IFCOPENINGELEMENT('0666CFtfr1Xf1db2ZGsb0P',#41,'Floor:Floor:300_43_cementde
kvloer_70:343800:10',$,'Opening',#2715753,#2715748,'343800');
#2715757= IFCQUANTITYLENGTH('Depth','',$,69.9999999999998);
#2715758= IFCQUANTITYAREA('Area','',$,0.0198529962080658);
#2715759=
IFCELEMENTQUANTITY('0zho$aNZD9zRIvk_n1Ft2Q',#41,'BaseQuantities',$,$,(#2715
757,#2715758));
#2715761=
IFCRELDEFINESBYPROPERTIES('2_pti_Qa1A_PDoey5uDM4L',#41,$,$,(#2715754),#2715
759);
#2715765=
IFCRELVOIDSELEMENT('1tiWtHmPPBwusC1$DQz$ZH',#41,$,$,#2715352,#2715754);
#2715767= IFCMATERIALLAYERSETUSAGE(#2713581,.AXIS3.,.POSITIVE.,0.);
#2715768=
IFCPROPERTYSET('0666CFtfr1Xf1ddzFGsb5L',#41,'Pset_SlabCommon',$,(#1696525,#
2656833,#2713585));
#2715770=
IFCRELDEFINESBYPROPERTIES('3LWd3xV5b2KeiR8FvDS3qg',#41,$,$,(#2715352),#2715
768);
#2715774= IFCQUANTITYAREA('GrossArea','',$,508.057062502168);
#2715775= IFCQUANTITYLENGTH('Perimeter','',$,254597.350621694);
```

```
#2715776=
IFCELEMENTQUANTITY('1gpbhrM3HCNvoL59Je7C$n',#41,'BaseQuantities','',$,(#271
5774,#2715775));
#2715778=
IFCRELDEFINESBYPROPERTIES('0Tr8__1qD3Jvo3OLVeJsea',#41,$,$,(#2715352),#2715
776);
#2715781= IFCPROPERTYSINGLEVALUE('Geluidsisolatie',$,IFCTEXT('50'),$);
#2715782= IFCPROPERTYSINGLEVALUE('Height Offset From
Level',$,IFCLENGTHMEASURE(69.9999999999668),$);
#2715783=
IFCPROPERTYSINGLEVALUE('Area',$,IFCAREAMEASURE(492.080657086163),$);
#2715784= IFCPROPERTYSINGLEVALUE('Elevation at
Bottom',$,IFCLENGTHMEASURE(3530.00000000001),$);
#2715785= IFCPROPERTYSINGLEVALUE('Elevation at
Top',$,IFCLENGTHMEASURE(3600.00000000001),$);
#2715786=
IFCPROPERTYSINGLEVALUE('Perimeter',$,IFCLENGTHMEASURE(254597.350621694),$);
#2715787= IFCPROPERTYSINGLEVALUE('Thickness',$,IFCLENGTHMEASURE(70.),$);
#2715788=
IFCPROPERTYSINGLEVALUE('Volume',$,IFCVOLUMEMEASURE(34.4455035692666),$);
[…]
#2738236= IFCWALLSTANDARDCASE('0666CFtfr1Xf1db2ZGslsi',#41,'Basic
Wall:Basic Wall:300_22_binnenwand_100:365057',$,'Basic Wall:Basic
Wall:300_22_binnenwand_100:340999',#2738216,#2738234,'365057');
#2738239= IFCQUANTITYLENGTH('Height',$,$,3599.99999999988);
#2738240= IFCQUANTITYLENGTH('Length',$,$,2976.4999999999);
#2738241= IFCQUANTITYLENGTH('Width',$,$,100.);
#2738242= IFCQUANTITYAREA('GrossSideArea',$,$,8.76278999999928);
#2738243= IFCQUANTITYAREA('GrossFootprintArea',$,$,0.29765);
#2738244= IFCQUANTITYVOLUME('GrossVolume',$,$,0.876278999999919);
#2738245=
IFCELEMENTQUANTITY('2DIlzHS51FGA3tqM2wy8jj',#41,'BaseQuantities',$,$,(#2738
239,#2738240,#2738241,#2738242,#2738243,#2738244));
#2738247=
IFCRELDEFINESBYPROPERTIES('3zBoe1qvrBu8VbUYcvo7rw',#41,$,$,(#2738236),#2738
245);
#2738251= IFCMATERIALLAYERSETUSAGE(#2697007,.AXIS2.,.NEGATIVE.,50.);
#2738252= IFCPROPERTYSINGLEVALUE('Geluidsisolatie',$,IFCTEXT('39'),$);
#2738253= IFCPROPERTYSINGLEVALUE('Top Constraint',$,IFCLABEL('Level: V2 bk.
ruwe dakvloer'),$);
#2738254= IFCPROPERTYSINGLEVALUE('Unconnected
Height',$,IFCLENGTHMEASURE(3599.99999999988),$);
#2738255=
IFCPROPERTYSINGLEVALUE('Area',$,IFCAREAMEASURE(8.76278999999928),$);
#2738256=
IFCPROPERTYSINGLEVALUE('Length',$,IFCLENGTHMEASURE(2976.4999999999),$);
#2738257=
IFCPROPERTYSINGLEVALUE('Volume',$,IFCVOLUMEMEASURE(0.876278999999919),$);
[…]
#2738316= IFCWALLSTANDARDCASE('0666CFtfr1Xf1db2ZGsloa',#41,'Basic
Wall:300_22_kozijnrekje_buitenkozijnen:5399280',$,'Basic
Wall:300_22_kozijnrekje_buitenkozijnen',#2738296,#2738314,'365321');
#2738319= IFCQUANTITYLENGTH('Height',$,$,3330.);
#2738320= IFCQUANTITYLENGTH('Length',$,$,2725.37608058247);
#2738321= IFCQUANTITYLENGTH('Width',$,$,100.);
#2738322= IFCQUANTITYAREA('GrossSideArea',$,$,5.72660234833962);
#2738323= IFCQUANTITYAREA('GrossFootprintArea',$,$,0.272537608058256);
#2738324= IFCQUANTITYVOLUME('GrossVolume',$,$,0.572660234833956);
#2738325=
IFCELEMENTQUANTITY('3bXgwNABzExgUECKU4ePjE',#41,'BaseQuantities',$,$,(#2738
319,#2738320,#2738321,#2738322,#2738323,#2738324));
```

```
#2738327=
IFCRELDEFINESBYPROPERTIES('0OuAESqa54pfzLY6FNSpbG',#41,$,$,(#2738316),#2738
325);
#2738331= IFCMATERIALLAYER(#2696380,90.,$);
#2738332= IFCMATERIALLAYER(#2702707,10.,$);
#2738333= IFCMATERIALLAYERSET((#2738331,#2738332),'Basic Wall:Basic
Wall:300_22_kozijnrekje_buitenkozijnen');
#2738337= IFCMATERIALLAYERSETUSAGE(#2738333,.AXIS2.,.NEGATIVE.,50.);
#2738338= IFCWALLTYPE('0666CFtfr1Xf1db2ZGsbZQ',#41,'Basic Wall:Basic
Wall:300_22_kozijnrekje_buitenkozijnen',$,$,(#2738382,#2738384,#2738386,#27
38388,#2738390),$,'341879',$,.STANDARD.);
#2738339= IFCPROPERTYSINGLEVALUE('Geluidsisolatie',$,IFCTEXT('40'),$);
#2738340= IFCPROPERTYSINGLEVALUE('Unconnected
Height',$,IFCLENGTHMEASURE(3330.),$);
#2738341=
IFCPROPERTYSINGLEVALUE('Area',$,IFCAREAMEASURE(5.72660234833962),$);
#2738342=
IFCPROPERTYSINGLEVALUE('Length',$,IFCLENGTHMEASURE(2725.37608058247),$);
#2738343=
IFCPROPERTYSINGLEVALUE('Volume',$,IFCVOLUMEMEASURE(0.572660234833956),$);
#2738344= IFCPROPERTYSINGLEVALUE('Family',$,IFCLABEL('Basic Wall: Basic
Wall:300_22_kozijnrekje_buitenkozijnen'),$);
#2738345= IFCPROPERTYSINGLEVALUE('Family and Type',$,IFCLABEL('Basic Wall:
Basic Wall:300_22_kozijnrekje_buitenkozijnen'),$);
#2738346= IFCPROPERTYSINGLEVALUE('Type',$,IFCLABEL('Basic Wall: Basic
Wall:300_22_kozijnrekje_buitenkozijnen'),$);
#2738347= IFCPROPERTYSINGLEVALUE('Type Id',$,IFCLABEL('Basic Wall: Basic
Wall:300_22_kozijnrekje_buitenkozijnen'),$);
#2738348= IFCPROPERTYSINGLEVALUE('Absorptance',$,IFCREAL(0.1),$);
#2738349= IFCPROPERTYSINGLEVALUE('Width',$,IFCLENGTHMEASURE(100.),$);
#2738350= IFCPROPERTYSINGLEVALUE('Type Name',$,IFCTEXT('Basic
Wall:300_22_kozijnrekje_buitenkozijnen'),$);
#2738351= IFCPROPERTYSINGLEVALUE('Family Name',$,IFCTEXT('Basic Wall'),$);
[…]
#2738427= IFCWALLSTANDARDCASE('0666CFtfr1Xf1db2ZGslnE',#41,'Basic
Wall:Basic Wall:300_22_binnenwand_100:365539',$,'Basic Wall:Basic
Wall:300_22_binnenwand_100:340999',#2738407,#2738425,'365539');
#2738430= IFCQUANTITYLENGTH('Height',$,$,3599.99999999988);
#2738431= IFCQUANTITYLENGTH('Length',$,$,2876.49999999989);
#2738432= IFCQUANTITYLENGTH('Width',$,$,100.);
#2738433= IFCQUANTITYAREA('GrossSideArea',$,$,8.40278999999927);
#2738434= IFCQUANTITYAREA('GrossFootprintArea',$,$,0.287649999999999);
#2738435= IFCQUANTITYVOLUME('GrossVolume',$,$,0.840278999999918);
#2738436=
IFCELEMENTQUANTITY('1CB2gymJvBJuqCtmQtVkfk',#41,'BaseQuantities',$,$,(#2738
430,#2738431,#2738432,#2738433,#2738434,#2738435));
#2738438=
IFCRELDEFINESBYPROPERTIES('3h0hpTZ4nCTQanU8gl_qfH',#41,$,$,(#2738427),#2738
436);
#2738442= IFCMATERIALLAYERSETUSAGE(#2697007,.AXIS2.,.NEGATIVE.,50.);
#2738443= IFCPROPERTYSINGLEVALUE('Geluidsisolatie',$,IFCTEXT('39'),$);
#2738444= IFCPROPERTYSINGLEVALUE('Unconnected
Height',$,IFCLENGTHMEASURE(3599.99999999988),$);
#2738445=
IFCPROPERTYSINGLEVALUE('Area',$,IFCAREAMEASURE(8.40278999999927),$);
#2738446=
IFCPROPERTYSINGLEVALUE('Length',$,IFCLENGTHMEASURE(2976.49999999989),$);
#2738447=
IFCPROPERTYSINGLEVALUE('Volume',$,IFCVOLUMEMEASURE(0.840278999999918),$);
[…]
```

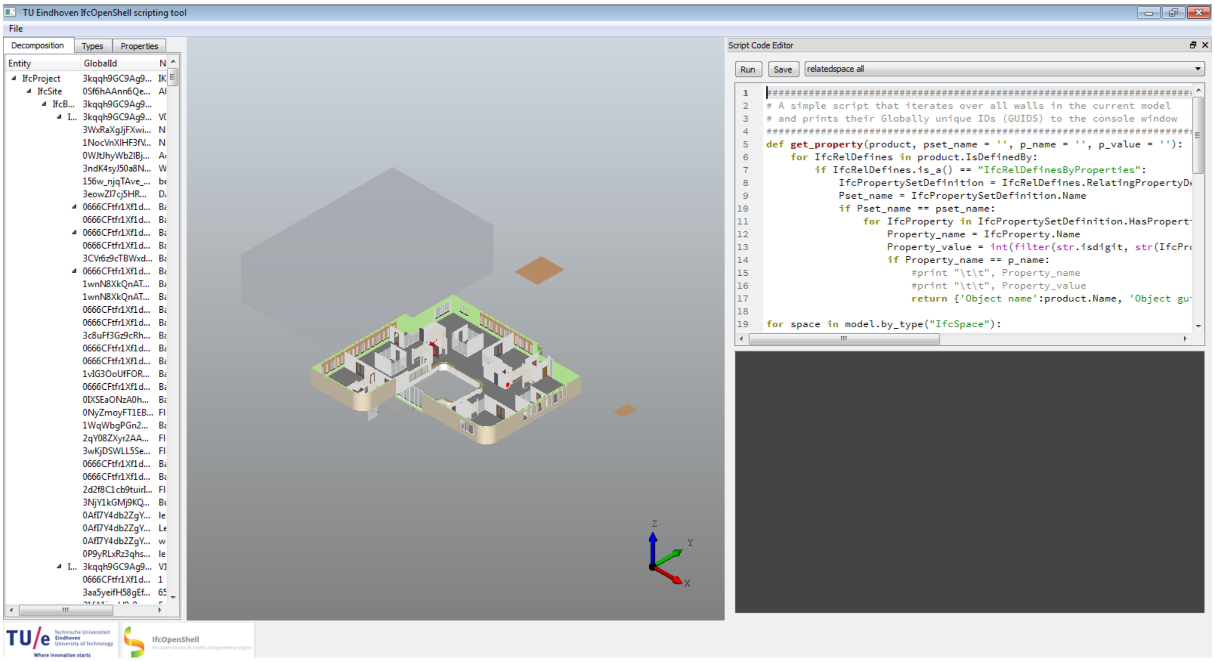
```
#2738642= IFCDOOR('0666CFtfr1Xf1db2ZGslbq',#41,'M_Single-Flush:0915 x 2134mm:366297',$,'0915 x 2134mm',#2875668,#2738636,'366297',2134.,915.);
#2738645= IFCMATERIALLIST((#2738606,#2738616));
#2738647= IFCPROPERTYSINGLEVALUE('Reference',$,IFCIDENTIFIER('0915 x 2134mm'),$);
#2738648= IFCPROPERTYSINGLEVALUE('ThermalTransmittance',$,IFCTHERMALTRANSMITTANCEMEASURE(3.7021),$);
#2738649= IFCPROPERTYSET('0666CFtfr1Xf1ddzZGslbq',#41,'Pset_DoorCommon',$,(#1696525,#2738647,#2738648));
#2738651= IFCRELDEFINESBYPROPERTIES('3yt0LRwYnC$8eorQ941eeN',#41,$,$,(#2738642),#2738649);
#2738655= IFCQUANTITYLENGTH('Height','',$,2134.);
#2738656= IFCQUANTITYLENGTH('Width','',$,915.);
#2738657= IFCQUANTITYAREA('Area','area measured in geometry',$,3.18957899999998);
#2738658= IFCELEMENTQUANTITY('3j$f765ZL9f9gMXmRkfW8h',#41,'BaseQuantities','',$,(#2738655,#2738656,#2738657));
#2738660= IFCRELDEFINESBYPROPERTIES('000lFa8E15pxljJiKgI7Mf',#41,$,$,(#2738642),#2738658);
#2738663= IFCPROPERTYSINGLEVALUE('Geluidsisolatie',$,IFCTEXT('37'),$);
#2738664= IFCPROPERTYSINGLEVALUE('Level',$,IFCLABEL('Level: V1 bk. ruwe vloer'),$);
#2738665= IFCPROPERTYSINGLEVALUE('Sill Height',$,IFCLENGTHMEASURE(0.),$);
#2738666= IFCPROPERTYSINGLEVALUE('Area',$,IFCAREAMEASURE(3.18957899999998),$);
#2738667= IFCPROPERTYSINGLEVALUE('Volume',$,IFCVOLUMEMEASURE(0.119856109999999),$);
#2738668= IFCPROPERTYSINGLEVALUE('Mark',$,IFCTEXT('307'),$);
#2738669= IFCPROPERTYSINGLEVALUE('Family',$,IFCLABEL('M_Single-Flush: 0915 x 2134mm'),$);
#2738670= IFCPROPERTYSINGLEVALUE('Family and Type',$,IFCLABEL('M_Single-Flush: 0915 x 2134mm'),$);
#2738671= IFCPROPERTYSINGLEVALUE('Head Height',$,IFCLENGTHMEASURE(2134.),$);
#2738672= IFCPROPERTYSINGLEVALUE('Host Id',$,IFCLABEL('Basic Wall: Basic Wall:300_22_binnenwand_100'),$);
#2738673= IFCPROPERTYSINGLEVALUE('Type',$,IFCLABEL('M_Single-Flush: 0915 x 2134mm'),$);
#2738674= IFCPROPERTYSINGLEVALUE('Type Id',$,IFCLABEL('M_Single-Flush: 0915 x 2134mm'),$);
#2738675= IFCPROPERTYSINGLEVALUE('Analytic Construction',$,IFCTEXT('Metal'),$);
#2738676= IFCPROPERTYSINGLEVALUE('Heat Transfer Coefficient (U)',$,IFCREAL(3.7021),$);
#2738677= IFCPROPERTYSINGLEVALUE('Solar Heat Gain Coefficient',$,IFCREAL(0.),$);
#2738678= IFCPROPERTYSINGLEVALUE('Thermal Resistance (R)',$,IFCREAL(0.270116960643959),$);
#2738679= IFCPROPERTYSINGLEVALUE('Visual Light Transmittance',$,IFCREAL(0.),$);
#2738680= IFCPROPERTYSINGLEVALUE('Door Material',$,IFCLABEL('Door - Panel'),$);
#2738681= IFCPROPERTYSINGLEVALUE('Frame Material',$,IFCLABEL('Door - Frame'),$);
#2738682= IFCPROPERTYSINGLEVALUE('Wall Closure',$,IFCIDENTIFIER('By host'),$);
```

```
#2738683= IFCPROPERTYSINGLEVALUE('Height',$,IFCLENGTHMEASURE(2134.),$);
#2738684= IFCPROPERTYSINGLEVALUE('Thickness',$,IFCLENGTHMEASURE(51.),$);
#2738685= IFCPROPERTYSINGLEVALUE('Trim Projection
Ext',$,IFCLENGTHMEASURE(25.),$);
#2738686= IFCPROPERTYSINGLEVALUE('Trim Projection
Int',$,IFCLENGTHMEASURE(25.),$);
#2738687= IFCPROPERTYSINGLEVALUE('Trim Width',$,IFCLENGTHMEASURE(76.),$);
#2738688= IFCPROPERTYSINGLEVALUE('Width',$,IFCLENGTHMEASURE(915.),$);
#2738689= IFCPROPERTYSINGLEVALUE('OmniClass
Number',$,IFCTEXT('23.30.10.00'),$);
#2738690= IFCPROPERTYSINGLEVALUE('OmniClass Title',$,IFCTEXT('Doors'),$);
#2738691= IFCPROPERTYSINGLEVALUE('Type Mark',$,IFCTEXT('10'),$);
#2738692= IFCPROPERTYSINGLEVALUE('Type Name',$,IFCTEXT('0915 x 2134mm'),$);
#2738693= IFCPROPERTYSINGLEVALUE('Family Name',$,IFCTEXT('M_Single-
Flush'),$);
[…]
#2738869= IFCDOOR('0666CFtfr1Xf1db2ZGslYr',#41,'M_Single-Flush:0915 x
2134mm:366360',$,'0915 x 2134mm',#2875777,#2738863,'366360',2134.,915.);
#2738872= IFCMATERIALLIST((#2738606,#2738616));
#2738874=
IFCPROPERTYSINGLEVALUE('ThermalTransmittance',$,IFCTHERMALTRANSMITTANCEMEAS
URE(3.7021),$);
#2738875=
IFCPROPERTYSET('0666CFtfr1Xf1ddzZGslYr',#41,'Pset_DoorCommon',$,(#1696525,#
2738647,#2738874));
#2738877=
IFCRELDEFINESBYPROPERTIES('36UvBjT3r7feSiMJwNXp4D',#41,$,$,(#2738869),#2738
875);
#2738881= IFCQUANTITYLENGTH('Height','',$,2134.);
#2738882= IFCQUANTITYLENGTH('Width','',$,915.);
#2738883= IFCQUANTITYAREA('Area','area measured in
geometry',$,3.18957899999998);
#2738884=
IFCELEMENTQUANTITY('0XReOtnHfEjf6oyi5lPoN0',#41,'BaseQuantities','',$,(#273
8881,#2738882,#2738883));
#2738886=
IFCRELDEFINESBYPROPERTIES('3gGs$_9aP3N8L4OKJeblaO',#41,$,$,(#2738869),#2738
884);
#2738889= IFCPROPERTYSINGLEVALUE('Geluidsisolatie',$,IFCTEXT('37'),$);
#2738890=
IFCPROPERTYSINGLEVALUE('Area',$,IFCAREAMEASURE(3.18957899999998),$);
#2738891=
IFCPROPERTYSINGLEVALUE('Volume',$,IFCVOLUMEMEASURE(0.119856109999999),$);
#2738892= IFCPROPERTYSINGLEVALUE('Mark',$,IFCTEXT('308'),$);
#2738893= IFCPROPERTYSINGLEVALUE('Head
Height',$,IFCLENGTHMEASURE(2134.),$);
[…]
#2739060= IFCWINDOW('0666CFtfr1Xf1db2ZGslZk',#41,'M_Fixed:0915 x
1830mm:366403',$,'0915 x 1830mm',#2875705,#2739054,'366403',1830.,915.);
#2739063= IFCMATERIALLIST((#1696473,#2739038));
#2739065= IFCPROPERTYSINGLEVALUE('Reference',$,IFCIDENTIFIER('0915 x
1830mm'),$);
#2739066=
IFCPROPERTYSINGLEVALUE('ThermalTransmittance',$,IFCTHERMALTRANSMITTANCEMEAS
URE(3.6886),$);
#2739067=
IFCPROPERTYSET('0666CFtfr1Xf1ddzRGslZk',#41,'Pset_WindowCommon',$,(#1696525
,#2739065,#2739066));
#2739069=
IFCRELDEFINESBYPROPERTIES('3pZ7GZeyX8MOCnpFiDkO1K',#41,$,$,(#2739060),#2739
067);
```

```
#2739073= IFCQUANTITYLENGTH('Height','',$,1830.);
#2739074= IFCQUANTITYLENGTH('Width','',$,915.);
#2739075= IFCQUANTITYAREA('Area','area measured in
geometry',$,2.64255400000013);
#2739076=
IFCELEMENTQUANTITY('0ufcodsjrDcB$oL1M3Nr1a',#41,'BaseQuantities','',$,(#273
9073,#2739074,#2739075));
#2739078=
IFCRELDEFINESBYPROPERTIES('3FBPD9Mz93kRMkjHFdic2g',#41,$,$,(#2739060),#2739
076);
#2739081= IFCPROPERTYSINGLEVALUE('Geluidsisolatie',$,IFCTEXT('32'),$);
#2739082= IFCPROPERTYSINGLEVALUE('Sill
Height',$,IFCLENGTHMEASURE(305.000000000004),$);
#2739083=
IFCPROPERTYSINGLEVALUE('Area',$,IFCAREAMEASURE(2.64255400000013),$);
#2739084=
IFCPROPERTYSINGLEVALUE('Volume',$,IFCVOLUMEMEASURE(0.0389562700000035),$);
#2739085= IFCPROPERTYSINGLEVALUE('Mark',$,IFCTEXT('341'),$);
#2739086= IFCPROPERTYSINGLEVALUE('Family',$,IFCLABEL('M_Fixed: 0915 x
1830mm'),$);
#2739087= IFCPROPERTYSINGLEVALUE('Family and Type',$,IFCLABEL('M_Fixed:
0915 x 1830mm'),$);
#2739088= IFCPROPERTYSINGLEVALUE('Head
Height',$,IFCLENGTHMEASURE(2135.),$);
#2739089= IFCPROPERTYSINGLEVALUE('Host Id',$,IFCLABEL('Basic Wall: Basic
Wall:300_22_kozijnrekje_buitenkozijnen'),$);
#2739090= IFCPROPERTYSINGLEVALUE('Type',$,IFCLABEL('M_Fixed: 0915 x
1830mm'),$);
#2739091= IFCPROPERTYSINGLEVALUE('Type Id',$,IFCLABEL('M_Fixed: 0915 x
1830mm'),$);
#2739092= IFCPROPERTYSINGLEVALUE('Analytic Construction',$,IFCTEXT('1/8 in
Pilkington single glazing'),$);
#2739093= IFCPROPERTYSINGLEVALUE('Heat Transfer Coefficient
(U)',$,IFCREAL(3.6886),$);
#2739094= IFCPROPERTYSINGLEVALUE('Solar Heat Gain
Coefficient',$,IFCREAL(0.78),$);
#2739095= IFCPROPERTYSINGLEVALUE('Thermal Resistance
(R)',$,IFCREAL(0.271105568508377),$);
#2739096= IFCPROPERTYSINGLEVALUE('Visual Light
Transmittance',$,IFCREAL(0.9),$);
#2739097= IFCPROPERTYSINGLEVALUE('Frame Exterior
Material',$,IFCLABEL('Sash'),$);
#2739098= IFCPROPERTYSINGLEVALUE('Frame Interior
Material',$,IFCLABEL('Sash'),$);
#2739099= IFCPROPERTYSINGLEVALUE('Glass Pane
Material',$,IFCLABEL('Glass'),$);
#2739100= IFCPROPERTYSINGLEVALUE('Sash',$,IFCLABEL('Sash'),$);
#2739101= IFCPROPERTYSINGLEVALUE('Default Sill
Height',$,IFCLENGTHMEASURE(305.),$);
#2739102= IFCPROPERTYSINGLEVALUE('Height',$,IFCLENGTHMEASURE(1830.),$);
#2739103= IFCPROPERTYSINGLEVALUE('Width',$,IFCLENGTHMEASURE(915.),$);
#2739104= IFCPROPERTYSINGLEVALUE('Window Inset',$,IFCLENGTHMEASURE(19.),$);
#2739105= IFCPROPERTYSINGLEVALUE('OmniClass
Number',$,IFCTEXT('23.30.20.17.11'),$);
#2739106= IFCPROPERTYSINGLEVALUE('OmniClass Title',$,IFCTEXT('Fixed
Windows'),$);
#2739107= IFCPROPERTYSINGLEVALUE('Type Mark',$,IFCTEXT('18'),$);
#2739108= IFCPROPERTYSINGLEVALUE('Type Name',$,IFCTEXT('0915 x 1830mm'),$);
#2739109= IFCPROPERTYSINGLEVALUE('Family Name',$,IFCTEXT('M_Fixed'),$);
[…]
```

```
#2739168= IFCWINDOW('0666CFtfr1Xf1db2ZGslZV',#41,'M_Fixed:0915 x 
1830mm:366450',$,'0915 x 1830mm',#2875740,#2739162,'366450',1830.,915.);
#2739171= IFCMATERIALLIST((#1696473,#2739038));
#2739173= 
IFCPROPERTYSINGLEVALUE('ThermalTransmittance',$,IFCTHERMALTRANSMITTANCEMEAS
URE(3.6886),$);
#2739174= 
IFCPROPERTYSET('0666CFtfr1Xf1ddzRGslZV',#41,'Pset_WindowCommon',$,(#1696525
,#2739065,#2739173));
#2739176= 
IFCRELDEFINESBYPROPERTIES('3E7ZZHtAHCCPrD7yzB8ekG',#41,$,$,(#2739168),#2739
174);
#2739180= IFCQUANTITYLENGTH('Height','',$,1830.);
#2739181= IFCQUANTITYLENGTH('Width','',$,915.);
#2739182= IFCQUANTITYAREA('Area','area measured in 
geometry',$,2.64255400000013);
#2739183= 
IFCELEMENTQUANTITY('1on_wtJ1z6Zx2X_iQqTxFT',#41,'BaseQuantities','',$,(#273
9180,#2739181,#2739182));
#2739185= 
IFCRELDEFINESBYPROPERTIES('20BVPKzVPFbg0iIEBIypfM',#41,$,$,(#2739168),#2739
183);
#2739188= IFCPROPERTYSINGLEVALUE('Geluidsisolatie',$,IFCTEXT('32'),$);
#2739189= IFCPROPERTYSINGLEVALUE('Sill 
Height',$,IFCLENGTHMEASURE(305.000000000004),$);
#2739190= 
IFCPROPERTYSINGLEVALUE('Area',$,IFCAREAMEASURE(2.64255400000013),$);
#2739191= 
IFCPROPERTYSINGLEVALUE('Volume',$,IFCVOLUMEMEASURE(0.0389562700000035),$);
#2739192= IFCPROPERTYSINGLEVALUE('Mark',$,IFCTEXT('342'),$);
#2739193= IFCPROPERTYSINGLEVALUE('Head 
Height',$,IFCLENGTHMEASURE(2135.),$);
```

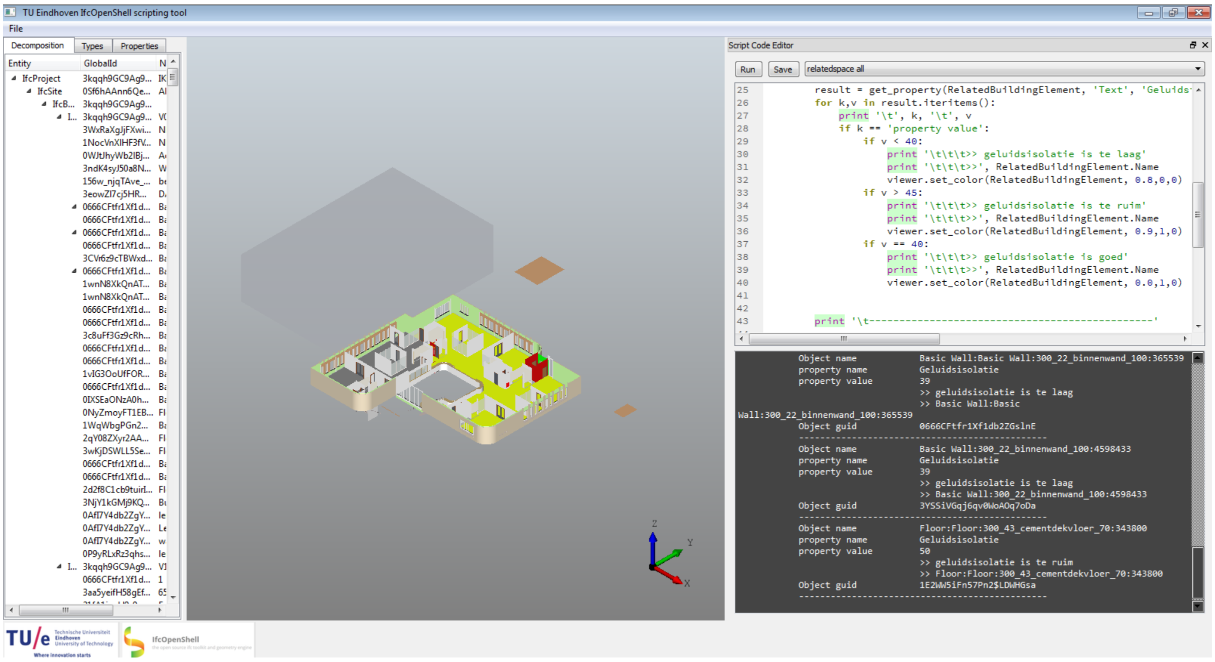
## 9.8 Script Verification Tool




```
def get_property(product, pset_name = '', p_name = '', p_value =
''):
for IfcRelDefines in product.IsDefinedBy:
if IfcRelDefines.is_a() == "IfcRelDefinesByProperties":
IfcPropertySetDefinition = IfcRelDefines.RelatingPropertyDefinition
Pset_name = IfcPropertySetDefinition.Name
if Pset_name == pset_name:
for IfcProperty in IfcPropertySetDefinition.HasProperties:
Property_name = IfcProperty.Name
Property_value = int(filter(str.isdigit,
str(IfcProperty.NominalValue)))
if Property_name == p_name:
#print "\t\t", Property_name
#print "\t\t", Property_value
return {'Object name':product.Name, 'Object guid': product.GlobalId,
'property name':Property_name, 'property value' :Property_value}

for space in model.by_type("IfcSpace"):
print ("space with global id: "+str(space.GlobalId))
print space, '\n'
for IfcRelSpaceBoundary in space.BoundedBy:
#print "\t", IfcRelSpaceBoundary
RelatedBuildingElement = IfcRelSpaceBoundary.RelatedBuildingElement
result = get_property(RelatedBuildingElement, 'Text',
'Geluidsisolatie')
for k,v in result.iteritems():
print '\t', k, '\t', v
if k == 'property value':
if v < 40:
print '\t\t\t>> geluidsisolatie is te laag'
print '\t\t\t>>', RelatedBuildingElement.Name
```

```
viewer.set_color(RelatedBuildingElement, 0.8,0,0)
if v > 45:
print '\t\t\t>> geluidsisolatie is te ruim'
print '\t\t\t>>', RelatedBuildingElement.Name
viewer.set_color(RelatedBuildingElement, 0.9,1,0)
if v == 40:
print '\t\t\t>> geluidsisolatie is goed'
print '\t\t\t>>', RelatedBuildingElement.Name
viewer.set_color(RelatedBuildingElement, 0.0,1,0)
print '\t----------------------------------------------'
```



```
space with global id: 0666CFtfr1Xf1db2ZGslbu
#147=IfcSpace('0666CFtfr1Xf1db2ZGslbu',#41,'1',$,$,#130,#143,'Space'
,.ELEMENT.,.INTERNAL.,$)

Object name M_Single-Flush:0915 x 2134mm:366297
property name Geluidsisolatie
property value 37
>> geluidsisolatie is te laag
>> M_Single-Flush:0915 x 2134mm:366297
Object guid 0666CFtfr1Xf1db2ZGslbq
----------------------------------------------
Object name Basic Wall:Basic Wall:300_22_binnenwand_100:365057
property name Geluidsisolatie
property value 39
>> geluidsisolatie is te laag
>> Basic Wall:Basic Wall:300_22_binnenwand_100:365057
Object guid 0666CFtfr1Xf1db2ZGslsi
----------------------------------------------
Object name Basic Wall:300_22_kozijnrekje_buitenkozijnen:5399280
property name Geluidsisolatie
property value 40
>> geluidsisolatie is goed
```

```
>> Basic Wall:300_22_kozijnrekje_buitenkozijnen:5399280
Object guid 0666CFtfr1Xf1db2ZGsloa
-----------------------------------------------
Object name M_Fixed:0915 x 1830mm:366450
property name Geluidsisolatie
property value 32
>> geluidsisolatie is te laag
>> M_Fixed:0915 x 1830mm:366450
Object guid 0666CFtfr1Xf1db2ZGslZV
-----------------------------------------------
Object name M_Fixed:0915 x 1830mm:366403
property name Geluidsisolatie
property value 32
>> geluidsisolatie is te laag
>> M_Fixed:0915 x 1830mm:366403
Object guid 0666CFtfr1Xf1db2ZGslZk
-----------------------------------------------
Object name Basic Wall:300_22_kozijnrekje_buitenkozijnen:5399280
property name Geluidsisolatie
property value 40
>> geluidsisolatie is goed
>> Basic Wall:300_22_kozijnrekje_buitenkozijnen:5399280
Object guid 0666CFtfr1Xf1db2ZGsloa
-----------------------------------------------
Object name Basic Wall:300_22_kozijnrekje_buitenkozijnen:5399280
property name Geluidsisolatie
property value 40
>> geluidsisolatie is goed
>> Basic Wall:300_22_kozijnrekje_buitenkozijnen:5399280
Object guid 0666CFtfr1Xf1db2ZGsloa
-----------------------------------------------
Object name Basic Wall:Basic Wall:300_22_binnenwand_100:365539
property name Geluidsisolatie
property value 39
>> geluidsisolatie is te laag
>> Basic Wall:Basic Wall:300_22_binnenwand_100:365539
Object guid 0666CFtfr1Xf1db2ZGslnE
-----------------------------------------------
Object name M_Single-Flush:0915 x 2134mm:366360
property name Geluidsisolatie
property value 37
>> geluidsisolatie is te laag
>> M_Single-Flush:0915 x 2134mm:366360
Object guid 0666CFtfr1Xf1db2ZGslYr
-----------------------------------------------
Object name Basic Wall:Basic Wall:300_22_binnenwand_100:365539
property name Geluidsisolatie
property value 39
>> geluidsisolatie is te laag
>> Basic Wall:Basic Wall:300_22_binnenwand_100:365539
Object guid 0666CFtfr1Xf1db2ZGslnE
-----------------------------------------------
Object name Basic Wall:300_22_binnenwand_100:4598433
property name Geluidsisolatie
property value 39
```

```
>> geluidsisolatie is te laag
>> Basic Wall:300_22_binnenwand_100:4598433
Object guid 3YSSiVGqj6qv0WoAOq7oDa
-----------------------------------------------
Object name Floor:Floor:300_43_cementdekvloer_70:343800
property name Geluidsisolatie
property value 50
>> geluidsisolatie is te ruim
>> Floor:Floor:300_43_cementdekvloer_70:343800
Object guid 1E2WW5iFn57Pn2$LDWHGsa
-----------------------------------------------
```