Technische Universität München
Fakultät für Architektur
Lehrstuhl für Architekturinformatik

Dissertation

A Framework for Using BIM in Mass-Customization and Prefabrication in the AEC Industry

Mohammed Refaat Mekawy Mohammed

Technische Universität München

Fakultät für Architektur

Lehrstuhl für Architekturinformatik

A Framework for Using BIM in Mass-Customization and Prefabrication in the AEC Industry

Mohammed Refaat Mekawy Mohammed

Vollständiger Abdruck der von der Fakultät für Architektur der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzende: Prof. Dr. Kathrin Dörfler

Prüfer der Dissertation: 1. Prof. Dr.-Ing. Frank Petzold
2. Prof. Dr. Eng. Zhilang Ma

Die Dissertation wurde am 27.05.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Architektur am 15.09.2020 angenommen.

Acknowledgments

The presented work was created during my research stay as a doctoral student at the chair of Architectural Informatics at the Technical University of Munich. I wish to express my gratitude to all the people whose support, assistance, and guidance was essential in the completion of this project.

First of all, I wish to express my deepest gratitude to my doctoral supervisor Prof. Dr.-Ing. Frank Petzold whose expertise, guidance, thoughtful critique, and support have been invaluable throughout this study, as well as on the personal level. I would also like to extend my gratitude to Prof. Dr.-Ing. Zhilang Ma for the thoughtful comments and recommendations regarding this dissertation.

I wish to acknowledge the support, both financial and personal, of the German Academic Exchange Service (DAAD), the Egyptian Ministry of Higher Education and Scientific Research, and the Technical University of Munich. Without their support, this journey would not have been possible.

Appreciation is also due to all my colleagues at the chair of Architectural Informatics. In particular, I would like to thank Dr.-Ing. Gerhard Schubert and Dr.-Ing. Christoph Langenhan for their insight and suggestions. Thanks are also due to Seongki Lee, Ata Zahedi, Ivan Bratoev, Sarah Jenney, Klaus Puchta, Michael Mühlhaus, and Nils Seifert. Their help, encouragement, and the discussions we had were deeply influential and inspiring. I am also indebted to my friend Mahmoud Islam whose advice, assistance, and sense of humor was crucial, especially during harder times.

Finally, my wholehearted appreciation goes without reservation to my family for all the support they have offered me through this research. My father Refaat and my mother Fatma to whom I am forever indebted for the opportunities they provided me with, for their invaluable advice, and for their incredible care, even when great distances kept us apart. My Brother Ahmed and my sister Ghada for their unyielding love and support. And last but not least, my wife Noor, my number one fan, who has stood by me through thick and thin and always kept my sanity in check.

Abstract

Design, pre-construction, and construction activities are becoming increasingly complex. Additionally, the construction industry as a whole still lags sharply in productivity behind other industries. One answer to this lag is to standardize and streamline the production of building components by moving the manufacturing process from the building site to the factory, a process commonly known as “Prefabrication”. Yet despite the benefits reaped from such approach, including improved quality, saved time, better accuracy, we still face the problem that the current construction process often relies on archaic business practices that intentionally segregate the design and construction activities. This is a problem that affects the entire spectrum of construction methods, including prefabrication. A dire consequence of such practice is the lack of expert input regarding prefabrication and construction soundness in earlier design phases, where the most important design decisions are made. Furthermore, manual checking for prefabrication considerations using pen-and-paper checklists proves to be costly in terms of time and money, especially in the earlier phases, where rapid design changes are the norm.

To approach this problem, this research focuses on precast concrete and proposes an expert system framework to check for prefabrication requirements that can be considered in earlier design stages. The idea is to exploit the data-rich Building Information Modeling design environment to obtain the design information needed to automatically perform checks to determine whether the design scheme in question fulfills prefabrication prerequisites or not. The checks are based on stored rule-sets that are curated and formulated from literature, human experts, and national and international standards. A preliminary prototype was developed to automatically perform these checks and was able to generate a report that highlights some early design issues that violate prefabrication considerations, presenting these results to the designer who still has the upper hand in determining the course of action. The idea is not to replace human experts, but to add an additional layer of “Intelligence” to design models, particularly Building Information Modeling design models, concerning construction and prefabrication limitations.

Keywords: Precast Concrete; Rule-based expert systems; Building Information Modeling; Model checking; Decision tables.

Zusammenfassung

Planungs- und Bautätigkeiten werden immer komplexer. Darüber hinaus liegt die Produktivität der Bauindustrie insgesamt immer noch stark hinter anderen Industriebranchen zurück. Eine Antwort auf dieses Problem besteht darin, die Produktion von Bauteilen zu standardisieren und zu rationalisieren, indem der Herstellungsprozess von der Baustelle in die Fabrik verlagert wird, was allgemein als „Vorfertigung" bekannt ist. Trotz der Vorteile, die sich aus einem solchen Ansatz ergeben, einschließlich verbesserter Qualität, Zeitersparnis und besserer Genauigkeit, stehen wir immer noch vor dem Problem, dass der aktuelle Bauprozess häufig auf archaischen Geschäftspraktiken beruht, die die Entwurfs- und Bautätigkeiten absichtlich voneinander trennen. Dies ist ein Problem, das das gesamte Spektrum der Bauweisen, einschließlich der Vorfertigung, betrifft. Eine schlimme Folge dieser Praxis ist der Mangel an fachkundigem Input in Bezug auf Vorfertigung und Konstruktionsfestigkeit in früheren Entwurfsphasen, in denen die wichtigsten Entwurfsentscheidungen getroffen werden. Darüber hinaus erweist sich die manuelle Überprüfung auf Konstruktionsüberlegungen mithilfe von Stift-Papier-Checklisten als zeit- und kostenintensiv, insbesondere in früheren Phasen, in denen schnelle Entwurfsänderungen die Norm sind.

Um dieses Problem anzugehen, konzentriert sich diese Forschung auf Betonfertigteile und schlägt ein Expertensystem vor, um die Vorfertigungsanforderungen zu überprüfen, die in früheren Entwurfsphasen berücksichtigt werden können. Die Idee besteht darin, die datenreiche Entwurfsumgebung in Bauwerkinformationsmodellen zu nutzen, um die Entwurfsinformationen zu erhalten, die für die automatische Überprüfung erforderlich sind, um festzustellen, ob das betreffende Entwurfsschema die Voraussetzungen für die Vorfertigung erfüllt oder nicht. Die Überprüfungen basieren auf gespeicherten Regelsätzen, die aus Literatur, menschlichen Experten sowie nationalen und internationalen Standards zusammengestellt und formuliert wurden. Ein vorläufiger Prototyp wurde entwickelt, um diese Überprüfungen automatisch durchzuführen, und konnte einen Bericht erstellen, der einige frühe Entwurfsprobleme hervorhebt, die gegen Vorfertigungsanforderungen verstoßen. Das Ziel ist nicht, menschliche Experten zu ersetzen, sondern den Entwurfsmodellen, insbesondere den Bauwerkinformationsmodellen, eine zusätzliche Ebene der „Intelligenz" hinzuzufügen, was die Einschränkungen bei Konstruktion und Vorfertigung betrifft. Diese Ebene kann dem Planer helfen, fundiertere Entwurfsentscheidungen zu treffen.

Table of contents

List of figures

List of Tables

List of Acronyms

AEC	Architecture, Engineering and Construction
AI	Artificial Intelligence
BIM	Building Information Modeling
BMC	BIM-based Model Checking
BOQ	Bill Of Quantities
CAD	Computer Aided Design
DBB	Design-Bid-Build
DIN	Deutsches Institut für Normung *[The German Institute for Standardization]*
ETO	Engineered-to-order
FDB	Fachvereinigung Deutscher Betonfertigteilbau *[The German association for precast concrete construction]*
FIP	The Fédération Internationale de la Précontrainte *[International Federation for Prestressing]*
GSA	General Services Administration (USA)
GUI	Graphical user Interface
HOAI	Honorarordnung für Architekten und Ingenieure *[Official scale of fees for services by architects & engineers]*
ICT	Information and Communications Technology
IFC	Industry Foundation Classes
IPD	Integrated Project Delivery
LOD	Level Of Development
OAV	Object-Attribute-Value
QTO	Quantity Take-Off
StVO	Straßenverkehrs-Ordnung *[The German Road Traffic Act]*
VCCL	Visual Code Checking Language
VOB	Vergabe- und Vertragsordnung für Bauleistungen *[The German construction contract procedures]*

Chapter 1: Introduction

1.1. Background

In today’s world, where there is an ever-increasing need for a faster, more efficient, and safer building construction environment, it is becoming more reasonable to adopt prefabrication and mass-customization in construction. Controlled production, improved job site safety, increased accuracy, and savings in schedule and cost are just a few examples of the benefits brought to the construction industry (Bachmann and Steinle, 2011, pp.14-15, Polat, 2008, pp.169-178). The idea, however, is not new and dates back to the nineteenth century in England all the way through the two world wars to the present day (Elliott and Jolly, 2013, p.1). Throughout these years, there has been undoubtedly great progress in the field of construction technology and prefabrication that enabled much more complicated structures to be built in much less time. In this regard, it is usually the responsibility of the architect to lead such an increasingly complex design and construction environment. The architect has to continuously create and evaluate design variants while lacking the expertise in various design-related fields (Zahedi and Petzold, 2018, pp.11-20). To be able to undertake such a task, the architect needs timely, well-informed expert input in a myriad of disciplines, including prefabrication expertise, throughout the phases of the project. Lack of such input in a timely fashion leads to conflicts and significant rework in later stages (Jiang, 2016, p.2).

But despite the great strides in the construction hardware and software sectors and the increased adoption of prefabrication and factory-built building parts, there has been a consistent stagnation of productivity in the construction sector, which has been attributed partially to the fragmentation of design and construction processes (Teicholz, 2013, p.9). Problems such as lack of coordination, design rework, and change orders can often arise from design decisions that lacked the essential knowledge of how a building is constructed on-site (Fischer and Tatum, 1997, pp.253-260). The problems could even be more apparent in a prefabrication scheme, where factors such as factory production, transportation, and mass-customization come into play. Manual reviews using checklists are sometimes implemented, but the large amount of resources required to undertake these tasks usually hinders such reviews (Jiang, 2016, p.16). Intervention of human prefabrication experts to manually review designs prior to construction could be another solution, but that is not always a viable solution due to the lack of such experts in design firms (Arditi et al., 2000, pp.79-86).

Among the innovations in the field of design and construction software that gained traction in the last decade is Building Information Modeling (BIM). It rapidly gained acceptance as the preferred medium to model and represent project data where the designers can efficiently communicate information to other team members, extract consistent sets of data, and most importantly, to plug this rich repository of project information to external tools and utilities to perform all sorts of automated analyses and calculations needed by the project team. Prior to BIM, in traditional Computer Aided Design (CAD) tools, the improvements in accuracy and productivity over manual checks were still not enough, because the cost of translating data from the graphic and textual building representation to analyses applications almost negates the effect of automation and is prone to human error (Sacks et al., 2004, pp.291-312). When BIM is used in earlier phases in design environments, it requires a lot of work and an intensive thought process upfront, but it also brings problematic design issues to the attention of the design team early on, so they can be solved before being further complicated down the line (Sacks et al., 2010, pp.419-432).

Utilizing the parametric, data-rich nature of BIM in automating modeling and checking tasks, especially for the construction and prefabrication sector can present significant advantages. BIM is “Visual”, meaning that any design issues that may arise can be very clearly communicated among the design team. BIM is “Informational”, meaning that better decisions could be made about design issues through the effective storage, management, and dissemination of project information. Finally, BIM is “Automational”, where it can support machine-automated review processes instead of manual checks using drawings and text (Jiang, 2016). BIM-centric workflows can specifically support prefabrication-themed projects. Benefits of such workflows include, among others, reduced cost of engineering, reducing design and drafting errors, enhanced cost estimation accuracy, and increased productivity (Sacks et al., 2005, pp.126-139, Sacks et al., 2010, pp.419-432).

1.2. Focus and problem definition

Prefabricated building components range from off-the-shelf parts such as pipe sections and dry-wall panels to custom-made components such as structural steel frames, precast concrete pieces, custom kitchens and cabinets (Sacks et al., 2018, p.278). By definition, the custom-made components are the ones that demand sophisticated engineering, pose greater challenges to traditional workflows, and require more innovation to solve the inherent problems in their design and production across all projects’ phases all the way down to handover. Some of these challenges can be generalized across all custom-prefabricated components, but every domain comes with its specific sets of standards, workflows, and also inefficiencies that need to be addressed. The versatility of concrete as a construction material and the overwhelming adoption of concrete in the construction practice, especially in large complex projects, makes it an excellent focal point in this research. Therefore, the focus of this research will be mainly on precast concrete components.

As discussed in the previous section, the usage of BIM can offer significant advantages to the prefabrication industry. However, oftentimes BIM-centric workflows are not brought to their full potential. A BIM model is “intelligent” in the sense that they are aware of the building components and their properties and categories. This “intelligence”, however, does not prevent users from taking actions that are “irrational” concerning construction and prefabrication (Bloch and Sacks, 2018, pp.256-272). In today’s design and construction practice, production and construction requirements are often not taken into account due to the separation of design and construction practices, and lack of machine-readable formalization of much of construction knowledge needed to perform automated checks on building design models (Nepal, 2011, p.2). This is specifically true in precast specific details and expertise, which are not usually represented in design decisions and design models (Aram, 2014, pp.33-34). That often leads to poor design decisions, rework, and even worse, unnecessarily increased expenses. In the German market, which is dominated by small enterprises with few employees in each enterprise, it is difficult to integrate personnel for all trades inside one company for a seamless exchange of expertise (including precast expertise) from the very start until the handover. Industry efforts to support precast data integration inside BIM models have been focused on using open data exchange means, namely Industry Foundation Classes (IFC)[1], to interface with production facilities (for example through the IFC4precast standard) to be able to extract production data directly from BIM models (Nemetschek, 2018a). However, this effort is directed to the very late stages of the precast process and does not deal with the early issues upstream.

The author’s careful examination of the precast practice in the design and construction space has led to a number of factors regarding precast construction that should be taken into consideration during the design phases. These factors sit generally in the following categories:

- Factors relating to the production requirements of precast components in the workshop

[1] Industry Foundation Classes (IFC) provide vendor-neutral, internationally standardized BIM data format that can be shared among various participants in building design, building construction or facility management project (buildingSMART, 2013)

- Factors relating to the structural limitations of precast components
- Factors relating to transport and installation limitations of precast components
- Factors relating to the cost-effectiveness of the designated precast components

So the specific problems we are facing can be summarized as follows:

- Ignoring some or all of the precast practice factors during the design phases of the project could lead to significant problems during the building construction.
- Manually checking for these requirements can be a better alternative, if a formalized checklist is readily available, but would take a significant amount of time for every design iteration and negate the benefit of design automation through BIM.
- In today's practice, a human precast expert is not often involved in the earlier design stages of a building.

1.3. Research questions

As will be explained in detail in chapter 2, a lot of research and some commercial solutions were developed to address these problems. These solutions were materialized either in the form of custom ready-made precast libraries for the designer to pick from, or in an Artificial intelligence (AI)-based computer system such as rule-based checking algorithms, automated detailing algorithms, or optimization algorithms. The problem with the reviewed literature is that the presented solutions either deal with the precast process very late in the project design phases after all the important design decisions have been made, or these solutions just present the designers with ready-made libraries of parts, which can really limit the creativity of the design. This research tries to address these gaps and is motivated by the inefficiencies in the design process of prefabricated construction, focusing on precast concrete. The broad research question is:

How could a BIM-centric process be used in the design stages of prefabricated buildings to help the designers make better-informed design decisions regarding prefabricated construction requirements, where actual expertise about these requirements is usually not introduced until the construction phase?

To answer this question, it needs to be broken down to several more specific questions, which can be detailed as follows:

1. What is the nature and mechanics of the current prefabricated construction practice, and what are the inefficiencies caused by the fragmented design and construction activities?
2. What are the current CAD and BIM tools being employed in the design process of prefabricated construction and how can we improve upon them?
3. What is the best way to capture and formalize expert knowledge about prefabrication-based construction considerations to support designers in earlier design stages?
4. How can a BIM-based system be devised and structured to support designers early on by automating checks for prefabrication requirements in design models?
5. How can the outcomes of such a system be available to the designer to utilize and interact with?

1.4. Research objectives and goals

The main goal of the research can be stated as follows:

To develop a system framework that enables an automated review process for prefabrication requirements, focusing on precast concrete construction, during the earlier design stages with the use of BIM, to provide proactive feedback for design decision support.

Specifically, the following objectives are defined to achieve that goal:

1. Identifying the current nature of CAD-enabled design and construction environments in prefabrication-based projects and their shortcomings.
2. Capturing and formulating expert prefabrication knowledge in a way that can be represented and implemented in computer systems.
3. Establishing the intricate links between design attributes and prefabrication considerations that affect decisions in the earlier phases of building design.
4. Developing a framework for a computer system that can be integrated in a BIM-centric design process, and that encapsulates expert prefabrication knowledge, to provide active feedback on building designs in the earlier design stages.

The example system selected for implementation in this research is precast concrete; however, the developed methodology and framework can be adapted to address the needs of other prefabrication methods. It should be noted that this research is largely interpretative and exploratory in nature, providing a coherent view of a problem that has not been previously clearly defined, and proposing a preliminary solution to it. The research does not intend to perform quantitative data analyses, such as statistical analyses, to justify hypothesis formulation and testing (Nepal, 2011, p.15).

1.5. Methodology

According to Blessing and Chakrabarti (2009, p.5), design-related research requires a model of the current situation (reference model), a vision of the improved situation (impact model), and a vision of a support system that could change the current situation to the improved one. Accordingly, it can be said that design research has simply two objectives:

1. First, to formulate models about design activities and features such as people, products, and design tools.
2. Second, to develop support based on these models and theories with the aim to improve the design process and its outcomes.

To realize these objectives, Blessing and Chakrabarti (2009, p.18) established a design research framework, and defined seven variants of that framework. The variants vary in length and sequence depending on the research type, questions, hypotheses, and available resources. One of the variants of this framework will be used as a basis of the methodology of this research. An overview of the framework variant can be viewed in Figure 1-1, and is composed of the following stages:

1. Research clarification: The aim here is to obtain a refined understanding of the area of research interest so as to be able to define a reasonable and realistic research problem and research goals. The research should be able to define the reference and impact models.
2. Descriptive study: In this stage, the research should demonstrate the relevance of the research topic, main lines of argumentation, and the factors to address in order to be able to realize the impact model.
3. Prescriptive study: In this stage, the research aims to develop the intended support that addresses the key factors identified in the descriptive study, explaining what it consists of and how it works.

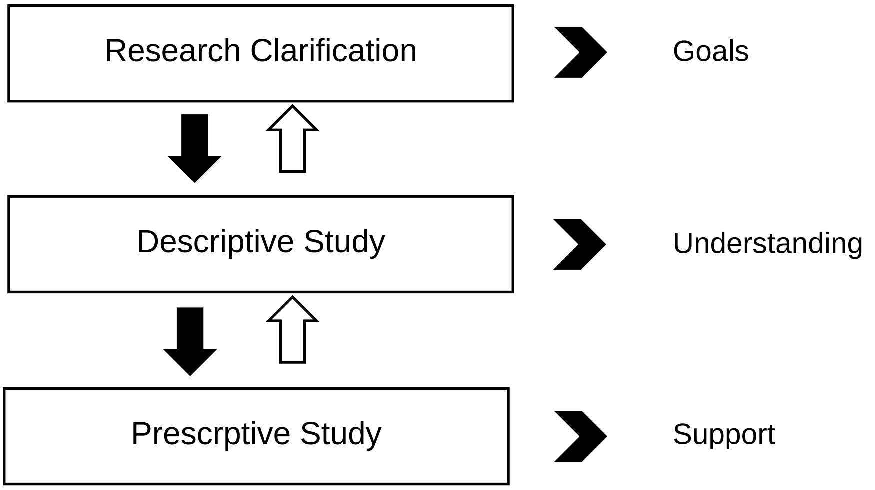

Figure 1-1 The design research framework adopted in this research (Adapted from Blessing and Chakrabarti (2009, p.18))

With this approach, the research tackles problems associated with the design process in the context of prefabricated construction, and the possibility of developing a computer system that enables the automated review of design models in earlier design stages to provide proactive feedback for designers concerning prefabrication requirements and limitations. In light of this statement, the previously mentioned research objectives can be pursued using the following steps, effectively representing the methodology of this research:

1. Process review :

   The objective here is to study the nature of prefabrication and mass-customization processes with a special focus on precast construction, and the usage of CAD systems in such practices. The aim is to identify the weaknesses in these practices and the opportunity of implementing a BIM-centric system to support the prefabrication construction practice in the earlier design stages.
2. Computer systems review:

   There are several approaches in computer systems that can be utilized to provide design decision support; Rule-based systems, design optimization algorithms, automated detailing algorithms, and ready-made detail libraries, are just a few examples. Therefore, the objective of this step is to explore the various systems, and then to study and analyze the structure of the ideal system in the context of this research, its development process, and the potential of such a system to be implemented in a BIM data-rich environment.

3. System framework development:

   This is the main part of this research. The aim here is to develop the system framework that enables checking design models in earlier stages for prefabrication “suitability”, to ensure that the design conforms to prefabrication requirements and limitations. The framework development process can be further subdivided into the following steps:

   i. Establishing the structure of the framework
   ii. Establishing the rule database
       a. Knowledge acquisition
       b. Rule-set structuring and categorization
       c. Knowledge formalization and representation
   iii. Prototyping and interface design

The thesis structure is shown in Figure 1-2. Chapter 1 provides an introduction to this research along with clarifying the focus, research problem, research questions, research objectives, and methodology. Chapter 2 reviews the basic concepts of prefabrication and precast practices and the use of computer systems in this context. Chapter 3 provides insight into rule-based, BIM-enabled expert systems, and their implementation in design and construction research. Chapter 4 introduces the proposed framework for the precast system, discussing its structure and requirements and the procedure of its development. Chapter 5 presents a prototypical implementation of the proposed system. Finally, chapter 6 discusses the findings of the research, its limitations, and potential areas for future development.

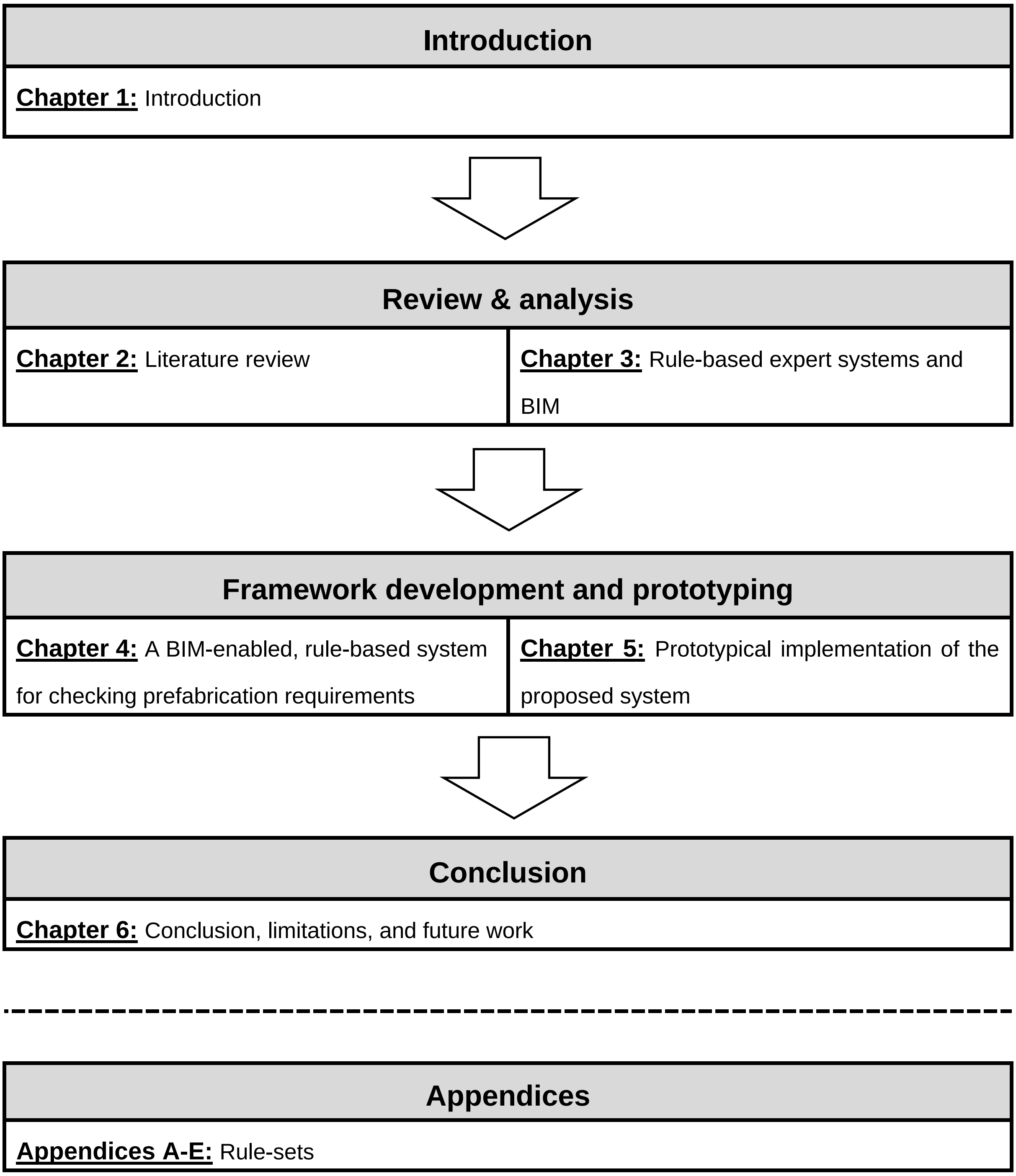

Figure 1-2 Structure of the dissertation

Chapter 2: Literature review

2.1. Introduction

The Architecture, Engineering, and Construction (AEC) industry is growing increasingly complex. In this competitive field, and with an ever-growing need for new construction, it emerges the urge to improve construction quality, save money, and shorten construction schedules. To address these issues, the construction industry began to prefabricate building components in- factory, under well-controlled conditions, to be later assembled on-site. This approach gave rise to precast concrete construction, utilizing concrete as a highly flexible construction material. This chapter discusses the basic concepts of prefabrication, mass-customization, and precast construction. More importantly, it reviews how commercial software companies and academic researchers have been tackling the issue of computer-aided prefabricated and precast concrete design to further enhance and optimize the process.

2.2. Prefabrication and mass-customization

2.2.1. Definition

According to Gibb (1999, p.1), prefabrication can be defined as a process that involves the design and manufacturing of modules, usually away from the site, and their assembly and installation afterward on-site to form the permanent structure. Prefabrication of building components has been present in one way or another since the year 1854 in the Crimean war when Isambard Kingdom Brunel was commissioned to design and build a 1,000 patient field hospital using factory-built timber units (Gibb, 1999, pp.10-11). The trend of using factory-built building components from wood, steel, and concrete continued during the 20th century to face housing shortages that prevailed after the two world wars (Huang et al., 2006, pp.203-208).

2.2.2. General concepts

In the present day, prefabricated building components can be broadly classified into three categories (Sacks et al., 2018, p.278):

1. Made-to-stock components: These are mass-produced components such as plumbing fixtures, drywall panels.

2. Made-to-order components: These are components produced after a client's order based on certain shapes and measurements defined in catalogs such as windows and doors
3. Engineered-to-order components (ETO): These components need to be custom designed and engineered prior to production such as precast concrete components and façade panels.

The focus of this research is on the third type, the ETO components, as they are the ones that require sophisticated engineering and collaboration workflows, where traditional mass-production strategies cannot be deployed. Prefabrication in the construction industry instead builds on the concepts of mass-customization. According to Joseph pine in his book "Mass Customization: The New Frontier in Business Competition" (as cited in (Piroozfar and Piller, 2013, p.4)), mass-customization is described as:

"Developing, producing, marketing, and delivering affordable goods and services with enough variety and customization that nearly everyone finds exactly what they want".

This new paradigm of mass-customization is a departure from the concepts of "mass-production", focusing on meeting individual needs rather than focusing solely on lower prices. The premise of mass-customization is to allow production driven by actual individual orders of a product variation (module) that exists typically within a larger family that accommodates these modules. In design and construction practice, mass-customization is tightly associated with the term "modular design", where the design scheme is based on a pre-specified, standardized dimensional array, commonly in the form of multiples of a given base. This array is what controls the sizing and location of prefabricated building components (Piroozfar and Piller, 2013, p.81). This is an essential concept in the construction industry that enables factory-built building components to be produced on-demand and with enough variety, yet with high efficiency, so as not to limit design possibilities. Table 2-1 presents a comparison between the concepts of mass-production and mass-customization.

Table 2-1 Mass-customization compared to mass-production. Adapted from "Mass Customization: The New Frontier in Business Competition", Joseph Pine, as cited in (Alfino et al., 1998, p.111)

Aspect	Mass-production	Mass-customization
Features	Low-cost, consistent quality, standardized goods and services	Low-cost, high-quality, customized goods and services
Focus	Efficiency through stability and control	Variety and customization through flexibility and quick responsiveness
Goal	To provide goods and services at lowest price	To provide affordable goods and services with enough variety to meet everyone's needs
Demand	Stable demand	Fragmented demand

2.2.3. Significance in the construction practice

A US study looked at the 48-year period between 1967 and 2015 and concluded that the productivity of the traditional construction industry has nearly remained the same, while off-site based construction has enjoyed increases in measured productivity (Figure 2-1) (Sacks et al., 2018, p.10). This confirms the increased value obtained from off-site construction. The benefits of off-site based construction have been extensively studied in literature, and they include (Mokk, 1964, pp.14-15, Bachmann and Steinle, 2011, pp.14-15, Polat, 2008, pp.169-178):

- Time savings
  Project delays due to unexpected site or weather conditions may incur significant costs due to the failure to meet the project schedule. Prefabrication enables reduced schedule durations and parallel production, assembly, and erection operations, which is often not affected by bad weather conditions (Hardin and McCool, 2015, p.29).
- Increased quality
  Factory-produced parts have better quality and tighter tolerances than site-built parts thanks to more robust quality control procedures (Rathnapala, 2009, p.22). Standardized factory products are also more guaranteed to fit-in together (Khalili, 2013, p.31).

- Improved cost-efficiency

  Factors to consider include decreased installation cost, lower site occupancy, increased mechanization, reduced construction duration, and reduction of scaffolding (Gibb, 1999, p.38, Khalili, 2013, pp.30-32).
- Better for the environment

  Relying on factory-produced components ensures less material wastage, which can happen more often on-site (Rathnapala, 2009, p.20), with reported waste reduction levels up to 52% compared to traditional construction (Jaillon et al., 2009, pp.309-320). Material savings could also be achieved by using more efficient components that can only be factory-built, like hollow-core slabs (instead of solid slabs) (Bachmann and Steinle, 2011, p.5).
- Safer working environment

  Factory conditions usually ensure a safer, more controlled environment for workers, where safety and security standards can be more strictly applied.
- Improved standardization and modularization

  Building parts can be structured in modules, where similar modules are grouped in one type (Mohamad et al., 2013, pp.289-298). Modularity is key to achieving mass-customization because it helps standardize repetitive building components (Farr et al., 2014, pp.119-125).
- Layout flexibility

  Some prefabricated components can offer construction solutions that are otherwise not doable on-site. For example, precast double tee slabs and hollow-core pre-stressed slabs provide longer spans that can open up interior spaces to maximize layout functionality (PCI Industry Handbook Committee, 2010, p.1B4).

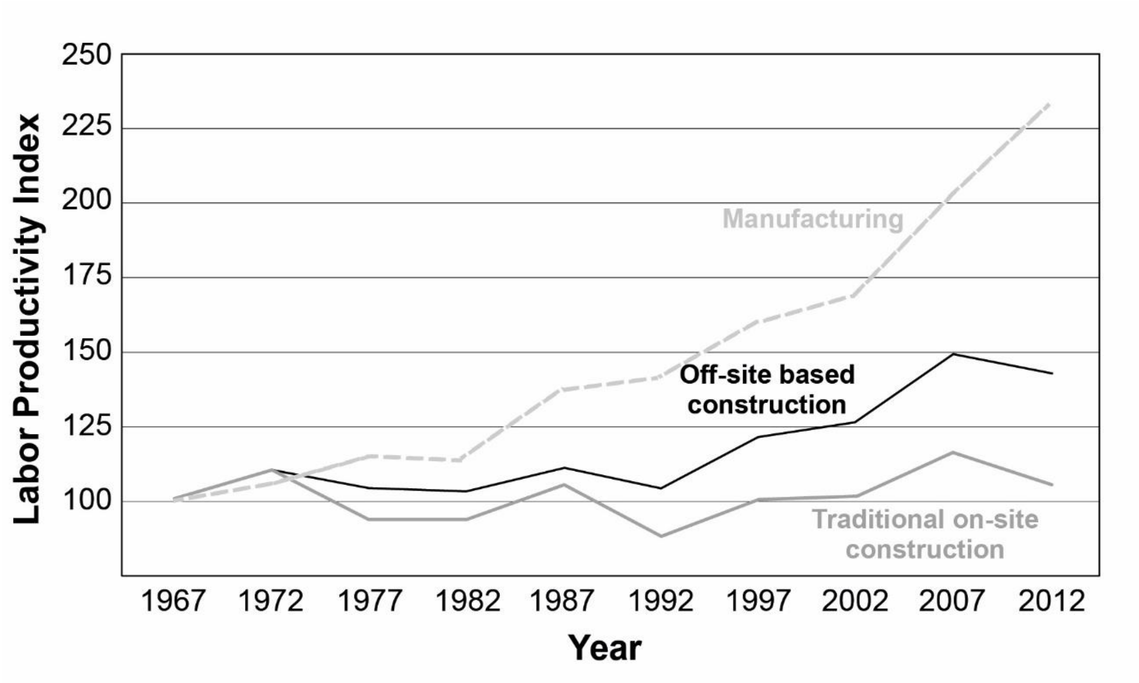

Figure 2-1 Labor productivity indices of manufacturing, on-site, and off-site construction in the period 1967-2015 in the American market. Adapted from (Sacks et al., 2018, p.10)

2.3. Precast concrete construction

2.3.1. Definition and history

Precast concrete construction is a construction type whereby concrete is cast into reusable moulds under factory conditions and then transported into the construction site for in-place erection, improving reliability, speed, and quality of construction (PCI Industry Handbook Committee, 2010, p.1A1). Precast concrete construction is not a “new” technology, it dates back to around the year 1900 in France when the first precast concrete structures were introduced, and was adopted in the First World War to construct storehouses for various uses (Elliott and Jolly, 2013, p.1). The next milestone came in 1918 when the American architect Grosvenor Atterbury designed a system of wall panels and slab panels and implemented it in a settlement in New York. The system was later transferred to Europe in 1923 through a Dutch construction company, which utilized the system to build 151 apartments in what was named the “Betondorp” or “Concrete village” (Plattenportal, 2018).

In the German market, precast concrete’s relevance increased due to the housing shortage after the first world war, which necessitated a more efficient method of construction, ultimately leading to the development of the “Frankfurt Slab System “ or the “Frankfurter Plattenbau” by Ernst May in the late 1920s (Figure 2-2). Although the system introduced factory-produced concrete slabs to the German market, it suffered afterward from a multitude of problems leading to its demise until the housing shortage after the Second World War, where the “Plattenbau” system dominated the residential construction scene in the German Democratic Republic (Knaack et al., 2012, pp.18-23). That approach, however, led to a repetitive, monotonous, mass-replicated architecture, painting a negative image of precast concrete construction (Ostrowska-Wawryniuk and Nazar, 2018, pp.247-256). It relied heavily on mass-production of components rather than mass-customization, where, for example, only 13 out of 39 possible components were used in construction in the German city of Leipzig to save costs (Knaack et al., 2012, p.22). However, since 1989, there has been still an ever-increasing demand for new housing and speedy construction in the German market to meet the needs of immigrants (Bachmann and Steinle, 2011, p.7). For example, it was estimated that there is a shortage of 40,000 residential units in the German city of Munich just to accommodate the increase in inhabitants since 2011, and this problem is likely to persist based on the current practices until 2030 (Möbert, 2018, p.4).

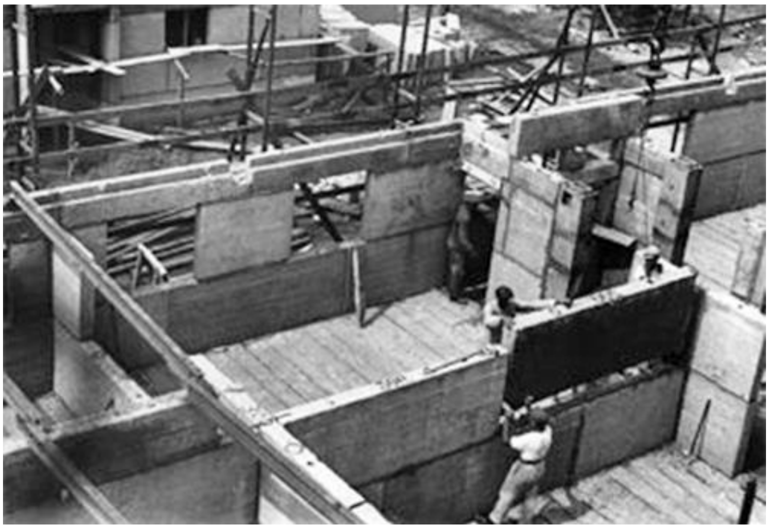

Figure 2-2 "Frankfurter Plattenbau" or the "Frankfurt slab system" under construction in the 1920s. Image from (Knaack et al., 2012, p.19)

2.3.2. The current design and construction practice

According to Eastman et al. (2009b, p.7), there are three scenarios for a precast project:

- Architectural precast (only) project:

  Precast is used only in the building façade to produce complex panels.
- Precast-led project:

  Here the general contractor, who is also the precast prefabricator, leads the project.

- Precast fabricator as a subcontractor:
  This is the most common case, where the precast fabricator is a subcontractor under the general contractor.

The focus in this research is on the third case, which is the most common case. Ideally, the project in this scenario progresses as shown in Figure 2-3, where the process in the earlier phases can be detailed as follows (Eastman et al., 2009b, pp.33-44):

1. The architect creates a schematic design model including space layout and use, structural system selection, structural grid, major architectural finishes, and expected thermal and acoustic functions.
2. Engineers use that concept model to provide feedback on the structural grid and other major issues. The precast expert also uses that model to propose major precast components and study the precast structural system, panelization, and site logistics.
3. Architects continue the design process after reviewing input from engineers and prefabrication experts and proceed to design development models, which are further reviewed by engineers and prefabricators

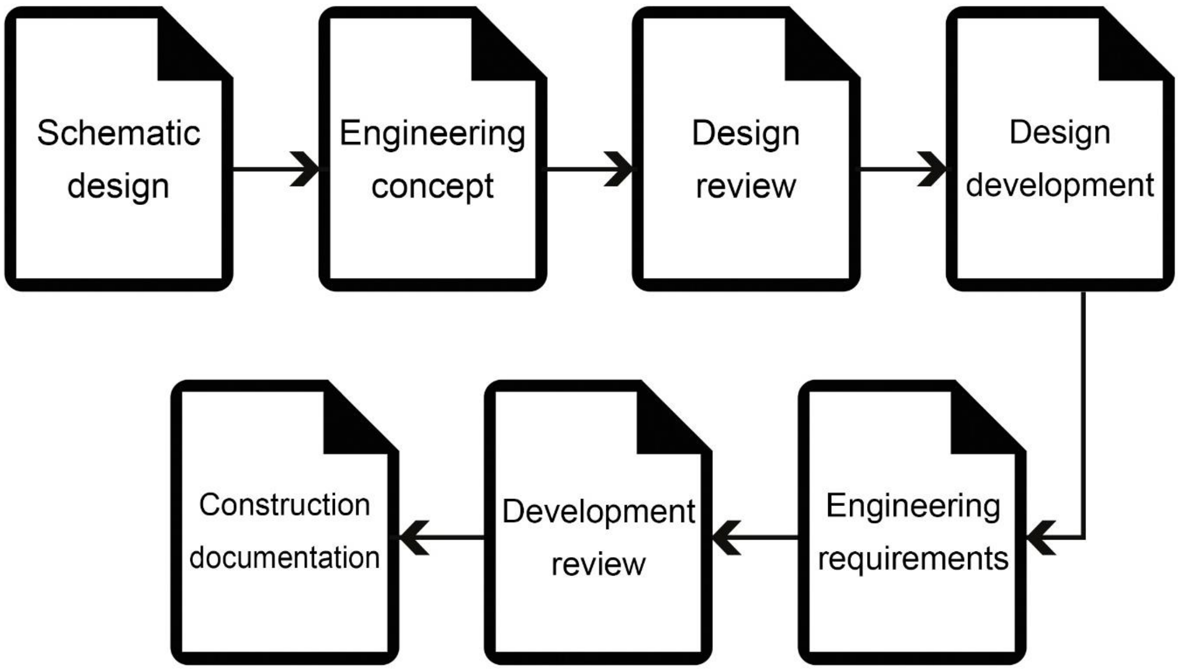

Figure 2-3 Ideal design process in a "Precast fabricator as subcontractor" model for design and construction. Adapted from(Eastman et al., 2009b).

However, in reality, this ideal case does not happen in an orderly manner, nor does it occur in an integrated, common design environment, where every project participant (e.g. Architect, structural engineer, cost estimator) provides his/her input exactly at the right point when needed. Current design practices, promote fragmented project delivery models (employing fragmented design and construction teams), like the Design-Bid-Build (DBB) model[2], over other, more collaborative models, like the Integrated Project Delivery (IPD) model[3]. In Germany, project delivery primarily rely on this traditional design-bid-build delivery model with a low-bid procurement (DeWitt et al., 2005, pp.10-11). In such a delivery model, the design team receives almost no input from construction experts until the end of the design process and the bidding process. In the German market, this fragmented model is further supported by two facts:

- First, two distinct regulatory documents exist to administer the practice of design (through HOAI[4] (Koeble and Zahn, 2013, p.4)), effectively regulating a "Designer space", and the practice of construction (through VOB[5] (DIN, 2016, p.8)), effectively creating a separate "Contractor space", in isolation from each other. Figure 2-4 highlights the different project tasks administered by each document.
- Second, the German market relies largely on small companies (< 50 employees), which employ around 64% of employees working in the construction sector (Figure 2-5) (Heinrich, 2017, p.2). This fragmented environment makes it even more difficult to integrate the expertise of all the necessary trades for precast building construction from schematic design to handover in one seamless environment. In this market in larger government projects, it may be even possible that a company becomes responsible for only one project phase[6].

[2] A design-bid-build (DBB) model is a project delivery model where the owner signs a contract with the architect, establish design objectives, and the architect then develops the project in phases commonly identified as schematic design (SD), design development (DD), and contract documents (CD). In a separate second stage, the project is offered for bidding from contractors (Sacks et al., 2018, p.4).
[3] An Integrated project delivery (IPD) model is a project delivery model where all the project participants come together from the early stages as an integrated team to design and construct a given project (AIA, 2007)
[4] Honorarordnung für Architekten und Ingenieure (HOAI): Official scale of fees for services by architects and engineers in Germany
[5] Vergabe- und Vertragsordnung für Bauleistungen (VOB): The German construction contract procedures
[6] Project phases: (German: Leistungsphasen) represent the different architectural and engineering services for a construction project as described in the Official scale of fees for services by architects and engineers in Germany (HOAI). They are divided into nine phases, from pre-planning to site supervision and as-built documentation.

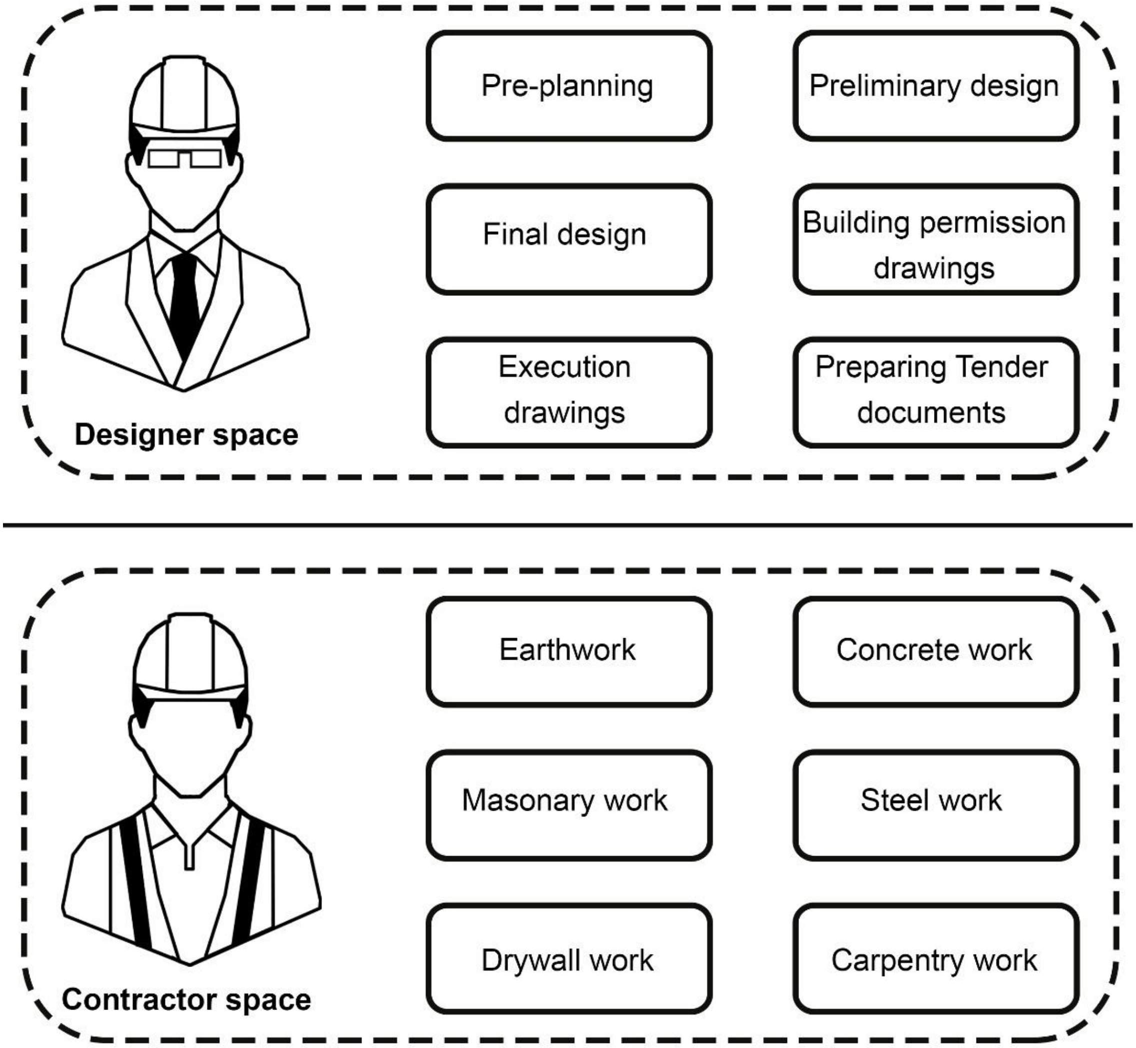

Figure 2-4 Exemplary tasks defined and administered by the HOAI for designers (above), vs exemplary tasks defined and administered by the VOB for contractors (below)

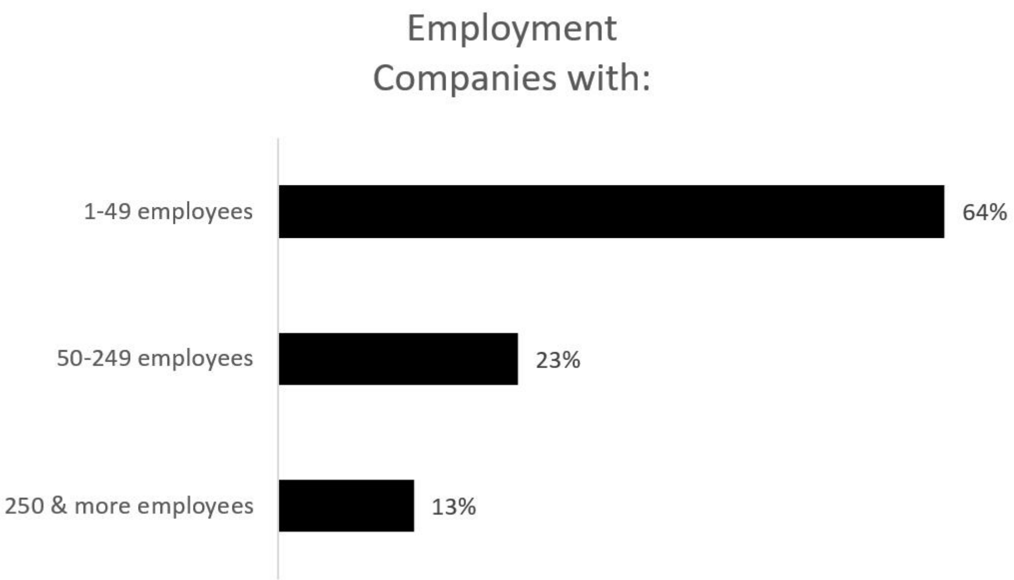

Figure 2-5 Percentage of employees in the German market working in the construction sector by company size. Based on data from (Heinrich, 2017, p.2)

2.3.3. Importance of planning for precast in the design phases

Such a project delivery model leads to problems in communication between precast concrete manufacturers and designers. This often creates situations in precast projects where prefabricated components are not recognized in design models, and where building parts are not based on precast considerations, leading to conflicts, rework and increased cost (Aram, 2014, p.3). In a survey conducted in the US market, manufacturers were asked to evaluate how often they found problems in production due to design ambiguities on a scale from 0 to 4 (0= never, 1= rarely, 2= sometimes, 3= often, 4= very often), the average score was 2.37 (which is higher than "sometimes") (Arditi et al., 2000, pp.79-86). The same survey was repeated eight years later, and the average score was 2.58 (closer to "often") (Polat, 2008, pp.169-178) (Figure 2-6). These results are consistent with the input received from precast experts in the German market, through interviews conducted by the author[7]. They expressed that architects usually work in isolation from precast experts and input from experts come much later in the project.

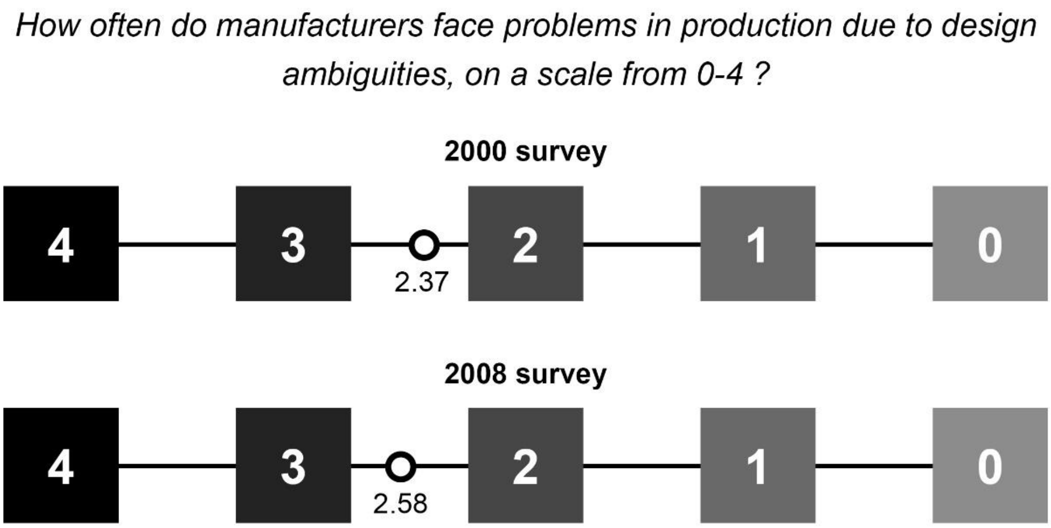

Figure 2-6 Comparing the results of a survey question in two surveys conducted in 2000 and 2008 in the American construction market regarding precast production problems due to design ambiguities. Based on data from (Arditi et al., 2000, pp.79-86) and (Polat, 2008, pp.169-178)

[7] Details about the conducted interviews are provided in section 4.3.1

In another experiment, where architectural design models developed by the designers were compared to those developed by precast experts, it was found that actual building geometry was quite different (Sacks et al., 2010, pp.419-432). For example, in a building's façade, it was found that the precast expert envisioned precast spandrels that cover the entire span between every couple of exterior columns, while the architect modeled it as three panels separated by steel mullions (Figure 2-7). Designs are often made with complete disregard to precast requirements, and then precast planning comes as an afterthought.

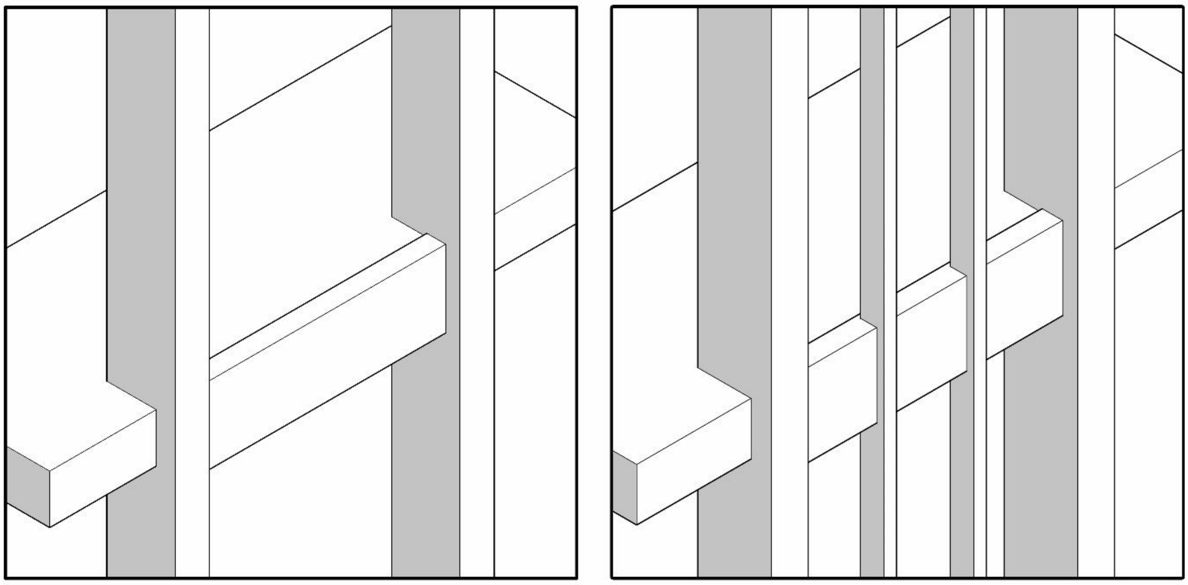

Figure 2-7 Different representations of a spandrel beam in the precast expert's model (left) and the architect's model (right). Adapted from (Sacks et al., 2010, pp.419-432)

Using precast concrete components to realize a design that was developed as a cast-in-situ project, requires changes, adjustments, and adaptations, resulting in conflicts and inefficiencies that could be avoided if it was planned as a precast project in the earlier phases (PCI Industry Handbook Committee, 2010, p.1D2). However, often times the design phase is carried out in the absence of a precast expert (Elliott and Jolly, 2013, p.44). When precast experts are integrated early in the design process, aspects such as component sizing, connections, transport, and erection considerations can be factored in the design, maximizing the value, cost efficiency, aesthetics, and functionality offered by precast concrete construction (PCI Industry Handbook Committee, 2010, p.1D3). Precast experts can provide valuable insight throughout the design process in conceptual, preliminary, and detailed design, on a variety of design aspects. An overview of these aspects is shown in Figure 2-8.

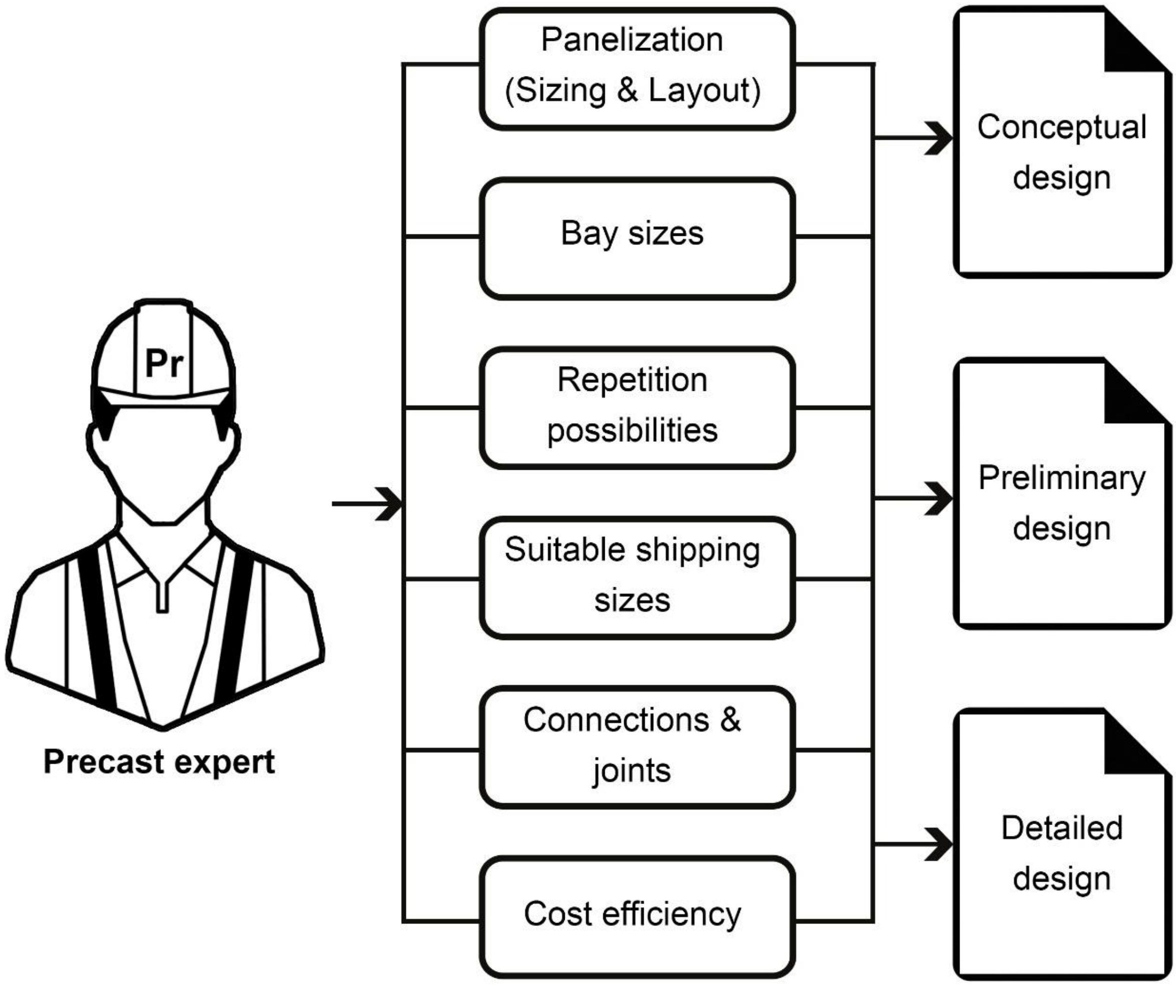

Figure 2-8 The precast expert can provide input on design aspects throughout the design process. Adapted from (PCI Industry Handbook Committee, 2010, p.1D3)

2.4. Computer aided Prefabrication design

2.4.1. Using BIM systems in prefabrication

According to Piroozfar and Piller (2013, p.35), the successful implementation of mass-customization in the construction industry requires a wider, smarter and more responsive use of Information and communications technology (ICT) such as parametric design, digital fabrication, and BIM models. The definition of a BIM model in this context is best explained through (Borrmann et al., 2018, p.4) as follows:

“A Building Information Model is a comprehensive digital representation of a built facility with great information depth. It typically includes the three-dimensional geometry of the building components at a defined level of detail. In addition, it also comprises non-physical objects, such as spaces and zones, a hierarchical project structure, or schedules. Objects are typically

associated with a well-defined set of semantic information, such as the component type, materials, technical properties, or costs, as well as the relationships between the components and other physical or logical entities"

BIM models are managed through "BIM applications" which include in a broad sense "BIM platforms" and "BIM tools". For the purposes of this research, it is useful to start by establishing the difference between the two terms. A "BIM platform" is a BIM information generator that has the necessary functionality to maintain the consistency and integrity of BIM models. A "BIM tool" is a BIM information sender, receiver, and processor that is used in association with BIM models authored by BIM platforms (Sacks et al., 2018, p.58). The idea of a BIM model (even if it involves a series of models) enables design, production, and construction participants to weigh in with their expertise to develop the model, making the whole process more efficient in many ways as follows (Kolarevic and Klinger, 2008, p.29):

- The model serves as a repository of smart systems and components that behave in a consistent manner and enable change propagation throughout the project to happen automatically without tedious manual edits.
- The model allows for simulations, analyses, and model checks using digital tools to evaluate, for example, the building performance, cost efficiency, or structural integrity, among other considerations.
- The model ideally provides the geometric and semantic information required to prefabricate building components. The data-rich model is an effective medium for design and construction teams to coordinate the prefabrication process (Nawari, 2012, pp.107-113).

When focusing on BIM use in prefabrication, unlike traditional 2D CAD tools, the parametric capabilities of BIM platforms enables the definition of semantic relationships between, and within building components, making them behave in an intelligent responsive way, relieving the designer from the tedious task of manually maintaining the model integrity (Sacks et al., 2004, pp.291-312). This does not only improve speed and productivity, but also reduces errors and clashes (Sacks et al., 2010, pp.419-432). Switching from 2D based workflows to BIM in prefabrication in actual projects yielded a lot of benefits including reduced time for design checks and detailing, better coordination, and accordingly much less required edits during the shop drawings phase (Kaner et al., 2008, pp.303-323). Table 2-2 summarizes the benefits of adopting a BIM-enabled environment in prefabrication in general, with a specific focus on precast concrete.

Table 2-2 Benefits of adopting BIM in prefabrication – Based on data from (Sacks et al., 2005, pp.126-139, Nath et al., 2015, pp.54-68, Sacks et al., 2018, p.22)

Benefit	**Description**
Reduced engineering cost	Due to the increased productivity of project participants in designing and replacing “drafting” with “parametric modeling”
Reduction of errors	Intelligent parametric model component and efficient change propagation enable consistent project documentation
Better Visualization	Improved 3D visualization enhances project presentation to clients and stakeholders
Improved project definition	More complete and better-coordinated project documentation is available throughout the project phases
Better cost estimation accuracy	Project costs could be easily and more accurately estimated throughout the project phases
Earlier collaboration	BIM models give earlier insight into design problems instead of waiting until the design is complete.

It is worth mentioning that BIM models are still by themselves “ignorant”, in the sense that they allow users to take irrational actions regarding design viability or constructability (Bloch and Sacks, 2018, pp.256-272). For example, if a designer devised a building design with irrationally huge column spans, no system alert would be raised unless a custom rule-checking routine is designed and built into the system to check for such inconsistencies. Generic BIM technology do not integrate relevant construction information in the design models out of the box, this can be attributed to a number of reasons (Nepal, 2011, pp.1-2):

- Designers cannot predict and manually integrate every intended use of the models they develop
- The separation of design and construction activities leads to a situation where designers do not care to take production and construction requirements into consideration

- Objective construction knowledge has largely not been formalized enough to enable its integration in information models.
- Current BIM technology does not offer adequate reasoning mechanisms to derive production/construction-related information automatically, and with a sufficient level of detail.

However, the “intelligence” of BIM can enable the integration of prefabrication domain knowledge in BIM authoring tools, reducing the effort of modeling, eliminating design inconsistencies, and streamlining the whole design process by generating coordinated drawings automatically and enabling design verification and review (Singh et al., 2015, pp.519-527). This is the subject of discussion of the following section.

2.4.2. Related work

The shortcomings mentioned in the previous section drove a lot of interest in developing tailored platforms for a multitude of building domains, including prefabrication. For example, in the precast concrete domain, commercial software such as Tekla Structures (Trimble, 2018), Structureworks (Structureworks, 2018), and Planbar (Nemetschek, 2018b) have been developed to suit the needs of precast experts in design and detailing. Accordingly, they are geared towards the detailed design phase and mainly used by precast experts for precast production and erection purposes only. Scientific literature has also tried to tackle the issue of the design of prefabricated buildings. A summary and classification of reviewed research can be found in Table 2-3. The criteria for choosing these publications for review can be summarized as follows:

- The research should deal with the “Design” of prefabricated structures/parts
- The research attempts a CAD/BIM/Web-Based approach for improving existing workflows in prefabrication design
- The research attempts to automate manual, laborious tasks and/or help produce better-informed designs for prefabrication.
- Similar approaches are only represented by choosing one representative publication, omitting the others for the sake of conciseness.

The following section provides a brief explanation followed by collective concluding remarks summarizing the research gaps, as well as the aspects relevant to this thesis. The reviewed research can be generally classified into research dealing with earlier design stages, and research dealing with detailed design and preconstruction phases.

a) Research dealing with earlier design stages

- A tool called "AUTMOD3" is developed in (Diez et al., 2007, pp.457-468) to help in the modular design of prefabricated houses. Two approaches were presented: The first lets architects devise architectural plans and then the tool processes the drawings to obtain the modules, proposing corrections along the way; in the second approach, the designer picks from a catalog of 3D parametric modules.
- A web-based design system for modular houses, called i_Prefab, is proposed in (Huang and Krawczyk, 2007, pp.679-686) to assist customers in the selection of appropriate design components through questionnaires. The system is capable of producing design options based on the client's answers and a catalog of modular components from a selected supplier, making it possible to simulate the design before ordering, manufacturing, and assembly.
- Authors in (Singh et al., 2015, pp.519-527) developed a system prototype, where the designer is guided by specific pre-designed prefabrication and mass-customization rules and constraints while developing a BIM model. The user will still have enough freedom to create design variations while guided with assistive messages.
- Research in (Collins and Gentry, 2017, pp.657-666) utilized solid modeling and visual/textual scripting to design a tool to bring together architects' designs with precast experts. The presented prototype was capable of generating a number of variations of architectural precast concrete panels from a base wall and some user input, which provides a visually-rich means of communication between architects and precast experts. Drawbacks of that approach were the usage of limited geometric models (using Rhino and Grasshopper) instead of Building Information models, and the total reliance on user input without any pre-embedded expertise in the system.
- An experimental tool was developed in (Ostrowska-Wawryniuk and Nazar, 2018, pp.247-256) in a BIM platform using visual scripting to aid the process of architectural design based on precast concrete components. The prototype tool takes a BIM model as an input, analyzes it, and proposes a number of split alternatives (with fit scores) based on input boundary conditions from the user, altering the building's geometry or

wall openings positioning (based on a desired tolerance parameter) to maximize production efficiency.

b) Research dealing with detailed design and preconstruction phases

- Research in (Retik and Warszawski, 1994, pp.421-436) presented a knowledge-based system to aid the design of prefabricated buildings. The system receives a given preliminary architectural building design, adjusts it to a modular grid, proposes the location for structural supports, divides the walls and floors into prefabrication-ready elements, and finally produces detailed drawings and the cost estimate.
- A system was developed in (Pastor et al., 2001, pp.216-227) for prefabricated façade panels, which takes a 3D building design from the architect and proposes an optimal façade partitioning solution taking into account process constraints.
- A tool is presented in (Poyatos et al., 2011, pp.1085-1090) that helps architects design façade panels that are viable in terms of dimensions, weight, location, and cost. The tool prompts the user for the façade geometry and some input parameters such as panels'' maximum size, then the algorithms divide the façade's surface into panels with panel-cut drawings and detailed costs.
- Research in (de Albuquerque et al., 2012, pp.348-356) presented a decision support system that automates and optimizes the structural layout of precast concrete floors. It takes a number of inputs such as the number of floors, floor dimensions, dimensional limitations, and floor loads. It then can provide a number of outputs that can help engineers in the structural design stage such as the number, direction, and dimensions of hollow-core slabs, reinforcement, and column positions.
- The approach in (Khalili and Chua, 2012, pp.243-253) was to move beyond individual building elements and to configure "groupings" of precast elements to minimize the number of components to reduce costs of production, transportation, and erection. The system uses an IFC model to extract a graph data model, generates all possible grouping configurations, and applies constructability rules to reduce the solution space to the feasible configurations only.
- Authors in (Jensen et al., 2012, pp.1-8) developed a configurable timber floor slab module, establishing the flow of design information, constraints, and rules governing the production between three viewpoints: customer view, engineering view, and production view in different engineering software platforms.

- In (Aram, 2014, pp.164-167), a knowledge-based system that encapsulates expert knowledge was proposed, devising rule libraries to automatically enrich BIM design models with details based on precast concrete considerations. The aim is to automate repetitive tasks in the detailed design phase and Quantity Take-Off (QTO) tasks.
- In (Nath et al., 2015, pp.54-68), a set of BIM precast components were developed, which enable the automated generation of shop drawings and reinforcement schedules, to streamline the process. Test results showed productivity improvement of up to 38%.
- A web-based catalog is presented in (Costa and Madrazo, 2015, pp.239-248) that compiles precast building components, enabling engineers to obtain components necessary to model a precast concrete structural frame in a BIM environment. A number of services were developed alongside the catalog, including a Revit plug-in that facilitates the component search.
- A pilot study is explained in (Patlakas et al., 2015, pp.597-604) that demonstrated the possibility of developing a library of smart BIM timber components, pre-programmed with standard allowable loading values and corresponding span and spacing values, along with manufacturing limitations. The result is a more intelligent BIM model that, for example, reacts to span changes and alters structural components accordingly.
- A prototype application called “SeeBIM” was implemented in (Belsky et al., 2016, pp.261-274) that comprises a set of rules for the precast concrete domain in the form of IF-THEN rules to test building models for geometrical and topological relationships. The aim is to supplement IFC-exported BIM models with semantic constructs, to ensure that the semantically enriched models conform to the requirements of the model’s receiving application.
- Authors in (Said et al., 2017, pp.1-13) presented an exterior, wall-panelization model that aims to balance the tradeoff between minimizing the total cost of panel fabrication and minimize the subsequent design deviation for a given design. Algorithms were developed to automate the processes of panel geometry manipulation, structural analysis and design, and the optimization process.
- A precast component database for prefabricated structures was established in (Bai et al., 2019, pp.13-21). Based on the BIM platform “Revit”, the authors built a parameterized component library and an accompanying plugin to customize the steel reinforcement details of a given component.

- Alwisy et al. (2019, pp.187-205) developed a methodology for automated design and drafting of wood-framed panels used in modular residential buildings to reduce design cost, improve accuracy, and enhance productivity. Based on 2D CAD drawings, the developed tool "MCMPro" is able to develop BIM models and then generate the necessary sets of shop drawings and QTOs of the panels.

In Table 2-3, the reviewed research is categorized and labeled for clarity, and to recognize and benefit from the different approaches each research has taken. Research has been labeled in general according to the field of application either as specific to precast concrete or relating to other practices of prefabrication. The aim of the developed models are classified as to either perform design checking, assist in the design process, provide automated cost estimations, or to detail/enhance/enrich design details. The practical implementation approach of the reviewed research was found to generally fall into Knowledge-based rule systems, design optimization tools, establishing product catalogs, or web-based tools and libraries for prefabrication configuration. Finally, the target project phase was labeled as either "Early design stages" or "Detailed design and/or construction documents".

Table 2-3 Summary of the review of literature

Publication	Field		Aim				Approach				Phase	
	Precast concrete	Other prefabricates	Design check	Design assist	Cost estimation	Design enrich/detail	Knowledge-based rules	Design optimization	Product catalogue	Web-based tech	Earlier design phases	Detailed design & preconstr.
(Retik and Warszawski, 1994) Automated design of prefabricated building.		x		x	x		x					x
(Pastor et al., 2001) Computer-Aided Architectural Design Oriented to Robotized Facade Panels Manufacturing	x		x	x			x					x
(Diez et al., 2007) The integration of design and planning tools for automatic modular construction.		x	x	x			x		x		x	
(Huang and Krawczyk, 2007) A Choice Model of Consumer Participatory Design for Modular Houses		x		x	x				x	x	x	
(Poyatos et al., 2011) Cad-based tool for automated panel cutting of prefabricated facades		x	x	x			x					x

Table 2-3 Summary of the review of literature

Publication	Field		Aim				Approach				Phase	
	Precast concrete	Other prefabricates	Design check	Design assist	Cost estimation	Design enrich/detail	Knowledge-based rules	Design optimization	Product catalogue	Web-based tech	Earlier design phases	Detailed design & preconstr.
(de Albuquerque et al., 2012) A cost optimization-based design of precast concrete floors using genetic algorithms	X			X		X		X				X
(Khalili and Chua, 2012) IFC-based framework to move beyond individual building elements toward configuring a higher level of prefabrication	X			X				X				X
(Jensen et al., 2012) Configuration through the parameterization of building components		X		X					X			X
(Aram, 2014) Knowledge-based system framework for semantic enrichment and automated detailed design in the AEC projects	X				X	X	X					X
(Nath et al., 2015) Productivity improvement of precast shop drawings generation through BIM-based process re-engineering	X			X					X			X
(Singh et al., 2015) Modular coordination and BIM: Development of rule based smart building components		X		X					X		X	
(Costa and Madrazo, 2015) Connecting building component catalogues with BIM models using semantic technologies	X			X					X	X		X
(Patlakas et al., 2015) A BIM Platform for Offsite Timber Construction		X		X					X			X
(Belsky et al., 2016) Semantic Enrichment for Building Information Modeling	X					X	X					X
(Said et al., 2017) Exterior prefabricated panelized walls platform optimization		X	X	X				X				X
(Collins and Gentry, 2017) KBAD-Knowledge Base for Architectural Detailing	X			X			X				X	
(Ostrowska-Wawryniuk and Nazar, 2018) Generative BIM Automation Strategies for Prefabricated Multi-Family Housing Design	X		X	X			X				X	
(Bai et al., 2019) Application of BIM in the creation of prefabricated structures local parameterized component database	X			X		X			X			X
(Alwisy et al., 2019) A BIM-based automation of design and drafting for manufacturing of wood panels for modular residential buildings		X				X	X					X

2.4.3. Concluding remarks on the literature review

Observing the results of the literature review, the following can be concluded:

- All reviewed research relies on some sort of “additional” digital tool that is added to traditional CAD or BIM platforms to enable process enhancements. These tools could take the form of custom libraries, rule-based checks, automated detailing algorithms, or design optimization algorithms, among others.
- Most of the reviewed research suffered one or more of the following drawbacks:
  - Lack of pre-embedded intelligence/expertise and reliance on manual user input.
  - Reliance on geometric models instead of energy-rich BIM models.
  - Being too specific and addressing only one type of building components in one building system.
  - Being too generic and trying to address all types of prefabricated components across all systems.
  - Reliance on some sort of library or “catalog” of prefabricated components, where the designer or the precast expert can just pick one or more of this limited selection of components. This is an inherently problematic approach, which could limit the design space available for designers and reduces the design to a Lego-like process.
- The use of a BIM process in design and construction can potentially allow for both design teams and precast experts to cooperate in the design, review, and optimization of prefabricated modules throughout the design process (Hardin and McCool, 2015, p.30). However, observing the available commercial prefabrication software, as well as most of the reviewed research, indicates the lack of tools that are capable of integrating prefabrication knowledge, or adapt prefabrication design tools, in the earlier design phases. Current prefabrication tools are mostly designed to function in late design development phases, raising the issue of how computer tools could be employed in earlier design phases to enable feedback early on in the project.

2.5. Conclusion

The construction industry has been lagging in productivity compared to other industries. It is also burdened by an increasing number of sophisticated building systems, without an equal parallel improvement in construction methods and design/construction automation.

Prefabrication and off-site based construction offer huge advantages to productivity, quality, and safety in the construction industry, especially when coupled with smart digital tools such as BIM. However, BIM in itself is not a universal solution that can be generically implemented in all domains. Individual BIM tools are, and should continue to be, developed to add a layer of “Domain-specific intelligence” for individual disciplines in the construction industry, including prefabrication and precast. A multitude of BIM tools and platforms has already been developed in the commercial market, as well as in academia. However, most of these applications have focused on the pre-construction and construction phases of project development. Attempts to tackle the issue in earlier design phases were limited to compiling a library of standard building parts that the designer has to pick from, significantly hindering design creativity. If the prefabrication expertise could be more intelligently integrated in the earlier design stages, it would prevent costly problems that may emerge down the line, without impeding design freedom. Unfortunately, it is not always easy to bring all project participants to work together from the earlier design stages and throughout the project. This is due to the prevalence of legacy models of building design and construction management, and due to the proliferation of fragmented, small-sized companies (especially in the German market). This is where an intelligent, domain-specific BIM system can potentially be used to help establish a prefabrication-aware design process.

Chapter 3: Rule-based expert systems and BIM

3.1. Introduction

With the lack of sufficient digitization and productivity growth in the AEC domain, including the prefabrication/precast sub-domain, it is essential to continue to develop the necessary digital tools that can drive up productivity. To face this challenge, utilizing rule-based expert systems emerges as one of the important approaches that were mentioned repeatedly in literature. These systems operate on an expert knowledge base that needs to be formulated in a logical structure to be easily transformed into machine-readable checking instructions. These instructions operate best on intelligent BIM models in a BIM-based model-checking environment, given that they possess sufficient detail. This chapter discusses rule-based expert systems, their definition, characteristics, and suitability in constructability checking in general, and in the context of this research. Then the chapter introduces different forms of knowledge representation inside these systems, with a specific focus on decision tables. Finally, the chapter explains BIM-based model-checking environments, on which an expert system can operate, and the standard used to measure the development level of BIM models needed in the checking process.

3.2. Rule-based expert systems

3.2.1. Lack of sufficient digitization in the AEC industry

The AEC industry suffers from the lack of digitization and digital tools, which in turn greatly hurt productivity and efficiency of construction practice. In a study conducted in 2015, McKinsey & Company[8] examined sectors across the US economy and compared them to capture how companies in each sector use digital tools and how does that affect productivity. The study found that different sectors in the US economy are digitizing unevenly, with the construction sector ranking in the bottom five sectors in terms of overall digitization (among 22 sectors), while ranking among the top five sectors among the employment share. The construction sector also scored the worst productivity growth rate among all examined sectors, with a value of -1.4% in the period between the years 2005-2014 (Manyika et al., 2015, p.5) (Figure 3-1). This is also apparent in the German market where a survey about the state of digitization was

[8] McKinsey & Company is an American consulting firm specializing in the evaluation of management decisions in both the public and the private sectors, with clients representing 80% of the largest world corporations.

carried out among professionals from various sectors. Two-thirds of the respondents see that their own company is not well situated for a digital transformation (BauenAktuell, 2019). This lack of digitization and productivity calls for more research and development in the field of expert systems and data-rich BIM models. The intelligent use of these two approaches together could help the industry eliminate a lot of the laborious work undertaken by individuals, and close the huge gap in productivity. The rest of this chapter will attempt to cover the relevant aspects in these two topics in the scope of this research.

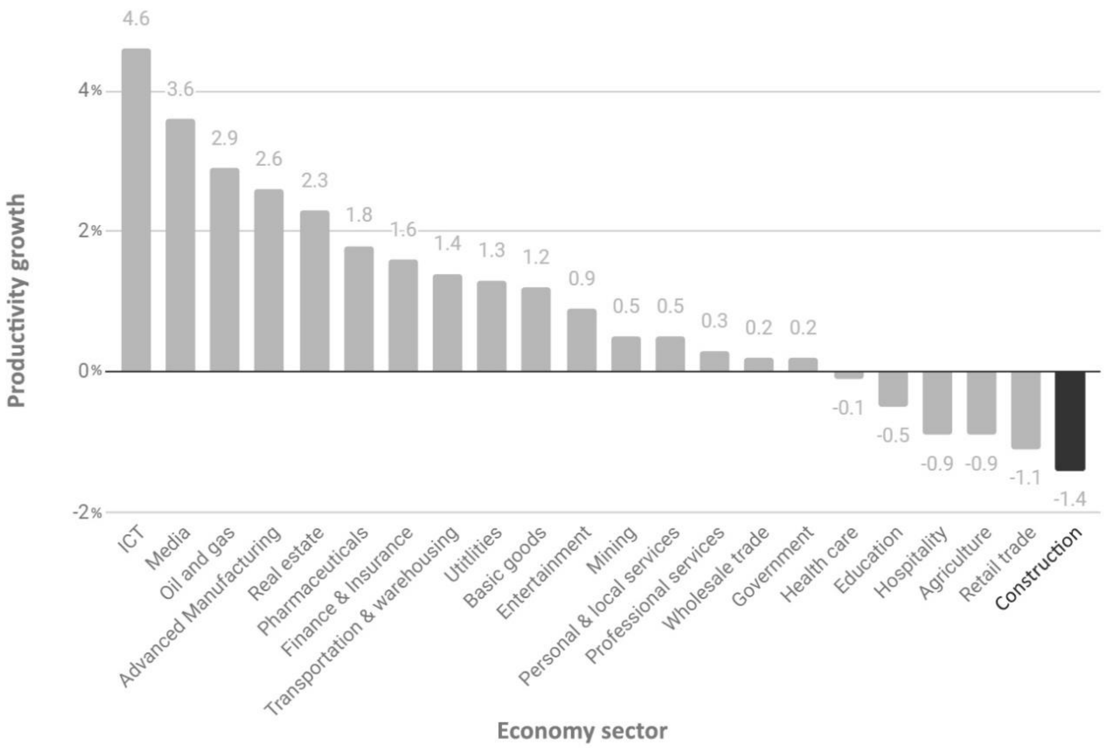

Figure 3-1 Productivity growth in different sectors of the US economy as measured in the period between 2004-2015. Based on data from (Manyika et al., 2015, p.5)

3.2.2. Expert systems for decision support in the AEC industry

As discussed before in the review of literature in chapter 2, many of the reviewed tools were either rule-based checking algorithms, automated detailing algorithms, or design optimization algorithms, among other approaches that can loosely fall under the umbrella of “Expert Systems”. An expert system is a specific branch of Artificial Intelligence (AI) that makes use of specialized knowledge to solve problems (Nagori and Trivedi, 2014, pp.20-33). It operates on a knowledge base of experts, mimicking their reasoning in practice, and achieving good performance in a specific domain (Nelson Ford, 1985, pp.21-26), eventually solving real-world problems, reaching the same conclusions that a human expert would reach in a similar situation (Adeli, 1988, p.5). To that end, expert systems require detailed input about a specific domain and the mechanisms for the application of this input in the domain (Puppe, 1993, p.3). This is usually done by extracting domain knowledge from the human expert “knowledge elicitation”, and then transferring it into a form that computers can deal with “knowledge representation” (Finlay, 1990, pp.535-543). The area of knowledge covered in the domain should have two main characteristics (Nelson Ford, 1985, pp.21-26):

- The domain is well defined and sufficiently specific to enable the successful transformation of knowledge into computer-readable code.
- The knowledge base can be obtained in an articulate form and the methods of applying it in practice are elaborate.

Expert systems could be either “Advisory systems”, where the users make the final decision, or “Dictatorial systems”, where the system makes final decisions without user consultation (Grosan and Abraham, 2011, p.176).

3.2.3. Definition and characteristics of rule-based expert systems

Rule-based expert systems are a type of expert systems, among many types such as fuzzy or neural network expert systems (Nagori and Trivedi, 2014, pp.20-33), in which knowledge is represented in the form of sets of rules that leads to different conclusions in different situations (Grosan and Abraham, 2011, p.149). Characteristics of rule-based expert systems can be summarized as follows (Adeli, 1988, pp.8-9, Grosan and Abraham, 2011, p.175):

- The system is not biased and makes only rational decisions based on predetermined logic, providing consistent results.
- The system acts based on predetermined logic, meaning that these systems cannot learn or develop on their own. However, the knowledge base can be expanded and incrementally developed over extended periods of time.
- It can hold and maintain significant amounts of information and expertise and process them quickly.
- It can reduce work time, cost, and errors by automating repetitive, tedious tasks.
- There is always the possibility that human experts are unable to explain their logic behind some decisions, or that these decisions are based on human intuition, which makes expert systems sometimes susceptible to incomplete knowledge and reasoning representation.

Rule-based expert systems, in their simplest form can be described in the form of If-Then statements. The “If” part represents the “Condition” and the “Then” part represents the “Action” or “Conclusion”. The rule tests the logic in the “Condition” and according to the evaluation (True or False) asserts a conclusion or takes an action. Additionally, rules can have multiple conditions, joined by logical operators, for example “And/Or”, resulting in one or more conclusion/action also joined by logical operators (Figure 3-2). Knowledge representation inside a rule-based expert system can also take different representational forms. The basic premise, however, remains the same: Conditions that are tested either to arrive at conclusions, or to take subsequent actions.

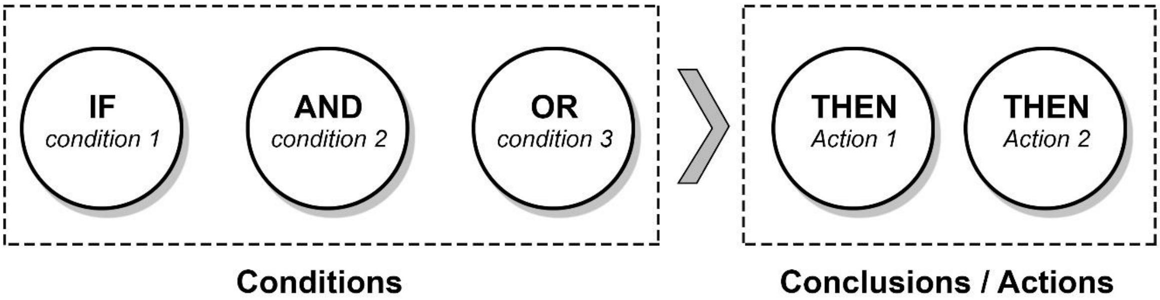

Figure 3-2 An example of the structure of a basic If-Then rule

3.2.4. Reasons for choosing rule-based expert system as the research approach

Rule-based expert systems have been extensively used for checking building design models for fire code requirements, building code requirements, design issues related to barrier-free access, initial feasibility assessments, among other issues (Eastman et al., 2009a, pp.1011-1033). Such systems have also been widely used in AEC applications for checking for construction-related issues. For example, a system known as “BERT” was developed by Cornick and Bull (1987, pp.117-126) as a design “advice” system, focusing on brickwork cladding. Designers insert their brickwork cladding layout, including the proposed movement joint layout, and the system presents comments and suggestions on the problematic parts, which the designer could willingly incorporate in his design. More recently Ma et al. (2013, pp.707-716) presented an approach to enable users to computerize the “Chinese Code of valuation with a bill of quantity of construction works” in the form of so-called “Discrimination rules” to automatically identify any given Bill of Quantities (BOQ) item. The authors realized their concept through the creation of a prototype tool, where users can graphically establish these rules to identify & filter BOQ items and their quantities, and then potentially use the results to obtain price calculations. A rule-based system was developed by Kim and Teizer (2014, pp.66-80) that recognizes geometric and semantic parameters in a building model, identifies the locations in need of scaffolding, and then produces a scaffolding system design automatically. Research in (Jiang and Leicht, 2015, p.A4014004) presented an approach to test cast-in-place concrete projects' models for formwork constructability, as well as to select the optimum formwork system, based on constructability rules developed from previous research. An approach to BIM-based compliance checking of deep foundation construction was presented by Luo and Gong (2015, pp.549-576). The research realized more than 300 checking rules comprising the knowledge library, and standardizing the information required by the system to operate, improving efficiency and precision instead of manual checking lists. A tool based on a Design-for-Safety rule-based knowledge library was demonstrated by Hossain et al. (2018, pp.290-302). Operating on BIM models, the system helps the designers identify risks and construction safety concerns associated with their designs, as well as required design features.

The following sections will focus specifically on advisory rule-based expert systems for design decision support, as they will be the approach of choice in this research. The reasoning behind choosing this type of systems is three-fold:

- They have been used extensively and successfully in many constructability-checking applications in the AEC industry as already shown.
- Nature of constructability checking relies mostly on dimensions of elements, distances between elements, quantities, material properties, and modularity (Fischer and Tatum, 1997, pp.253-260). These aspects and their relationships can be easily coded into computer-readable rules, and these rules can be easily applied in intelligent models.
- These systems can provide the flexibility needed in design operations, when their role is limited to "decision support" or put another way an "advisory" role, instead of "dictatorial" role like intrusive automated detailing or optimization algorithms.

3.2.5. Structure of rule-based expert systems and the rule-checking process

At its core, a rule-based expert system consists of three main components (Jiang and Leicht, 2015, p.A4014004):

- A knowledge base (rule base): This part stores expert knowledge from domain experts and/or literature in a logical codified form.
- Inference engine: This part performs actual reasoning on the inputs based on the rules in the rule base, arriving at the relevant conclusions.
- User Interface: This part acts as the means of a two-way communication channel between the end-user and the expert system.

Such systems depend on extracting accurate design information from design models, with the appropriate level of detail in the proper design stage that the system addresses, passing this information to its inference engine, which tests this information against its knowledge base, and finally presents the user with the results via the user interface. A clear structure is usually followed in order to implement a functional rule-based checking and reporting process. An overview of the structure can be found in Figure 3-3. Based on (Eastman et al., 2009a, pp.1011-1033), this structure can be broadly described in four stages as follows:

- Rule interpretation

It was already mentioned that rule-based expert systems operate on knowledge bases that should be easily converted into machine-readable code. However, before being able to write that code, knowledge that is typically expressed in the form of natural written or spoken language has to be translated into an intermediate clear logical format to ensure consistency and avoid conflict. The simplest form of this intermediate format is logical IF-THEN statements, but they can also take more sophisticated forms such as tables or trees. This point will be subject to a more in-depth discussion in section 3.3 of this chapter.

- Building model preparation

This depends on the form of the “model” being used. Using 2D design models (2D drawings) means that these models should be visually correct and have sufficient, consistent textual information attached to them. In parametric, information-rich BIM models, they have to be prepared so that they meet the sufficient level of development (LOD) necessary to perform the required checks, which will be the subject of discussion in section 3.4.3 of this chapter. Sometimes the system also needs to do some filtering of model elements and/or pre-checks for information completeness and consistency, or even derive new models from the base model.

- Rule execution

Here the rules are applied to the prepared design model to be able to obtain the required results, managing a consistent process of rule implementation, and capturing the outputs to report them to the user or to use them on dependent rules.

- Reporting

This is the final step of the rule checking process. The results are presented to the user with sufficient categorization, contextualization, and visual feedback (for example using dedicated 3D or 2D views) that enables the user to identify the problematic parts with relative ease. The feedback may be reactive (reporting only), or proactive (editing and changing) (Jiang and Leicht, 2015, p.A4014004).

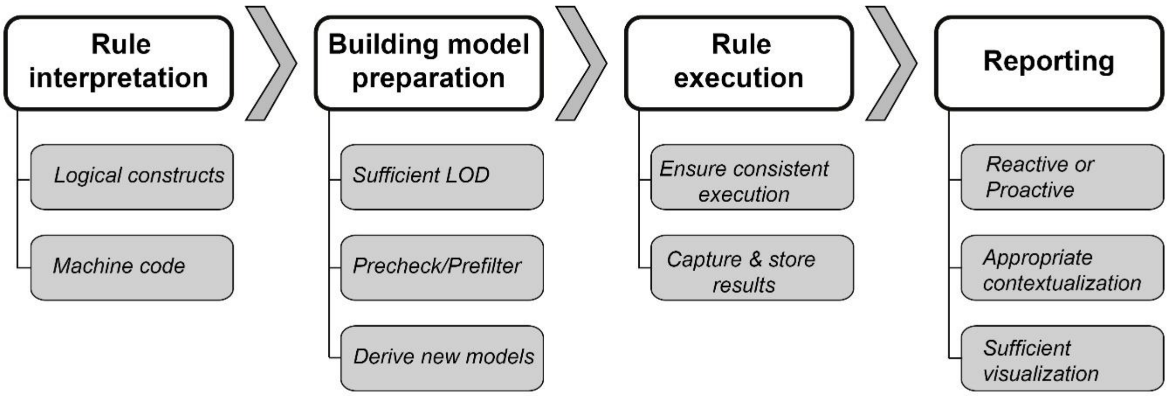

Figure 3-3 An overview of the structure of the rule checking process. Adapted from (Eastman et al., 2009a, pp.1011-1033)

3.3. Knowledge representation and formalization in a rule-based expert system

3.3.1. Forms of logical knowledge representation

It can be safely said that the most important part of software development is not writing the actual code for systems, but rather the capture and transformation of vague, unclear, and sometimes contradictory business rules into consistent, clear logic (Lieberman, 2012, p.1). As discussed before, the "IF-THEN" form is the simplest form of rule representation logic, but the essence of rules is that they are structured as well-defined pairs of condition-action statements. Compiling an exhaustive list of knowledge representation and formulation systems is out of the scope of this research; however, the following text shows some examples of the more popular examples (Adeli, 1988, p.15, Lieberman, 2012, pp.2-5):

- Object-Attribute-Value (OAV) triplets

One of the simpler forms of rule representation is OAV triplets. It is a special case of semantic representation where a specific value, property, or feature of an object is specified and the purpose is merely only limited to get a True (T)/False (F) result on the following query: Does the actual value of a specific attribute of an object matches a preset standard value? A representation of OAV triplets can be found in Figure 3-4.

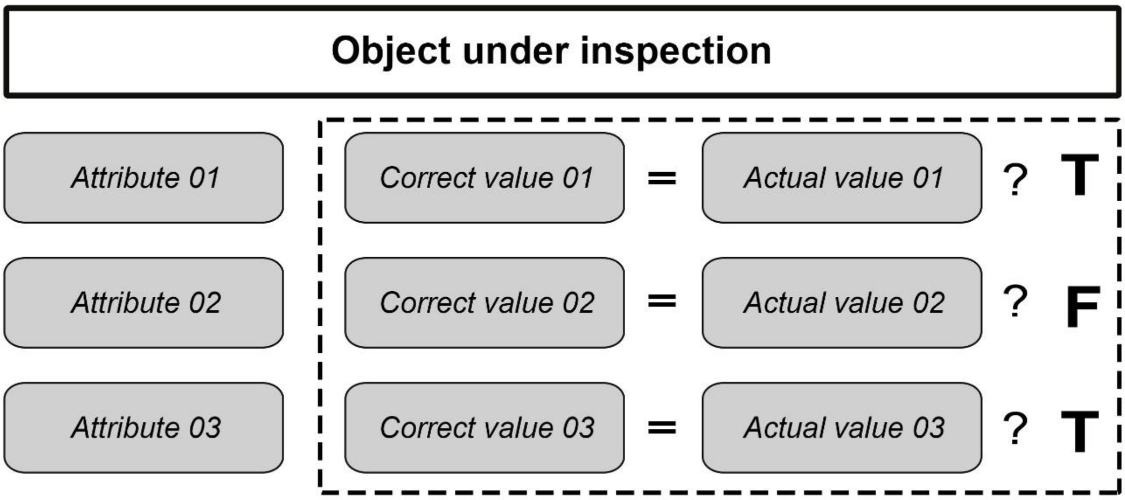

Figure 3-4 A generic representation of OAV triplets

- Tabular form

The tabular form is widely supported in commercial and open-source rule engines. This form of rule-representation also operates on condition-action representation, with the added benefit of being able to represent multiple conditions associated with multiple actions. It is also possible to link multiple tables together to be able to represent rule-groups that are more complex in an easy-to-follow and consistent manner. There are many forms of the tabular format, the most popular of which is a "Decision table", which will be the subject of discussion in section 3.3.2. A generic representation of rules in the tabular form can be found in

Table 3-1.

Table 3-1 A generic representation of the tabular form of rule representation

Object under inspection

Conditions	*Case 1*	*Case 2*	*Case 3*
Value A > Value B	**T**	**F**	**F**
Value B > Value C	**T or F**	**T**	**F**
	Take action X	Take action Y	Go to table ()

- Decision trees

This form of representation is used for representing multiple interconnected, dependent rules, but unlike the tabular form, it depicts the full decision/rule path from start to end in a single diagram. This could be viewed as an advantage in simpler situations where rules are of manageable size. However, in larger, more complicated rules, the tree could grow very large, and discerning different rules becomes very difficult. A representation of the basic form of a decision tree can be found in Figure 3-5.

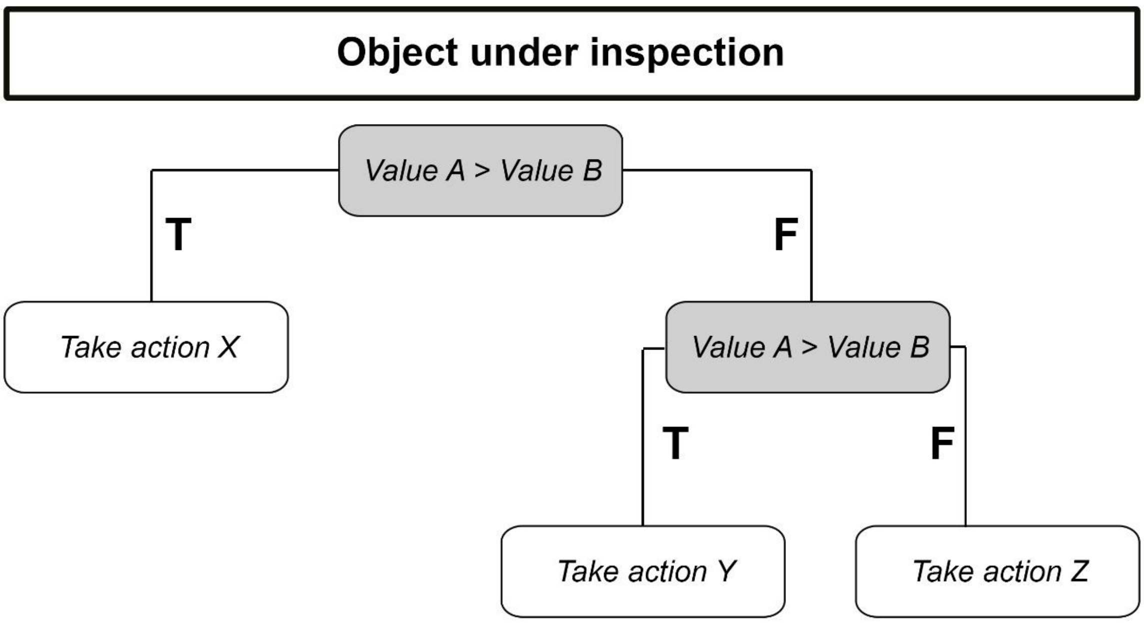

Figure 3-5 A basic representation of a decision tree consisting of only two levels

3.3.2. Definition and structure of decision tables

As discussed previously, the tabular format is considered one of the most extensible, efficient, and consistent ways of rule representation. A well-established form of this format is a "Decision table", which is essentially a concise tabular representation of related rules in the form of logical conditions and the corresponding actions to be taken based on the outputs of these conditions (Fenves et al., 1969, p.6). Decision tables have been used since the sixties in the engineering domain (Harris, 1981, p.20). For example, they have been used to translate the AISC[9] specification of the design of structural steel buildings (Fenves et al., 1969, p.6). They were also used to represent code requirements for reinforced concrete (ACI, 1973, p.10), to formalize seismic design provisions for buildings (Wright and Fenves, 1979, p.23), to represent design provisions of structural components (Cronembold and Law, 1988, pp.255-273), as well as to represent building envelope design codes (Tan et al., 2010, pp.203-211). More recently, they are also used in computerized systems of financial institutions for decision support in loan requests based on the client's credit history, assets, employment, and so forth (Lieberman, 2012, p.4).

The structure and components of a decision table are shown in Figure 3-6. The basic layout of a decision table is divided into two horizontal sections and two vertical sections, essentially forming four quadrants. The two vertical sections are called "Stubs" (to the left) and "Entries" (to the right), and the two horizontal sections are "Conditions" (in the upper part) and "Actions" in the lower part. Therefore, we have the following quadrants:

- Condition stubs: Show the questions or attributes of objects relevant to the rule.
- Condition entries: Represent the possible answers to the question, or attribute values, in the scope of the designated rule-set.
- Action stubs: List of all actions that could be taken in the scope of the rule-set.
- Action entries: Specify the particular action(s) to be taken based on the rule.

In simpler terms, we could say that each column in the right section of the table represents a complete rule, with an imaginary "IF" on top of it. The column, in its upper section, represents the condition(s) part of the rule and, in its lower section, specifies the action(s) to be taken according to these conditions. Not all individual, unique sets of conditions need to be represented in the table, sometimes an "Else" rule is used to shorten the table, resulting in a

[9] AISC: American Institute of Steel Construction

single action for all other combinations of conditions that are not explicitly mentioned in the table. The term “action” is generic and may be used, for example, to indicate the assignment a value to a variable, printing a message, or linking to another table, among other functions (Fenves et al., 1969, pp.6-7). The “entries”, whether a condition entry or an action entry, can be classified as follows (Pollack et al., 1972, pp.8-9):

- Limited entry

In the condition entries: Here entries are limited to a simple “Y” or “N” (True or False) answer to a Yes/No query in the condition stub.

In the action entries: Here entries are limited to a simple “X” or “-”. “X” means that the action is to be executed if the conditions of the rule are met, while “-“ means that action is to be ignored in that case.

- Extended entry

In the condition entries: They enable the conditions to be represented in a much more concise manner, representing many limited entries in one entry, using values other than “Y” and “N”. This is better explained through a practical example, which can be viewed in Figure 3-7.

In the action entries: Same as condition entries, condensing many limited entries into one entry. See Figure 3-7.

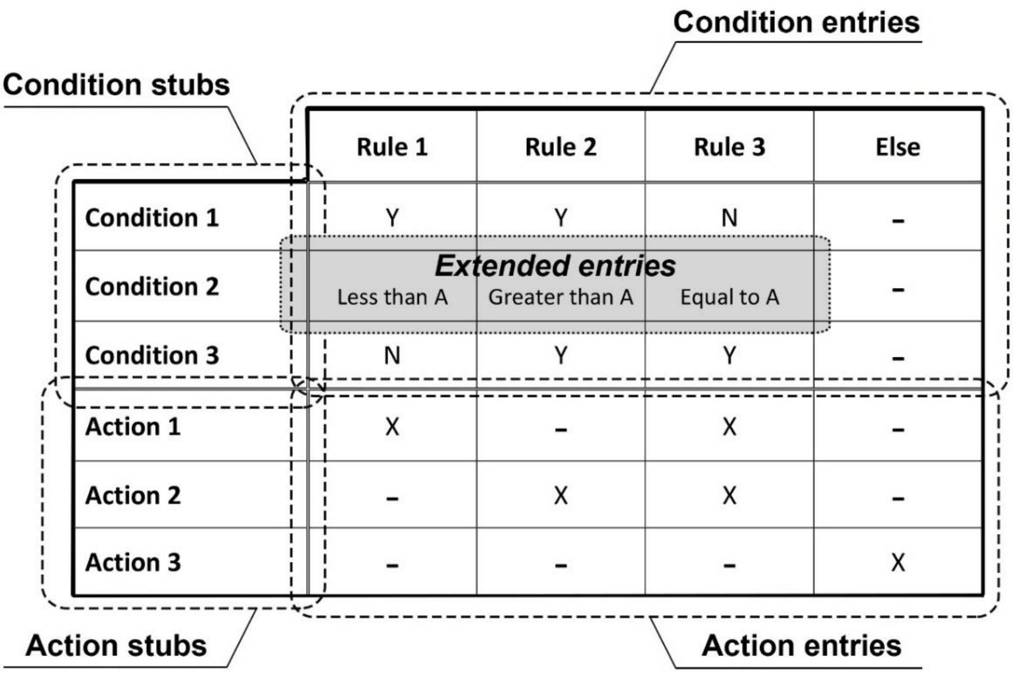

Figure 3-6 The basic components of a generic, mixed entry decision table. Reproduced from (Mekawy and Petzold, 2018, pp.71-80)

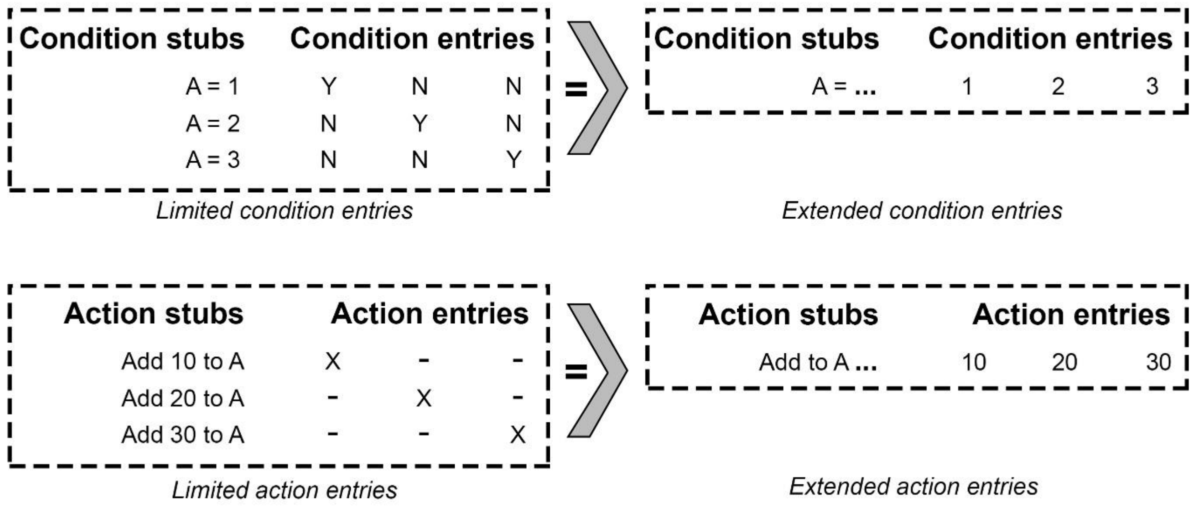

Figure 3-7 Difference between limited entries (the left part) and extended entries (the right part) in a decision table. The same rules can be expressed in a more concise manner. Adapted from ((Pollack et al., 1972, p.9)

3.3.3. Types of decision tables

In a rule-based system for design checking based on decision tables, the tables come in three different varieties (Fenves et al., 1969, p.17):

- Switching tables:

These tables act as a directory that indexes which table from the database is relevant for execution under the given inputs and the current situation. An example is shown in Table 3-2

- Working tables:

Having received direction from switching tables, the system may need to calculate intermediate results or generate information, which is needed to complete the actual rule(s), both of which are done using working tables (Table 3-3).

- Testing tables:

These tables perform the actual test, generating the actual results and/or outputting the appropriate messages to the end user like Pass/Fail messages (Table 3-4).

Table 3-2 Shows an exemplary "switching" decision table

	Rule 1	Rule 2	Else
Condition 1	Y	N	-
Condition 2	N	Y	-
Execute Table …	(A)	(B)	(C)

Table 3-3 Shows an exemplary "working" decision table

	Rule 1	Rule 2	Else
Condition 1	Y	N	-
Condition 2	N	Y	-
Take Value of X = …	10	20	30

Table 3-4 Shows an exemplary "reporting" decision table

	Rule 1	Rule 2	Else
Condition 1	Y	N	-
Condition 2	N	Y	-
"Design is …"	Satisfactory	Partially satisfactory	Not satisfactory

3.3.4. Reasons for choosing decision tables as the knowledge formalization approach

Having developed an understanding of the structure and properties of decision tables, it is worth explaining why they are suitable as a knowledge representation method in the context of this research. Based on research and implementation of decision tables, their advantages can be detailed as follows (Fenves, 1966, pp.473-490, Fenves et al., 1969, p.3, Harris, 1981, p.19):

- It is much easier to locate logical errors in a rule-set than to locate syntax/numerical errors in a program. Decision tables allow for a clear logical formulation of rules in a way that provides all possible conditions, which makes the rule-set easier to "Debug" in comparison to other forms of rule formalization and in comparison to translating knowledge directly to code.
- Their structure, which enables the representation of all conditions in a certain rule, makes it much easier to identify gaps, inconsistencies, and situations where knowledge is not defined/provided.
- They enable the description of processes involving parallel reasoning, in contrast to flow charts or decision trees that only describe reasoning in sequential patterns.
- They can be easily analyzed and tested to ensure that the reasoning process always operates on unique rules, providing unique results, lowering the probability of writing contradictory rules (taking different action for the same conditions), and redundant rules (same conditions with same actions) (Pollack et al., 1972, pp.53-54).
- Decision tables are extensible and can be easily ratified to formulate future editions of specifications and to cover existing gaps
- Finally, they complete the essential phases of computer programming; the translation of these rules into actual computer code is practically a mechanical process, which can even be sometimes done automatically using digital tools.

3.4. Towards an automated, BIM-enabled, rule-based approach in Prefab design

3.4.1. BIM-based model checking (BMC)

Having explored the benefits of using BIM-enabled workflows in many aspects in precast design in chapter 2, it is worth focusing here on BIM-based model checking (BMC) because it is the approach of choice in this research. In almost any design-checking situation, it has been found that using automated reviews using BIM models is far more superior than using a drawing-based checking. A study done by Lee et al. (2013, pp.144-156) found out that experienced professionals using drawings found as low as 0%, and a maximum of 4.6%, of the total number of errors discovered through automated BIM checking.

BMC can be broadly defined as a procedure that processes input obtained from BIM files according to pre-defined rules, essentially operating based on a system composed of the following (Hjelseth, 2015, pp.33-61):

- Rule-sets: Collections of rules in a certain domain.
- BIM content: The necessary BIM files, developed to a sufficient level, which are needed to perform the checks.
- Software: Includes the platform(s) providing the functionality necessary to process and implement the rules, manage the BIM content, and visualize and report the results.

3.4.2. BMC implementation examples

BMC has been used extensively in recent years for a myriad of purposes. One of the most prominent examples of the usage of BMC on a wide scale is the CORENET platform, which is operated by the Building Construction Authority in Singapore. The platform allows firms to submit projects for license approval in the IFC standard, providing code compliance checking for Singaporean regulations concerning accessibility, fire safety, among other aspects (Preidel and Borrmann, 2018, p.373). Similar approaches were adopted in Australia with the DesignCheck system for checking for requirements of the building code in Australia, as well as in USA, where the GSA (General services administration) and a team from Georgia Tech developed a platform to check for circulation and security rules in BIM models of US federal courthouses (Lee et al., 2016, pp.49-61). To provide few other examples for the usage of BMC for design, Tan et al. (2010, pp.203-211) presented an approach that relied on the BIM-based simulation of facades' thermal performance; the values obtained are then automatically compared to values specified in design codes using a rule-based engine to present feedback to the designers. A more general approach was introduced by Preidel and Borrmann (2016, pp.402-421), where a Visual Code Checking Language (VCCL) was developed to flexibly check BIM models against different sets of design rules. The authors demonstrated the use of the developed language using two examples: the rule for required ventilation area in rooms according to the German fire code, and the rule for staircase location for fire escape scenarios in the Korean building act. Shaohua and Zheng (2017, pp.156-163) developed an approach, based on IFC building models and rule-based code checking, to check for the requirements from the “Code for green construction of building” and the “Evaluation standard for green construction of building” in China to ensure building compatibility with green practices. In the field of BMC for construction, Ji and Leite (2018, pp.78-90) proposed a framework that integrates BIM models, construction schedules, and rule-based checking to provide feedback on tower crane plans based on a rule-set for tower crane specification.

3.4.3. BIM Level of Development (LOD)

As shown in section 3.4.1, BIM content with sufficient detail is necessary for a successful BMC process. This concept is better explained in BIM practice through the viewpoint of LODs. LOD specifications provide a reference that enables the accurate articulation of the prerequisites of BIM content in BIM models at various design and construction stages (Bimforum, 2016, p.9). This acts as a more accurate measure of design maturity than the concept of a “Design phase”, whose definition can greatly differ among AEC firms. An overview of the LOD levels can be viewed in Figure 3-8. According to (Weygant, 2011, pp.80-84, Borrmann et al., 2018, p.136, Bimforum, 2016, pp.12-13), the specification for each LOD level could be described as follows:

- LOD 100: Model elements have mere symbolic representation or basic masses without accurate shape, size, or location. Basic preliminary volume and area calculations and cost estimations can be derived from the model.
- LOD 200: Model elements are represented in the model as generic systems or assemblies with approximate sizes, shapes, locations, and orientations. Some non-graphic information may also be represented as attachments to model elements.
- LOD 300: Model elements are well defined and represented in the model as specific systems with specific sizes, shapes, locations, and orientations and with non-graphic information attached to them.
- LOD 350: In addition to the specifications of LOD 300, elements’ interfaces with other systems, and parts necessary for coordination with nearby elements are provided
- LOD 400: In addition to the specifications of LOD 350, elements are also modeled with sufficient detail, information, and accuracy to enable its fabrication and installation.
- LOD 500: This level does not represent progression to a higher level of design maturity; it just represents a field-verified representation of model elements concerning their sizes, quantities, shapes, locations, and orientations.

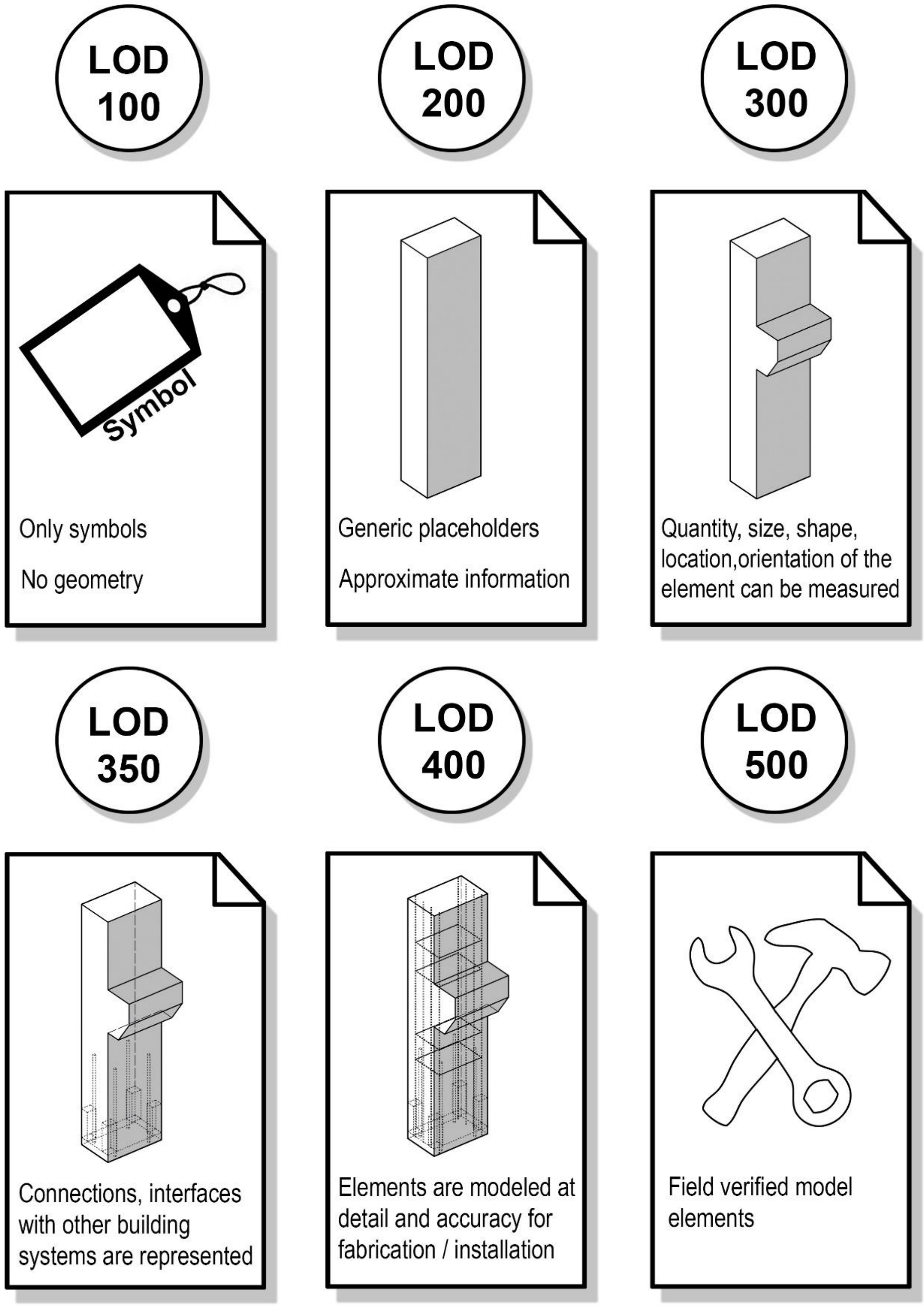

Figure 3-8 The different levels in the LOD specification. Based on input from (Bimforum, 2016, pp.39-41)

To pinpoint the LOD level appropriate for the context of this research, it is worth spending some time discussing the difference between the roles of the architect and the precast expert in AEC projects. In Germany, as already mentioned in section 2.3.2, the practice of design is regulated through the HOAI (Official Scale of Fees for Services by Architects and Engineers).

It details the tasks and outcomes expected from architects and engineers from concept to site supervision and documentation. It includes nine phases in total, of which phases 1-5 are concerned with the actual design activities as follows (HOAI, 2013, pp.41-49):

- Phase one (Consultation): Fundamental tasks and scope are clarified.
- Phase two (Preliminary Design): A design concept is developed with drawings and a preliminary cost estimate.
- Phase three (Design development): Feedback from the client is integrated, overall planning is developed, and a detailed cost estimate is developed.
- Phase four (Building permit drawings): All the detailed documentation necessary for receiving permits is prepared and submitted and the permit is obtained.
- Phase five (Detailed building design): All the drawings and details needed for the project construction are prepared.

The precast expert’s role, on the other hand, comes in preparing “Erection drawings” and “Production drawings”. Erection drawings are prepared from the contract documents from the architect and provide the precast expert’s view of the shape and configuration of beams, columns, slabs, etc. Production drawings are prepared from erection drawings and act as instructions to plant workers, providing accurate details that are needed to build the concrete moulds, reinforcement, and to prepare material lists (PCI Industry Handbook Committee, 1975, pp.20-21).

Since the aim of this research is concerned with the role of the architect and improvements in his practice in precast projects, the focus will not be on the fabrication and installation details. Instead, it will be on general layout planning, sizing, and placement of precast building components. The juxtaposition of the LOD levels with HOAI phases is shown in Figure 3-9. It shows that LOD 200 and LOD 300, corresponding to phases two, three, and four in the HOAI, represent an appropriate benchmark for the BIM model development level and the project phase in which this research is relevant. These are the phases in which general planning and element sizing, orientation, shape, and location are still rapidly changing, with just about the sufficient amount of detail and information needed to carry out the preliminary precast checks designated in this research.

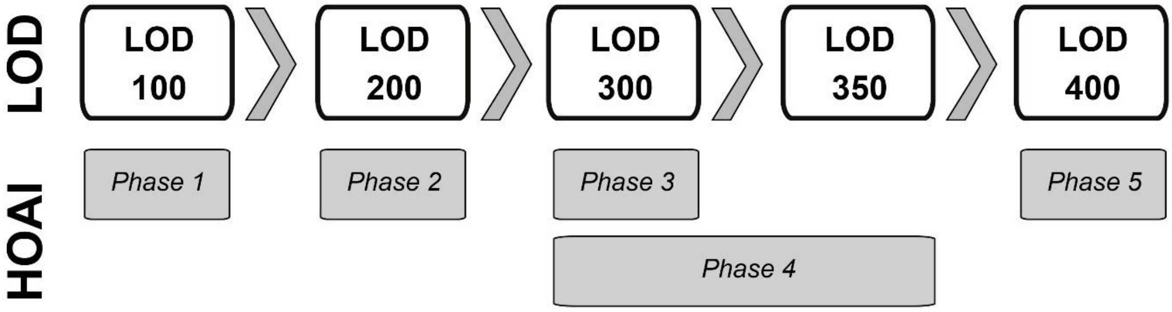

Figure 3-9 The relationship between HOAI phases and LOD levels. Adapted from (Mini, 2016, p.33)

3.5. Conclusion

Rule-based expert systems emerge as one of the most important approaches to drive forward the digitization and automation in the AEC domain. This is especially relevant in the field of constructability checking, where they have been demonstrated to work very well. The idea behind these systems is to replace the laborious, repetitive, error-prone operations undertaken by human experts. For these systems to work, knowledge, on which humans base their judgments, needs to be formalized in the form of logical rules that can easily be transformed into computer-executable code. One of the most efficient ways used to represent expert knowledge is decision tables. They can be easily extended, amended, and tested for logical errors, making them ideal for constructability knowledge formalization. A rule engine that administers the formalized rules, in their digital form, needs also intelligent design models with sufficient detail and information for the execution of these rules. To that end, BIM-based model checking is widely used in a myriad of design checking applications, and it will be the logical approach of choice in this research. For the purposes of this research, it was determined that BIM models with a level of development 200 to 300 are ideal for the aim of checking for precast requirements in earlier design stages. Having established the main approach to tackle the objectives of this thesis, the next chapter will provide the structure and the details of the framework proposed, how it is developed, and how the precast rule-sets are formulated.

Chapter 4: A BIM-enabled, rule-based system for checking prefabrication requirements

4.1. Introduction

After tackling the first two steps in the methodology established in chapter 1, namely “Process review” in chapter 2, and “Computer systems review” in chapter 3, it is time to tackle the third and main step to realize the objectives of this thesis, which is “System framework development”. This framework defines the requirements of the computer system that checks early design models for precast considerations and reports the results to the user. The first step that will be discussed in this chapter is how to establish the main structure of the framework and how it works. Then the chapter will dive into the steps followed to establish the rule database, on which the system operates. Establishing the database involved many steps including knowledge acquisition, knowledge structuring and categorization, and knowledge formalization and representation. Finally, the chapter will show exemplary rule-sets from the established database, in decision-table format, involving all main layers of the precast database, and explaining the reasoning behind these rules.

4.2. Establishing the structure and procedure of the framework

As mentioned in the methodology section in chapter one, design-related research is based on the definition of a “current situation” model, an “improved” model, and a “support” to transform the current model to the improved one. In the context of this research, the “current situation” mostly does not take early design checks for precast requirements into consideration, and when it does, it is usually a manual process that is performed much later in the design process. The purpose of this research is to establish the framework for a support system that enables an “improved” situation where early design schemes can be checked and problematic design decisions are automatically determined and pointed out to the designer in an iterative, interactive manner (Figure 4-1). The proposed system should comprise structural requirements, cost requirements, transport requirements, and production requirements for precast concrete in a specific market. The system should then apply these requirements to digital building models to screen the models’ components for “incompatible” parts and report them back to the user (Figure 4-2).

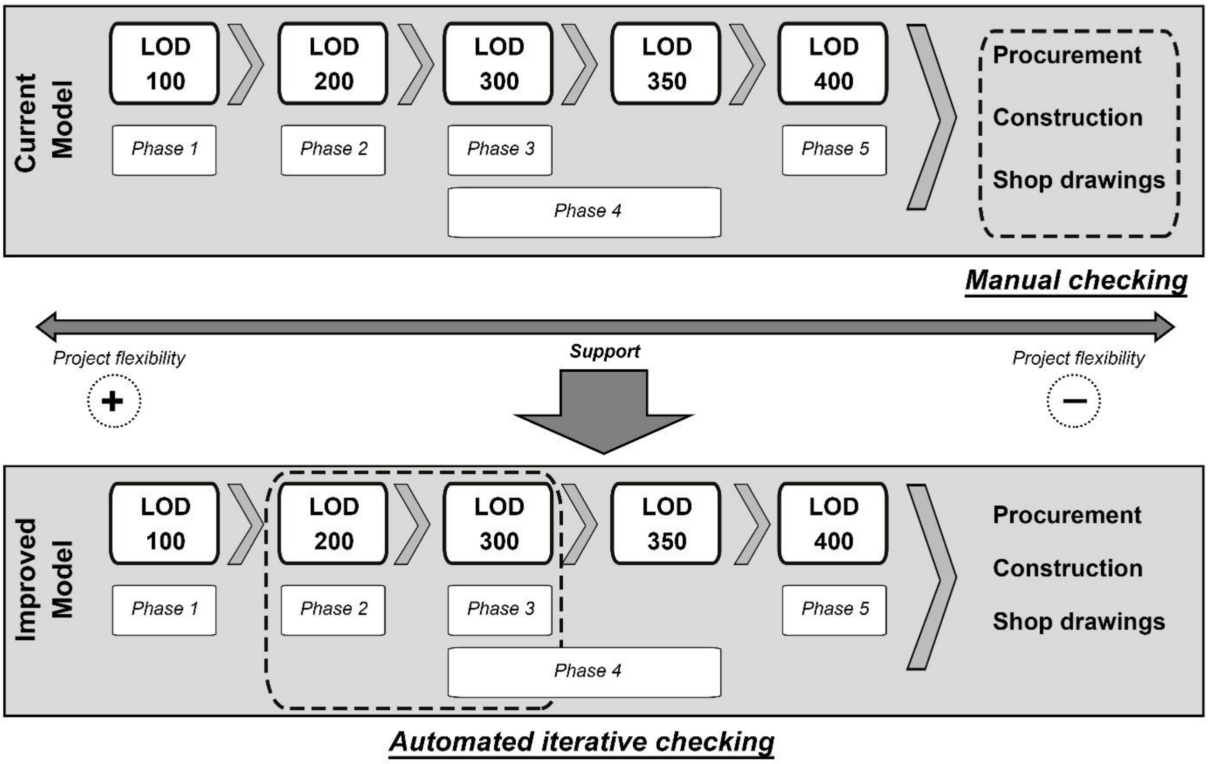

Figure 4-1 An overview of the "Current" vs "Improved" models

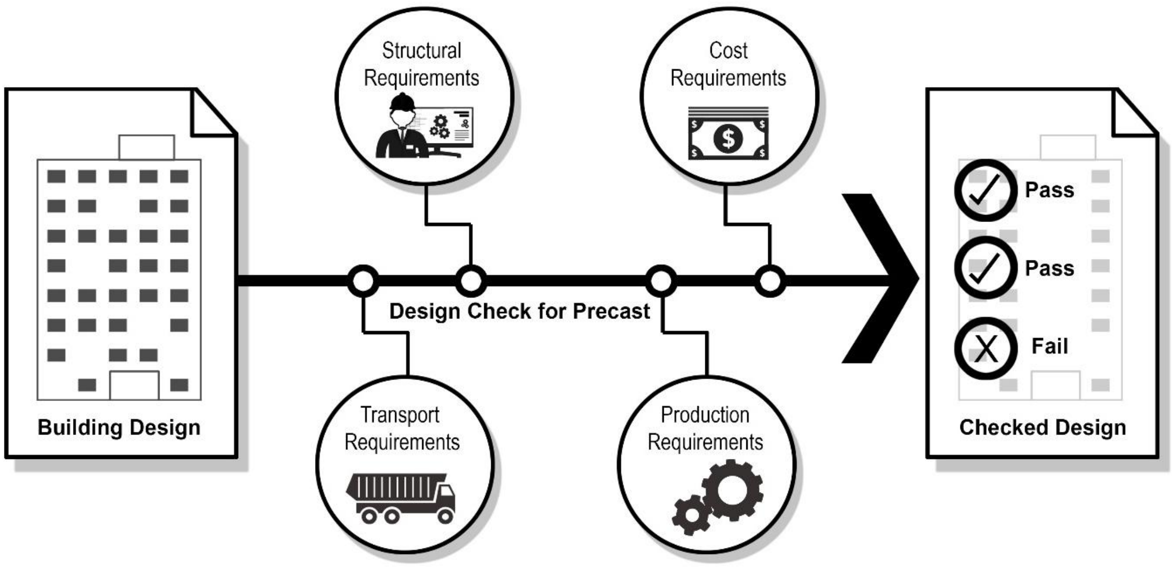

Figure 4-2 An overview of the sequence of the proposed system

As previously mentioned in chapter 3, rule-checking systems are composed of three components: a rule/knowledge base, an inference engine, and a user interface. According to Sacks et al. (2018, p.80), such systems need to have the following functionality:

- Recognize the rules and rule-sets
- Recognize and extract the necessary geometry and/or data in the model needed as input to test the rules
- Apply the rules to the building model
- Identify all instances in the model that do not pass the test(s).

Following this structure, the framework established in this research can be described in three layers as follows (Figure 4-3):

4.2.1. Input

The input to the system is two-fold

- A BIM model in LOD 200 or LOD 300: The model in this stage should have enough detail and level of information for walls, slabs, columns, beams, and other model elements to carry out the checking process. It is also early enough in the project when the design is more flexible and changes are less costly.
- Manual user input: BIM models in general, and in early stages in particular, may still lack some information that is crucial to the checking process in this context. The user will have to complement the information present in the model with location data, building use, and construction system-related information.

4.2.2. Filtering and rule execution

A rule engine should be able to:

- Filter and sort the components of the BIM model and establish the dependencies between various precast building elements.
- Store the predesigned precast concrete rule database.
- Apply the rule-sets to the user input and arrive at the results.

4.2.3. Output

The output should consist of a report summarizing the problematic model parts to the designer via a user interface. The designer should be able to easily navigate through the results and spot the individual model issues. The designer should also be able to differentiate between problematic elements that need immediate attention and less important situations, where elements could be designed or situated in a different way only to facilitate production or to reduce overall cost.

The rest of the chapter will discuss the precast concrete rule database and its formulation, and how specific examples of the precast rule-set are applied to arrive at the results. The next chapter will deal with the issue of the practical implementation of these rules into a BIM model to check model elements, selecting which model elements need to be filtered and stored, and presenting the output to the user through a working prototype.

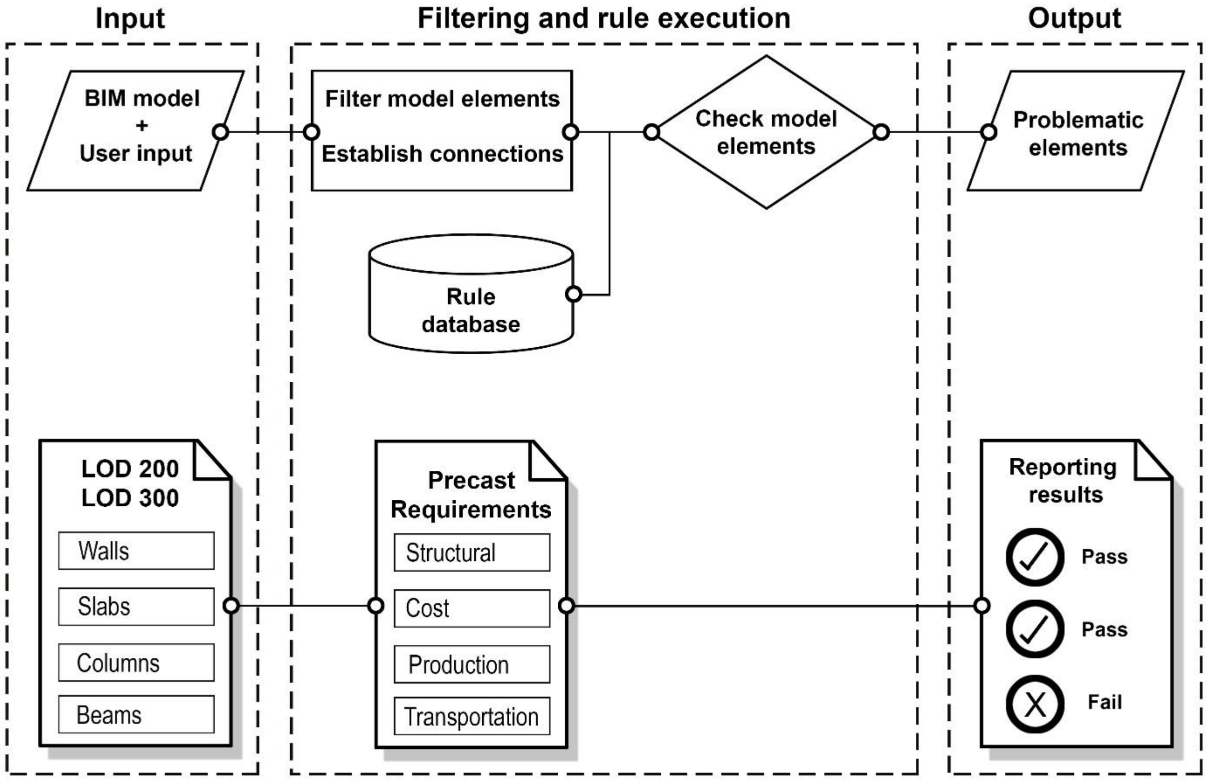

Figure 4-3 An overview of the framework proposed in this research

4.3. Establishing the rule database

The key part of the established framework is a solid, computer-interpretable rule database. To develop such a database, the research goes through three main steps, namely knowledge acquisition, rule-set structuring and categorization, and knowledge formalization and representation. These will be explained in detail in the following subsections.

4.3.1. Knowledge acquisition

The first step to build a knowledge base is to gather the expert knowledge necessary from available resources. In the context of this research, expert knowledge was gathered from four main resources:

- Semi-structured interviews with three precast experts with experience in projects in the German market. The interviews were conducted in the earlier phases of this research, they were exploratory in nature and limited to help understand the basics of precast-based design and construction practice in Germany, the nature of the communication process between the architects and precast specialists, and how to bring issues related to precast requirements to the earlier design phases. They were not meant for gathering all the technical details needed to formulate the rule-sets. Rather, they were meant as a guide for the literature review process[10].
- An extensive literature review to identify similar scientific research and publications that dealt with precast knowledge, constructability knowledge, and design decisions and conditions that are particularly important to precast construction.
- Textbooks and International standards: Including for example publications of the Precast Concrete Institute (PCI) (PCI Industry Handbook Committee, 2010), and publications of the International Federation for Structural Concrete (FIB/FIP) (Fédération Internationale de la Précontrainte "FIP", 1994), among others.
- National standards: Including, most importantly the publications and guidelines of the German association for precast concrete construction (Fachvereinigung Deutscher Betonfertigteilbau (FDB)) (Brandt, 1994, Borchardt et al., 2002, Hierlein et al., 2009, FDB, 2016). Also, relevant German DIN specifications were carefully studied such as the German DIN standard for live loads (DIN, 2010).

4.3.2. Rule-set structuring and categorization

As a part of the described framework, it is important to filter, organize, and establish dependencies between model elements to enable and facilitate rule execution. According to Lieberman (2012, pp.5-6), rule-sets themselves need to be organized in rule-groups with well-defined categories to permit a logical rule flow and allow for a clear representation of the overall

[10] For the list of questions posed during the interviews, see Appendix F

rule engine behavior, and the validation of rules in isolation. Such organization also helps humans experts audit, edit, complement, and transform the rule-set into computer code.

In the context of this research, the rules in the rule-set are categorized into two layers as follows (A graphical representation of all categorization layers can be seen in Figure 4-4):

a. Layer 1: By Precast element type

This is the uppermost layer that will be made visible to the user. Rules are grouped together according to the building element they relate to. Such an approach makes it easier to develop algorithms to filter elements for checks, establish intra- and inter-dependencies between element categories, and most importantly, enable the end-user to view and filter the results of the application of these rules in a simple, easy-to-understand manner.

According to (PCI Industry Handbook Committee, 2010, pp.1c-1 - 1c-5), precast elements are classified as follows:

i. Beams
ii. Columns
iii. Slabs
iv. Wall panels
v. Piles
vi. Stairs
vii. Shear walls
viii. Column covers

This research will focus on the first four items (i to iv) in addition to the creation of a fifth category called “General planning considerations”. This category contains rules that do not pertain to any specific precast element, and/or can be applied to all precast elements and was placed in this group to avoid duplication. This makes the final categories in this layer as follows:

i. General planning considerations
ii. Slabs
iii. Beams
iv. Wall panels
v. Columns

b. Layer 2: By Requirement category

This is the second layer beneath the previous layer, which labels the designated rule in one of the following categories:

i. Production requirements: These types of rules deals with dimensions, shape, size limitations, and preferences in precast concrete fabrication.
ii. Cost requirements: These rules represent best practice in the time/cost-efficient production of precast elements at scale.
vi. Transportation requirements: These rules determine requirements and limitations in the transportation of precast components, with a specific focus on the German market according to the German Road Traffic Act (StVO, Straßenverkehrs-Ordnung). These rules apply for all precast elements and therefore are all placed in the “General planning considerations” first layer category.
iii. Structural requirements: These types of rules deals with structural design and limitations of specific individual precast elements and cannot be generalized. The primary concern here in this research, dealing with LODs 200 and 300, is the sizing, placement, and shape of the elements only. These rules address issues directly

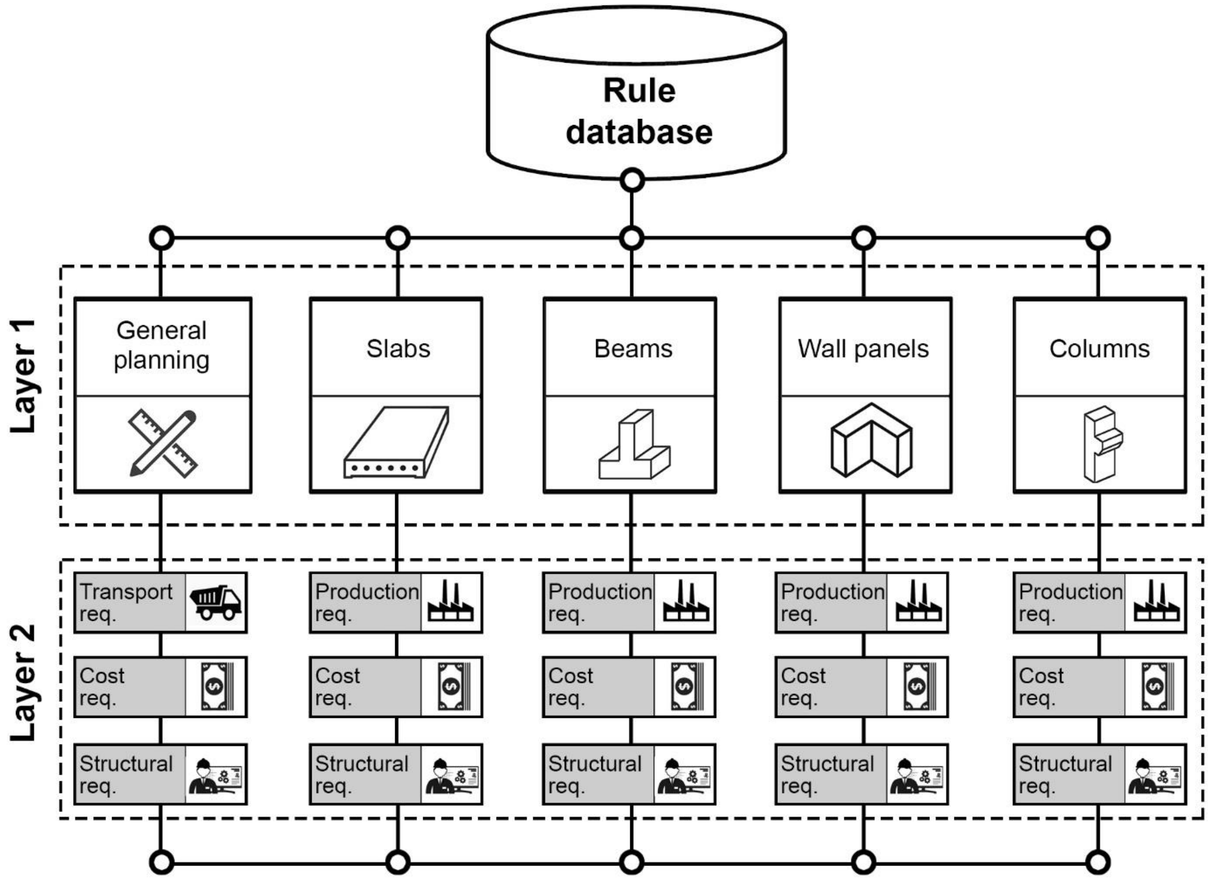

Figure 4-4 An overview of the layered structure of the Precast knowledge database

relevant to the architect's work (e.g. Sizing) and is not meant to go into greater structural detail (e.g. reinforcement details).

4.3.3. Knowledge formalization & representation

Having classified the expert knowledge in categories in the knowledge base, loose knowledge and guidelines need to be formalized into structured knowledge and represented into a machine-programmable format that can be implemented consistently, leaving no room for conflicting interpretations. A computer system could utilize this formalized knowledge, implementing different types of reasoning, to infer new knowledge about a given design model. According to Fischer (1993, pp.179-192), such an approach for constructability reasoning in automated systems is divided into three types (Table 4-1 provides a comparison between these three types of reasoning) :

- Reasoning about attributes of objects
- Reasoning about relationships between attributes of objects
- Spatial reasoning

Table 4-1 Comparison between types of Constructability Reasoning in an automated system

Types of constructability reasoning	Definition	Example
Reasoning about attributes of objects	Simple comparison between a value of an object attribute to a value from the database	The length of any component should not exceed the value of x
Reasoning about relationships between attributes of objects	Reasoning about the relationship of multiple attributes before comparing a project value with a value from the database	The building use is residential and the span between columns is "a" meters, then the thickness of the slab should be between x & y
Spatial reasoning	Reasoning about a big number of objects, usually in terms of location, orientation, spacing and clearance	Column spacing should be equally distributed to maximize modularization possibilities

Through these types of reasoning and using a rule-based approach, as previously discussed in chapter 3, it is possible to formalize and represent the precast knowledge base in a network of decision tables in a clear, auditable, extensible system. Every rule needs to follow a unique path, providing unique results, without redundancy or contradiction with other rules. To achieve this sort of consistency, it is impossible to simply represent the rules in the form of individual monolithic tables that encompass each rule from start to end. A structure has to be established to organize the rules and manage the interconnections between various rule-paths. Building on the two-layered knowledge base described in the previous section, along with the typology explained in section 3.3.3, a knowledge formalization approach is adopted using a three-tiered structure as follows:

a. Referencing tables (REF)

These tables fulfill one or both of two possible functions:

- They point the system to specific formulas for further calculations or an action table to perform the final action.
- Based on the model and manual inputs, they set specific variables that will be used in further rules.

b. Calculation tables (CLC)

These tables get values from referencing tables, derive a value based on a formula, then feed the derived value into another referencing table to determine the value of further variables or an action table to provide the final outcome.

c. Action tables (ACT)

These tables are responsible for determining issues that should be reported to the user based on model/manual inputs and/or variables set from referencing tables and/or calculation tables. The outcome of the rule from action tables could be one of three possible results:

I. Severe warning (S)

This type of output relates to building elements, or design considerations, that need immediate attention from the designer because they violate a fundamental production, transport, or structural requirement.

II. Recommendation (R)

This type of output relates to building elements, or design considerations, that could be changed to improve cost/time/production efficiency. They can, however, be ignored if there is a sound justification for such design decision.

III. Element meets Criteria (O)

This type of output relates to building elements, or design considerations, that meet all the precast design criteria that are designated in the system.

The following figure (Figure 4-5) presents an overview for an exemplary rule-path from input to output utilizing the three types of tables, highlighting how rules may branch into several tables to cover all the possible conditions in a building model.

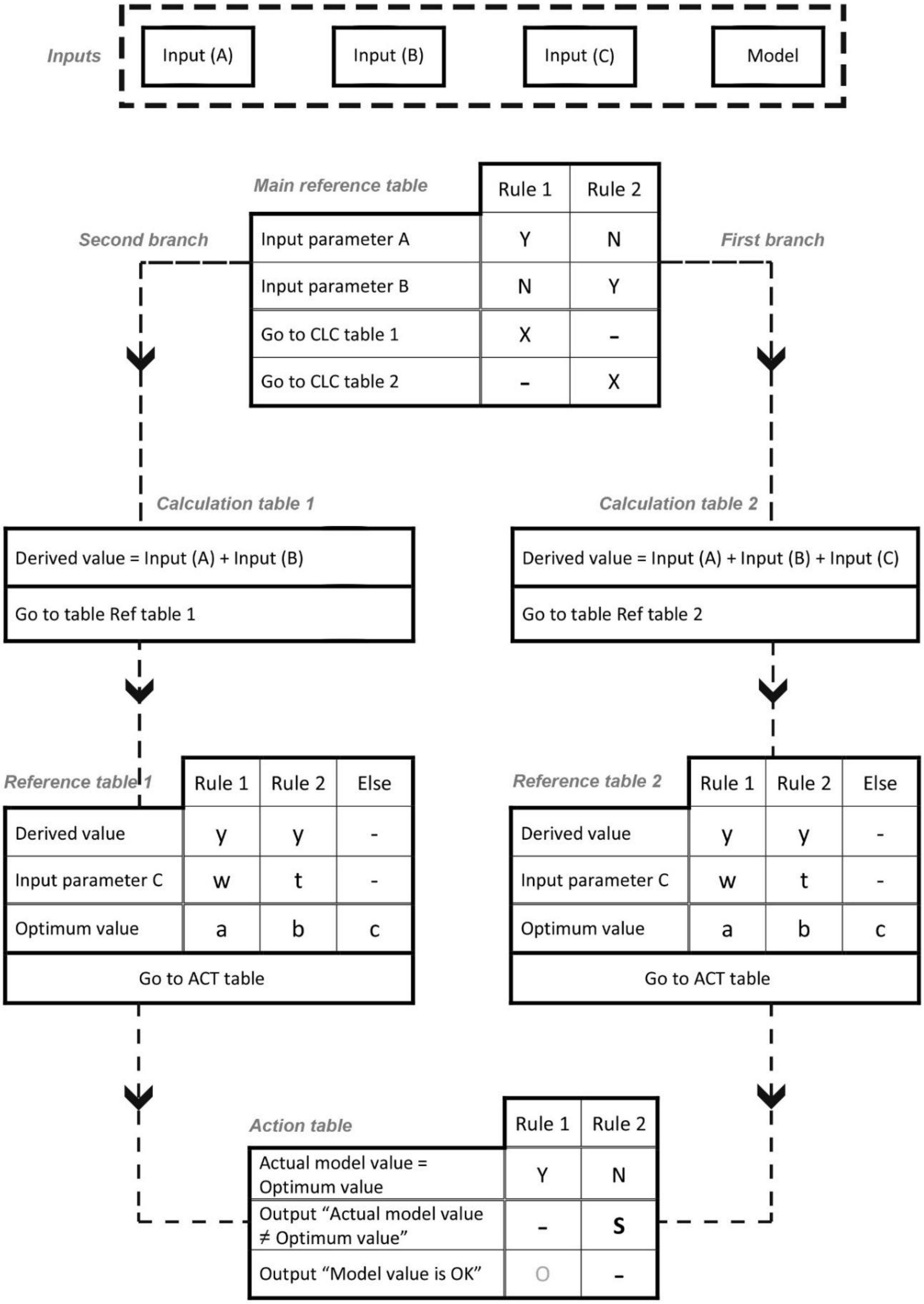

Figure 4-5 Schematic representation of a single rule represented as a network of decision tables in the rule database

It is important to point out that the established rule-set is not meant to provide an exhaustive directory of every corner of precast expertise available. A careful selection process was implemented to make sure that any given rule satisfies the following criteria:

- The rule should be suitable for implementation using the information typically available in BIM with LOD 200-300 only.
- The rule should be relevant to the activities and considerations that are typically associated with architectural design. This excludes, for example, rules related to steel reinforcement or panel steel embeds.
- The rule can be transformed into a tangible, computer-readable/interpretable piece of logic. That includes clear numerical, dimensional, and geometrical rules and excludes considerations based on intuition and/or qualitative assessment.
- The rule should be generic to enable industry-wide implementation (at least within the German market). That excludes any manufacturer-specific preferences or limitations.

4.4. Rule-sets (examples)

It is important to reiterate that this research is concerned only with precast construction considerations that are relevant in the design phase of the building. Through an extensive literature review, Nepal (2011, pp.31-32) summarized the design conditions that impact construction in general. These conditions generally fall under one of the following categories:

- Component dimensions (Length, width, height, thickness)
- Maximum/ Minimum dimensions/ spacing
- Modular layout
- Adjacency of building components
- Repetition of same/similar sized components
- Variation in the size and/or location of the components
- Shape of the components
- Clear height and elevation of the bottom of the slabs

This served as a guide to identify the knowledge and rules relevant to the purposes of this research among the vast repository of precast knowledge. The following subsections present examples from the five categories present in the upper layer of the database (namely general planning considerations, slabs, beams, wall panels, columns). The complete knowledge base is provided in appendices A to E.

4.4.1. General planning considerations

Under this category, two examples will be presented. The first example is a simple case that relates to the general concept of “Similarity and grouping”. The cost of precast concrete moulds represents a big part of the fabrication cost (Kolarevic and Klinger, 2008, p.50). The idea is to rationalize building geometry, using precast components that are similar or identical, thus minimizing the number of moulds needed in the fabrication process, and making the process of fabrication far more cost-effective (PCI Industry Handbook Committee, 2010, p.3D2). Therefore, the primary task here is to establish an economical layout of components to optimize the number of similar items. The optimum layout is usually a rectangular grid where the distances between columns and beams are similar as far as possible to be able to subdivide the building into equal bay widths (Elliott and Jolly, 2013, p.74). Ideally, to maximize the repetition of components, floor plans should be based on a 1.2m modular grid (Fédération Internationale de la Précontrainte "FIP", 1994, p.81). This also translates to floor heights, where equal floor heights simplifies the production of precast components.

The formulation of such concepts can be captured in a single decision table (Action table), which is represented in Table 4-2. The system should be able to interpret the column spacing and floor heights, compare them, and determine whether they are equal or unequal. Feedback in the case of violation of the rule is on the “Recommendation” level because variations are always expected and they do not violate a core principle of precast production/construction.

Table 4-2 Decision table representation of the "Similarity and grouping" rule

Action table (ACT)	Rule 1	Rule 2	Rule 3	Rule 4
Are the typical floor heights equal?	Y	N	Y	N
Is the column spacing equal based on a 1.2m grid?	N	Y	Y	N
Display “Column spacing and floor heights check OK”	-	-	O	-
Display “Please consider creating evenly distributed columns based on a 1.2m”	R	-	-	R
Display “Please consider creating equal floor heights”	-	R	-	R

The second example relates to the restrictions on the size of individual precast components due to transport limitations from the production facility to the construction site. Since transport by rail is quite rare, the focus here is on transport by tractor units with trailers. The road transport dimensions are regulated by the German Road Traffic Act (StVO, Straßenverkehrs-Ordnung). The outcome is a limit of 2.55m on transported object's width, 15.5m for length, and 4m for height in absence of special permits and police escorts. Special permits can be issued by local authorities and extend the limit to 3m for width, 24m for length (Bachmann and Steinle, 2011, p.21). Such permits are essential for when the dimensions of individual precast components exceed the allowed values and must be taken into consideration in an early stage to avoid project delays because of route restrictions (Dawar et al., 2014, pp.1771-1781). The formulation of such concepts can be captured in the reference and action tables, which are represented in Table 4-3 a-c.

Table 4-3 a-c: Decision table representation of the "Transportation limitations" rule

a) Reference table (REF)	Rule 1	Rule 2
Transport without special license	Y	N
Transport with special license	N	Y
Go to table ACT.1	X	-
Go to table ACT.2	-	X

b) Action table (ACT.1)	Rule 1	Else
Length of individual component (m)	≤ 15.50	-
Width of individual component (m)	≤ 2.55	-
Height of individual component (m)	≤ 4.00	-
Display "Element sizing for transport is OK"	O	-
Display "Warning! Maximum dimensions are 15.50*2.55*4.00 m"	-	**S**

c) Action table (ACT.2)	Rule 1	Else
Length of individual component (m)	≤ 24.00	-
Width of individual component (m)	≤ 3.00	-
Height of individual component (m)	≤ 4.00	-
Display “Element sizing for transport is OK”	O	-
Display “Warning! Maximum dimensions are 24.00*3.00*4.00 m”	-	**S**

4.4.2. Slabs

Under this category, one example will be presented that relates to the slab thickness. Determining the exact slab thickness in early design stages and without consulting human expert is unrealistic. The aim, however, is to arrive at an approximate informed estimate of the slab thickness based on the information available in this design stage. Determining an approximate slab thickness based on precast standards is essential to determine the approximate floor clear height and space available for dropped ceilings before proceeding in design detailing. It is also essential to determine the type of precast slab used based on building use and the required spans (s). Slabs in precast concrete construction in the German market can be broadly divided into three types (Hierlein et al., 2009, pp.23-25) (Figure 4-6):

- Element slabs: Can cover spans up to 7m. A semi-finished precast slab with the necessary reinforcement, the slab is supplemented on-site with in-situ concrete.
- Hollow core slabs: Can cover spans up to 12m. A precast slab of pre-stressed concrete with tubular voids to reduce its own weight.
- TT slabs: Can cover spans up to 25m. It resembles the shape of two (T) letters connected with each other. The structure can withstand higher loads and longer spans.

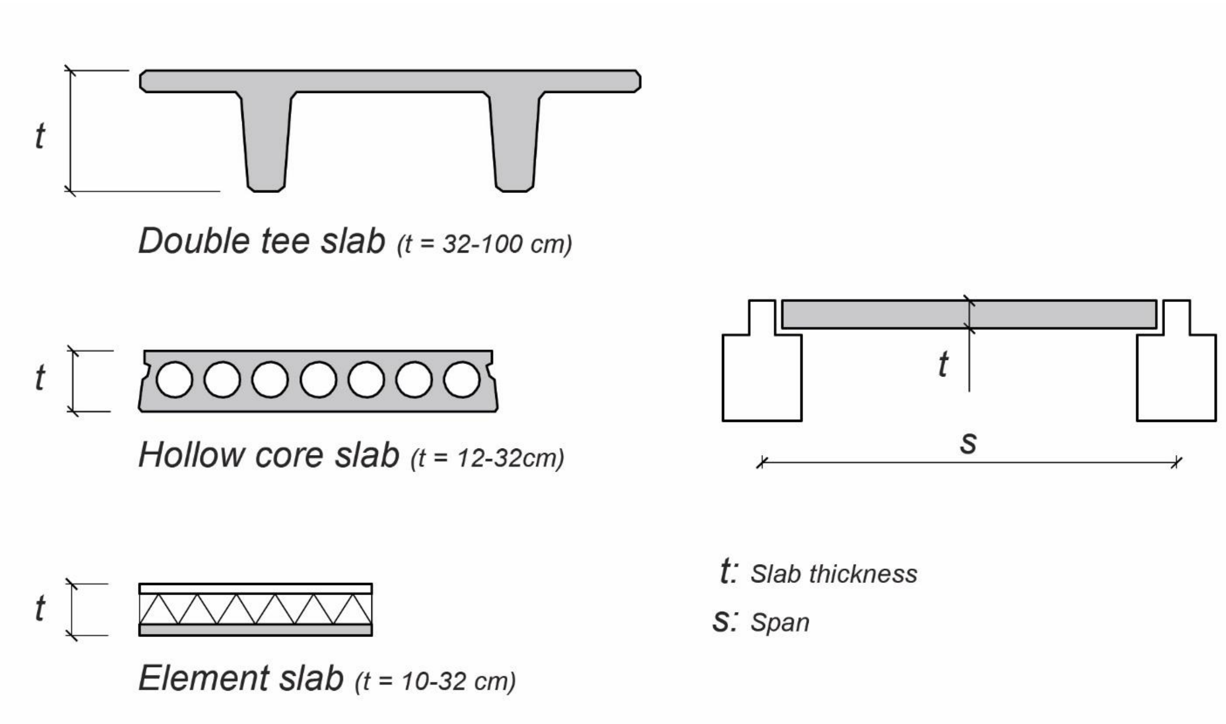

Figure 4-6 Cross-sections in the three main types of precast slabs and the range of their thickness (t). The value of (t) depends on the span (s) and design loads on the slab.

Design loads acting on the slabs are usually divided into dead loads (g_k) and live loads (q_k).

- Dead loads (g_k) are the permanent loads including own-weight, floor finishes, and partitions. In the context of this research, where the exact floor finishes are not yet determined and the precise location of wall partitions is subject to change, assumptions have to be made to arrive at approximate values for element sizing. The load of partitions can be assumed to be 2.5 kN/m2 spread across the entire floor area (O'Brien, 1995, p.71). The load of floor cover can be assumed to be around 1.5 kN/m2 in case of typical floor slabs, and 0.5 kN/m2 in the case of light floor cover on the roof (Hierlein et al., 2009, p.68).
- Live loads (q_k) comprise all occupancy loads and snow loads on roof slabs (Chanakya and Arya, 2009, pp.9-10). Occupancy loads vary according to building type and can be obtained from the DIN EN 1991-1-1:2010-12 document[11] (DIN, 2010, pp.6-7). Snow loads vary according to location and can be obtained from the data of the German institute for building technology (DIBt, 2019).

The German association for precast concrete construction (FDB) offers extensive guidelines, providing estimations on the recommended slab thickness based on the slab's type, loads on

[11] Eurocode 1: Actions on structures - Part 1-1: General actions - Densities, self-weight, imposed loads for buildings

the slab, and the span of the slab[12]. In the context of this research, the inputs and outputs will be as follows

- **Input 1** (user input): Slab type
- **Input 2** (user input): Location (to determine snow load)
- **Input 3** (user input) Building type (to determine occupancy load)
- **Input 4** (assumed): Floor cover load, partitions load
- **Input 5** (model input): Model elements to obtain span and actual slab thickness
- **Output:** Recommended slab thickness

The calculations on which the FDB bases its recommendations take into consideration the own-weight of the slabs so it does not need to be calculated separately in the rule-set. The formulation of these recommendations can be captured in the decision tables provided in Table 4-4 a-k. Please note that the entire network of reference tables is not provided for the sake of simplification. The complete rule-set can be found in Appendix B. Due to the complexity of this rule, a schematic overview of this rule-set is provided in Figure 4-7.

Table 4-4 a-k: Decision table representation of the "Slab thickness" rule

a) REF	Rule 1	Rule 2
Roof slab	Y	N
Typical slab	N	Y
Go to table REF1	X	-
Go to table REF3	-	X

b) REF.1	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
City	Berlin	Hamburg	Munich	Cologne	Frankfurt
Set snow load to (KN/m2)	1.95	1.95	2.47	0.49	0.6
Go to table REF.2					

[12] It is important to note that the information and values provided by the FDB in this section and the following sections were specifically developed and intended for the earlier design stages before a detailed structural analysis model is possible. Values of element sizing can be fetched based on basic inputs, like dead and live loads, while reasonable assumptions about wind seismic loads are implicitly taken into consideration.

c) REF.2	Rule 1	Rule 2
Light floor cover (unused roof)	Y	N
Normal floor cover (Used roof)	N	Y
Set Floor cover load to 0.5 KN/m2	X	-
Set Floor cover load to 2.0 KN/m2	-	X
Go to table CLC.1		

d) CLC.1	
Roof slab load (KN/m2)= Floor cover load + Snow load	
For TT slabs	Go to table REF.4
For Hollow core slabs	Go to table REF.5
For Element slabs	Go to table REF.6

e) REF.3	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Building use	Residential Hotel	office	School Kindgart.	Exhibit space	Shopping
Set live load to (KN/m2)	1.5	2.0	3.0	5.0	5.0
Go to table CLC.2					

f) CLC.2	
Typical slab load (KN/m2)= Floor cover load (1.5 kN/M2) + Partitions load (2.5 kN/M2) + Live load	
For TT slabs	Go to table REF.7
For Hollow core slabs	Go to table REF.5
For Element slabs	Go to table REF.6

This table is just a sample. The full table is available in the Appendix B

g) REF.4	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Span (m)	≤ 7.5	7.5 < x ≤ 10.0	7.5 < x ≤ 10.0	10.0 < x ≤ 12.5	12.5 < x ≤ 15.0	12.5 < x ≤ 15.0	15.0 < x ≤ 17.5
Load (KN/m2)	-	1.0 ≤ y ≤ 3.5	3.5 < y ≤ 5.0	-	1.0 ≤ y ≤ 3.0	3.0 < y ≤ 5.0	1.0 ≤ y ≤ 2.5
Set Slab thick. (t) to (mm)	260	360	460	460	460	560	560
Display "Reduce the span to 25m"	-	-	-	-	-	-	-
Continued (In the end, go to table ACT)							

This table is just a sample. The full table is available in the Appendix B

h) REF.5	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Span (m)	x ≤ 3.0	x ≤ 3.0	x ≤ 3.0	3.0 < x ≤ 4.0	3.0 < x ≤ 4.0	3.0 < x ≤ 4.0	4.0 < x ≤ 5.0
Load (KN/m2)	1.0 ≤ y ≤ 5.0	5.0 < y ≤ 7.5	7.5 < y ≤ 10.0	1.0 ≤ y ≤ 4.5	4.5 < y ≤ 7.5	7.5 < y ≤ 10.0	1.0 ≤ y ≤ 3.0
Set Slab thick. (t) to (mm)	120	140	160	120	140	160	120
Display "Reduce span to 12.5m"	-	-	-	-	-	-	-
Continued (In the end, go to table ACT)							

This table is just a sample. The full table is available in the Appendix B

i) REF.6	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Span (m)	x ≤ 3.0	x ≤ 3.0	x ≤ 3.0	3.0 < x ≤ 4.0	3.0 < x ≤ 4.0	3.0 < x ≤ 4.0	4.0 < x ≤ 5.0
Load (KN/m2)	1.0 ≤ y ≤ 7.5	7.5 < y ≤ 15.0	15.0 < y ≤ 25.0	1.0 ≤ y ≤ 7.5	7.5 < y ≤ 15.0	15.0 < y ≤ 25.0	1.0 ≤ y ≤ 7.5
Set Slab thick. (t) to (mm)	120	140	160	140	160	180	180
Display "Reduce span to 7.5m"	-	-	-	-	-	-	-
Continued (In the end, go to table ACT)							

This table is just a sample. The full table is available in the Appendix B

j) REF.7	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Span (m)	x ≤ 6.0	x ≤ 6.0	x ≤ 6.0	6.0 < x ≤ 7.5	6.0 < x ≤ 7.5	6.0 < x ≤ 7.5	7.5 < x ≤ 10.0
Load (KN/m2)	y ≤ 5.0	5.0 < y ≤ 15.0	15.0 < y ≤ 25.0	y ≤ 5.0	5.0 < y ≤ 15.0	15.0 < y ≤ 25.0	y ≤ 3.5
Set Slab thick. (t) to (mm)	320	350	400	420	450	500	420
Display "Reduce span to 20.0m"	-	-	-	-	-	-	-
Continued (In the end, go to table ACT)							

k) ACT	Rule 1	Rule 2
Actual slab thickness = t	Y	N
Display "Warning! Slab thickness should be equal (t)"	-	S
Display "Slab thickness is OK"	O	-

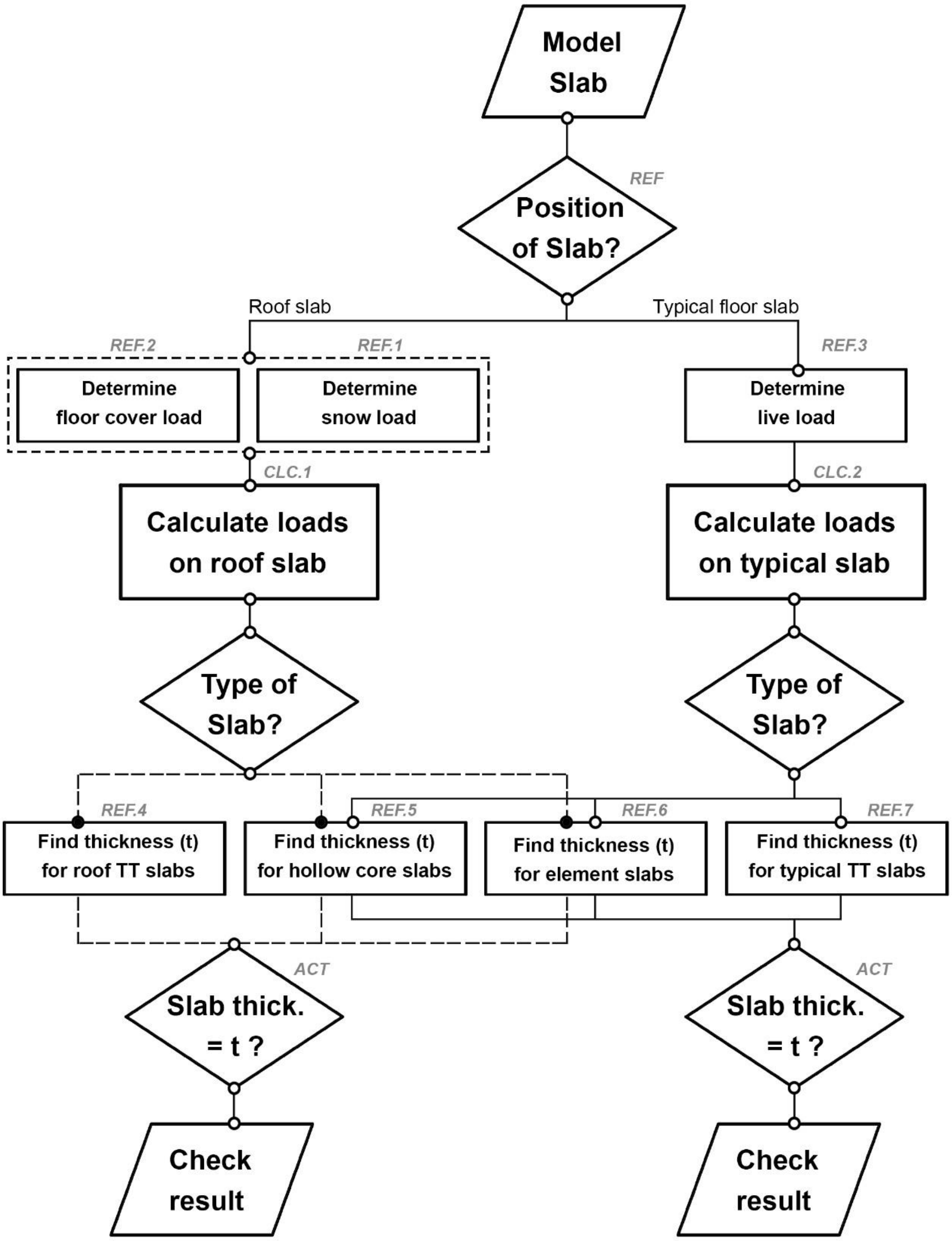

Figure 4-7 A schematic overview for the decision table structure for the rule "Slab thickness"

4.4.3. Beams

Under this category, one example will be presented that relates to the beam depth. Determining the exact depth in early design stages without consulting human experts is also unrealistic. The aim, however, is to arrive at an approximate informed estimate of the beam depth based on the information available in this design stage. Determining an approximate beam depth based on precast standards is essential to determine clear heights under the beams and limitations of window and door openings inside the building and in the building's shell. Hierlein et al. (2009, pp.74-75) provides an estimate of the beam depth based on the beam span, span of the slab carried by the beam (both can be obtained from the BIM model), and assumptions for the dead load together with live load on the building. Beam span is always assumed in a direction parallel with the greater dimension of the building (Elliott and Jolly, 2013, p.34). Own weight of the beam and the slab are already taken into consideration in the recommendations. In the context of this research, the inputs and outputs will be as follows:

- **Input 1** (user input): Location (to determine snow load)
- **Input 2** (user input) Building type (to determine occupancy load)
- **Input 3** (assumed): Floor cover load, partitions load
- **Input 4** (model input): Model elements to obtain slab span
- **Input 5** (model input): Model elements to obtain beam span, actual depth
- **Output:** Recommended beam depth

The formulation of these recommendations can be captured in the decisions tables provided in Table 4-5 a-c. Please note that the entire network of reference tables is not provided for the sake of simplification. The complete rule-set can be found in Appendix C

Table 4-5 a-c: Decision table representation of the "Beam depth" rule

a) CLC
Load (KN/m2)= Floor cover load + live load (Loads can be fetched from table 4-4f)
Go to table REF

This table is just a sample. The full table is available in the Appendix C

b) REF	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Span of beam l (m)	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0
Span of slab L (m)	y ≤ 6.0	y ≤ 6.0	y ≤ 6.0	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5
Load (KN/m2)	z ≤ 5.0	5.0 < z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 3.0	3.0 < z ≤ 15.0	15.0 < z ≤ 20.0	20.0 < z ≤ 25.0
Set beam height (h) to (mm)	400	500	600	400	500	600	700
Display "Consider reducing the span to 10.0m"	-	-	-	-	-	-	-
Continued (In the end, go to table ACT)							

c) ACT	Rule 1	Rule 2
Actual beam depth = h?	Y	N
Display "Warning! Beam depth should be equal (h)"	-	S
Display "Beam depth is OK"	O	-

4.4.4. Wall panels

Rules in this category are applied based on the assumption that external wall panels are made of sandwich precast concrete panels with a "Punched openings" façade type, effectively halting the checks if the user chooses other types of facades (e.g. ribbon facades, glass façades). Under this category, one example will be presented that relates to the vertical joints in external façade wall panels. The designer must take into consideration that the length of individual wall panels should not exceed a certain value, and that value determines the width of the joint between individual panels. This can have a huge effect on the façade design and should be taken care of in the early design stages.

These considerations are regulated in Germany by the standard DIN 18540 (Sealing of exterior wall joints in building using joint sealants), and states that the width of the joints should not be less than 15mm and the length of individual panels should not exceed 7.0-8.0m (FDB, 2016, p.3). The connection detail between panels is provided in Figure 4-8. The detailed requirements of the standard are summarized in Table 4-6 a-b.

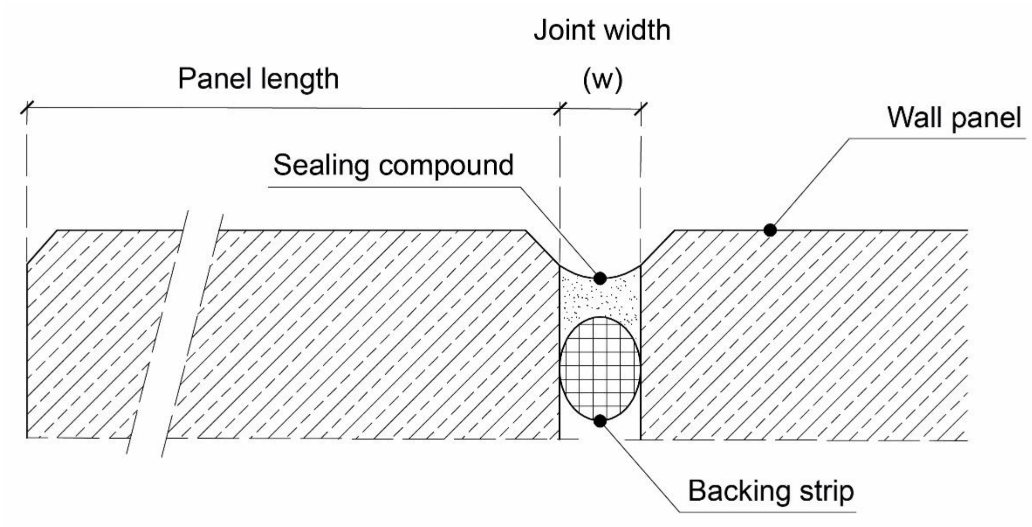

Figure 4-8 Detail of a typical external wall joint. Adapted from (Bachmann et al., 2009, p.232)

Table 4-6 a-b Decision table representation of the "Facade joint width" rule

a) REF	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Else
Spacing between joints/panel length (m)	≤ 2.0	2.0 < x ≤ 3.5	3.5 < x ≤ 5.0	5.0 < x ≤ 6.5	6.5 < x ≤ 8.0	-
Set joint width [w] to (mm)	15	20	25	30	35	-
Display "Please reduce joint spacing to 8.0 m or less"	-	-	-	-	-	**S**
Go to table ACT						

b) ACT	Rule 1	Rule 2
Actual joint width ≥ w	Y	N
Actual joint width < w	N	Y
Display "Warning! Minimum joint width is [w]"	-	**S**
Display "Joint width is OK"	O	-

4.4.5. Columns

Columns in precast production and construction should have a uniform rectangular cross-section to facilitate production, transport, and installation. These are very simple checks to undertake in a BIM model and will not be explained here, decision tables for these checks are available, however, in appendix E. The example that will be shown here relates to estimating the dimensions of the columns' cross-section in a floor plan. According to Brandt et al. (1993, p.42), an estimation of the edge and middle columns can be given knowing the effective area of the column (Beam span * Slab span), and slab loads (Roof slab load + Typical slab load * no of typical slabs). This data can be extracted from the model and using informed assumptions as previously explained in previous sections. In the context of this research, the inputs and outputs will be as follows

- **Input 1** (calculated): ∑ Roof slab loads
- **Input 2** (calculated): ∑ Typical slab loads
- **Input 3** (model input): Model elements to obtain slab span
- **Input 4** (model input): Model elements to obtain beam span
- **Output:** Recommended column cross-section

The formulation of these recommendations can be captured in the decision tables provided in Table 4-7 a-e. Please note that the entire network of reference tables is not provided for the sake of simplification. The complete rule-set can be found in Appendix E.

Table 4-7 Decision table representation of the "Column cross-section" rule

a) CLC.1
Effective area (m2)= Span of beam l (m) * Span of slab L (m)
Go to table CLC.2

b) CLC.2	
Loads (KN/m2) = ∑ Roof slab loads (cover + snow) + ((∑ Typical slab loads (live + cover))*no of floors)	
For edge columns	Go to table REF.1
For middle columns	Go to table REF.2

This table is just a sample. The full table is available in the Appendix E

c) REF.1	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Effective area (m2)	x ≤ 100.0	100.0 < x ≤ 125.0	100.0 < x ≤ 125.0	125.0 < x ≤ 150.0	125.0 < x ≤ 150.0	150.0 < x ≤ 175.0	150.0 < x ≤ 175.0
Load (KN/m2)	-	y ≤ 20.0	20.0 < y ≤ 30.0	y ≤ 15.0	15.0 < y ≤ 30.0	y ≤ 10.0	10.0 < y ≤ 20.0
Set column cross section (b*h) to (mm)	300*400	300*400	400*400	300*400	400*400	300*400	400*400
Display “Consider reducing the span”	-	-	-	-	-	-	-
Continued (In the end, go to table ACT)							

This table is just a sample. The full table is available in the Appendix E

d) REF.2	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Effective area (m2)	x ≤ 75.0	75.0 < x ≤ 100.0	75.0 < x ≤ 100.0	75.0 < x ≤ 100.0	100.0 < x ≤ 125.0	100.0 < x ≤ 125.0	100.0 < x ≤ 125.0
Load (KN/m2)	-	y ≤ 15.0	15.0 < y ≤ 25.0	25.0 < y ≤ 30.0	y ≤ 7.5	7.5.0 < y ≤ 15.0	15.0 < y ≤ 25.0
Set column cross section (b*h) to (mm)	300*400	300*400	400*400	400*500	300*400	400*400	400*500
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued (In the end, go to table ACT)							

e) ACT	Rule 1	Rule 2
Actual column cross section = b*h	Y	N
Display "Warning! Column cross section should be equal (b*h)"	-	**S**
Display "Column cross section is OK"	O	-

4.4.6. Conclusion

The basic premise of the proposed system depends on filtering and sorting model input, as well as manual user input, then executing the rules stored in the database, providing feedback as an output to the user. To establish the rule-base needed for the system, precast knowledge was gathered from many resources, consolidated, filtered for relevant expertise, and categorized in a layered structure to facilitate rule auditing and execution. Finally, knowledge was formalized in a tabular form (Decision tables), which can be easily and logically organized and checked for completeness, uniqueness, and consistency. This form also acts as an intermediate logical format of rules that can be easily implemented in an actual computer system. The chapter finally presented examples of these rule-sets for each of the categories specified in the rule-set structure. The process of the actual implementation of this rule-set in a computer system, including managing system input and output, will be the subject of the next chapter.

Chapter 5: Prototypical implementation of the proposed system

5.1. Introduction

After establishing the system framework and its components, this chapter presents an exemplary practical application inside a BIM platform. The idea is to showcase the capabilities of the proposed approach in real-life scenarios, not to present a polished, commercial software package. This chapter begins by describing the components of the software system and how they operate. It then describes the criteria used for the selection of the software platform, in which the prototype tool is developed. The chapter then walks the reader through the inner workings of the tool and how it receives, manages, and stores information. It also explains how the rule-sets are stored, managed, and implemented in order to detect issues. Finally, the chapter presents a “Test run” of the prototype tool to introduce the Graphical User Interface (GUI) of the tool and to demonstrate how the user can interact with the tool and view the results.

5.2. System overview

5.2.1. BIM model requirements

Previously, in chapter 3, it was determined that design schemes with correspondent BIM models in LOD 200 and LOD 300 will be suitable for the purposes of this research. In particular, the following is expected to be present in the BIM models in order for the tool to function properly:

- Columns and column grid
- Levels
- Walls
- Wall openings
- Wall joints (Reveals)
- Windows
- Doors
- Slabs
- Beams
- Staircase(s)

For each of these elements, the sizing, orientation, shape, and location should correspond to the requirements of LOD 200 or LOD 300 to provide the checking system with the necessary information to carry out the needed precast checks. Additionally, the building design should be based on a rectangular grid in order for the system to be able to make appropriate assumptions. Finally, standard fenestrate facades are the only type of facades that could be taken in consideration in the checks, other types (like ribbon facades) will not be taken into consideration in the precast checks.

5.2.2. System components

For the system to run successfully, it operates the checking process through its internal structure, which can be viewed in Figure 5-1. The practical implementation of this structure inside of a software solution is detailed later in section 5.3. The components of that structure can be described as follows:

a. Input

The inputs to the system consist of two categories:

- BIM model: The model requirements were previously discussed in section 5.2.1
- Manual user inputs: These are inputs needed to perform the necessary calculations of the checking process, but they cannot be directly obtained from the BIM model. These include:
  - Building type: To determine live load estimation on the building
  - City: To determine snow load estimation on the roof
  - Transport license: To determine max. dimensions of precast elements
  - Slab type: To determine optimal slab thickness value for each type
  - Roof slab type: To determine floor cover load estimation on the roof
  - Façade type: To determine whether to run façade checks or not

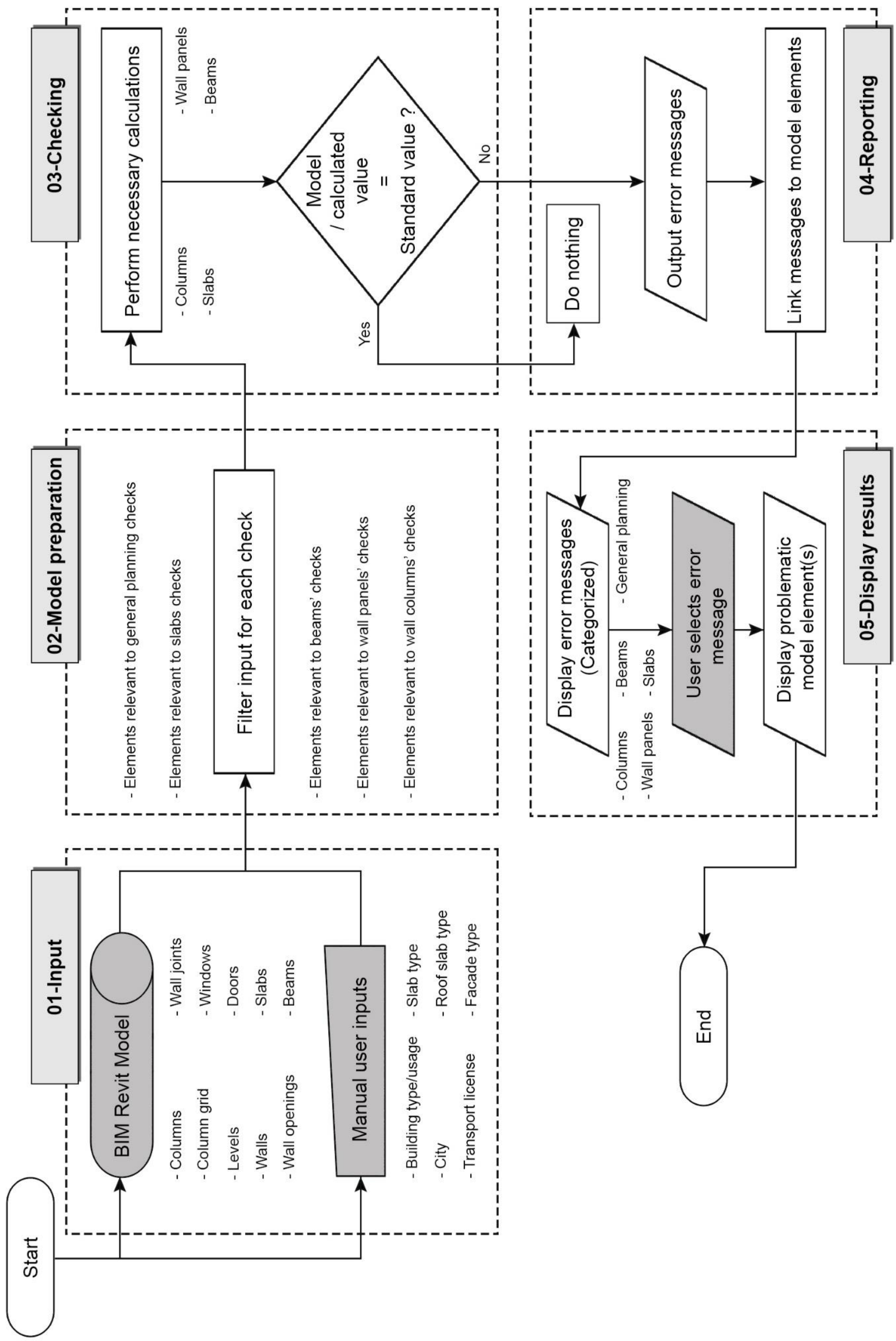

Figure 5-1 The components and the interior structure of the implementation of the proposed system

b. Model preparation

For each individual check, specific model elements need to be filtered out to obtain relevant information from them and/or to eventually report on problematic elements.

- Elements relevant to general planning checks:
  - Grids
  - Levels
- Elements relevant to slabs' checks:
  - Roofs
  - Slabs
  - Beams
- Elements relevant to beams' checks:
  - Columns
  - Beams
  - Slabs
  - Roofs
- Elements relevant to wall panels' checks:
  - Walls
  - Wall reveals
  - Windows
- Elements relevant to columns' checks:
  - Structural columns
  - Walls
  - Slabs
  - Roofs
  - Beams

c. Checking

This is the main component of the system, where the stored rules are implemented, calculations are carried out when necessary, and checks are performed in order to detect issues with the design. The actual checks are carried out in the same five categories described in section 4.3.2

d. Reporting

After performing the checks, the system may detect issues with the design. These issues have to be presented to the architect in an orderly manner, to facilitate navigation through these issues, and to determine which ones require immediate attention. In order to do this, the results are presented in a categorized manner under the same checking categories, namely "General planning checks", "Slabs", "Beams", "Wall panels", "Columns". Under each category, each issue is numbered, described in full, and presented with the appropriate solution. Each issue is also labeled either as "Critical", meaning it needs immediate attention because of an obvious violation of precast building considerations, or "Mild", meaning that it represents a mere recommendation to save time/cost, but can be ignored when justification is available.

e. Display results

Having reported the results in textual form to the user, the next step is to display the affected parts in the BIM model for the user to pinpoint the problematic elements. When the user picks a certain issue to take a closer look at it, the system can zoom and isolate the affected element in order to give the user the possibility to begin dealing with it. The user is also able to return at any time to the original results' report, browse it, and pick another element to preview.

5.2.3. Platform selection

After establishing the precast rule-database and system components, it is time to integrate these into an actual software system to realize a prototype of the proposed system. The software of choice here is Autodesk Revit and its visual/textual programming platform Dynamo. The reason behind this choice can be better explained through a detailed layout of the system requirements that the software platform should fulfill. These requirements can be detailed as follows:

a. The software should be able to manage BIM content

This is the most basic requirement, which can be found in almost every BIM related software. The software should be able to store, manage, and filter geometry and related information attached to it. It should also be able to receive user input and use this input to perform actions in the model.

b. The software should seamlessly integrate into the design process

The design process is an iterative process with the design evolving with each iteration. In this environment, continuous design changes and alterations are the norm and with most changes comes the need for a potential re-check of design requirements, including precast design requirements. For this reason, it makes sense to build the checking tools right into the design platform itself, to provide a seamless experience. This is not the case in external BIM checking platforms where users would have to shuttle back and forth between different, sometimes incompatible, applications. A popular example is Solibri Model Checker (SMC), which is a java-based platform that is designed to perform rule-based checks, accepting IFC models as input (Jiang and Leicht, 2015, p.A4014004). In addition to the problem of having to continuously re-export models after design changes and re-import the model to SMC to perform checks, there are also problems with data loss and misinterpretation in IFC imports and exports (Borrmann et al., 2018, p.13), where the researcher has also found out, for example, that wall joints are not correctly recognized in SMC.

c. The software capabilities should be extensible and customizable

The software should be flexible enough to provide the possibility to build on its out-of-the-box capabilities in order to be able to do the following:

- To build a custom, easy to use user interface for the user to be able to navigate the system, provide input, and implement the needed checks.
- To store, manage, and implement the rule-base and filter relevant BIM content to apply these rules to it.
- To visualize and report the results of the check.

These capabilities are available via Autodesk Revit's visual/textual programming platform Dynamo. It is a tool that gives programmers the ability to extend Revit's functionality by making them able to define custom logic and behavior, making use of external code libraries as well as Revit's own Application Programming Interface (API), to drive sequences of actions that constitute a custom algorithm (Autodesk, 2019). The process of developing these capabilities will be the subject of discussion of the following section.

5.3. System prototype implementation

Having determined the appropriate platform to implement the system, this section and the next one will describe the implementation of a preliminary prototype of the described system. The prototype is implemented entirely in Revit, Dynamo, and Python. It provides a GUI for the user to manage the whole process, without having to deal with programming and visual programming concepts. The prototype is a direct implementation of the theoretical system framework described in section 4.2 and the associated rule database described in section 4.3. Following the process demonstrated in Figure 5-1, the prototype's internal core can be detailed as follows:

5.3.1. Input capturing and filtering/Model preparation

The inputs to the tool are divided into two categories: manual user input and model input. For receiving manual user inputs, a series of custom Dynamo Python scripts were developed to provide a GUI, where the user can provide these inputs. Dynamo's custom python nodes operate on IronPython, which is an implementation of Python with .NET Framework integration (IronPython, 2019). Such integration enabled the usage of the "Windows Forms" graphical class library (part of the .NET Framework) to design a complete user interface with features such as dropdown lists, radio buttons, and input boxes to capture manual user input. This input is stored in local data structures inside Dynamo, ready for filtering and classification.

In parallel, relevant elements of a BIM model are collected using a subset of Dynamo nodes called "Selection nodes" or "Collector nodes". These elements are then channeled to the respective section of the Dynamo graph, along with the relevant manual user input, fulfilling the input requirements for each individual check as highlighted in section 4.4. Relevant input is extracted directly from model elements' properties and geometries. Others, such as wall panels and floor slabs, need to be further deconstructed into their basic layers for calculation purposes (Figure 5-2), which was achieved with the help of Dynamo's custom package Archilab (Sobon, 2019), and the clockwork package (Dieckmann, 2019). Generally speaking, this entire section is the practical implementation of the first type of "Referencing tables" system described in section 4.3.3 in this research. The obtained input is either used in intermediate formulas or is used directly in designated rules.

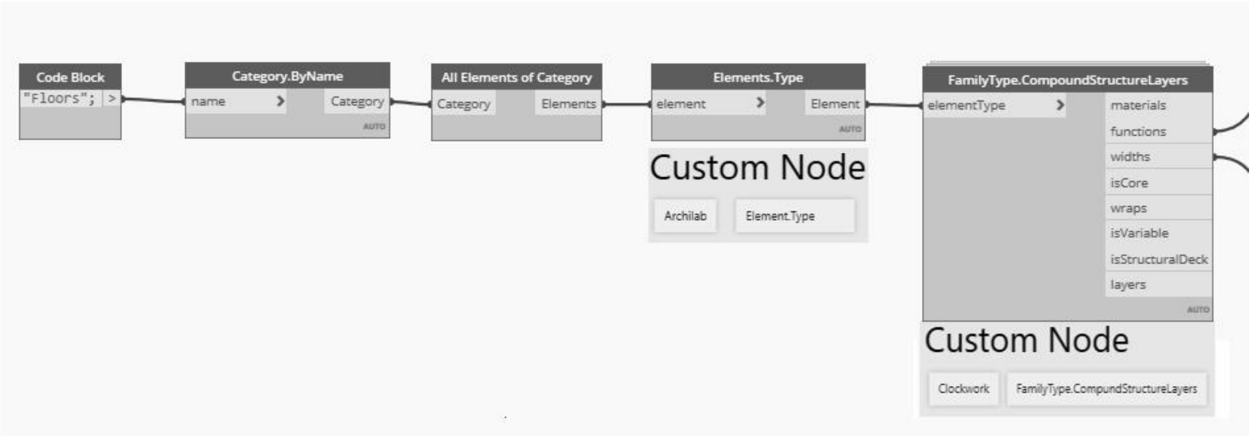

Figure 5-2 A snippet from the Dynamo graph showing the procedure of collecting model elements, in this case slabs, and extracting the widths of their different layers.

5.3.2. Calculations

After receiving input, the system needs to translate that input into concrete numerical values. These values are stored either to be implemented directly in rule-sets, or to be used in further calculations. For example, the system asks the user to determine the building use in order for it to be able to make assumptions regarding the live loads on the building slabs using the DIN EN 1991-1-1:2010-12: “General actions - Densities, self-weight, imposed loads for buildings” (DIN, 2010). In the same example, an intermediate calculation is carried out to determine the actual loads on a slab by adding the assumed live load corresponding to the chosen building use with an assumed floor cover load of 1.5kN/m (Figure 5-3). The calculated load is used further in the rule-sets to determine the optimal slab thickness according to standards for precast considerations. Other intermediary calculations are carried out to determine, for example, the actual loads on roof slabs, beams, and columns, among others. Generally speaking, this entire section is the practical implementation of the “Calculation tables” system described in section 4.3.3 in this research.

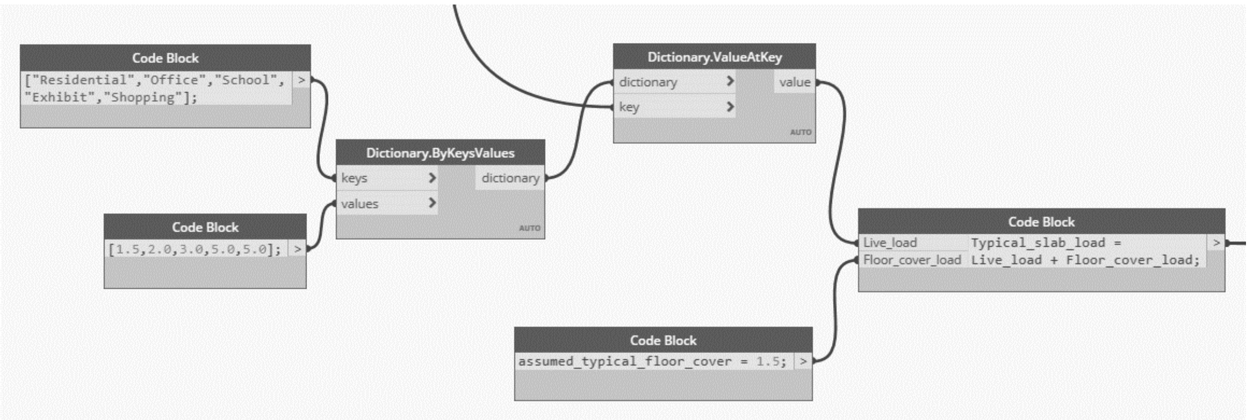

Figure 5-3 A snippet from the Dynamo graph showing the intermediary calculation needed to determine the actual loads on a typical floor slab

5.3.3. Rule translation, storage, and implementation

Having extracted the relevant input from the model, from manual user input, and from the output generated from intermediately calculations, the system uses that input in the designated rule-sets. The first set of rules ranges from a simple value comparison (e.g. is a=b?), to more complex ones where multiple value pairs have to be compared simultaneously. An example of the simpler value comparisons is the determination of the appropriate joint width in façade panels based on the panel length. The example was explained previously in section 4.4.4, and its practical implementation in python code is represented in (Figure 5-4). Custom python nodes are used to represent all the rules involving sets of value comparisons, providing the optimum value of a given precast element parameter to be compared to the value of the actual parameter value in the building model. An example of a more complex rule representation is the rule-set of slab thickness, where the slab type, load on a unit of the slab, and the span have to be checked simultaneously to determine the optimum thickness (Figure 5-5). Generally speaking, this subset of nodes, and the written code within, is the practical implementation of the second type of “Referencing tables” system described in section 4.3.3 in this research.

Python Script

```
spacing = IN[0]
wdth = IN[1]
actwdth = []

stat = []
for i in spacing:
    if i <= 2000:
        actwdth.append(15)
    elif 2000<i<=3500:
        actwdth.append(20)
    elif 3500<i<=5000:
        actwdth.append(25)
    elif 5000<i<=6500:
        actwdth.append(30)
    elif 6500<i<=8000:
        actwdth.append(35)

```

Figure 5-4 A snippet from the python code used to represent the “Joint width” rule-set.

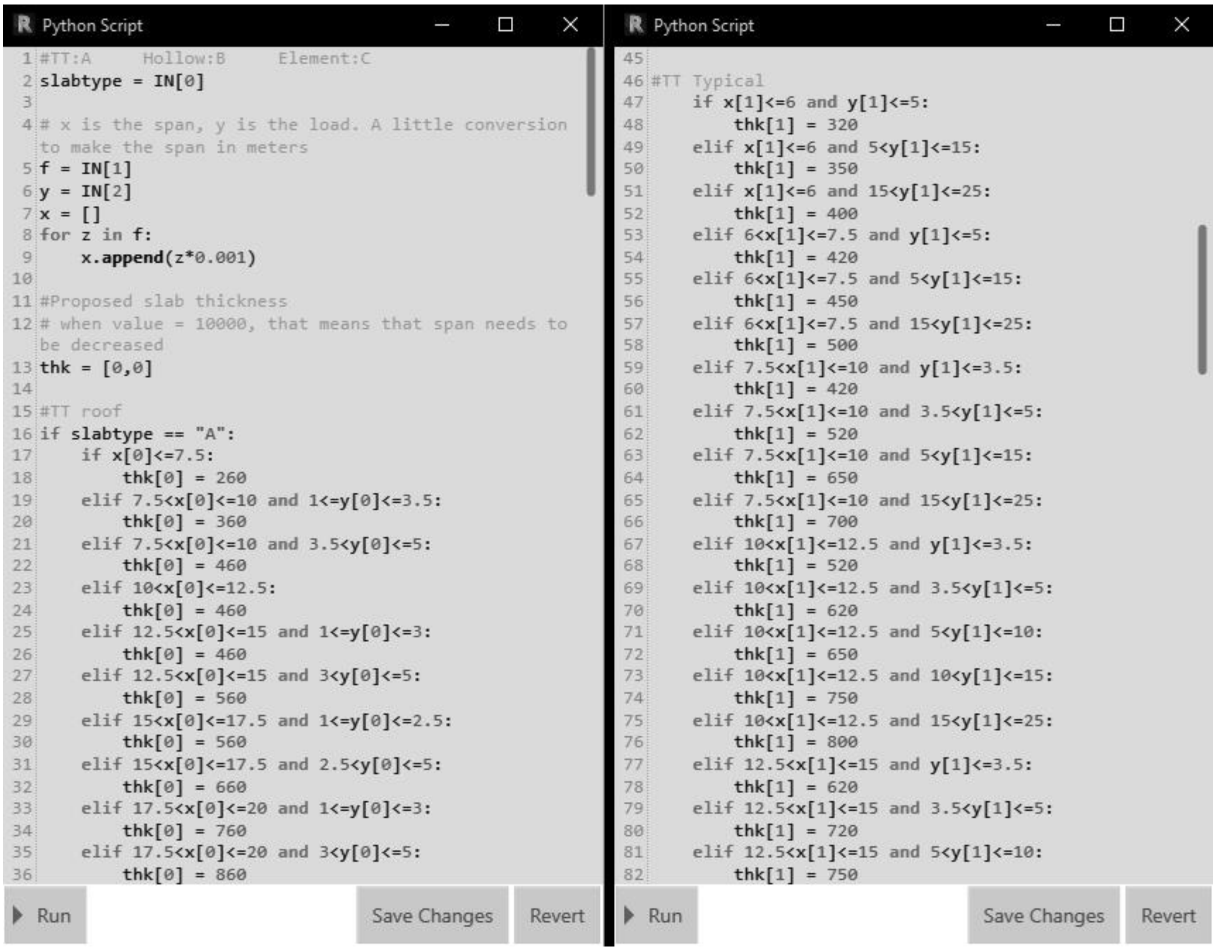

Figure 5-5 Snippets from the python code used to represent the "Slab thickness" rule

The second set of rules is implemented to compare the extracted model values to the recommended values that are either hard-coded in the system or are determined from the first set of rules. The output of these rules is basic Boolean values that determine if the design decisions follow precast considerations in a given case or not. To continue with the previous example, the next rule is implemented to determine whether the actual joint width in the BIM model is within the range of the optimum joint width (Figure 5-6). In the second example, a rule is implemented to determine whether an actual slab's thickness is within the range of the optimum value (Figure 5-7). The outcome of these rules is a simple Boolean value, accompanied by a recommendation describing the optimum value to the user in case of a "No" output. Generally speaking, this subset of nodes, and the written code within, is the practical implementation of the "Action tables" system described in section 4.3.3 in this research.

```python
wdth = IN[1]
actwdth = []

for j,k in enumerate (wdth):
    if k != actwdth[j]:
        stat.append("Warning! Joint width should be around %d mm" % (actwdth[j]))
    else:
        stat.append("Ok")

OUT = stat
```

Figure 5-6 A snippet from the python code used to determine the compatibility of the actual in-model precast panels joint width with the recommended value.

```python
actual_roof = IN[0]
calc_roof = IN[1]
actual_typ = IN[2]
calc_typ = IN[3]
iterate = len(IN[4])
stat = ["Slab thickness ok"]

if calc_roof == 10000:
    stat[0] = "Warning!The designated span is not allowable. Please decrease floor span."
elif actual_roof > calc_roof or calc_roof > actual_roof:
    stat[0] = "Warning! Slab structural thickness should be equal to %d mm instead of %d mm" %(calc_roof,actual_roof)

if calc_typ == 10000:
    stat.extend(["Warning!The designated span is not allowable. Please decrease floor span." for i in range(iterate)])
elif actual_typ>calc_typ or actual_typ<calc_typ:
    stat.extend(["Warning! Slab structural thickness should be equal to %d mm instead of %d mm" %(calc_typ,actual_typ) for i in range(iterate)])

OUT = stat
```

Figure 5-7 A snippet from the python code used to determine the compatibility of the actual in-model slab thickness with the recommended value.

5.3.4. Output

As already mentioned in section 5.2.2, the “Reporting” and “Display results” parts of the system constitute the “Output” portion of the system, presenting the results of the checking process in the five categories described before. Results of the check are presented in such a way, so that only problematic elements are listed and labeled with either a “Critical” or a “Mild” status. In addition, each issue is accompanied by a short description and the suggested course of action. This report is automatically generated at the end of the rule-checking process and the individual issues' description, along with the issues' number, status, and a custom 3D view, are embedded as custom parameters inside the affected Revit elements. Here, IronPython code was used once more to design a GUI for reporting the results and for the user to interact with the interface and open a 3D view for any given issue.

5.4. System test run

5.4.1. Test model description

To test the system prototype, a Revit test model was developed to showcase a realistic use case of the system. The system assumes a model of a LOD of 200-300 as an input. A simple design of a typical L-Shaped office building of five floors and a total built-up area of 1595m^2 was implemented (Figure 5-8). To fulfill the information needs of the developed system, the test model was implemented so it includes the following elements (as highlighted in section 5.2.1):

- Concrete columns [13]
- Column grid
- Concrete Beams [13]
- Concrete Slabs [13] [14]
- Concrete Roof Slab [13] [14]
- Walls/Wall panels [14]
- External wall reveals/joints
- Staircase
- Openings (Doors and windows)
- Levels

[13] No reinforcement information and/or details were necessary
[14] Modeled with detailed layers

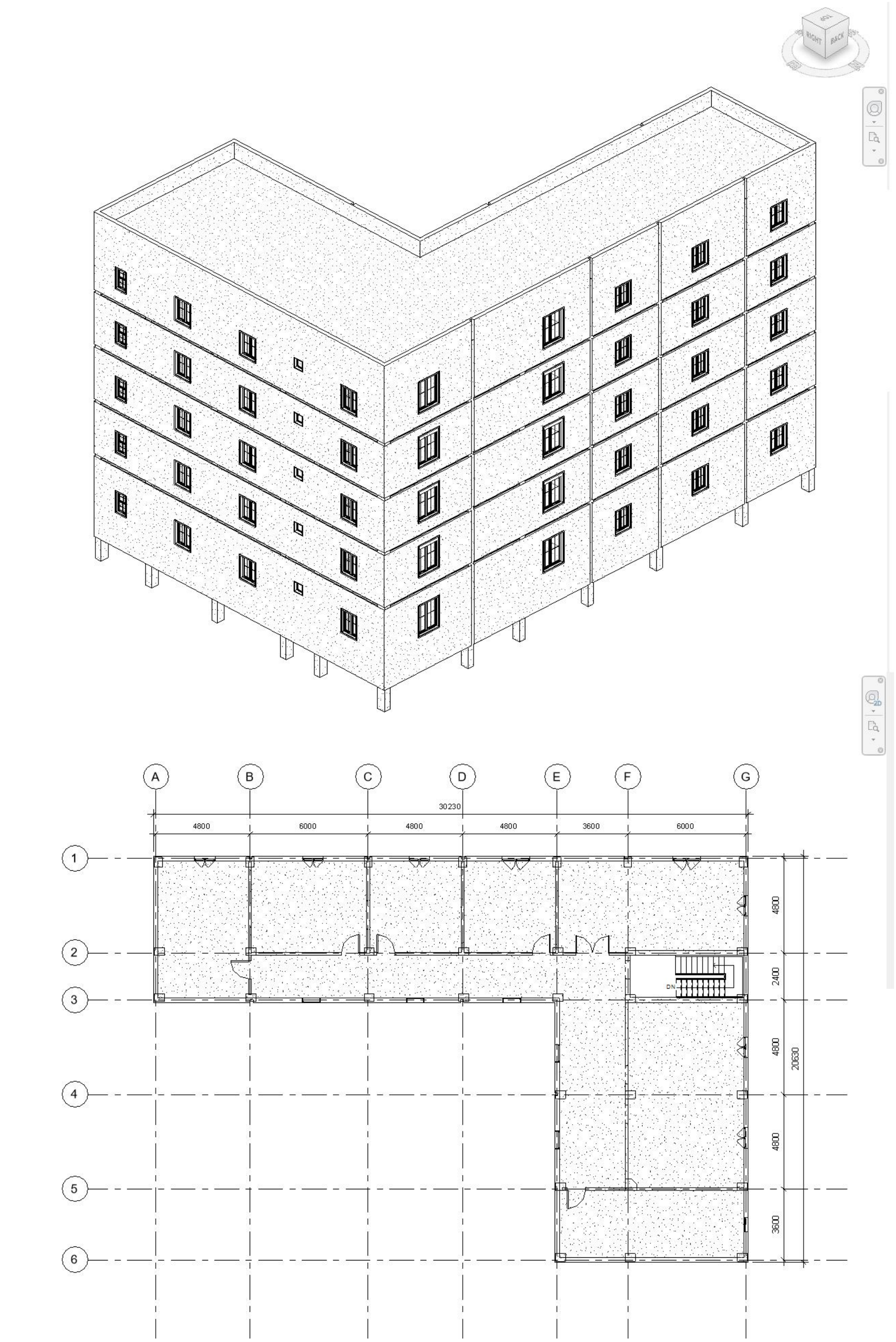

Figure 5-8 The test model 3D (Above), typical floor plan (Below)

Furthermore, the model was developed in such a way to fulfill the following requirements:

- The model follows a design based on a modular grid with fairly uniform bay sizes, typical of precast projects.
- The model also exhibits some variation in room sizes and openings to mimic a real design situation and to test the system capabilities in recognizing patterns
- Other design decisions were made to test the full range of the prototype's capabilities (e.g. Absence of vertical joints on a façade, columns with irregular cross-sections)

5.4.2. Interface

a. Input

Before the actual checks take place, the system requires some manual input from the user to complement the information available inside the model. The whole process can be managed using three simple buttons in "Dynamo player", which enables the user to run pre-made Dynamo graphs in the background, without ever needing to interact directly with them. As a first step, the user should run the first script "PC_01 Run from scratch" in order to initiate the manual input process (Figure 5-9). A "Welcome screen" pops up, providing the detailed steps, which the user has to follow to successfully run the checking process (Figure 5-10).

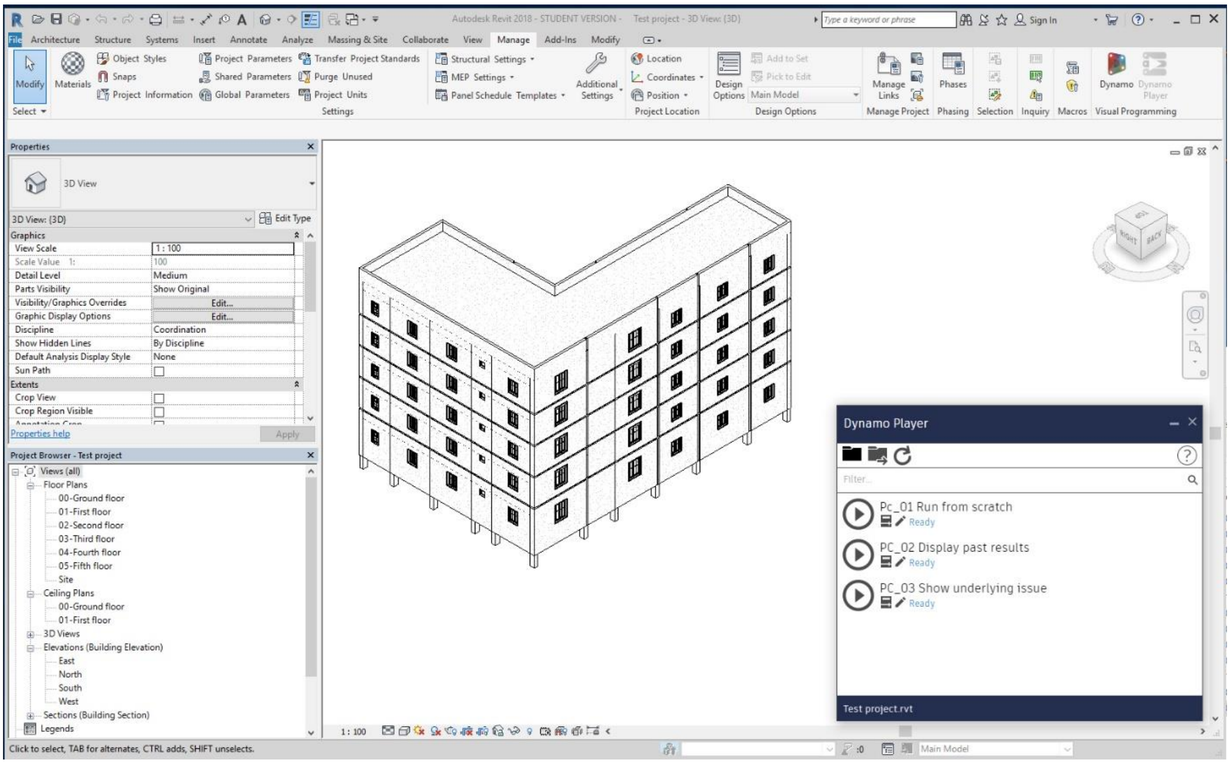

Figure 5-9 The user can control the entire system through three buttons available in Dynamo player

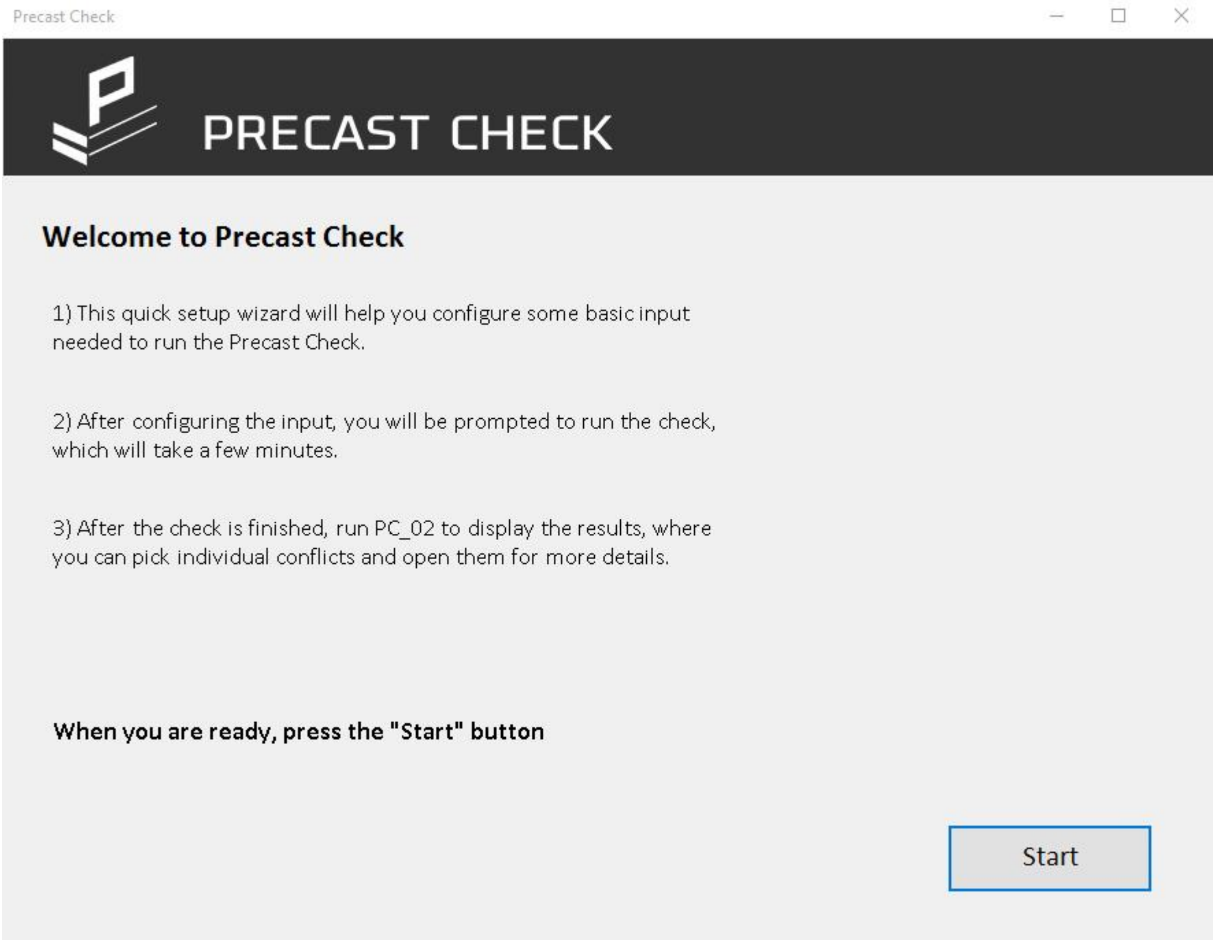

Figure 5-10 The welcome screen of the tool, providing the steps to follow to use the tool

Afterward, a series of screens are dedicated to receiving the user's input (Figure 5-11 to Figure 5-14). These inputs are meant to capture information related to the building's use, type of slabs used, and the city in which the project is located, among other inputs. This information is essential to perform some of the calculations the tool makes in order to present the user with design recommendations. Skipping one or more of these inputs could cause run problems or inaccurate outputs.

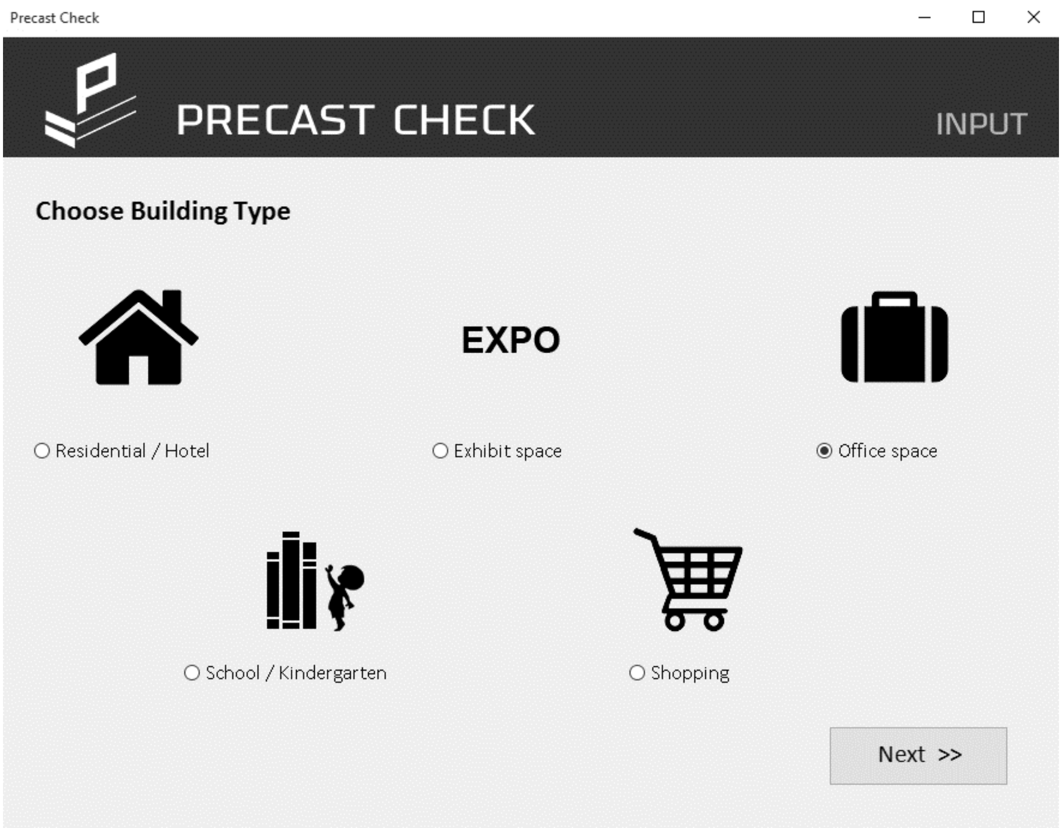

Figure 5-11 Input screen to capture the building type

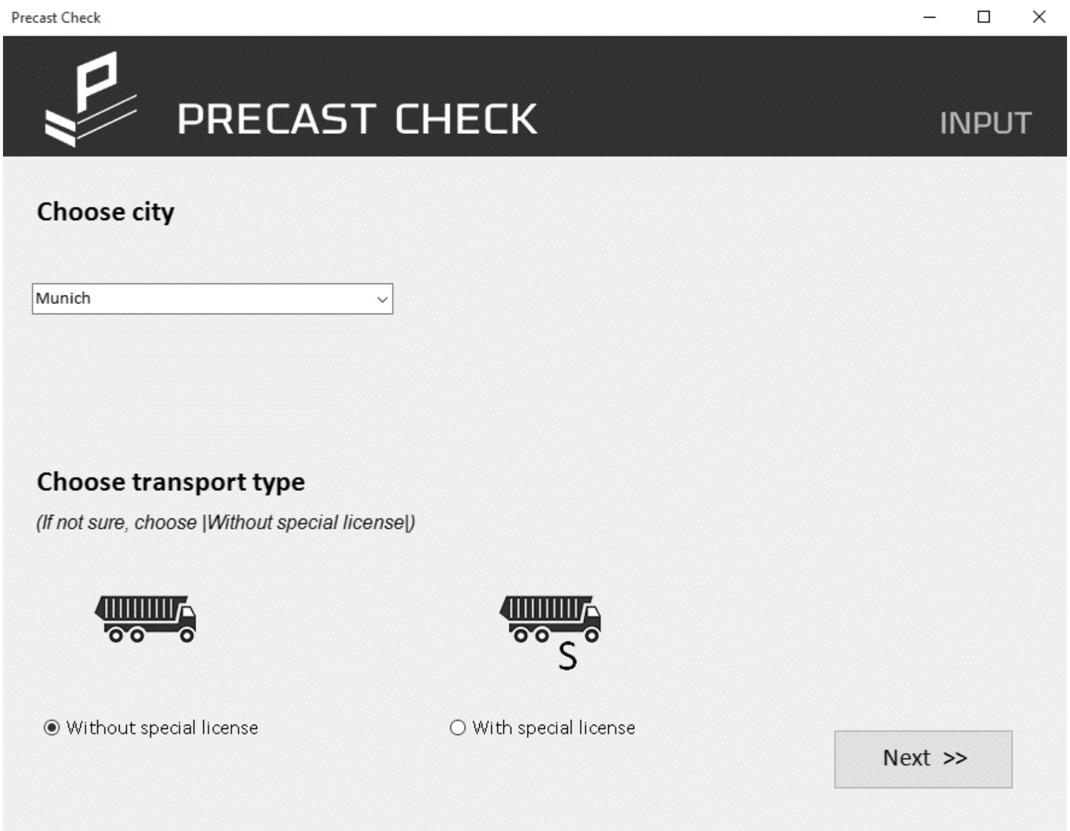

Figure 5-12 Input screen to capture the project's location and transport type

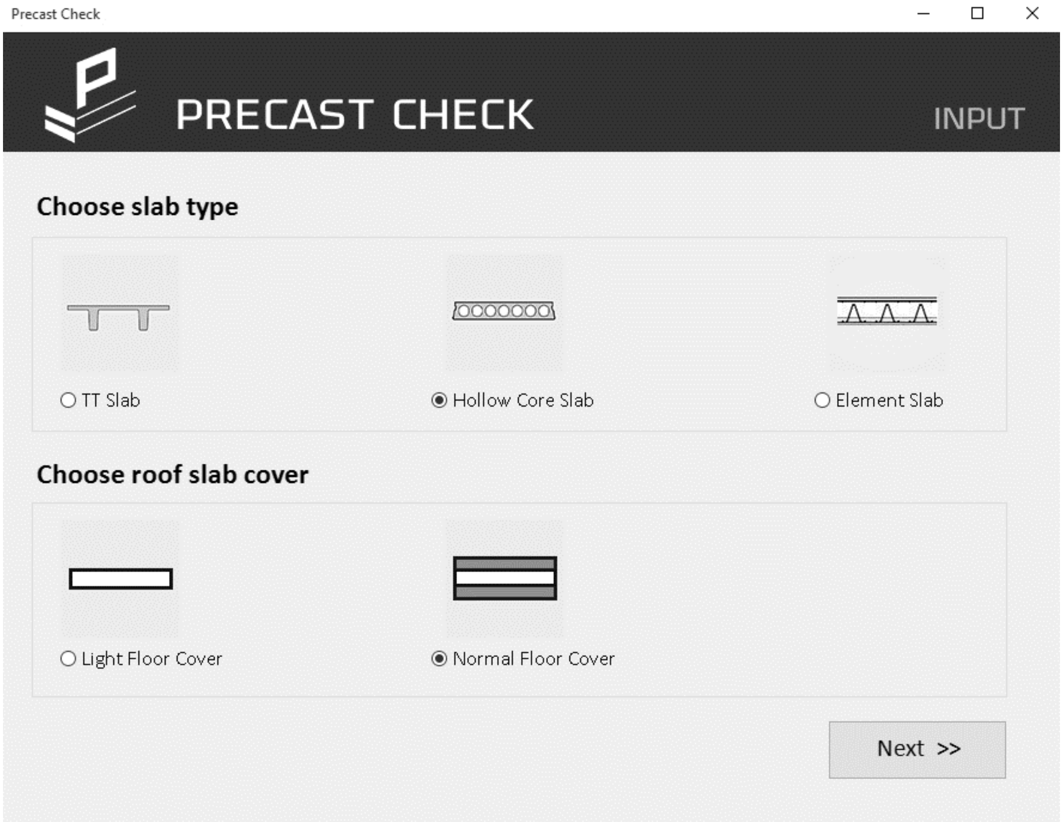

Figure 5-13 Input screen to capture the typical and roof slab types used

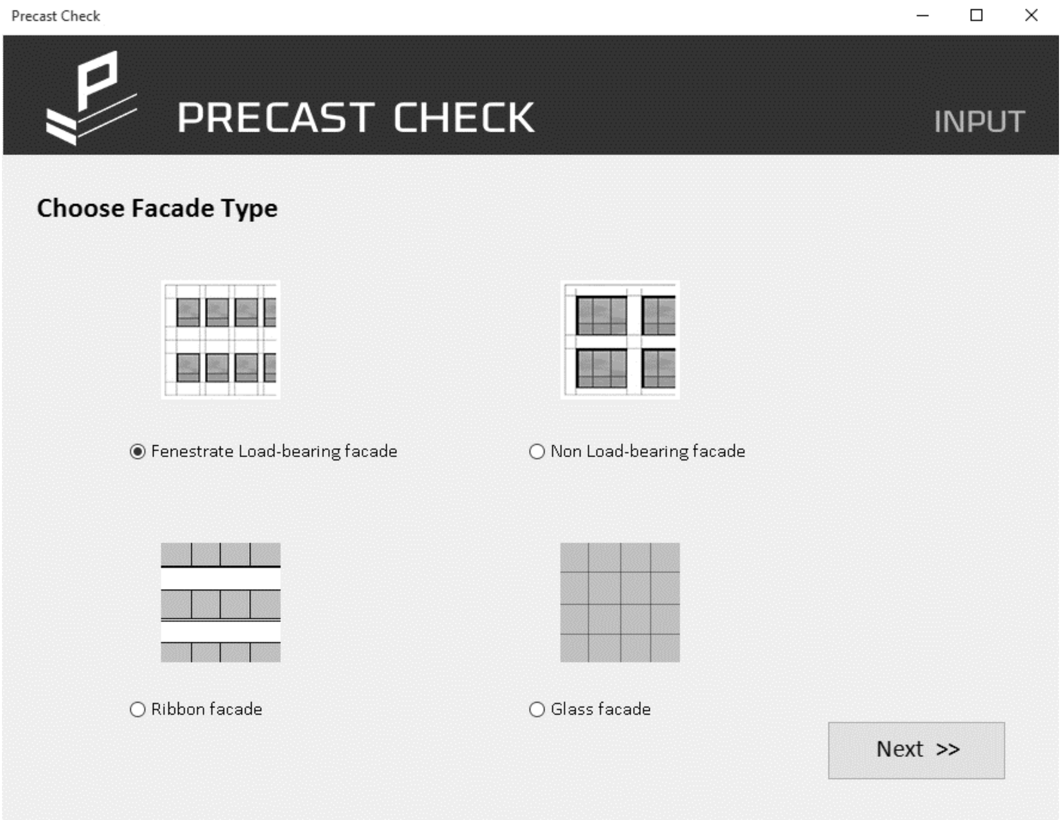

Figure 5-14 Input screen to capture the Facade type

b. Reporting/Output

After receiving the input, the tool uses this input, along with the input from the model, to perform the checks. After the checking process is done, the user needs to run the second script from the Dynamo Player named “PC_02 Display results”. Now the checking results pop up as a multi-tabbed window that classifies the results in the five categories discussed in section 5.3.4 in this research. Each issue is numbered and tagged either as a “Critical” issue that necessarily needs to be addressed, or as a “Mild” issue that is merely a design recommendation to optimize the manufacturing process (But can be ignored). Examples of the checking results can be viewed in Figure 5-15 and Figure 5-16. In the background, the tool creates customized 3D views, which zooms in and isolates every single issue, so that when the user picks an issue to “Show in model”, the tool automatically opens the corresponding dedicated 3D view Figure 5-17.

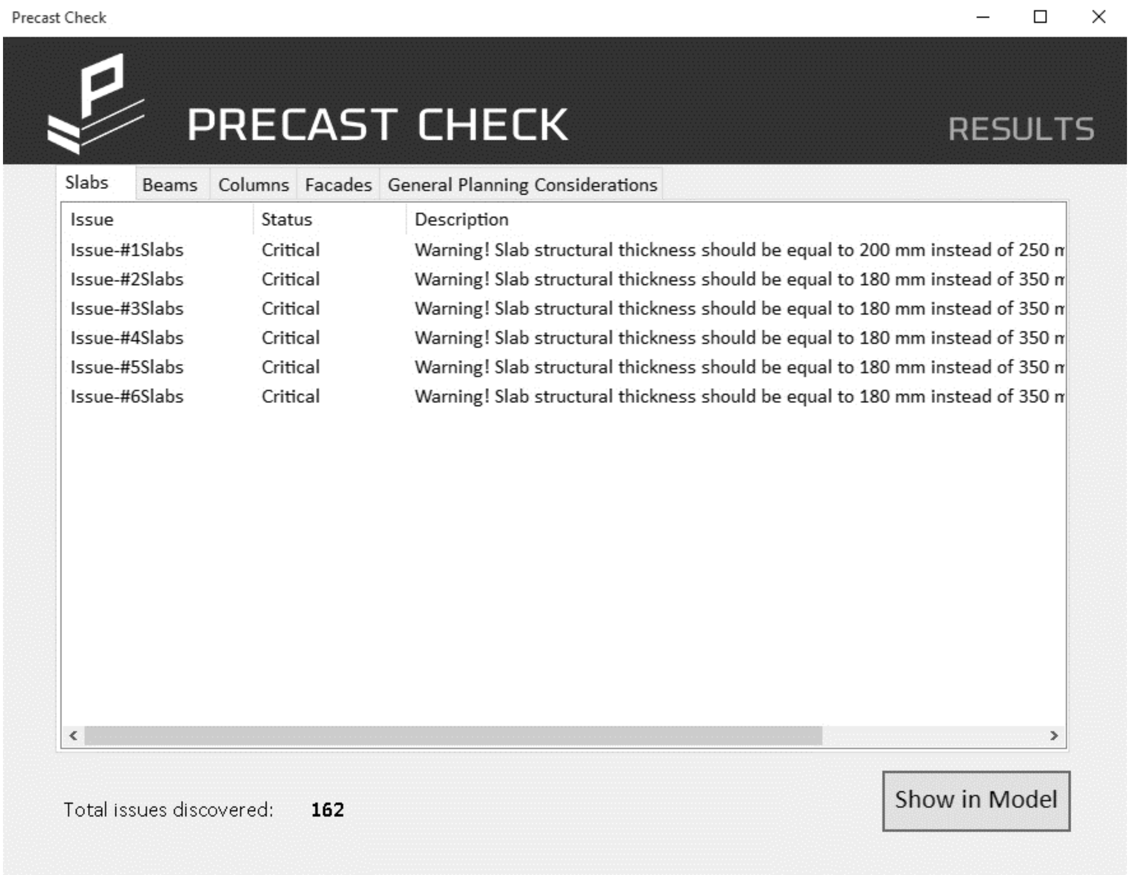

Figure 5-15 The results window displaying issues detected in the model's slabs

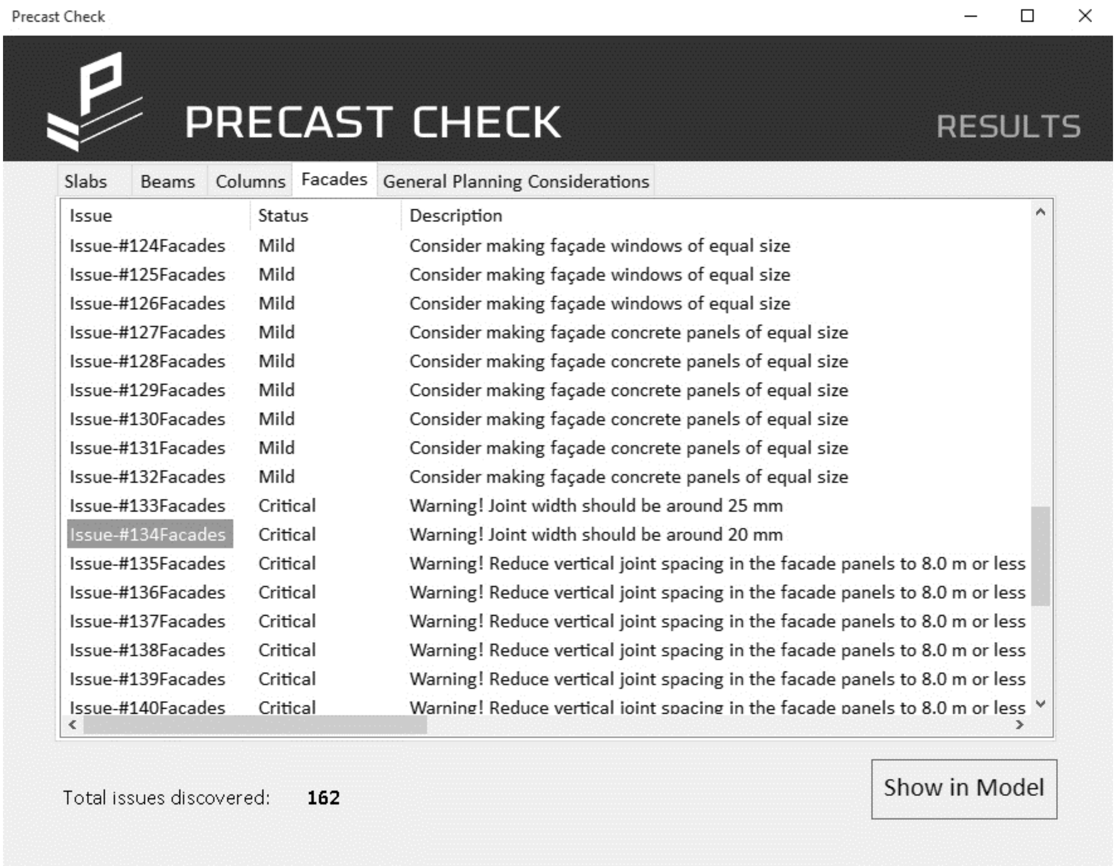

Figure 5-16 The results window displaying issues detected in the model's facade panels

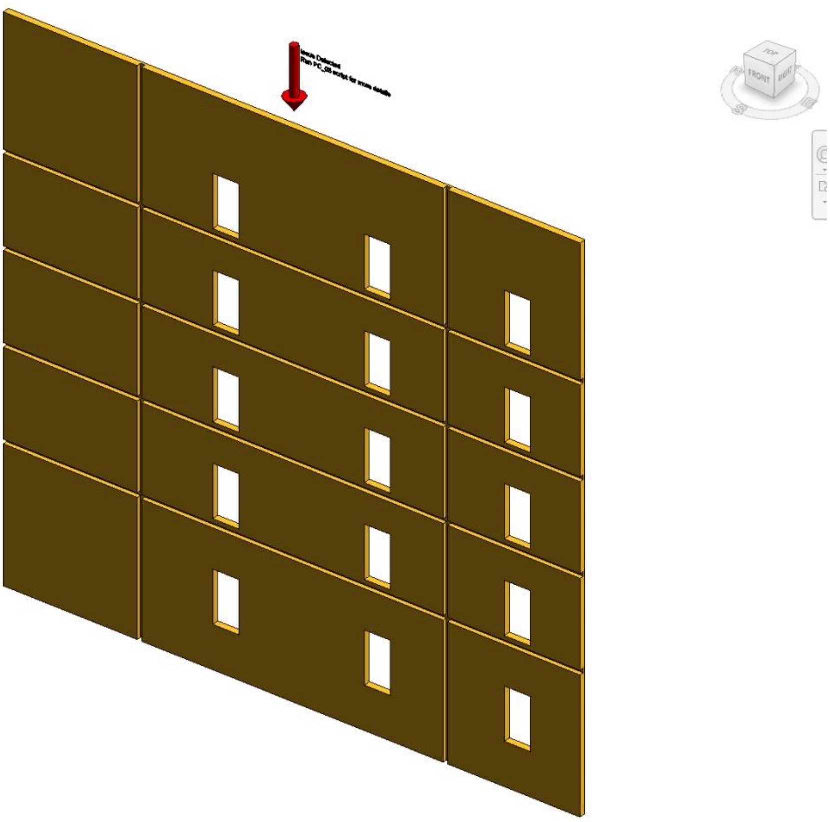

Figure 5-17 A dedicated 3D view for the issue picked to be viewed, in this case to zoom in the joints of an external facade

The user can run the second script to recall the results window as many times as needed to review each issue. Additionally, to recall the information about a specific issue after opening it, the user can use the third and final script “PC_03 Show underlying issue”, pick the arrow accompanying the affected element in view, and then click on the “Show” button (Figure 5-18).

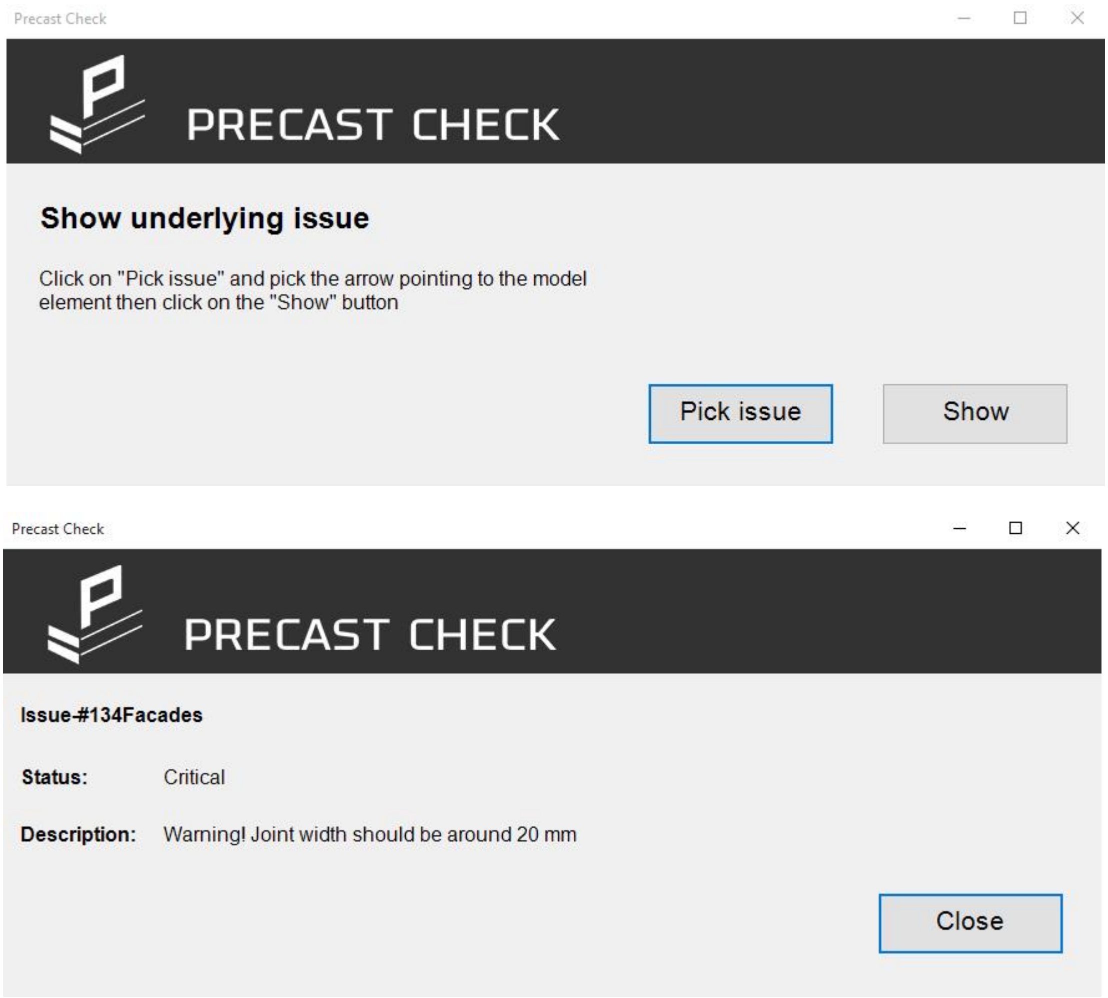

Figure 5-18 The pop-up window that appears when running the third script (Above), and the window showing the display of recalled information (Below)

5.4.3. System prototype’s remarks

The prototype was not meant in itself to be a final polished tool, which is ready for commercial use, but rather a way to demonstrate the capabilities of the developed system framework, which is the main contribution of this research. Upon the implementation and the actual testing of the prototype, the following issues need to be emphasized:

- The choice of Revit's open visual programming tool "Dynamo" makes it possible, and fairly easy, to extend the incorporated rule-sets to encompass new rules.
- The prototype can be implemented only in models that fulfill the requirements described in sections 5.2.1 and 5.4.1. Models that do not fulfill these requirements could result in a system error and failure of the checking process and/or inaccurate output.
- The checking process not instantaneous, exhibiting runtimes of approximately two and a half minutes for every run of the checking process for the chosen test model. The ability to disable one or more of the checking modules is currently unavailable.
- For the prototype tool to run without errors, a number of custom packages need to be installed to Dynamo[15] and Revit[16]. These are "Archilab"[17], "Ampersand"[18], and "Clockwork"[19].
- For the purposes of data management and ease of use, a custom "Arrow" family is inserted in all the model parts that warrant a system "Issue detected" warning. These arrows remain in the file, so the user can revisit the detected issues at any time, until manually deleted.
- The values and recommendations given by the system are meant only as guidelines and approximations to the optimum values in earlier design stages. These values should be revisited and fine-tuned by human experts in later design stages.

5.5. Conclusion

To demonstrate the potential of the developed system framework, a prototype software tool was implemented inside Revit and its visual programming language "Dynamo". This software environment makes it possible to manage BIM content, along with manual input, while seamlessly integrating in the natural flow of design activities. It also has the capabilities needed to build a custom user interface, manage and store rule-sets, and to build a reporting interface. The tool provides an easy-to-use GUI for architects, with few simple inputs, to automatically check for precast requirements. The tool assumes a model with a specific set of characteristics in order to carry out the necessary checks, and a test model of a simple office building was used to test the tool's capabilities. The tool was able to detect a number of issues and reported them back to the user, with the recommended design edits needed to

[15] Version 2.0.1 at the time of implementation
[16] Version 2018 at the time of implementation
[17] Version 2019.2.2 at the time of implementation
[18] Version 2018.10.28 at the time of implementation
[19] Version 2.1.0 at the time of implementation

comply with precast considerations, and with the ability to revisit each issue and resolve it separately. It should be emphasized that the prototype tool is merely a proof-of-concept to showcase the potential of the developed system and its applications in real life, and is not meant to be an “End product” in itself.

Chapter 6: Conclusion, limitations, and future work

In spite of the great advancements in the design and construction sector in recent years, a clear productivity stagnation problem still plagues the entire construction industry. This problem can be traced back to the fragmented nature of today's practices, which often leads to coordination issues, design rework, and ill-informed design decisions. This is particularly apparent in prefabrication-based construction, which is supposed to save time, increase quality, and improve productivity. This process fragmentation across the industry often leads to a situation where designers carry on their design activities and make design decisions with very little input, if any, from precast experts concerning the nature of prefabricated construction and its requirements. This leads to badly informed design decisions, which in turn leads to a host of problems in the detailing and constructions phase, potentially increasing cost, extending schedules, and ultimately eroding the expected productivity gains from prefabricated construction. Focusing on precast construction, this research attempted to address this problem and to provide a system framework to enable architects in the early design stages to make well-informed design decisions concerning prefabrication-based construction projects. In the following few pages, the conclusion and the limitations of this research will be presented, and proposals for further development will be offered.

6.1. Conclusion

To review the conclusions and the primary findings of this research, it is useful to revisit the research questions and to go over the research's attempt to address each question and the overall objective of the research.

- *"What is the nature and mechanics of the current prefabricated construction practice and what are the inefficiencies caused by the fragmented design and construction activities?"*
    - This question was addressed in chapter 2 of this research. It was found that although labor productivity in off-site construction has improved over the years in comparison to traditional construction, it still enormously lags behind the manufacturing sector.
    - The AEC practice still largely relies on an inefficient process, namely the design-bid-build model, where the designers and contractors carry on their tasks and

activities in complete isolation. This is true in traditional construction projects as well as prefabricated construction.

- This problem is further complicated in the German market by the fact that this market relies heavily on small companies with a small number of personnel, which makes it very difficult to employ a wide variety of experts, including prefabrication experts, in design firms.
- Precast experts, for example, have to provide expertise concerning panelization sizing and layout, bay sizes, modularization possibilities, shipping sizes limitations, among other issues in the earlier design stages. The absence of this expert input has been repeatedly proven through surveys to cause ambiguities and problems in the production phase of the prefabricated parts.

- *"What are the current CAD and BIM tools being employed in the design process of prefabrication-based construction and how can we improve upon them?"*
    - This question was also addressed in chapter 2 of this research. It was found that BIM systems are still on their own "ignorant". They provide an output that is only as good as the information you feed the system. In other words, "Generic BIM technology" still inherently lacks domain-specific human expertise, including prefabrication expertise.
    - Furthermore, available commercial software solutions are entirely geared towards the later stages of prefabrication detailing, not the early stages of design, and are, understandably, tailored to the needs of precast experts, not the architects.
    - After reviewing current and previous scientific literature that tried to tackle this issue, it was found that the main approach was to designate some sort of a digital tool or a digital platform that relies on either custom libraries, rule-based checks, automated detailing algorithms, or design optimization algorithms, among others.
    - The reviewed research suffered one or more of the following drawbacks:
        - Reliance on geometric models rather than data-rich BIM models
        - Reliance on manual user input, assuming the presence of a human expert.

      - Reliance merely on a predefined library of components, limiting the design space.
      - Being too generic, trying to address all types of prefabrication across all systems, which is not a realistic goal.
      - Focusing on the detailed design phase only, rather than trying to bring the prefabrication expertise to the earlier phases.

- *"What is the best way to capture and formalize expert knowledge about prefabricated construction considerations to support the designer in earlier design stages?"*
   - This question was addressed in chapter 3 of this research. Expert systems as a branch of AI is one of the most dominant approaches for computer-aided design decision support in the AEC domain. That is usually because the AEC domain is well defined and information could be easily captured and articulated to enable the transformation of knowledge into computer-readable code.
   - Rule-based expert systems, as a special type of expert systems, offer the possibility to take rational, non-biased decisions, providing consistent results. They can also maintain, store, and process a huge amount of information, thus saving time, cost, and reducing errors. They can also be incrementally expanded over time.
   - This type of systems is particularly suitable for the purposes of this research because:
      - They have been tested and used extensively in similar applications in the AEC industry
      - Prefabrication knowledge can be easily transformed into computer-readable rule-based logic
      - These systems provide the possibility to serve as an "Advisory" system, where it only provide feedback and recommendations, rather than acting intrusively and making what could be "undesired" edits to the design model.
   - A "Rule database" lies in the heart of every rule-based expert system. There are many approaches to knowledge formalization to establish the rule database. In the context of this research, the tabular form (Decision tables) are chosen for the following reasons:

        - Their clear structure makes it easy to identify gaps, inconsistencies, and logical errors in rule formulation.
        - They enable rule description involving parallel reasoning, as opposed to other rule representation methods, which use sequential flows.
        - They can easily be analyzed and tested to avoid contradictory and/or redundant rules.
    - An implementation of such an expert system in a BIM environment, it was necessary to define the LOD of a given model, which is necessary to use the system. It was found that models with LOD 200-300 are best suited for the purposes of this research. At these levels, the design is developed well enough to start carrying out the preliminary checks needed in this context, yet not too developed and detailed to the degree of making the edits too costly.

- *“How can a BIM-based system be devised and structured to support designers early on by automating checks for prefabrication requirements in design models?”*
    - This question was addressed in chapter 4 of this research. The answer proposed to this question is to establish a decision support system that checks design models and detects problematic design decisions regarding prefabrication considerations, and reports them back to the user in an iterative manner with every design change.
    - To begin specifying the structure and the contents of that system, it was necessary not to fall into the trap of being too generic or specific in the research’s approach. Accordingly, it was determined from the beginning that the focus will be on precast concrete.
    - The proposed system should comprise structural requirements, cost requirements, transport requirements, and production requirements for precast concrete. Some of these requirements can be generalized and others had to stick to a specific market, and the German market was chosen in this case.
    - The system, which is proposed to be a rule-based expert system, operates on a precast rule database and can be described in terms of:
        - Input: A BIM model with LOD 200-300 and manual user input
        - Filtering and rule execution: Filter and establish dependencies between precast building components then apply the rule-sets.

    - Output: Compile a report summarizing the problematic model parts and recommend how to address them.
- The key part of this system framework is the rule database. To establish this database, three steps were followed:
    - Knowledge acquisition: Through an extensive literature review, semi-structured interviews, textbooks, international, and national standards.
    - Rule-set structuring and categorization: The rule-set is classified into two layers. The first layer, which is visible to the user for the ease of use, classifies the rules according to the precast element type to slabs, beams, wall panels, columns, and general planning considerations. The second layer tags this rules for clarity in rule archiving with one of the following tags: Production requirements, cost requirements, transportation requirements, and structural requirements.
    - Knowledge formalization & representation: The rules are represented in its final form, as decision tables, providing a concise, machine transferable middle format. A network of decision tables was classified, as described in the previous step, and developed in a three-tiered structure: Referencing tables, calculation tables, and action tables.
- The established system framework simulates the generic reasoning necessary to establish a rule-based expert system to check for precast requirements in the reviewed categories. The rule-base and the whole system can be expanded to accommodate other rules, for example for a specific manufacturer, or different rule, for example for a different market.
- Some examples of the rule database were provided in the main text, the entire database is provided in appendices A to E. The rule-set is not meant to be exhaustive or to provide every reviewed precast requirement. Specifically, rules included are only the ones that are:
    - Suitable for implementation with the information available in BIM LODs 200-300
    - Relevant to the architectural design considerations
    - Possible to be transformed into computer-readable/interpretable logic.
    - Generalizable in most precast projects.

- *"How can such a system be practically available to the designer to utilize and interact with?"*
  - This question was addressed in chapter 5 of this research. The chapter presented a prototype software tool to demonstrate the potential of the practical implementation of the developed system framework. It was determined that "Autodesk Revit" along with its visual programming tool "Dynamo" and its textual programing language "IronPython" comprise the most suitable environment to implement such a tool.
  - The presented tool is composed of several components that work together to eventually provide the user with design recommendations concerning precast requirements. These components can be explained as follows:
    - In addition to a BIM model with LOD200-300, the tool requires additional information from the user like the building type, project location (city), slabs' types, among other inputs.
    - The tool captures and filters information accordingly. The tool extracts and filters the relevant information for each individual check from the BIM model and manual user inputs.
    - Using this information, the tool calculates intermediary information needed to carry on the checking process.
    - The developed rule-sets, which are stored in the form of computer code inside the tool, are recalled to check the design and scan it for issues that do not comply with the rules.
    - The reporting component classifies the detected issues by type, labels them either as "Critical" or "Mild", and pushes them to the GUI.
    - The final component is the GUI that presents the results to the user, enabling him/her to open a 3D dedicated view for each issue and recalling its description.
  - Finally, the research presents a "Test run" to examine the actual software implementation of the presented approach. The tool was able to run smoothly, albeit not instantaneously, completing the design-check in approximately two and a half minutes and providing the user with design recommendations to make the design more "Precast compatible".

Furthermore, it is important to consider the methodology, components, and the contributions of this research in a broader context.

- *Producing more "intelligent" design models*

  The AEC industry's current design and construction models suffer from an inherent "Segregation" problem. There is often a misalignment between design models and construction models that goes far beyond "Detailing" activities. More Intelligent, better-informed design models are needed from the earlier design phases can save a lot of time, effort, and money in later design and construction phases. The application presented in this research is but a small step in filling the void of the inherent lack of intelligence in current BIM models and BIM platforms in the domain of construction readiness/soundness.
- *A step closer to full automation*

  The design process has become increasingly complex and lacks significantly in productivity when compared to other industrial domains. The increasing number of disciplines, which are becoming a fundamental part of the design process, are adding to the problem's complexity. Each domain, system, or discipline comes with a specific set of needs and requirements that need to be integrated in design decision making. Utilizing the information-rich BIM environments in automating the checking process for these requirements increases productivity in design processes, even in domains, such as the prefabrication/precast domain, where a human expert is still needed in later stages to check and detail design schemes.
- *System extendibility*

  Possible Parallels can be drawn to this research's problem. There are many similarities between the workflow in other building systems and precast concrete. If we consider, for example, other types of prefabrication-based construction, or even other types of construction, such as Cast-in-place concrete, we could find similarities in design and construction processes, workflows, and information exchange requirements. Taking into consideration the specific nature of each system, and the different rules that govern each of them, the same building blocks and components utilized in the system framework developed in this research could be re-engineered and re-calibrated to suit a diverse array of other applications in design and construction. The prototype tool demonstrated here is just a proof of the system concept, which is the main contribution of this research.

6.2. Limitations

During the course of work in this research, a number of limitations were encountered in designing the proposed framework as well as the developed prototype.

- Limitations in the proposed framework
    - In order to be able to compile rule-sets for prefabrication/precast, it was necessary to limit the scope to a specific market. Different countries and different markets usually follow different practices and standards can vary greatly. Accordingly, the German market was the one considered in this research. However, the approach could be tailored to any other market.
    - Different prefabrication/precast factories have different internal standards. For example, two factories in the German market may use different machinery, moulds, or transport and lifting equipment. The research's approach was to try not to be contrived by such limitations, although they can be important factors, and to stick, instead, to an industry-wide core of standards. This decision is especially important because one of the main problems of the industry, especially in Germany, is the uncertainty concerning the specific construction and/or the prefabrication firm that will win the contract. However, it is important to restate that the presented approach could be tailored to the standards of specific prefabrication companies when possible.
    - The intent of the proposed system from the beginning is to improve current conditions, to provide some expert input to earlier design schemes, and to do so in an automated way. The approach, however, does not eliminate the need for human experts in later design stages, where much more accurate values have to be determined and when all prefabrication aspects have to be considered, including the ones that a computer system cannot currently handle.

- Limitations in the developed prototype
    - The prototype was developed in a closed BIM system, namely the Autodesk Revit platform, in order to avoid inconsistencies found in platforms that deal with BIM models in IFC format. It was found that IFC exported models typically exhibit irregularities and misclassifications that would render the checking process useless. Another reason for using such a closed system is the necessity of streamlining the checking process, and making it an integral part of design activities, instead of shuffling back and forth between numerous platforms.
    - The utilized rule-sets can be fairly considered a “Black Box”. The end-user does not see the rules themselves in action, although he can add, remove, and edit them if he/she possesses the necessary technical expertise to do so. It is important to restate that Autodesk Dynamo is an open visual programming tool that is easily accessible. However, a tradeoff had to be made between simplicity/ease-of-use and maximum flexibility.
    - To be able to use the tool, a BIM model has to be developed in such a way that it follows a set of criteria, which were described in section 5.4.1. Furthermore, the tool cannot handle the following cases:
        - Curved walls
        - Custom made precast walls with custom ornaments
        - Ribbon facades

6.3. Future work

- With the release of more developed IFC standards, better designed and faster exporters and importers, it may be possible to develop applications based on the proposed system that are based on open BIM information exchange workflows. This could help develop the added level of intelligence on a Meta level instead of being bound to a specific BIM platform.
- In future iterations, it may be possible to enable the end-user to modify or adapt the rules using the GUI instead of having to open the Black Box environment behind the system, which could prove challenging to normal users. It may also be possible to

enable the system to intervene and directly edit the problematic design/model elements upon request from the end-user.

- The design and implementation of the system are based mainly on precast concrete. However, the system concept is flexible and can accommodate other prefabrication applications. Future iterations of the system can be designed to check for another type of prefabrication-based systems such as prefabricated steel structures, prefabricated timber structures, and “Box modules” or “Room modules” like the ones from “Max Bögl” (maxmodul) (MaxBögl, 2019) or Room modules from “Kaufmann Bausysteme” (Kaufmannbausysteme, 2019).
- Future iterations of the system could also be implemented to be specific to other countries and markets if needed. Some rules are generic and applicable to every market, but others need to be re-curated and re-formulated to suit the needs of individual markets.
- Finally, the idea of adding an additional layer of intelligence to BIM models concerning construction considerations can be extended using the same basis of the proposed system to include other construction systems and construction-related applications. Having an automated checking system to guide the designer through the complex issues of construction in the earlier phases could prove to greatly enhance productivity and decrease conflict in preconstruction and construction activities.

References

1. ACI 1973. *Decision Logic Table Format for Building Code Requirements for Reinforced Concrete*, American Concrete Institute.

2. Adeli, H. 1988. *Expert Systems in Construction and Structural Engineering,* London, UK, CRC Press.

3. AIA 2007. *Integrated Project Delivery: A Guide*, The American Institute of Architects. Available: https://www.aia.org/resources/64146-integrated-project-delivery-a-guide [Accessed 05/09/2018].

4. Alfino, M., Caputo, J. S. & Wynyard, R. 1998. *Mcdonaldization Revisited: Critical Essays on Consumer Culture,* Connecticut, US, Greenwood Publishing Group.

5. Alwisy, A., Hamdan, S. B., Barkokebas, B., Bouferguene, A. & Al-Hussein, M. 2019. *A BIM-Based Automation of Design and Drafting for Manufacturing of Wood Panels for Modular Residential Buildings*. International Journal of Construction Management, 19**,** 187-205.

6. Aram, S. 2014. *A Knowledge-Based System Framework for Semantic Enrichment and Automated Detailed Design in the AEC Projects.* Doctoral dissertation, Georgia Institute of Technology.

7. Arditi, D., Ergin, U. & Günhan, S. 2000. *Factors Affecting the Use of Precast Concrete Systems*. Journal of Architectural Engineering, 6**,** 79-86.

8. Autodesk. 2019. *What Is Dynamo?* [Online]. Available: https://primer.dynamobim.org/01_Introduction/1-2_what_is_dynamo.html [Accessed 24/06/2019].

9. Bachmann, H. & Steinle, A. 2011. *Precast Concrete Structures,* Berlin, Germany, Ernst & Sohn.

10. Bachmann, H., Steinle, A. & Hahn, V. 2009. *Bauen Mit Betonfertigteilen Im Hochbau,* Berlin, Germany, Ernst & Sohn.

11. Bai, Q., Deng, S., Li, C. & Qie, Z. 2019. *Application of BIM in the Creation of Prefabricated Structures Local Parameterized Component Database*. Architecture and Engineering, 4**,** 13-21.

12. BauenAktuell. 2019. *Deutsche Unternehmen Laufen Digitalisierung Hinterher* [Online]. Available: https://www.bauen-aktuell.eu/node/35954 [Accessed 23/04/2019].

13. Belsky, M., Sacks, R. & Brilakis, I. 2016. *Semantic Enrichment for Building Information Modeling*. Computer-Aided Civil and Infrastructure Engineering, 31**,** 261-274.

14. Bimforum, b. 2016. *Level of Development Specification 2016*, BIMForum.

15. Blessing, L. T. M. & Chakrabarti, A. 2009. *DRM, a Design Research Methodology,* London, UK, Springer.

16. Bloch, T. & Sacks, R. 2018. *Comparing Machine Learning and Rule-Based Inferencing for Semantic Enrichment of BIM Models*. Automation in Construction, 91**,** 256-272.

17. Borchardt, A., Reise, H., Schwerm, D. & Treppke, D. 2002. *Betonfertigteile Für Den Wohnungsbau,* Bonn, Germany, Fachvereinigung Deutscher Betonfertigteilbau e. V.

18. Borrmann, A., König, M., Koch, C. & Beetz, J. 2018. Building Information Modeling: Why? What? How? *In:* Borrmann, A., König, M., Koch, C. & Beetz, J. (eds.) *Building Information Modeling - Technology Foundations and Industry Practice.* 1st edition ed. Cham, Switzerland: Springer nature Switzerland AG.

19. Brandt, J. 1994. *Beton-Bauteile Für Den Industrie- Und Verwaltungsbau,* Düsseldorf, Germany, Beton-Verlag GmbH.

20. Brandt, J., Röse, W., Schwerm, D. & Stöffler, J. 1993. *Betonfertigteile Im Skelett- Und Hallenbau,* Bonn, Germany, FDB Fachvereinigung Deutscher Betonfertigteilbau.

21. buildingSMART. 2013. *Industry Foundation Classes Release 4 (Ifc4)* [Online]. Available: http://www.buildingsmart-tech.org/ifc/IFC4/final/html/ [Accessed 22/11/2018].

22. Chanakya, A. & Arya, C. 2009. *Design of Structural Elements: Concrete, Steelwork, Masonry and Timber Designs to British Standards and Eurocodes,* London, UK, Taylor & Francis.

23. Collins, J. & Gentry, R. *Kbad - Knowledge Base for Architectural Detailing*. ShoCK! – Sharing of Computable Knowledge!: Proceedings of the 35th eCAADe Conference, 2017 Rome, Italy. 657-666.

24. Cornick, T. & Bull, S. *Expert Systems for Detail Design in Building*. Proceedings of the second International CAADFutures Conference, 1987 Amsterdam, Netherlands. 117-126.

25. Costa, G. & Madrazo, L. 2015. *Connecting Building Component Catalogues with BIM Models Using Semantic Technologies: An Application for Precast Concrete Components*. Automation in Construction, 57**,** 239-248.

26. Cronembold, J. R. & Law, K. H. 1988. *Automated Processing of Design Standards*. Journal of Computing in Civil Engineering, 2**,** 255-273.

27. Dawar, N., Eric, W., Justin, M., Matthew, M. & Tony, N. 2014. *Transportation Considerations in Module Design*. Structures Congress 2014**,** 1771-1781.

28. de Albuquerque, A. T., El Debs, M. K. & Melo, A. M. C. 2012. *A Cost Optimization-Based Design of Precast Concrete Floors Using Genetic Algorithms*. Automation in Construction, 22**,** 348-356.

29. DeWitt, S., Yakowenko, G., Bohuslav, T., Ferguson, T., Hoelker, E., Molenaar, K., Schiess, G., Smythe, J., Triplett, J. & Wagman, R. 2005. *Construction Management Practices in Canada and Europe*. Available: https://international.fhwa.dot.gov/construction_mgmt/pl05010.pdf [Accessed 26/11/2016].

30. DIBt. 2019. *Deutschen Instituts Für Bautechnik* [Online]. Available: https://www.dibt.de/ [Accessed 04/05/2018].

31. Dieckmann, A. 2019. *Clockwork for Dynamo* [Online]. Available: https://github.com/andydandy74/ClockworkForDynamo [Accessed 30/06/2019].

32. Diez, R., Padrón, V. M., Abderrahim, M. & Balaguer, C. 2007. *Autmod3: The Integration of Design and Planning Tools for Automatic Modular Construction*. International Journal of Advanced Robotic Systems, 4**,** 457-468.

33. DIN 2010. *Eurocode 1: Actions on Structures - Part 1-1: General Actions - Densities, Self-Weight, Imposed Loads for Buildings; German Version En 1991-1-1:2002 + Ac:2009*, Deutsches Institut für Normung

34. DIN 2016. *VOB 2016 Gesamtausgabe: Vergabe- Und Vertragsordnung Für Bauleistungen,* Berlin, Germany, Beuth Verlag GmbH.

35. Eastman, C., Lee, J. M., Jeong, Y. S. & Lee, J. K. 2009a. *Automatic Rule-Based Checking of Building Designs*. Automation in Construction, 18**,** 1011-1033.

36. Eastman, C., Sacks, R., Panushev, I., Aram, V. & Yagmur, E. 2009b. *Information Delivery Manual for Precast Concrete*. Available: http://dcom.arch.gatech.edu/pcibim/documents/IDM_for_Precast.pdf [Accessed 28/03/2016].

37. Elliott, K. S. & Jolly, C. 2013. *Multi-Storey Precast Concrete Framed Structures,* West Sussex, UK, Wiley-Blackwell.

38. Farr, E. R. P., Piroozfar, P. A. E. & Robinson, D. 2014. *BIM as a Generic Configurator for Facilitation of Customisation in the AEC Industry*. Automation in Construction, 45**,** 119-125.

39. FDB 2016. *FDB Merkblatt Nr. 3: Zur Planung Vorgefertigter Stahlbetonfassaden,* Bonn, Germany, Fachvereinigung Deutscher Betonfertigteilbau (FDB).

40. Fédération Internationale de la Précontrainte "FIP" 1994. *Planning and Design Handbook on Precast Building Structures*, SETO.

41. Fenves, S. J. 1966. *Tabular Decision Logic for Structural Design*. Journal of the Structural Division, 92**,** 473-490.

42. Fenves, S. J., Gaylord, E. H. & Goel, S. K. 1969. *Decision Table Formulation of the 1969 Aisc Specification,* Illinois, US, University of Illinois.

43. Finlay, P. N. 1990. *Decision Support Systems and Expert Systems: A Comparison of Their Components and Design Methodologies*. Computers & Operations Research, 17**,** 535-543.

44. Fischer, M. & Tatum, C. B. 1997. *Characteristics of Design-Relevant Constructability Knowledge*. Journal of Construction Engineering and Management, 123**,** 253-260.

45. Fischer, M. A. 1993. *Automating Constructibility Reasoning with a Geometrical and Topological Project Model*. Computing Systems in Engineering, 4**,** 179-192.

46. Gibb, A. G. F. 1999. *Off-Site Fabrication: Pre-Fabrication, Pre-Assembly and Modularisation,* Scotland, UK, Whittles Publishing.

47. Grosan, C. & Abraham, A. 2011. *Intelligent Systems: A Modern Approach,* Berlin, Germany, Springer.

48. Hardin, B. & McCool, D. 2015. *BIM and Construction Management: Proven Tools, Methods, and Workflows,* Indiana, US, Wiley.

49. Harris, J. R. 1981. *Organization of Building Standards: Systematic Techniques for Scope and Arrangement,* Washington, DC, US, National Bureau of Standards (NBS).

50. Heinrich, W. 2017. *German Construction Figures 2017*. Available: https://www.bauindustrie.de/zahlen-fakten/statistik-anschaulich/facts-and-charts-english/ [Accessed 15/11/2018].

51. Hierlein, E., Tillmann, M., Brandt, J., Röse, W., Schwerm, D. & Stöffler, J. 2009. *Betonfertigteile Im Geschoss- Und Hallenbau,* Bonn, Germany, FDB Fachvereinigung Deutscher Betonfertigteilbau.

52. Hjelseth, E. 2015. BIM-Based Model Checking (BMC). *Building Information Modeling: Applications and Practices.* Virginia, UK: American Society of Civil Engineers.

53. HOAI. 2013. *HOAI 2013* [Online]. Available: https://www.hoai.de/online/HOAI_2013/HOAI_2013.php [Accessed 03/11/2018].

54. Hossain, M. A., Abbott, E. L. S., Chua, D. K. H., Nguyen, T. Q. & Goh, Y. M. 2018. *Design-for-Safety Knowledge Library for BIM-Integrated Safety Risk Reviews*. Automation in Construction, 94**,** 290-302.

55. Huang, C.-h. & Krawczyk, R. *A Choice Model of Consumer Participatory Design for Modular Houses*. Predicting the future: Proceedings of the 25th eCAADe Conference 2007 Frankfurt am Main, Germany. 679-686.

56. Huang, C.-h., Krawczyk, R. J. & Schipporeit, G. *Mass Customizing Prefabricated Modular Housing by Internet-Aided Design*. Rhythm and Harmony in Digital Space: Proceedings of the 11th CAADRIA conference, 2006 Kumamoto, Japan. 203-208.

57. IronPython. 2019. *Ironpython, the Python Programming Language for The .Net Framework* [Online]. Available: https://ironpython.net/ [Accessed 25/06/2019].

58. Jaillon, L., Poon, C. S. & Chiang, Y. H. 2009. *Quantifying the Waste Reduction Potential of Using Prefabrication in Building Construction in Hong Kong*. Waste Management, 29**,** 309-320.

59. Jensen, P., Olofsson, T. & Johnsson, H. 2012. *Configuration through the Parameterization of Building Components*. Automation in Construction, 23**,** 1-8.

60. Ji, Y. & Leite, F. 2018. *Automated Tower Crane Planning: Leveraging 4-Dimensional BIM and Rule-Based Checking*. Automation in Construction, 93**,** 78-90.

61. Jiang, L. 2016. *A Constructability Review Ontology to Support Automated Rule Checking Leveraging Building Information Models.* Doctoral dissertation, The Pennsylvania State University.

62. Jiang, L. & Leicht, R. M. 2015. *Automated Rule-Based Constructability Checking: Case Study of Formwork*. Journal of Management in Engineering, 31**,** 10.

63. Kaner, I., Sacks, R., Kassian, W. & Quitt, T. 2008. *Case Studies of BIM Adoption for Precast Concrete Design by Mid-Sized Structural Engineering Firms*. Journal of Information Technology in Construction, 13**,** 303-323.

64. Kaufmannbausysteme. 2019. *Kaufmann Raummodule Für Bildung (Schulen/Kitas), Studentenheime, Wohnbau, Hotellerie, Sozialbau* [Online]. Available: https://kaufmannbausysteme.at/de/system-1 [Accessed 29/07/2019].

65. Khalili, A. 2013. *An IFC-Based Framework for Optimizing Level of Prefabrication in Industrialized Building Systems.* Doctoral dissertation, National University of Singapore.

66. Khalili, A. & Chua, D. K. H. 2012. *IFC-Based Framework to Move Beyond Individual Building Elements toward Configuring a Higher Level of Prefabrication*. Journal of Computing in Civil Engineering, 27**,** 243-253.

67. Kim, K. & Teizer, J. 2014. *Automatic Design and Planning of Scaffolding Systems Using Building Information Modeling*. Advanced Engineering Informatics, 28**,** 66-80.

68. Knaack, U., Chung-Klatte, S. & Hasselbach, R. 2012. *Prefabricated Systems: Principles of Construction,* Basel, Switzerland, Walter de Gruyter.

69. Koeble, W. & Zahn, A. 2013. *Die Neue HOAI 2013: Text - Erläuterung - Synopse*, Werner Verlag.

70. Kolarevic, B. & Klinger, K. 2008. *Manufacturing Material Effects: Rethinking Design and Making in Architecture,* New York, US, Routledge.

71. Lee, G., Parkү, J., Won, J., Park, H. K., Uhm, M. & Lee, Y. *Can Experience Overcome the Cognitive Challenges in Drawing-Based Design Review?* Proceedings of the 16th International Conference on Construction Applications of Virtual Reality "CONVR", 2013 Hong Kong, China. 144-156.

72. 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.

73. Lieberman, B. A. 2012. *Requirements for Rule Engines - Capture and Communication of Complex Business Rules* [Online]. Available: https://www.ibm.com/developerworks/library/os-rulesengines/index.html [Accessed 23/05/2017].

74. Luo, H. & Gong, P. 2015. *A BIM-Based Code Compliance Checking Process of Deep Foundation Construction Plans*. Journal of Intelligent & Robotic Systems, 79**,** 549-576.

75. Ma, Z., Wei, Z. & Liu, Z. *Ontology-Based Computerized Representation of Specifications for Construction Cost Estimation*. Proceedings of the CIB W78 2013: 30th International Conference, 2013 2013 Beijing, China. 707-716.

76. Manyika, J., Ramaswamy, S., Khanna, S., Sarrazin, H., Pinkus, G., Sethupathy, G. & Yaffe, A. 2015. *Digital America: A Tale of the Haves and Have-Mores*. Available: https://www.mckinsey.com [Accessed 05/11/2018].

77. MaxBögl. 2019. *Maxmodul* [Online]. Available: https://www.maxmodul.de/en/ [Accessed 29/07/2019].

78. Mekawy, M. & Petzold, F. *BIM-Based Model Checking in the Early Design Phases of Precast Concrete Structures*. Learning, Prototyping & Adapting: Proceedings of the 23rd CAADRIA conference, 2018 Beijing, China. 71-80.

79. Mini, F. 2016. *Entwicklung Eines Lod Konzepts Fur Digitale Bauwerksmodelle Von Brucken.* Masters dissertation, Technical University of Munich.

80. Möbert, J. 2018. *The German Housing Market in 2018*. Available: https://www.dbresearch.com/PROD/RPS_EN-PROD/RPSHOME.alias [Accessed 31/08/2018].

81. Mohamad, A., Hickethier, G., Hovestadt, V. & Gehbauer, F. *Use of Modularization in Design as a Strategy to Reduce Component Variety in One-Off Projects*. Proceedings of the 21th Annual Conference of the International Group for Lean Construction, 2013 Fortaleza, Brazil. 289-298.

82. Mokk, L. 1964. *Prefabricated Concrete for Industrial and Public Structures,* Budapest, Hungary, Akadémiai Kiadó.

83. Nagori, V. & Trivedi, B. 2014. *Types of Expert System: Comparative Study*. Asian Journal of Computer and Information Systems, 2**,** 20-33.

84. Nath, T., Attarzadeh, M., Tiong, R. L. K., Chidambaram, C. & Yu, Z. 2015. *Productivity Improvement of Precast Shop Drawings Generation through BIM-Based Process Re-Engineering*. Automation in Construction, 54**,** 54-68.

85. Nawari, N. O. 2012. *BIM Standard in Off-Site Construction*. Journal of Architectural Engineering, 18**,** 107-113.

86. Nelson Ford, F. 1985. *Decision Support Systems and Expert Systems: A Comparison*. Information & Management, 8**,** 21-26.

87. Nemetschek. 2018a. *Ifc4precast* [Online]. Available: https://www.precast-software.com/en/bim-process/ifc4precast/ [Accessed 22/11/2018].

88. Nemetschek. 2018b. *Planbar* [Online]. Available: http://www.precast-software.com/en/solutions/planbar/ [Accessed 07/09/2018].

89. Nepal, M. 2011. *Automated Extraction and Querying of Construction-Specific Design Features from a Building Information Model.* Doctoral dissertation, University of British Columbia.

90. O'Brien, E. J. 1995. *Reinforced and Prestressed Concrete Design,* Harlow, UK, Longman Scientific & Technical.

91. Ostrowska-Wawryniuk, K. & Nazar, K. *Generative BIM Automation Strategies for Prefabricated Multi-Family Housing Design*. Computing for a better tomorrow: Proceedings of the 36th eCAADe Conference, 2018 Lodz, Poland. 247-256.

92. Pastor, J. M., Balaguer, C., Rodriguez, F. J. & Diez, R. 2001. *Computer-Aided Architectural Design Oriented to Robotized Facade Panels Manufacturing*. Computer-Aided Civil and Infrastructure Engineering, 16**,** 216-227.

93. Patlakas, P., Livingstone, A. & Hairstans, R. *A BIM Platform for Offsite Timber Construction*. Real Time: Proceedings of the 33rd eCAADe Conference, 2015 Vienna, Austria. Vienna University of Technology, 597-604.

94. PCI Industry Handbook Committee 1975. *Architectural Precast Concrete Drafting Handbook,* New Jersey, US, Prentice-Hall.

95. PCI Industry Handbook Committee 2010. *Designing with Precast and Prestressed Concrete,* Illinois, US, Precast/Prestressed Concrete Institute PCI.

96. Piroozfar, P. A. E. & Piller, F. T. 2013. *Mass Customisation and Personalisation in Architecture and Construction,* New York, US, Routledge.

97. Plattenportal. 2018. *Ursprünge Des Industriellen Bauens - Häuser Am Fließband* [Online]. Available: https://www.jeder-qm-du.de/ueber-die-platte/detail/vorlaeufer-der-platte/ [Accessed 05/12/2018].

98. Polat, G. 2008. *Factors Affecting the Use of Precast Concrete Systems in the United States*. Journal of Construction Engineering and Management-Asce, 134**,** 169-178.

99. Pollack, S. L., Hicks, H. T. & Harrison, W. J. 1972. *Decision Tables: Theory and Practice,* New York, US, John Wiley & Sons Inc.

100. Poyatos, M., De Dios, J. J., Zangroniz, R. & Pastor, J. M. *Cad-Based Tool for Automated Panel Cutting of Prefabricated Facades*. Proceedings of the 28th International Symposium on Automation and Robotics in Construction (ISARC), 2011 Seoul, Korea. 1085-1090.

101. Preidel, C. & Borrmann, A. 2016. *Towards Code Compliance Checking on the Basis of a Visual Programming Language*. Journal of Information Technology in Construction (ITcon), 21**,** 402-421.

102. Preidel, C. & Borrmann, A. 2018. BIM-Based Code Compliance Checking. *In:* Borrmann, A., König, M., Koch, C. & Beetz, J. (eds.) *Building Information Modeling -*

Technology Foundations and Industry Practice. 1st edition ed. Cham, Switzerland: Springer nature Switzerland AG.

103. Puppe, F. 1993. *Systematic Introduction to Expert Systems: Knowledge Representations and Problem-Solving Methods,* Berlin Heidelberg, Springer-Verlag.

104. Rathnapala, T. 2009. *Incorporating Prefabrication Processes into Building Information Modeling.* Doctoral dissertation, University of Wolverhampton.

105. Retik, A. & Warszawski, A. 1994. *Automated Design of Prefabricated Building*. Building and Environment, 29**,** 421-436.

106. Sacks, R., Eastman, C., Lee, G. & Teicholz, P. 2018. *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers,* New Jersey, US, Wiley.

107. Sacks, R., Eastman, C. M. & Lee, G. 2004. *Parametric 3d Modeling in Building Construction with Examples from Precast Concrete*. Automation in Construction, 13**,** 291-312.

108. Sacks, R., Eastman, C. M., Lee, G. & Orndorff, D. 2005. *A Target Benchmark of the Impact of Three-Dimensional Parametric Modeling in Precast Construction*. Pci Journal, 50**,** 126-139.

109. Sacks, R., Kaner, I., Eastman, C. M. & Jeong, Y.-S. 2010. *The Rosewood Experiment — Building Information Modeling and Interoperability for Architectural Precast Facades*. Automation in Construction, 19**,** 419-432.

110. Said, H. M., Chalasani, T. & Logan, S. 2017. *Exterior Prefabricated Panelized Walls Platform Optimization*. Automation in Construction, 76**,** 1-13.

111. Shaohua, J. & Zheng, W. *A BIM-Based Code Checking Approach for Green Construction*. Proceedings of the International Conference on Construction and Real Estate Management 2017 Charleston, USA. 156-163.

112. Singh, M. M., Sawhney, A. & Borrmann, A. *Modular Coordination and BIM: Development of Rule Based Smart Building Components*. Proceedings of the Creative Construction Conference 2015 Krakow, Poland. 519-527.

113. Sobon, K. 2019. *Archilab* [Online]. Available: http://archi-lab.net [Accessed 30/06/2019].

114. Structureworks, L. L. C. 2018. *Structureworks* [Online]. Available: https://structureworks.net/structureworks/ [Accessed 07/09/2018].

115. Tan, X., Hammad, A. & Fazio, P. 2010. *Automated Code Compliance Checking for Building Envelope Design*. Journal of Computing in Civil Engineering, 24**,** 203-211.

116. Teicholz, P. M. 2013. *Labor-Productivity Declines in the Construction Industry: Causes and Remedies (a Second Look)* [Online]. Available: http://www.aecbytes.com/viewpoint/2013/issue_67.html [Accessed 30/11/2018].

117. Trimble. 2018. *Tekla Structures BIM Software* [Online]. Available: https://www.tekla.com/products/tekla-structures [Accessed 07/09/2018].

118. Weygant, R. S. 2011. *BIM Content Development: Standards, Strategies, and Best Practices,* New Jersey, US, Wiley.

119. Wright, R. & Fenves, S. 1979. *Analysis of Tentative Seismic Design Provisions for Buildings,* Washington, U.S, U.S. Government printing office.

120. Zahedi, A. & Petzold, F. *Utilization of Simulation Tools in Early Design Phases through Adaptive Detailing Strategies*. Learning, Prototyping & Adapting: Proceedings of the 23rd CAADRIA conference, 2018 Beijing, China. 11-20.

Appendix A – Rule-set for checking general precast considerations

Directory
1.1- Transport requirements
1.2- Similarity and grouping
1.3- Core placement for Bracing
1.4- Expansion joints
1.5- Corbel placement

1.1 Transport requirements

1.1.REF

	Rule 1	Rule 2
Transport without special license	Y	N
Transport with special license	N	Y
Go to table 1.1.ACT.1	X	-
Go to table 1.1.ACT.2	-	X

1.1.ACT.1

	Rule 1	Else
Length of individual component (m)	≤ 15.50	-
Width of individual component (m)	≤ 2.55	-
Height of individual component (m)	≤ 4.00	-
Display “Element sizing for transport is OK”	O	-
Display “Warning! Maximum dimensions are 15.50*2.55*4.00 m”	-	**S**

1.1.ACT.2

	Rule 1	Else
Length of individual component (m)	≤ 24.00	-
Width of individual component (m)	≤ 3.00	-
Height of individual component (m)	≤ 4.00	-
Display “Element sizing for transport is OK”	O	-
Display “Warning! Maximum dimensions are 24.00*3.00*4.00 m”	-	**S**

1.2 Similarity and grouping

1.2.ACT.1

	Rule 1	Rule 2	Rule 3	Rule 4
Are the typical floor heights equal?	Y	N	Y	N
Is the column spacing equal based on a 1.2m grid?	N	Y	Y	N
Display "Column spacing and floor heights check OK"	-	-	O	-
Display "Please consider creating evenly distributed columns based on a 1.2m"	R	-	-	R
Display "Please consider creating equal floor heights"	-	R	-	R

1.3 Core placement for Bracing

1.3.ACT.1

	Rule 1	Rule 2
Building core is centrally placed	N	Y
Display "Consider placing building core in the center to help carry horizontal loads economically"	R	-
Display "Central core placement is ideal"	-	O

1.4 Expansion joints

1.4.ACT.1

	Rule 1	Rule 2	Rule 3
Building length exceeds 60m	Y	Y	N
Expansion joints are placed	N	Y	-
Display "Building length exceeds 60m, an expansion joint should be placed"	S	-	-
Display "Expansion joint placement OK"	-	O	-

1.5 Corbel placement

1.5.ACT.1

	Rule 1	Rule 2	Rule 3	Rule 4
Column carry four beams	Y	N	N	N
Column carry three beams	N	Y	N	N
Column carry two beams	N	N	Y	N
Column carry one beams	N	N	N	Y
Display “Consider redesigning the column grid for each column to carry only one or two beams”	**R**	**R**	-	-
Display “Column Corbel loading is OK”	-	-	O	O

Appendix B – Rule-set for checking precast slabs considerations

Directory
2.1- Slab thickness
TT Slabs
Hollow core slabs
Element slabs

2.1 Slab thickness based on loads

2.1.REF

	Rule 1	Rule 2
Roof slab	Y	N
Typical slab	N	Y
Go to table 2.1.REF.a.1	X	-
Go to table 2.1.REF.b.1	-	X

2.1.REF.a.1

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
City	Berlin	Hamburg	Munich	Cologne	Frankfurt
Set snow load to (KN/m2)	1.95	1.95	2.47	0.49	0.6
Continued					

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
City	Essen	Stuttgart	Dortmund	Düsseldorf	Bremen
Set snow load to (KN/m2)	0.56	0.85	0.58	0.47	1.95
Go to table 2.1.REF.a.2					

2.1.REF.a.2

	Rule 1	Rule 2
Light floor cover (unused roof)	Y	N
Normal floor cover (Used roof)	N	Y
Set Floor cover load to 0.5 KN/m2	X	-
Set Floor cover load to 2.0 KN/m2	-	X
Go to 2.1.CLC.1		

2.1.CLC.1

Roof slab load (KN/m2)= Floor cover load + Snow load	
For TT slabs	Go to table 2.1.REF.1
For Hollow core slabs	Go to table 2.1.REF.2
For Element slabs	Go to table 2.1.REF.3

2.1.REF.b.1

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Building use	Residential Hotel	office	School Kindgart.	Exhibit space	Shopping
Set live load to (KN/m2)	1.5	2.0	3.0	5.0	5.0
Go to table 2.1.CLC.2					

2.1.CLC.2

Typical slab load (KN/m2)= Floor cover load (1.5 kN/M2) + Partitions load (2.5 kN/M2) + Live load	
For TT slabs	Go to table 2.1.REF.4
For Hollow core slabs	Go to table 2.1.REF.2
For Element slabs	Go to table 2.1.REF.3

2.1.REF.1

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Span (m)	≤ 7.5	7.5 < x ≤ 10.0	7.5 < x ≤ 10.0	10.0 < x ≤ 12.5	12.5 < x ≤ 15.0	12.5 < x ≤ 15.0	15.0 < x ≤ 17.5
Load (KN/m2)	-	1.0 ≤ y ≤ 3.5	3.5 < y ≤ 5.0	-	1.0 ≤ y ≤ 3.0	3.0 < y ≤ 5.0	1.0 ≤ y ≤ 2.5
Set Slab thickness (t) to (mm)	260	360	460	460	460	560	560
Display "Consider reducing the span to 25m"	-	-	-	-	-	-	-
Continued							

	Rule 8	Rule 9	Rule 10	Rule 11	Rule 12	Rule 13	Rule 14
Span (m)	15.0 < x ≤ 17.5	17.5 < x ≤ 20	17.5 < x ≤ 20	20.0 < x ≤ 22.5	20.0 < x ≤ 22.5	22.5 < x ≤ 25.0	x > 25
Load (KN/m2)	2.5 < y ≤ 5.0	1.0 ≤ y ≤ 3.0	3.0 < y ≤ 5.0	1.0 ≤ y ≤ 2.5	2.5 < y ≤ 5.0	-	-
Set Slab thickness (t) to (mm)	660	760	860	760	860	860	-
Display "Consider reducing the span to 25m"	-	-	-	-	-	-	**S**
Go to table 2.1.ACT.1							

2.1.REF.2

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Span (m)	x ≤ 3.0	x ≤ 3.0	x ≤ 3.0	3.0 < x ≤ 4.0	3.0 < x ≤ 4.0	3.0 < x ≤ 4.0	4.0 < x ≤ 5.0
Load (KN/m2)	1.0 ≤ y ≤ 5.0	5.0 < y ≤ 7.5	7.5 < y ≤ 10.0	1.0 ≤ y ≤ 4.5	4.5 < y ≤ 7.5	7.5 < y ≤ 10.0	1.0 ≤ y ≤ 3.0
Set Slab thickness (t) to (mm)	120	140	160	120	140	160	120
Display “Consider reducing the span to 12.5m”	-	-	-	-	-	-	-
Continued							

	Rule 8	Rule 9	Rule 10	Rule 11	Rule 12	Rule 13	Rule 14
Span (m)	4.0 < x ≤ 5.0	4.0 < x ≤ 5.0	4.0 < x ≤ 5.0	4.0 < x ≤ 5.0	5.0 < x ≤ 6.0	5.0 < x ≤ 6.0	5.0 < x ≤ 6.0
Load (KN/m2)	3.0 < x ≤ 4.5	4.5 < y ≤ 5.0	5.0 < y ≤ 7.5	7.5 < y ≤ 10.0	1.0 ≤ y ≤ 1.5	1.5 < y ≤ 2.0	2.0 < y ≤ 2.5
Set Slab thickness (t) to (mm)	140	160	180	220	120	140	160
Display “Consider reducing the span to 12.5m”	-	-	-	-	-	-	-
Continued							

	Rule 15	Rule 16	Rule 17	Rule 18	Rule 19	Rule 20	Rule 21
Span (m)	5.0 < x ≤ 6.0	5.0 < x ≤ 6.0	5.0 < x ≤ 6.0	5.0 < x ≤ 6.0	6.0 < x ≤ 7.5	6.0 < x ≤ 7.5	6.0 < x ≤ 7.5
Load (KN/m2)	2.5 < y ≤ 4.0	4.0 < y ≤ 5.0	5.0 < y ≤ 7.5	7.5 < y ≤ 10.0	1.0 ≤ y ≤ 2.0	2.0 < y ≤ 3.0	3.0 < y ≤ 4.5
Set Slab thickness (t) to (mm)	180	200	220	260	180	200	220
Display “Consider reducing the span to 12.5m”	-	-	-	-	-	-	-
Continued							

	Rule 22	Rule 23	Rule 24	Rule 25	Rule 26	Rule 27	Rule 28
Span (m)	6.0 < x ≤ 7.5	6.0 < x ≤ 7.5	7.5 < x ≤ 10.0	7.5 < x ≤ 10.0	7.5 < x ≤ 10.0	10 < x ≤ 12.5	10 < x ≤ 12.5
Load (KN/m2)	4.5 < y ≤ 7.5	7.5 < y ≤ 10.0	1.0 ≤ y ≤ 4.0	4.0 < y ≤ 5.0	5.0 < y ≤ 7.5	y ≤ 1.0	1.0 < y ≤ 1.5
Set Slab thickness (t) to (mm)	240	300	260	280	320	260	280
Display "Consider reducing the span to 12.5m"	-	-	-	-	-	-	-
Continued							

	Rule 29	Rule 30	Rule 31	Rule 32
Span (m)	10 < x ≤ 12.5	10 < x ≤ 12.5	10 < x ≤ 12.5	x > 12.5
Load (KN/m2)	1.5 < y ≤ 2.0	2.0 < y ≤ 2.5	y > 2.5	-
Set Slab thickness (t) to (mm)	300	320	-	-
Display "Consider reducing the span to 10.0m"	-	-	**S**	-
Display "Consider reducing the span to 12.5m"	-	-	-	**S**
Go to table 2.1.ACT.1				

2.1.REF.3

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Span (m)	x ≤ 3.0	x ≤ 3.0	x ≤ 3.0	3.0 < x ≤ 4.0	3.0 < x ≤ 4.0	3.0 < x ≤ 4.0	4.0 < x ≤ 5.0
Load (KN/m2)	1.0 ≤ y ≤ 7.5	7.5 < y ≤ 15.0	15.0 < y ≤ 25.0	1.0 ≤ y ≤ 7.5	7.5 < y ≤ 15.0	15.0 < y ≤ 25.0	1.0 ≤ y ≤ 7.5
Set Slab thickness (t) to (mm)	120	140	160	140	160	180	180
Display "Consider reducing the span to 7.5m"	-	-	-	-	-	-	-
Continued							

	Rule 8	Rule 9	Rule 10	Rule 11	Rule 12	Rule 13	Rule 14
Span (m)	4.0 < x ≤ 5.0	4.0 < x ≤ 5.0	5.0 < x ≤ 6.0	5.0 < x ≤ 6.0	6.0 < x ≤ 7.5	6.0 < x ≤ 7.5	6.0 < x ≤ 7.5
Load (KN/m2)	7.5 < y ≤ 15.0	15.0 < y ≤ 25.0	1.0 ≤ y ≤ 7.5	7.5 < y ≤ 25.0	1.0 ≤ y ≤ 3.5	3.5 < y ≤ 7.5	7.5 < y ≤ 15.0
Set Slab thickness (t) to (mm)	200	220	220	240	240	260	280
Display "Consider reducing the span to 7.5m"	-	-	-	-	-	-	-
Continued							

	Rule 15	Rule 16
Span (m)	6.0 < x ≤ 7.5	x > 7.5
Load (KN/m2)	15.0 < y ≤ 25.0	-
Set Slab thickness (t) to (mm)	300	-
Display "Consider reducing the span to 7.5m"	-	**S**
Go to table 2.1.ACT.1		

2.1.REF.4

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Span (m)	x ≤ 6.0	x ≤ 6.0	x ≤ 6.0	6.0 < x ≤ 7.5	6.0 < x ≤ 7.5	6.0 < x ≤ 7.5	7.5 < x ≤ 10.0
Load (KN/m2)	y ≤ 5.0	5.0 < y ≤ 15.0	15.0 < y ≤ 25.0	y ≤ 5.0	5.0 < y ≤ 15.0	15.0 < y ≤ 25.0	y ≤ 3.5
Set Slab thickness (t) to (mm)	320	350	400	420	450	500	420
Display "Consider reducing the span to 20.0m"	-	-	-	-	-	-	-
Continued							

	Rule 8	Rule 9	Rule 10	Rule 11	Rule 12	Rule 13	Rule 14
Span (m)	7.5 < x ≤ 10.0	7.5 < x ≤ 10.0	7.5 < x ≤ 10.0	10.0 < x ≤ 12.5	10.0 < x ≤ 12.5	10.0 < x ≤ 12.5	10.0 < x ≤ 12.5
Load (KN/m2)	3.5 < y ≤ 5.0	5.0 < y ≤ 15.0	15.0 < y ≤ 25.0	y ≤ 3.5	3.5 < y ≤ 5.0	5.0 < y ≤ 10.0	10.0 < y ≤ 15.0
Set Slab thickness (t) to (mm)	520	650	700	520	620	650	750
Display "Consider reducing the span to 20.0m"	-	-	-	-	-	-	-
Continued							

	Rule 15	Rule 16	Rule 17	Rule 18	Rule 19	Rule 20	Rule 21
Span (m)	10.0 < x ≤ 12.5	12.5 < x ≤ 15.0	12.5 < x ≤ 15.0	12.5 < x ≤ 15.0	12.5 < x ≤ 15.0	12.5 < x ≤ 15.0	12.5 < x ≤ 15.0
Load (KN/m2)	15.0 < y ≤ 25.0	y ≤ 3.5	3.5 < y ≤ 5.0	5.0 < y ≤ 10.0	10.0 < y ≤ 15.0	15.0 < y ≤ 20.0	20.0 < y ≤ 25.0
Set Slab thickness (t) to (mm)	800	620	720	750	850	900	1000
Display "Consider reducing the span to 20.0m"	-	-	-	-	-	-	-
Continued							

	Rule 22	Rule 23	Rule 24	Rule 25	Rule 26	Rule 27	Rule 28
Span (m)	15.0 < x ≤ 17.5	15.0 < x ≤ 17.5	15.0 < x ≤ 17.5	15.0 < x ≤ 17.5	17.5 < x ≤ 20.0	17.5 < x ≤ 20.0	x > 20.0
Load (KN/m2)	y ≤ 3.5	3.5 < y ≤ 5.0	5.0 < y ≤ 7.5	7.5 < y ≤ 10.0	y ≤ 5.0	5.0 < y ≤ 7.5	-
Set Slab thickness (t) to (mm)	720	820	850	950	920	950	-
Display "Consider reducing the span to 20.0m"	-	-	-	-	-	-	**S**
Go to table 2.1.ACT.1							

2.1.ACT.1

	Rule 1	Rule 2
Actual slab thickness = t	Y	N
Display "Warning! Slab thickness should be equal (t)"	-	**S**
Display "Slab thickness is OK"	O	-

Appendix C – Rule-set for checking precast beams considerations

Directory
3.1- Beam depth

3.1 Beam depth

3.1.CLC

Load (KN/m2)= Floor cover load (assume 1.5 KN/M2) + live load (Fetch from table 2.1.REF.b.1)
Go to table 3.1.Ref

3.1.REF

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Span of beam l (m)	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0
Span of slab L (m)	y ≤ 6.0	y ≤ 6.0	y ≤ 6.0	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5
Load (KN/m2)	z ≤ 5.0	5.0 < z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 3.0	3.0 < z ≤ 15.0	15.0 < z ≤ 20.0	20.0 < z ≤ 25.0
Set beam height (h) to (mm)	400	500	600	400	500	600	700
Display "Consider reducing the span to 10.0m"	-	-	-	-	-	-	-
Continued							

	Rule 8	Rule 9	Rule 10	Rule 11	Rule 12	Rule 13	Rule 14
Span of beam l (m)	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0
Span of slab L (m)	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5
Load (KN/m2)	z ≤ 2.0	2.0 < z ≤ 10.0	10.0 < z ≤ 15.0	15.0 < z ≤ 25.0	z ≤ 7.5	7.5 < z ≤ 10.0	10.0 < z ≤ 15.0
Set beam height (h) to (mm)	400	500	600	700	500	600	700
Display "Consider reducing the span to 10.0m"	-	-	-	-	-	-	-
Continued							

	Rule 15	Rule 16	Rule 17	Rule 18	Rule 19	Rule 20	Rule 21
Span of beam l (m)	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0
Span of slab L (m)	10.0 < y ≤ 12.5	12.5 < y ≤ 15.0	12.5 < y ≤ 15.0	12.5 < y ≤ 15.0	12.5 < y ≤ 15.0	12.5 < y ≤ 15.0	15.0 < y ≤ 17.5
Load (KN/m2)	15.0 < z ≤ 25.0	z ≤ 5.0	5.0 < z ≤ 10.0	10.0 < z ≤ 15.0	15.0 < z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 4.0
Set beam height (h) to (mm)	800	500	600	700	800	1000	500
Display "Consider reducing the span to 10.0m"	-	-	-	-	-	-	-
Continued							

	Rule 22	Rule 23	Rule 24	Rule 25	Rule 26	Rule 27	Rule 28
Span of beam l (m)	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0
Span of slab L (m)	15.0 < y ≤ 17.5	15.0 < y ≤ 17.5	15.0 < y ≤ 17.5	15.0 < y ≤ 17.5	17.5 < y ≤ 20.0	17.5 < y ≤ 20.0	17.5 < y ≤ 20.0
Load (KN/m2)	4.0 < z ≤ 7.5	7.5 < z ≤ 10.0	10.0 < z ≤ 15.0	15.0 < z ≤ 25.0	z ≤ 3.0	3.0 < z ≤ 5.0	5.0 < z ≤ 10.0
Set beam height (h) to (mm)	600	700	800	1000	500	600	700
Display "Consider reducing the span to 10.0m"	-	-	-	-	-	-	-
Continued							

	Rule 29	Rule 30	Rule 31	Rule 32	Rule 33	Rule 34	Rule 35
Span of beam l (m)	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0	x ≤ 5.0
Span of slab L (m)	17.5 < y ≤ 20.0	17.5 < y ≤ 20.0	20.0 < y ≤ 25.0	20.0 < y ≤ 25.0	20.0 < y ≤ 25.0	20.0 < y ≤ 25.0	20.0 < y ≤ 25.0
Load (KN/m2)	10.0 < z ≤ 15.0	15.0 < z ≤ 25.0	z ≤ 1.0	1.0 < z ≤ 3.0	3.0 < z ≤ 5.0	5.0 < z ≤ 10.0	10.0 < z ≤ 25.0
Set beam height (h) to (mm)	800	1000	500	600	700	800	1000
Display "Consider reducing the span to 10.0m"	-	-	-	-	-	-	-
Continued							

	Rule 36	Rule 37	Rule 38	Rule 39	Rule 40	Rule 41	Rule 42
Span of beam l (m)	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25
Span of slab L (m)	y ≤ 6.0	y ≤ 6.0	y ≤ 6.0	y ≤ 6.0	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5
Load (KN/m2)	z ≤ 4.0	4.0 < z ≤ 10.0	10.0 < z ≤ 15.0	15.0 < z ≤ 25.0	z ≤ 3.0	3.0 < z ≤ 7.5	7.5 < z ≤ 10.0
Set beam height (h) to (mm)	400	500	600	700	400	500	600
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 43	Rule 44	Rule 45	Rule 46	Rule 47	Rule 48	Rule 49
Span of beam l (m)	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25
Span of slab L (m)	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0
Load (KN/m2)	10.0 < z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 1.0	1.0 < z ≤ 4.0	4.0 < z ≤ 7.5	7.5 < z ≤ 15	15.0 < z ≤ 25.0
Set beam height (h) to (mm)	700	900	400	500	600	700	900
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 50	Rule 51	Rule 52	Rule 53	Rule 54	Rule 55	Rule 56
Span of beam l (m)	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25
Span of slab L (m)	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	12.5 < y ≤ 15.0	12.5 < y ≤ 15.0
Load (KN/m2)	z ≤ 3.0	3.0 < z ≤ 5.0	5.0 < z ≤ 10.0	10.0 < z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 2.0	2.0 < z ≤ 4.0
Set beam height (h) to (mm)	500	600	700	900	1000	500	600
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 57	Rule 58	Rule 59	Rule 60	Rule 61	Rule 62	Rule 63
Span of beam l (m)	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25
Span of slab L (m)	12.5 < y ≤ 15.0	12.5 < y ≤ 15.0	12.5 < y ≤ 15.0	12.5 < y ≤ 15.0	15.0 < y ≤ 17.5	15.0 < y ≤ 17.5	15.0 < y ≤ 17.5
Load (KN/m2)	4.0 < z ≤ 7.5	7.5 < z ≤ 15.0	15.0 < z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 1.0	1.0 < z ≤ 3.0	3.0 < z ≤ 5.0
Set beam height (h) to (mm)	700	900	1000	1200	500	600	700
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 64	Rule 65	Rule 66	Rule 67	Rule 68	Rule 69	Rule 70
Span of beam l (m)	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25
Span of slab L (m)	15.0 < y ≤ 17.5	15.0 < y ≤ 17.5	15.0 < y ≤ 17.5	17.5 < y ≤ 20.0	17.5 < y ≤ 20.0	17.5 < y ≤ 20.0	17.5 < y ≤ 20.0
Load (KN/m2)	5.0 < z ≤ 10.0	10.0 < z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 2.0	2.0 < z ≤ 4.0	4.0 < z ≤ 7.5	7.5 < z ≤ 15.0
Set beam height (h) to (mm)	900	1000	1200	600	700	900	1000
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 71	Rule 72	Rule 73	Rule 74	Rule 75	Rule 76	Rule 77
Span of beam l (m)	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	5.0 < x ≤ 6.25	6.25< x ≤ 7.50	6.25< x ≤ 7.50
Span of slab L (m)	17.5 < y ≤ 20.0	20.0 < y ≤ 25.0	20.0 < y ≤ 25.0	20.0 < y ≤ 25.0	20.0 < y ≤ 25.0	y ≤ 6.0	y ≤ 6.0
Load (KN/m2)	15.0 < z ≤ 25.0	z ≤ 3.0	3.0 < z ≤ 5.0	5.0 < z ≤ 10.0	10.0 < z ≤ 25.0	z ≤ 7.5	7.5 < z ≤ 15.0
Set beam height (h) to (mm)	1200	700	900	1000	1200	500	700
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 78	Rule 79	Rule 80	Rule 81	Rule 82	Rule 83	Rule 84
Span of beam l (m)	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50
Span of slab L (m)	y ≤ 6.0	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0
Load (KN/m2)	15.0 < z ≤ 25.0	z ≤ 5.0	5.0 < z ≤ 10.0	10.0 < z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 3.0	3.0 < z ≤ 7.5
Set beam height (h) to (mm)	800	500	700	800	1000	500	700
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 85	Rule 86	Rule 87	Rule 88	Rule 89	Rule 90	Rule 91
Span of beam l (m)	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50
Span of slab L (m)	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	12.5< y ≤ 15.0
Load (KN/m2)	7.5 < z ≤ 15.0	15.0 < z ≤ 25.0	z ≤ 2.0	2.0 < z ≤ 5.0	5.0 < z ≤ 10.0	10.0 < z ≤ 25.0	z ≤ 1.0
Set beam height (h) to (mm)	800	1000	500	700	800	1000	500
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 92	Rule 93	Rule 94	Rule 95	Rule 96	Rule 97	Rule 98
Span of beam l (m)	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50
Span of slab L (m)	12.5< y ≤ 15.0	12.5< y ≤ 15.0	12.5< y ≤ 15.0	12.5< y ≤ 15.0	15.0< y ≤ 17.5	15.0< y ≤ 17.5	15.0< y ≤ 17.5
Load (KN/m2)	1.0 < z ≤ 3.0	3.0 < z ≤ 7.5	7.5 < z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 2.0	2.0 < z ≤ 5.0	5.0 < z ≤ 15.0
Set beam height (h) to (mm)	700	800	1000	1200	700	800	1000
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 99	Rule 100	Rule 101	Rule 102	Rule 103	Rule 104	Rule 105
Span of beam l (m)	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50
Span of slab L (m)	15.0< y ≤ 17.5	17.5< y ≤ 20.0	17.5< y ≤ 20.0	17.5< y ≤ 20.0	17.5< y ≤ 20.0	17.5< y ≤ 20.0	20.0< y ≤ 25.0
Load (KN/m2)	15.0 < z ≤ 25.0	z ≤ 1.0	1.0 < z ≤ 4.0	4.0 < z ≤ 10.0	10 < z ≤ 20.0	20 < z ≤ 25.0	z ≤ 2.0
Set beam height (h) to (mm)	1200	700	800	1000	1200	1400	800
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 106	Rule 107	Rule 108	Rule 109	Rule 110	Rule 111	Rule 112
Span of beam l (m)	6.25< x ≤ 7.50	6.25< x ≤ 7.50	6.25< x ≤ 7.50	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75
Span of slab L (m)	20.0< y ≤ 25.0	20.0< y ≤ 25.0	20.0< y ≤ 25.0	y ≤ 6.0	y ≤ 6.0	y ≤ 6.0	y ≤ 6.0
Load (KN/m2)	2.0 < z ≤ 7.5	7.5 < z ≤ 15.0	15.0 < z ≤ 25.0	z ≤ 5.0	5.0 < z ≤ 15.0	15.0 < z ≤ 20.0	20.0 < z ≤ 25.0
Set beam height (h) to (mm)	1000	1200	1400	600	800	900	1000
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 113	Rule 114	Rule 115	Rule 116	Rule 117	Rule 118	Rule 119
Span of beam l (m)	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75
Span of slab L (m)	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0
Load (KN/m2)	z ≤ 4.0	4.0 < z ≤ 10.0	10 < z ≤ 15.0	15 < z ≤ 25.0	z ≤ 3.0	3.0 < z ≤ 7.5	7.5 < z ≤ 10.0
Set beam height (h) to (mm)	600	800	900	1000	600	800	900
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 120	Rule 121	Rule 122	Rule 123	Rule 124	Rule 125	Rule 126
Span of beam l (m)	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75
Span of slab L (m)	7.5 < y ≤ 10.0	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	12.5< y ≤ 15.0
Load (KN/m2)	10.0 < z ≤ 25.0	z ≤ 2.0	2.0 < z ≤ 5.0	5.0 < z ≤ 7.5	7.5 < z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 1.0
Set beam height (h) to (mm)	1000	600	800	900	1000	1200	600
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 127	Rule 128	Rule 129	Rule 130	Rule 131	Rule 132	Rule 133
Span of beam l (m)	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75
Span of slab L (m)	12.5< y ≤ 15.0	12.5< y ≤ 15.0	12.5< y ≤ 15.0	12.5< y ≤ 15.0	12.5< y ≤ 15.0	15.0< y ≤ 17.5	15.0< y ≤ 17.5
Load (KN/m2)	1.0 < z ≤ 4.0	4.0 < z ≤ 5.0	5.0< z ≤ 15.0	15.0< z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 3.0	3.0 < z ≤ 5.0
Set beam height (h) to (mm)	800	900	1000	1200	1400	800	900
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 134	Rule 135	Rule 136	Rule 137	Rule 138	Rule 139	Rule 140
Span of beam l (m)	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75
Span of slab L (m)	15.0< y ≤ 17.5	15.0< y ≤ 17.5	15.0< y ≤ 17.5	17.5< y ≤ 20.0	17.5< y ≤ 20.0	17.5< y ≤ 20.0	17.5< y ≤ 20.0
Load (KN/m2)	5.0 < z ≤ 10.0	10.0< z ≤ 15.0	15.0< z ≤ 25.0	z ≤ 2.0	2.0 < z ≤ 4.0	4.0 < z ≤ 7.5	7.5 < z ≤ 10.0
Set beam height (h) to (mm)	1000	1200	1400	800	900	1000	1200
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 141	Rule 142	Rule 143	Rule 144	Rule 145	Rule 146	Rule 147
Span of beam l (m)	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	7.50< x ≤ 8.75	8.75< x ≤ 10.0	8.75< x ≤ 10.0
Span of slab L (m)	17.5< y ≤ 20.0	20.0< y ≤ 25.0	20.0< y ≤ 25.0	20.0< y ≤ 25.0	20.0< y ≤ 25.0	y ≤ 6.0	y ≤ 6.0
Load (KN/m2)	10.0< z ≤ 20.0	z ≤ 2.0	2.0 < z ≤ 5.0	5.0 < z ≤ 10.0	10.0< z ≤ 15.0	z ≤ 7.5	7.5 < z ≤ 10.0
Set beam height (h) to (mm)	1400	900	1000	1200	1400	700	800
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 148	Rule 149	Rule 150	Rule 151	Rule 152	Rule 153	Rule 154
Span of beam l (m)	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0
Span of slab L (m)	y ≤ 6.0	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	6.0 < y ≤ 7.5	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0
Load (KN/m2)	10.0 < z ≤ 25.0	z ≤ 5.0	5.0 < z ≤ 7.5	7.5 < z ≤ 20.0	20.0 < z ≤ 25.0	z ≤ 3.0	3.0 < z ≤ 5.0
Set beam height (h) to (mm)	1000	700	800	1000	1200	700	800
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 155	Rule 156	Rule 157	Rule 158	Rule 159	Rule 160	Rule 161
Span of beam l (m)	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0
Span of slab L (m)	7.5 < y ≤ 10.0	7.5 < y ≤ 10.0	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5	10.0 < y ≤ 12.5
Load (KN/m2)	5.0< z ≤ 15.0	15.0 < z ≤ 25.0	z ≤ 2.0	2.0 < z ≤ 4.0	4.0 < z ≤ 10.0	10.0< z ≤ 15.0	15.0 < z ≤ 25.0
Set beam height (h) to (mm)	1000	1200	700	800	1000	1200	1400
Display “Consider reducing the span to 10.0m”	-	-	-	-	-	-	-
Continued							

	Rule 162	Rule 163	Rule 164	Rule 165	Rule 166	Rule 167	Rule 168
Span of beam l (m)	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0
Span of slab L (m)	12.5< y ≤ 15.0	12.5< y ≤ 15.0	12.5< y ≤ 15.0	12.5< y ≤ 15.0	12.5< y ≤ 15.0	15.0< y ≤ 17.5	15.0< y ≤ 17.5
Load (KN/m2)	z ≤ 1.0	1.0 < z ≤ 3.0	3.0 < z ≤ 7.5	7.5 < z ≤ 10.0	10.0< z ≤ 20.0	z ≤ 2.0	2.0 < z ≤ 5.0
Set beam height (h) to (mm)	700	800	1000	1200	1400	800	1000
Display "Consider reducing the span to 10.0m"	–	–	–	–	–	–	–
Continued							

	Rule 169	Rule 170	Rule 171	Rule 172	Rule 173	Rule 174	Rule 175
Span of beam l (m)	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0	8.75< x ≤ 10.0
Span of slab L (m)	15.0< y ≤ 17.5	15.0< y ≤ 17.5	17.5< y ≤ 20.0	17.5< y ≤ 20.0	17.5< y ≤ 20.0	17.5< y ≤ 20.0	20.0< y ≤ 25.0
Load (KN/m2)	5.0 < z ≤ 7.5	7.5 < z ≤ 15.0	z ≤ 1.0	1.0 < z ≤ 5.0	5.0 < z ≤ 7.5	7.5 < z ≤ 15.0	z ≤ 3.0
Set beam height (h) to (mm)	1200	1400	800	1000	1200	1400	1000
Display "Consider reducing the span to 10.0m"	–	–	–	–	–	–	–
Continued							

	Rule 176	Rule 177	Rule 178
Span of beam l (m)	8.75< x ≤ 10.0	8.75< x ≤ 10.0	x>10
Span of slab L (m)	20.0< y ≤ 25.0	20.0< y ≤ 25.0	–
Load (KN/m2)	3.0 < z ≤ 5.0	5.0 < z ≤ 7.5	–
Set beam height (h) to (mm)	1200	1400	–
Display "Consider reducing the span to 10.0m"	–	–	**S**
Go to table 3.1.ACT.1			

3.1.ACT.1

	Rule 1	Rule 2
Actual beam depth = h	Y	N
Display "Warning! Beam depth should be equal (h)"	-	S
Display "Beam depth is OK"	O	-

Appendix D – Rule-set for checking precast façade panels considerations

Directory
4.1- Vertical joint spacing and joint width
4.2- Panel’s Layers
4.3- Façade panel and window repetition

4.1 Vertical joint spacing and joint width

4.1.REF

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Else
Spacing between joints (m)	≤ 2.0	2.0 < x ≤ 3.5	3.5 < x ≤ 5.0	5.0 < x ≤ 6.5	6.5 < x ≤ 8.0	-
Set joint width [w] to (mm)	15	20	25	30	35	-
Display “Please reduce joint spacing to 8.0 m or less”	-	-	-	-	-	S
Go to table 4.1.ACT.1						

4.1.ACT.1

	Rule 1	Rule 2
Actual joint width ≥ w	Y	N
Actual joint width < w	N	Y
Display “Warning! Minimum joint width is [w]”	-	S
Display “Joint width is OK”	O	-

4.2 Panel's layers

4.2.ACT.1

Facing layer

	Rule 1	Rule 2	Rule 3
Facing layer thickness (mm)	< 70	> 100	70 ≤ x ≤ 100
Display "Minimum thickness of facing layer is 70 mm"	R	-	-
Display "Maximum thickness of facing layer is 100 mm"	-	R	-
Display "Facing layer thickness is OK"	-	-	O
Go to table 4.2.ACT.2			

4.2.ACT.2

Insulation layer

	Rule 1	Rule 2	Rule 3
Insulation layer thickness (mm)	< 60	> 100	60 ≤ x ≤ 100
Display "Minimum thickness of insulation layer is 40 mm"	R	-	-
Display "Maximum thickness of insulation layer is 100 mm"	-	R	-
Display "Insulation layer thickness is OK"	-	-	O
Go to table 4.2.ACT.3			

4.2.ACT.3

Load bearing layer

	Rule 1	Rule 2	Rule 3
Loadbearing layer thickness (mm)	< 140	> 200	140 ≤ x ≤ 200
Display “Minimum thickness of Load bearing layer is 140 mm”	R	-	-
Display “Maximum thickness of Load bearing layer is 200 mm”	-	R	-
Display “Load bearing layer thickness is OK”	-	-	O

4.3 Façade panel and window repetition

4.3.ACT.1

	Rule 1	Rule 2	Rule 3	Rule 4
Façade windows are of equal size	Y	N	N	Y
Façade panels are of equal size	N	Y	N	Y
Display "Consider making façade panels of equal size"	R	-	R	-
Display "Consider making façade windows of equal size"	-	R	R	-
Display "Façade panels' uniform sizing OK"	O	O	-	O
Display "Façade windows' uniform sizing OK"	O	-	-	O

Appendix E – Rule-set for checking precast columns considerations

Directory
5.1- Column rectangular cross-section
5.2- Column uniform cross-section
5.3- Column positioning to avoid thermal bridges
5.4- Column cross-section size

5.1- Column rectangular cross-section

5.1.ACT.1

	Rule 1	Rule 2
Column has a rectangular cross-section	Y	N
Display "Column cross-section OK"	O	-
Display "Consider designing columns with rectangular cross-sections"	-	**R**

5.2- Column uniform cross-section

5.2.ACT.1

	Rule 1	Rule 2
Column has a uniform cross-section across floors	Y	N
Display "Column cross-section across floors OK"	O	-
Display "Consider making columns with uniform cross section across floors"	-	**R**

5.3- Column positioning to avoid thermal bridges

(Position in relation to Façade panels)

5.3.ACT.1

	Rule 1	Rule 2	Rule 3	Rule 4
Column is entirely in between façade panels	Y	N	N	N
Column is in between the load bearing and insulation layers	N	Y	N	N
Column is entirely behind the façade panels	N	N	Y	N
Column is in between the load bearing layer and behind the insulation layer	N	N	N	Y
Display "Column position may cause thermal bridges, consider moving the column"	**R**	**R**	-	-
Display "Column positioning OK"	-	-	O	O

5.4- Column cross-section size

5.4.CLC.1

Effective area (m2)= Span of beam l (m) * Span of slab L (m)
Go to 5.4.CLC.2

5.4.CLC.2

Loads (KN/m2) = ∑ Roof slab loads (cover + snow) + ((∑ typical slab loads (live + cover))*no of floors)	
For edge columns	Go to table 5.4.REF.1
For middle columns	Go to table 5.4.REF.2

5.4.REF.1

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Effective area (m2)	x ≤ 100.0	100.0 < x ≤ 125.0	100.0 < x ≤ 125.0	125.0 < x ≤ 150.0	125.0 < x ≤ 150.0	150.0 < x ≤ 175.0	150.0 < x ≤ 175.0
Load (KN/m2)	-	y ≤ 20.0	20.0 < y ≤ 30.0	y ≤ 15.0	15.0 < y ≤ 30.0	y ≤ 10.0	10.0 < y ≤ 20.0
Set column cross section (b*h) to (mm)	300*400	300*400	400*400	300*400	400*400	300*400	400*400
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 8	Rule 9	Rule 10	Rule 11	Rule 12	Rule 13	Rule 14
Effective area (m2)	150.0 < x ≤ 175.0	175.0 < x ≤ 200.0	175.0 < x ≤ 200.0	175.0 < x ≤ 200.0	175.0 < x ≤ 200.0	200.0 < x ≤ 225.0	200.0 < x ≤ 225.0
Load (KN/m2)	20.0 < y ≤ 30.0	y ≤ 10.0	10.0 < y ≤ 15.0	15.0 < y ≤ 25.0	25.0 < y ≤ 30.0	y ≤ 7.5	7.5.0 < y ≤ 15.0
Set column cross section (b*h) to (mm)	400*500	300*400	400*400	400*500	500*500	300*400	400*400
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 15	Rule 16	Rule 17	Rule 18	Rule 19	Rule 20	Rule 21
Effective area (m2)	200.0 < x ≤ 225.0	200.0 < x ≤ 225.0	225.0 < x ≤ 250.0	225.0 < x ≤ 250.0	225.0 < x ≤ 250.0	225.0 < x ≤ 250.0	225.0 < x ≤ 250.0
Load (KN/m2)	15.0 < y ≤ 20.0	20.0 < y ≤ 30.0	y ≤ 5.0	5.0 < y ≤ 10.0	10.0 < y ≤ 15.0	15.0 < y ≤ 25.0	25.0 < y ≤ 30.0
Set column cross section (b*h) to (mm)	400*500	500*500	300*400	400*400	400*500	500*500	500*600
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 22	Rule 23	Rule 24	Rule 25	Rule 26	Rule 27	Rule 28
Effective area (m2)	250.0 < x ≤ 275.0	250.0 < x ≤ 275.0	250.0 < x ≤ 275.0	250.0 < x ≤ 275.0	250.0 < x ≤ 275.0	275.0 < x ≤ 300.0	275.0 < x ≤ 300.0
Load (KN/m2)	y ≤ 3.0	3.0 < y ≤ 10.0	10.0 < y ≤ 15.0	15.0 < y ≤ 20.0	20.0 < y ≤ 30.0	y ≤ 7.5	7.5 < y ≤ 10.0
Set column cross section (b*h) to (mm)	300*400	400*400	400*500	500*500	500*600	400*400	400*500
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 29	Rule 30	Rule 31	Rule 32	Rule 33	Rule 34	Rule 35
Effective area (m2)	275.0 < x ≤ 300.0	275.0 < x ≤ 300.0	275.0 < x ≤ 300.0	300.0 < x ≤ 325.0	300.0 < x ≤ 325.0	300.0 < x ≤ 325.0	300.0 < x ≤ 325.0
Load (KN/m2)	10.0 < y ≤ 20.0	20.0 < y ≤ 25.0	25.0 < y ≤ 30.0	y ≤ 5.0	5.0 < y ≤ 10.0	10.0 < y ≤ 15.0	15.0 < y ≤ 20.0
Set column cross section (b*h) to (mm)	500*500	500*600	600*600	400*400	400*500	500*500	500*600
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 36	Rule 37
Effective area (m2)	300.0 < x ≤ 325.0	x > 325
Load (KN/m2)	20.0 < y ≤ 30.0	-
Set column cross section (b*h) to (mm)	600*600	-
Display "Consider reducing the span"	-	**S**
Go to table 5.4.ACT.1		

5.4.REF.2

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
Effective area (m2)	x ≤ 75.0	75.0 < x ≤ 100.0	75.0 < x ≤ 100.0	75.0 < x ≤ 100.0	100.0 < x ≤ 125.0	100.0 < x ≤ 125.0	100.0 < x ≤ 125.0
Load (KN/m2)	-	y ≤ 15.0	15.0 < y ≤ 25.0	25.0 < y ≤ 30.0	y ≤ 7.5	7.5.0 < y ≤ 15.0	15.0 < y ≤ 25.0
Set column cross section (b*h) to (mm)	300*400	300*400	400*400	400*500	300*400	400*400	400*500
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 8	Rule 9	Rule 10	Rule 11	Rule 12	Rule 13	Rule 14
Effective area (m2)	100.0 < x ≤ 125.0	125.0 < x ≤ 150.0	125.0 < x ≤ 150.0	125.0 < x ≤ 150.0	125.0 < x ≤ 150.0	125.0 < x ≤ 150.0	150.0 < x ≤ 175.0
Load (KN/m2)	25.0 < y ≤ 30.0	y ≤ 3.0	3.0 < y ≤ 10.0	10.0 < y ≤ 15.0	15.0 < y ≤ 20.0	20.0 < y ≤ 30.0	y ≤ 5.0
Set column cross section (b*h) to (mm)	500*500	300*400	400*400	400*500	500*500	500*600	400*400
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 15	Rule 16	Rule 17	Rule 18	Rule 19	Rule 20	Rule 21
Effective area (m2)	150.0 < x ≤ 175.0	150.0 < x ≤ 175.0	150.0 < x ≤ 175.0	150.0 < x ≤ 175.0	175.0 < x ≤ 200.0	175.0 < x ≤ 200.0	175.0 < x ≤ 200.0
Load (KN/m2)	5.0 < y ≤ 10.0	10.0 < y ≤ 15.0	15.0 < y ≤ 20.0	20.0 < y ≤ 30.0	y ≤ 3.0	3.0 < y ≤ 7.5	7.5 < y ≤ 10.0
Set column cross section (b*h) to (mm)	400*500	500*500	500*600	600*600	400*400	400*500	500*500
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 22	Rule 23	Rule 24	Rule 25	Rule 26	Rule 27	Rule 28
Effective area (m2)	175.0 < x ≤ 200.0	175.0 < x ≤ 200.0	175.0 < x ≤ 200.0	200.0 < x ≤ 225.0	200.0 < x ≤ 225.0	200.0 < x ≤ 225.0	200.0 < x ≤ 225.0
Load (KN/m2)	10.0 < y ≤ 20.0	20.0 < y ≤ 25.0	25.0 < y ≤ 30.0	y ≤ 5.0	5.0 < y ≤ 10.0	10.0 < y ≤ 15.0	15.0 < y ≤ 20.0
Set column cross section (b*h) to (mm)	500*600	600*600	600*800	400*500	500*500	500*600	600*600
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 29	Rule 30	Rule 31	Rule 32	Rule 33	Rule 34	Rule 35
Effective area (m2)	200.0 < x ≤ 225.0	225.0 < x ≤ 250.0	225.0 < x ≤ 250.0	225.0 < x ≤ 250.0	225.0 < x ≤ 250.0	225.0 < x ≤ 250.0	225.0 < x ≤ 250.0
Load (KN/m2)	20.0 < y ≤ 30.0	y ≤ 3.0	3.0 < y ≤ 7.5	7.5 < y ≤ 10.0	10.0 < y ≤ 15.0	15.0 < y ≤ 25.0	25.0 < y ≤ 30.0
Set column cross section (b*h) to (mm)	600*800	400*500	500*500	500*600	600*600	600*800	700*800
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 36	Rule 37	Rule 38	Rule 39	Rule 40	Rule 41	Rule 42
Effective area (m2)	250.0 < x ≤ 275.0	250.0 < x ≤ 275.0	250.0 < x ≤ 275.0	250.0 < x ≤ 275.0	250.0 < x ≤ 275.0	275.0 < x ≤ 300.0	275.0 < x ≤ 300.0
Load (KN/m2)	y ≤ 5.0	5.0 < y ≤ 10.0	10.0 < y ≤ 15.0	15.0 < y ≤ 20.0	20.0 < y ≤ 30.0	y ≤ 5.0	5.0 < y ≤ 7.5
Set column cross section (b*h) to (mm)	500*500	500*600	600*600	600*800	700*800	500*500	500*600
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 43	Rule 44	Rule 45	Rule 46	Rule 47	Rule 48	Rule 49
Effective area (m2)	275.0 < x ≤ 300.0	275.0 < x ≤ 300.0	275.0 < x ≤ 300.0	275.0 < x ≤ 300.0	300.0 < x ≤ 325.0	300.0 < x ≤ 325.0	300.0 < x ≤ 325.0
Load (KN/m2)	7.5 < y ≤ 10.0	10.0 < y ≤ 20.0	20.0 < y ≤ 25.0	25.0 < y ≤ 30.0	y ≤ 3.0	3.0 < y ≤ 5.0	5.0 < y ≤ 10.0
Set column cross section (b*h) to (mm)	600*600	600*800	700*800	800*800	500*500	500*600	600*600
Display "Consider reducing the span"	-	-	-	-	-	-	-
Continued							

	Rule 50	Rule 51	Rule 52	Rule 53
Effective area (m2)	300.0 < x ≤ 325.0	300.0 < x ≤ 325.0	300.0 < x ≤ 325.0	x > 325
Load (KN/m2)	10.0 < y ≤ 15.0	15.0 < y ≤ 20.0	20.0 < y ≤ 30.0	-
Set column cross section (b*h) to (mm)	600*800	700*800	800*800	-
Display "Consider reducing the span"	-	-	-	**S**
Go to table 5.4.ACT.1				

5.4.ACT.1

	Rule 1	Rule 2
Actual column cross section = b*h	Y	N
Display "Warning! Column cross section should be equal (b*h)"	-	**S**
Display "Column cross section is OK"	O	-

Appendix F – List of questions posed in the exploratory interviews

1. What is the typical process in precast concrete projects, from design to precast concrete design and detailing? What type of drawings and documentation are typically developed?

2. In your own experience, how often do communication between precast experts/ structural engineers and architects occur in the early design stages regarding precast requirements happen in practice?

3. What are the architectural design factors that can benefit from expert precast knowledge input? What design modifications may precast contractors propose?

4. What are the design conditions that may affect precast planning in the architectural design phase? Do you think the following parameters are relevant? And How?

- Component sizing
- Repetition of components vs variation of size and shape
- Component spacing
- Modular layout of components (Equal bay sizes)
- Floor heights
- Irregularity in the shape of the components (Curved walls – Tapered columns)

5. What are the types of slabs – walls – beams – columns most frequently used in the German market?

6. How are prefabricated façade walls typically layered and subdivided? What to prioritize more: Identical units or bigger parts?

7. To ensure optimum costs and structural integrity in precast practice, what are the general, rule-of-thumb, design limitations regarding the following?

- Window and openings’ dimensions in the external façade
- Slab openings
- Limitations due to transport
- Limitations due to lifting and erection on-site
- Production in the workshop

8. Can you provide any final comments or remarks regarding the research?