Radboud University Nijmegen

Faculty of Science

Conceptualisation of problem-solution patterns

Knowledge sharing across professional boundaries

Master's Thesis

Supervisor:
Stijn Hoppenbrouwers

Author:
Joris Sunnotel

Second Supervisor:
Eelco Herder

July 2021

Abstract

A problem-solution chain is a combination of problem-solution links, a specific development path from a high-level business problem or need to lower-level solutions. All chains combined form the pattern language of problem-solution links. In this exploratory study, we elaborate on how to represent and elicit such chains into a knowledge system, in order to explicate the patterns and the modelling discussion that leads to problem-solution patterns. This can help to effectively communicate complex structures across professional knowledge boundaries. By doing so, this could possibly increase Business-IT Alignment through effective communication in and between organisations.
We have established a syntactic and semantic structure for problem-solution chains. We analysed different modelling situations that resulted in problem-solution chains, by applying Focused Conceptualisations (FoCons). The FoCons describe the conversation that led to a conceptual model of linked problems and solutions. The FoCon is the beginnings of a design for a dialogue game to elicit problem-solution chains from knowledge experts. In this dialogue game, the FoCon supports the modellers with defining solutions to a particular problem, creating problem-solution links. The discussion and outcome of the FoCons are stored in a proof-of-concept knowledge system that in the future should transform the FoCon output in visualisations of problem-solution chains and store the chains in the repository. We evaluated our proposed concepts by applying three case examples to our proposed concepts.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Today's organizations are forced to build cross-departmental and cross-organisational collaborations that involve significant investments in information technology. Research indicates that IT investments can enhance business performance given specific conditions, for instance, high Business-IT Alignment [1]. Improving the alignment of business and IT is of vital concern to contemporary leaders because it can increase the effectiveness of IT investments [2]. There is a significant amount of literature on alignment strategies to achieve the full potential of technology in organisations. However, the vast majority of alignment approaches are resource-intensive and therefore not feasible for all organisations [3].

In the increasingly complex collaborative environment organisations are forced to work with third-party companies and individual external workers. Even though organisations try to bridge this gap with dedicated employee roles, communication remains an important obstacle in these collaborations [1]. Professionals from different backgrounds experience noise in conversations, resulting in distinct perceptions between groups. While collaboration asks for a mutual understanding of subjects to achieve a certain goal or task [4].

This thesis is written on behalf of the research group Model-Based Information System of HAN University of Applied Sciences. The research group addressed the need for a framework structuring high-level discussions about innovation, abstract enough to be understood by all, specific enough to provide meaning to all. This discussion concerns solutions to a problem and problems solved by a solution. This study explores the assertion made by the HAN research group, that abstract reasoning about problems and solutions is the most practical way for distinct profession groups to innovate by means of collaboration. As Stijn Hoppenbrouwers, professor of the research group, states: "the basic idea is that people, in particular when discussing an application of some approach or technique that may be a solution to some problem, are inclined to think and talk in terms of solutions to problems. This is also the core of Design Pattern thinking. My belief, as an applied researcher that spends much time talking to non-IT stakeholders interested in applying some IT solution, it would be helpful to use 'Problem-Solution Chains' in exploratory conversations concerning Business-IT Alignment, or similar activities."

1.2 Solution

In the previous section, we briefly addressed that there are numerous approaches to increase effective communication in organisations through a shared language. We also noted that these techniques are resource-intensive and not feasible in all contexts. This study aims to explore possibilities to develop a tool of basic cognitive complexity and low cost that constitutes a shared language for effective communication across professional knowledge boundaries. We aim to do so through the conceptualisation of problem-solution chains in organisations. A problem-solution chains links high-level business problems to concrete lower-level solutions. Moreover, the chains include the underlying modelling discussion that resulted in the drawn concepts. The representations of such chains aim to support communication between the business and IT, within and across different organisations. This solution creates a common language for professionals from different backgrounds and expertise. Our solution shows how to represent problem-solution chains and proposes a method to elicit chains from knowledge experts. Our solution has a theoretical basis and is evaluated by three case examples. To develop the solution the following research questions are answered:

RQ1: How to represent problem-solution chains?

RQ2: How to elicit problem-solution chains from knowledge experts?

The theoretical framework that is the structure that supports the solution of this study is depicted in 1.1. Not illustrated in this figure, but part of Business-IT Alignment is Capability Thinking. Moreover, KAOS is a specific Requirements Engineering technique we take great interest in during this study. We elaborate on the different theories in Chapter 3 and how the theories relate to problem-solution chains.

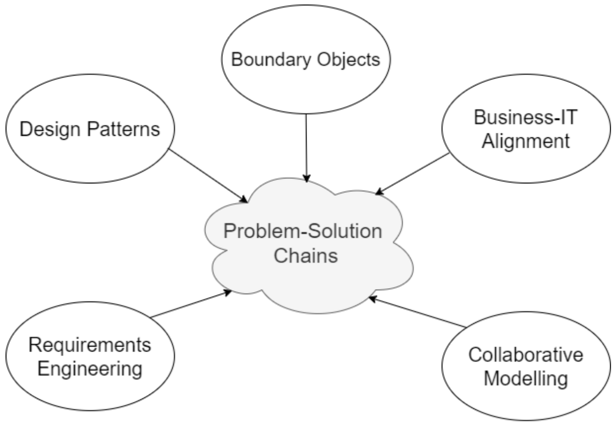


Figure 1.1: Theoretical Framework

1.3 Significance of the Research

As discussed in section 1.1, little research is performed concerning a simple language that bridges the gap between the business and IT vocabulary. The existing methods take a resource-intensive approach not applicable in all situations. The concepts in this thesis are developed taking a human-centric rather than a knowledge-centric approach, reasoning about abstract problems and solutions rather than return on investment and technical specifications. If organisations are provided with a language understood by all stakeholders, the improved communication leads to more effective collaboration between distinct profession groups, vital in modern enterprises. The findings of this study may in the future be used for research projects within the research group Model-Based Information System of the HAN University of Applied Sciences.

1.4 Structure of the Thesis

In Chapter 2 we elaborate on the methodology applied in our study, during the creation of our concepts. Chapter 3 provides the theoretical background that is the basis of our developed concepts. We set the context and reasons for this study. Finally, we look at different fields that influenced our work and from which we derived the representation of our concepts. Chapter 4 presents the problem-solution chains and the proof-of-concept knowledge system that implements the chains. Chapter 5 proposes a method for the elicitation of problem-solution chains from knowledge experts. Then, in Chapter 6 the proposed concepts are evaluated by means of three case examples. Chapter 7 contains the discussion of our research results and limitations to our study. Finally, Chapter 8 concludes our study and presents possible future research directions.

Chapter 2

Methodology

In this study, we create an artefact to address a certain problem following a design science approach. Design science is "the scientific study and creation of artefacts as they are developed and used by people with the goal of solving practical problems of general interest [5]."

2.1 Research Activities

The design science method consists of five main activities, from problem and requirements formulation to design and development until the demonstration and evaluation of the artefact [5]. A design science project is performed in an iterative way, alternating between different activities. Many design science projects have a specific focus and do not perform all activities. This study did perform all activities, we elaborate on the specifics of all activities below.

1. Explicate Problem - the focus of this activity is on the analysis of a practical problem. The problem should be formulated accurately and the significance of the problem should be addressed.
Section 1.1 introduces the problem experienced by stakeholders that thrives the development of the artefact. We further elaborate on the significance of the problem in section 1.3 and section 3.1.

2. Define Requirements - this activity outlines the solution to the proposed problem in the shape of an artefact and its requirements. The problem is translated to demands for the artefact that is to solve the problem.
The requirements for the solution are formulated based on the literature review pre-

sented in Chapter 3. In this chapter, we evaluate the best practices of related theories to gain a better understanding of what our artefact should conform to. For this, we look at the design principles of Boundary Objects and Design Patterns. Moreover, we also formulate requirements concerning the elicitation method that is to establish the information input for our artefact. The latter is performed through an analysis of the theory of Collaborative Modelling.

3. Design and Develop Artefact - the third activity entails the creations of the artefact that addresses the problem and satisfies the stated requirements.
In Chapter 4 we elaborate on the functionality and structure of the artefact. Moreover, we present the design of concepts representing the problem-solution chains: the repository and the visualisations. In Chapter 5 we demonstrate the design of the information flow to the artefact, the FoCon. The design and development of the artefact satisfies the requirements defined in the previous activity.

4. Demonstrate Artefact - the demonstration is to test the way in which and the feasibility that the artefact is able to solve the problem at hand.
In this activity, three case examples are applied to our artefact, as a proof of concept. The case examples are used to further redefine our concepts and explore the applicability. In Chapter 6 we first present a fabricated instantiation we applied to our artefact. Subsequently, we apply two real-life cases to test the feasibility of our artefact. A case study provides a detailed description of a specific instance [5]. The second case we developed together with a knowledge expert to replicate a real-life situation. The information was gathered through an observation during which the knowledge professional conceptualised problems and solutions in a research project. The third case is developed based on the slides of a workshop about AI innovation.

5. Evaluate Artefact - the last activity of the design science approach aims to explain, to what extent the artefact solves or alleviates the explicated problem that was the main force behind the study.
In Chapter 7 we evaluate our work by discussing the findings of our case examples and existing limitations to our study. In addition, in Chapter 8 we conclude on the realisation of requirements by the artefact.

Chapter 3

Theoretical Framework

To be able to answer the research questions, a decent understanding of the certain theoretical concepts involved is required. This chapter presents the theory that drives our research and upon which our solution is built. First, Business-IT Alignment is introduced and in what way our proposed solution is to increase alignment. Subsequently, we elaborate on an alignment technique called Capability Thinking, to demonstrate how our solution is different from the existing approaches. To better understand how our concepts should be represented, we address the theory of Design Patterns and Boundary Objects. Moreover, this chapter presents KAOS, a goal-oriented requirements engineering approach that inspired the visuals and structure of this study's proposed solution. Finally, this chapter introduces the theory of Collaborative Modelling which forms the basis of our proposed method to elicit problem-solution chains from knowledge experts.

3.1 Business-IT Alignment

Business-IT Alignment is the main force behind this study. Alignment of IT with the business exists when "goals, activities and processes of a business organisation are in harmony with the information systems supporting them. [6]" We see problem-solution chains as a tool to bridge the knowledge gap between professionals with different backgrounds, business domain experts and on the other side IT experts. We recognise the obstacles towards effective communication that enable Business-IT Alignment, due to the vocabulary and knowledge variance in the distinct profession groups. Problem-solution chains can act as a tool to enhance innovation across the distinct knowledge groups through its support of communication in collaboration initiatives.

In the contemporary business environment it is vital for organisations to implement IT in an effective and cost-efficient manner in harmony with the business strategy [7].

This concerns both IT alignment with the business and how the business is aligned with IT. According to [7], the six most important enablers and inhibitors for Business-IT Alignment are illustrated in figure 3.1.

Figure 3.1: Enablers and Inhibitors for Business-IT Alignment [7]

Enablers	**Inhibitors**
• Senior executive support for IT	• IT/business lack close relationships
• IT involved in strategy development	• IT does not prioritize well
• IT understands the business	• IT fails to meet its commitments
• Business/IT partnership	• IT does not understand business
• Well-prioritized IT projects	• Senior executives do not support IT
• IT demonstrates leadership	• IT management lacks leadership

3.1.1 alignment Approach

[7] propose a six-step approach to enhance Business-IT Alignment. The approach begins with *setting the goals and establishing a team.* In the second step: *understand the business-IT linkage*, individuals of each relevant team in an organisation discuss the problems and opportunities in the business and IT environment. This enhances the mutual understanding in the organisation and improves the relationship between business and IT. The third step is to *analyse and prioritise gaps*, between the as-is state and the desired to-be state of alignment. The fourth step, *specify actions*, aims to identify what must be done and who is responsible. In the subsequent step, *choose and evaluate success criteria*, measure criteria is set to evaluate the actions on the strategy. Finally, *sustain alignment* emphasises the importance of behaviour to increase the potential of Business-IT Alignment. According to [7], vital for alignment is clear communication via strong partnerships between IT personnel and line managers.

3.1.2 Relation to Problem-Solution Chains

We believe problem-solution chains could be a tool to support Business-IT Alignment. The foremost advantage is that it establishes mutual understanding in organisations across departments. By linking higher business problems to lower-level technical solutions it is communicated how the IT supports the business or might do so in the future. Moreover, the IT personnel are made aware of the business problems or needs apparent in organisations. IT understanding the business and an effective Business/IT partnership are recognised as enablers for alignment, see figure 3.1. Moreover, by linking business issues to concrete technical solutions, more support for IT projects can be

harvested at executive level in an organisation. The latter realises the first enabler of Business-IT Alignment, senior executive support for IT.
The problem-solution chains support steps 2-4 in the six-step approach of [7]. Regarding the second step, the business-IT linkage is shown by the problem-solution patterns from high-level (business) to lower-level (IT). Related to the third step, alignment gaps become easier to identify because the current state (as-is) can be compared with the desired state (to-be). The as-is state are the existing business problems or needs. The to-be state are the proposed solutions to these high-level problems that are not yet realised. Finally, for the fourth step in the alignment process, chains are also a tool to communicate individual deliverables and ownership to specific employees or departments, by assigning ownership of solutions to actors.

3.2 Capability Thinking

The solution developed in this study is inspired by the research project: Capability as a Service (CaaS) for Digital Enterprises. This project is built in the paradigm of Capability Thinking, implemented through the capability management approach: Capability Driven Development (CDD) [8]. To understand why this project is relevant we analyse the Capability Thinking approach. Subsequently, we discuss how Capability Thinking relates to the Problem-Solution chains.

3.2.1 Key Elements Capability

The CaaS project defines a capability as: "an ability and capacity that enables an enterprise to achieve a business goal in a certain context [9]." The ability is the level of available competence to achieve an objective, the capacity is the available resources such as money and personnel [10].

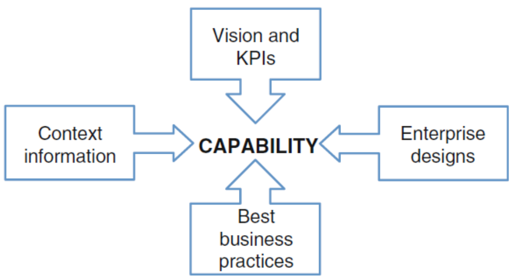


Figure 3.2: Elements of Capability [10]

The main elements of capabilities proposed by [10] are depicted in figure 3.2. *Vision* *and KPIs* contain the foreseen vision and objectives to realise the vision. Common techniques to capture this element are balanced scorecards and the business model canvas. *The existing Enterprise Designs* depict the organisational structure, e.g. what services are generated and in what way IT provides support. *Context Information* is captured during context modelling, with the aim to record and analyse what environmental properties influence and create the need for adaptations to the enterprise's structure. *Best Business Practices* are proven methods organisations have performed in the past and then recorded. Adjusting operations can typically be done by adapting known best practices, without the need for innovative solutions. The contemporary business environment requires a swift response to changes, this can be realised by utilising the best business practices represented by: organisational patterns, process variants, services and IT components.

3.2.2 Motivation for Capability Thinking

Agility and Flexibility - Contemporary organisations must adapt swiftly to economic and regulatory changes in their environment, thus become agile and flexible to continue to be competitive. Traditional change management methods do not satisfy the current demands for responsiveness in digital enterprises [10]. Capability Thinking is proposed as an approach to deal with the varying circumstances digital enterprises operate in. It reduces the amount of time and effort for delivering a business service in a new context [11].

Business-IT Alignment through Capability Thinking - Digital enterprises must have an increased awareness of digital potentials and aim at technology as a competitive advantage, more than in traditional organisations. Therefore, in digital organisations technology innovation requires a strong connection with the business model [10]. Each stakeholder group has a certain view of the organisation, often different from the other groups. This view entails what is vital in the organisation in order to realise the company's objectives, also called the local view. This view reduces the understanding stakeholders have, regarding the effects of their actions on other stakeholders and their activities. Capability Thinking helps to reason from a shared view about what must be achieved, more detailed than the company's goals, however abstract enough to be placed in the local view of the stakeholders. This is done by linking organisational representations such as business process models to strategic objectives. Capabilities integrate the strategy view with the operational view establishing a shared viewpoint. Through its creation of a shared viewpoint, Capability Thinking is recognised as an effective approach to align IT with the business [10]. A capability model acts as a common definition for both business and technology professionals, through which it supports reasoning about IT enabling of the business [12].

3.2.3 Delivery Patterns

The Capability Driven Development (CDD) process as proposed in the CaaS project consists of three cycles: capability design, capability delivery and capability refinement. The main deliverable of the CDD process are reusable patterns [13]. The patterns or so-called capability delivery patterns are: ”reusable solutions for reaching business goals under specific contexts” [14]. For each capability, one or more goals exists. The context is measured through context indicators which are represented in a context set. A capability delivery pattern is delivered when its context set is equal to the context set of the capability [15]. The context is of vital importance to the CDD approach. However, in this study the delivery pattern is what interests us. For more information on context modelling in Capability Thinking see [10]. A capability pattern repository (CPR) enables the storage and retrieval of delivery patterns [16]. The repository is a tool to reuse best practices, in this case: patterns. For which each pattern is a solution to a problem. The patterns can be represented in text or models [17]. Patterns are determined by completing the form template of figure 3.1.

Table 3.1: Delivery Pattern Template [13]

Name of the field	**Purpose of the field**
Name:	Each pattern should have a `name` that reflects the problem/solution that it addresses. Names of patterns are also used for indexing purposes.
Problem:	Describes the issues that the pattern wishes to address within the given context and forces.
Context:	Describes the preconditions under which the problem and the proposed solution seem to occur. This can initially be expressed in free text and later represented by creating a `Context Set` that encompasses the permitted `Context Element Ranges` of `Context Elements` that influence the applicability and variability of the solution proposed by the `Pattern`.
Solution:	Describes how to solve the `problem` and to achieve the desired result. The solution field consists of a textual `solution description` and a `Solution Body` in the form of a model fragment. Currently we focus of `Process Variants` expressed using the BPMN. The `Process Variants` may also contain `Variation Points`. Other model types are also possible to use in order to represent the solution

3.2.4 Capability Thinking in Practice

The capability management approach capability driven development proposed in [10], is a resource-intensive effort. Especially, for enterprises that do not yet have formalised their organisational structure on a mature level. Hence, we investigate how feasible this approach is for smaller sized organisations. In this section, we elaborate on how the elements of capability as illustrated in figure 3.2 fit into the reality of small and medium-sized enterprises (SMEs).

Elements of Capability

Vision and KPIs: Studies have indicated the beneficial effects of strategy formalisation on innovation in SMEs [18]. A formal strategy increases the enterprise's flexibility and innovation [19]. Even though the advantages are evident, a significant part of the SMEs do not perform strategic planning [3].

The Existing Enterprise Designs: Smaller organisations do often not have the resources to define all of their assets through Enterprise Modelling techniques. Moreover, motivation is lacking because of the lower variety of IT applications [20]. Consequently, these organisations lack the knowledge about their organisation to satisfy this condition to construct capability models.

Context information: Capability driven development takes into consideration the context organisations operate in. The strong focus on context-awareness is important for large organisations, because of their complex environment [9]. Multinationals must be able to adapt to new legislation in a variety of countries, while many smaller organisations deal with the rules of just a single nation. Thus, context information is on average less important to SMEs.

Best Business Practices: Process modelling is a commonly used technique to capture processes and their variants. A process variant is an example of a best business practice that helps organisations to innovate and improve their operations [10]. Small and medium-sized companies often decide not to model best business practices because of scarce resources [21].

3.2.5 Relation to Problem-Solution Chains

From section 3.2.4 it can be inferred that a complete capability driven approach, as developed in the project Capability as a Service for Digital Enterprises, is infeasible to adopt in smaller and medium-sized organisations. Problem Solution chains can offer these organisations a method to perform a minimal form of Capability Thinking. The delivery patterns of Capability Thinking explained in section 3.2.3, can be transformed into patterns for solutions to a problem part of a problem-solution chain. For each problem in the chain, a pattern is developed. All patterns combined constitute the problem-solution chain(s). Problems are mapped to a suitable solution, without the assumption that the enterprise architecture must be represented in models. Moreover, without the formalisation of the enterprise vision and context information. The problem-solution patterns fit into the best business practices aspect of Capability Thinking as adapted delivery patterns. While the high-level problems in a chain come near the representations of capabilities, they are defined on a lower-level. In the future, these high-level problems could be directly linked to capabilities to integrate the chains with the Capability Thinking approach. This is out of the scope of this study.

3.3 Design Patterns

The previous section addressed the fact that we desire to transform the capability delivery patterns to problem-solution patterns. To support this endeavour we study design patterns theory. A design pattern is "the re-usable form of a solution to a design problem. [22]" Design Patterns specific to a domain is a pattern language. A pattern describes a common problem, and the core of the solution, such that the solution can be used repeatedly producing a distinct solution every time.

Design patterns were first introduced in the book: *Design Patterns: Elements of Reusable Object-Oriented Software* [23]. This is written in the context of object-oriented software development. Generally speaking, a pattern has four elements:

- The **pattern name** describes the design problem at hand and its solution. A name makes communicating about patterns easier.
- The **problem** describes specific design problems. Often, conditions are listed that must be met for the pattern to be applicable.
- The **solution** illustrates the aspects of the design solution, the relationships, responsibilities and collaborations. This is an abstract description so that the pattern can be applied in different contextual situations.
- The **consequences** are the results of the pattern. Consequences are defined to analyse the costs and benefits of a pattern.

Table 3.2: Design Pattern Input Form

Element	**Description**
Pattern Name and Classification	The name contains the essence of the pattern.
Intent	Describes the issue or problem the design addresses.
Also Known As	Synonyms for the pattern name.
Motivation	The context in which the pattern is applicable.
Structure	Graphical representations of the design solution. Such as interaction diagrams and class diagrams.
Participants	The classes or objects in the pattern and these element's responsibilities.
Collaborations	Describes the way participants collaborate to perform their responsibilities.
Consequences	The trade-offs and results of the pattern.
Related Patterns	Patterns that are related to this pattern. Describes the differences and possible combinations.

Patterns can be useful by themselves to a certain extent as a specific design to a problem. More relevant patterns become when multiple are linked by a relationship [24]. For instance, when an agent is faced with a problem, multiple solutions offered enable the decision-maker to select the better option. In this way, patterns can be complementary since together they constitute a complete overview of solutions to a particular problem. Furthermore, patterns can also be cooperative when a solution enhances the effect of another solution and both patterns are realised together for a problem. For instance, when a pattern leads to another pattern. This type of relationship is best captured in the resulting context or solution [25]. Pattern sequences are specific development paths. The context of each pattern is explained by its predecessors. All pattern sequences combined form a pattern language. Table 3.2 shows an example of a more detailed input form aimed at the creation of design patterns. Design patterns should be described using a consistent format for them to be reused [23].

3.3.1 Relation to Problem-Solution Chains

We view the problem-solution links part of the chain as re-usable problem-solution patterns. We have studied design patterns to better understand the best practices for structuring a conceptual pattern. In section 4.1, we discuss how the elements of the design pattern are mapped to the problem-solution link.

3.4 Boundary Objects

It is widely acknowledged that innovation comes from the collaboration between actors of different departments through knowledge sharing [26]. The collaboration across functions leads to so-called problematic knowledge boundaries [27]. The main cause of what creates the obstacles in cross-boundary knowledge transfer is also what produces its value, the different backgrounds and expertise of the collaboration participants involved. The knowledge transfer is complicated by the different perspectives and realities. Our knowledge system communicates with recipients of distinct backgrounds. Consequently, a sociological perspective is much needed to understand how the developed artefact should help overcome communication obstacles. For this, we look into the concept of boundary objects.

In their study, [28] raised the question: "how do heterogeneity and cooperation coexist, and with what consequences for managing information?" For this purpose, they studied a process of knowledge sharing between distinct groups with different backgrounds in a Zoology museum in Berkeley. They found that the actors made use of what they phrased boundary objects. Boundary objects serve as means of coordination between stakeholders with distinct knowledge [29]. [28] define boundary objects as "an object that lives in multiple social worlds and which has different identities in each". Boundary objects are robust enough to maintain their abstract meaning across boundaries or actors, while the objects do adapt to the local perceptions and constraints of the different actors using them, satisfying the informational requirements [30]. In this way, boundary objects are artefacts that link distinct groups. An example of a boundary object is a strategic planning document that links business goals to technical IT solutions. The document helps to align the business with IT. The business is made aware in what way IT innovation can help their organisation, the IT professionals come to the understanding in what way IT solutions contribute to the enterprise's objectives, strengthening the Business-IT linkage.

3.4.1 The Coordinating Role

Boundary objects act within a certain role in order to support the coordination required for cross-boundary collaboration [26]. In a routine situation, a minimal boundary object such as a single word is sufficient to coordinate collaboration. This coordination is defined as *syntactic coordination.* In a more complex situation, common meanings shared by all actors must be established, which requires the boundary object to include more information. This process of *semantic coordination* is to translate information such that all stakeholders have a shared meaning of the information and understand one another. In situations where negotiation by means of a boundary object is apparent, there is *pragmatic coordination.* In this role, the boundary object enables change or transformation as a broker or intermediary. A boundary object can have one or more roles simultaneously.

3.4.2 Design Principles

In their paper, [31] argue that boundary objects have four essential features:

- **1. Capability to promote shared representation -** focus on common syntax and semantics.
- **2. Capability to transform design knowledge -** transform representations from abstract to concrete solutions.
- **3. Capability to mobilise for action -** facilitate design solutions and improvements.
- **4. Capability to legitimise design knowledge -** legitimise through validation by stakeholders.

3.4.3 Relation to Problem-Solution Chains

To relate the problem-solution chain to the theory of boundary objects, we must first understand more of the different elements of the proposed solution that is to present and elicit the chains. For that reason, this discussion is held in section 4.4, there we comment on the role of problem-solution chains as boundary objects and to what extent the chains satisfy the functional requirements of boundary objects.

The four essential features of boundary objects have functional and political aspects [31]:

Table 3.3: Requirements Boundary Objects

Feature	Functional	Political
1.	Shared functional representations, e.g. data models or specifications	Shared political representations, e.g. agreements or contracts
2.	Knowledge from ambiguous to specific, e.g. business goal to problem, idea or solution	Reallocate power from provider to recipient, because the recipient has more control over the designed solution
3.	Enable experts to exercise review designs, problem analysis and solution discovery.	Establish high decision-making participation and allocation of design tasks.
4.	Validate truthfulness and correctness of the boundary object's content	Demonstrate acceptance of stakeholders for the boundary object's elements, e.g. problems and solutions.

3.5 Requirements Engineering

Requirements Engineering (RE) can be best described as: "the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behaviour, and to their evolution over time and across software families. [32]" RE aims to define a complete enough overview of the requirements of a to be developed software system. In RE there are business, user and system requirements [33]. A business requirement explains the high-level purpose of a software product such as a business goal it helps to achieve. A user requirement represents a property in a business process that the new software system is to realise. A system requirement represents a system property that is to achieve a user requirement. In Agile development processes, a user requirement is typically formulated as: as a User, I want to do something. The later format is also referred to as, a user story.

3.5.1 KAOS

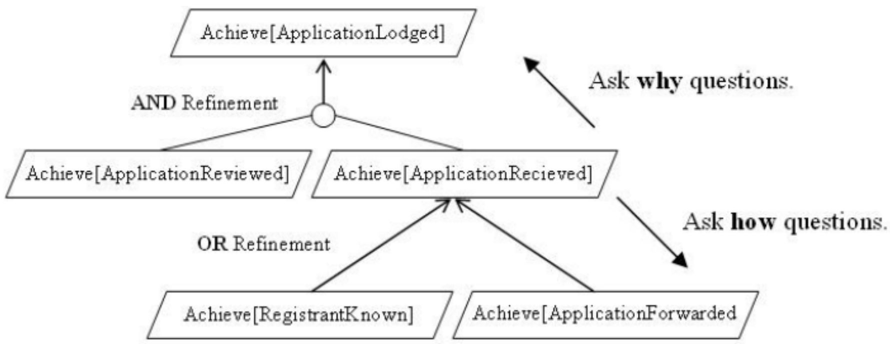
KAOS is a goal-oriented requirements engineering technique [34]. KAOS is to enhance the problem analysis process, by the discovery and structuring of requirements repre-

sented as goals. Also, to effectively communicate responsibilities to all stakeholders involved. KAOS provides: "a high-level view on the system-to-be: what it does, on what, why, how, by whom and when [35]". We take interest in KAOS for the way the concepts are presented in visualisations and information is consistently stored in different models as input for the requirements document.
The KAOS approach is introduced by [35], using the example of a design for an elevator system. However, the KAOS approach can be applied to the design of all sorts of systems. In KAOS there exist four distinct models together form the KAOS model: the Goal model, the Responsibility model, the Object model and the Operation model. The requirements document derives all information from the models.

Goal Model

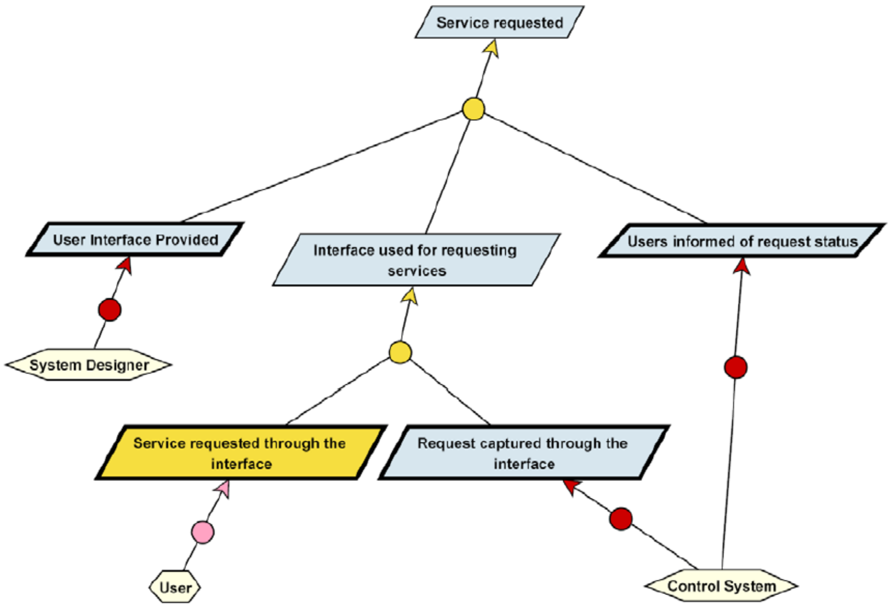
[35] define the KAOS Goal Model as: "the set of interrelated goal diagrams that have been put together for tackling a particular problem." Goals are expressed in the stakeholders' vocabulary of whom it concerns. Higher goals in business language, lower goals in more technical terms. Identification of goals takes place via a top-down approach, from business goals to technical goals, or a bottom-up approach. In the predominant part of KAOS initiatives, both approaches are applied [35]. Usually, KAOS analysts begin with defining intermediate goals. Then, to understand the higher strategic goals, they consider the reasons for having these goals. This is also called the 'why' question. The more sub-goals are captured by thinking about how the intermediate goals can be actualised. This results in more concrete technical goals. Figure 3.3 depicts such a KAOS Goal Model. AND indicates all child goals must be achieved to satisfy the parent goal, OR only requires a single child goal to be actualised to gratify the parent goal [36].

Figure 3.3: Modelling goals in KAOS [36]



A KAOS goal model is a directed graph, indicating that a goal can be a sub-goal of multiple higher-level goals and be in multiple goal diagrams [35]. Goals refer to system states and not state transitions. For instance, 'elevator called' and not 'call elevator'. This is reached by first naming the subject, then a past tense verb, finally a further description if needed. Figure 3.4 is an example of such a goal diagram. In this diagram, we also see a new concept not yet discussed, agents: *System Designer, User* and *Control System*. The agents are related to the goals they are connected with. In the next section, we discuss this in greater detail. In KAOS it is also possible to present alternative ways of realising a goal, by adding more yellow circles pointing from a sub-goal to a higher-level goal.

Figure 3.4: KAOS Goal Diagram [35]



Responsibility Model

Agents being humans or automated components are dominant concepts in the KAOS approach, they are responsible for achieving certain goals [35]. A requirement is a goal that is the responsibility of at least one agent. Lower-level goals are usually the responsibility of automated components, also called software agents. Another type of goal is expectation, displayed as the yellow box in the goal diagram, which is a requirement for agents interacting with the system. This goal type is introduced to emphasise how a system and its environment must cooperate to achieve goals. A responsibility diagram depicts the responsibilities of an individual agent. All responsibility diagrams together constitute the responsibility model.

Object Model

The object model is a glossary that stores the relevant concepts and static variables, applicable to the system domain. A KAOS object model contains elements such as agents, entities and associations [35]. In an elevator system, entities are *elevator* and *floor*. Agents can perform operations on entities for instance, *call(p,f)* elevator is called by *passenger p* at *floor f*. Associations are dependent objects. For example '*At*' is an association between the *elevator* and the *floor* it is at. The object model is visualised using the Unified Modelling Language (UML) for class diagrams.

Operation Model

Besides responsibilities, agents also have capabilities or behaviours [35]. The operation model shows all behaviours agents must perform to satisfy the goals. Examples of behaviours in an elevator system are: open doors, move up or move down. A typical KAOS operation diagram contains the operations performed by an individual or group of agents. KAOS integrates the operation model with the goal model, the operations are legitimised by the goals they realise. Whereas, goals would never become reality were they not operationalised by operations.

Requirements Document

In KAOS, information for the requirements document is extracted from the model [35]. Starting from the top for the strategic goals, down to the sub-goals for the requirements. The responsibility model is the input for the architecture requirements, while the operation model contains requirements for the system behaviour. Modifications are applied to the model, not the document, ensuring the document's consistency with the model. To update the requirements document one must refresh the model input.

3.5.2 Relation to Problem-Solution Chains

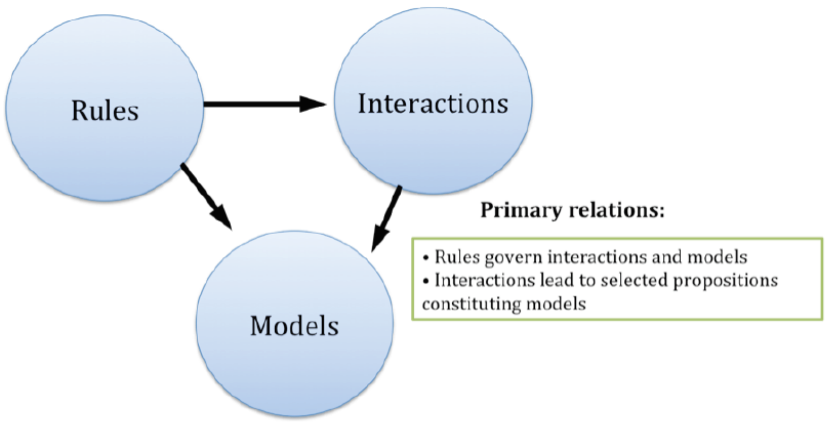
The problem-solution chains are generically defined concepts of proven problem and solution combinations. The chains are more abstract than RE which is to define detailed requirements concerning a system. The notion of a business and user requirements from RE is relevant for the problem elements in the chain. As a problem element in the chain is sometimes more a need than a typical problem. In Chapter 6, we present the observations that led to this assertion. KAOS helps to visualise the chain following a proven method. Moreover, it helps to reason about higher and lower-level elements in the chain. Since, it provides us with valuable insight concerning questions, the sequence of questions and answer formats in the elicitation method. Finally, KAOS demonstrates a method to ensure information consistency in a system with distinct components, relevant to our proposed solution.

3.6 Collaborative Modelling

Besides exploring how to represent problem-solution chains, this study also aims to answer the question: *RQ2: how to elicit problem-solution chains from knowledge experts?* In this section, we describe the collaborative modelling theory in order to design the beginnings of a method to instantiate problem-solution chains answering the second research question.

We use the following notion of modelling: "the purposeful creation of structured and coherent texts or graphical artefacts and subject to strong conceptual (and other) constraints" [37]. A model is composed of related propositions represented as text and graphics. The general elements of a modelling process according to [37], are illustrated in figure 3.5. Collaborative modelling is a model-oriented conversation in which propositions are brought forward and analysed [38]. We recognise problem-solution chains as a possible outcome of a collaborative modelling initiative.

Figure 3.5: RIM Framework [37]



A modelling process is constrained by different rule types: goal, interaction and procedural [37]. Goals constrain through the requirements they enforce on the content, syntax, validation and argumentation of models. Interaction rules are implicit or explicit conventions for acceptable conversation patterns producing unique outcomes. Procedural rules determine the sequence of activities. Interactions entail presenting, discussing and accepting or rejecting propositions [38]. Together, accepted propositions constitute a model.

In collaborative modelling, more attention is brought to the synergy of cross-department communication in modelling [38]. This focus leads to an intensified collaboration of expert modellers with business actors inexperienced in formal modelling techniques. This approach is operationalised by so-called dialogue games, in which collaborative modelling discussions and decisions are explicitly stated. Dialogue games consist of rules and interactions, two of the RIM framework's elements. The output is models as sets of propositions, realising the third element of the RIM framework.

3.6.1 Focused Conceptualisation (FoCon)

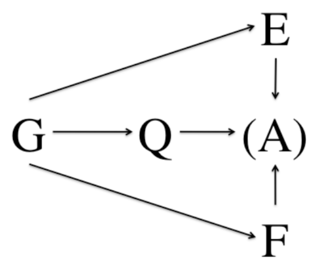
[37] understood the necessity for a theoretical instrument to support the analysis and design of dialogue games. While expert modellers usually develop a model all at once, novice modellers prefer to divide the modelling process into smaller parts. These mini dialogue games are also referred to as Focused Conceptualisations (FoCons). The FoCon concept is established to reason about the questions asked in a modelling process and the answers given to those questions. Novice modellers also called players, are guided by an expert modeller, the facilitator.

In the FoCon theory, a distinction is made between the pragmatic focus of modelling and the semantic-syntactic focus. The pragmatics of a model emphasises the informational and communicational objectives it aims to achieve. For instance, questions modellers should answer in order to gain knowledge about a process and subsequently realise process optimisation. The semantic-syntactic focus addresses the constraints such as the modelling language decided for. The former focus should have priority over the latter focus. These two goal types of modelling are essential to the design of a modelling initiative [38].

3.6.2 Question Asking Framework

The heuristic Question Asking Framework (QAF) is developed to coherently combine questions and answers in FoCons. Figure 3.6 illustrates the concepts with an arrow indicating the 'generative route' [38].

Figure 3.6: Concepts of the Question Asking Framework (QAF) [38]



Goal Questions (G)

Goal questions explain the pragmatic goals of a modelling initiative [38]. The semantic-syntactic goals are integrated into the focus questions and forms constraining the answers. A goal question consists of the main question, to inform the participants about the topic, and the purpose. For instance:
Main Question: Please describe all problems and proposed solutions to those problems during the timeline of project x.
Purpose: The stated problems and solutions will be used in future projects to consider possibilities to overcome certain obstacles.
The goal questions are not part of the individual FoCons, they give the motivation for the modelling session as a whole. Thus, for all FoCons combined that form the conversation-for-modelling. The goal questions are to be understood by the participants before the start of a session.

Focus Questions (Q)

A focus question consists of two elements: the question part and the topic part [38]. The topic part is context-dependent and can refer to an answer expressed just before the question is asked. The question formulation might be influenced by the syntax of a modelling language in use. For instance, the syntax prescribes that entities of type A must have a 'cause' relationship with entities of type B.

Forms (F)

A form as a conceptual frame that limits the variety of shapes and structures of answers, can be helpful to reach modelling objectives. [38] argue that: "the use of standard openers does not impede the flow of communication, and helps focus conceptualisation and structure conversation". Another possible form restriction is to limit the number of characters for answers.

Examples (E)

Examples of answers are helpful to effectively communicate the existing constraints on answers [38]. Examples can also be negative, such that they illustrate wrong input. At the start of a modelling initiative, it is recommended to use examples.

Sequence of Interactions

With focus questions and forms restricting answers, we aim to structure the modelling conversation. However, in collaborative modelling, the dialogue that takes place is not a fully static standard sequence of interactions [38]. Variation is common and it should be allowed to revert to a previously defined FoCon, for instance, to clarify or modify an answer given.

3.6.3 Relation to Problem-Solution Chains

Collaborative Modelling is the cornerstone of our proposed method to elicit problem-solution chains from knowledge experts. The problem-solution chains are defined during a cross-department modelling conversation about problems and solutions. Guiding such a modelling conversation is the primary focus of Collaborative Modelling. The FoCon specific to our method helps to split the modelling task that produces the chain into smaller segments, more specifically, problem-solution patterns. The QAF framework is used to connect human knowledge to concrete problems and solutions by structuring the questions and answers of the FoCon. We present the elicitation method based on the theory of Collaborative Modelling in Chapter 5.

Chapter 4

Representation of Problem-Solution Chains

In this chapter, we build on the theoretical framework to answer the research question: *RQ1: How to represent problem-solution chains?* First, the different concepts of a problem-solution chains are presented. Then, the rules of the problem-solution chain language are demonstrated. Thereafter, the knowledge system that is to present, store and elicit the chains is displayed. Finally, this chapter elaborates on the proposed solution as a boundary object.

4.1 Concepts of the Chain

Problem-Solution Link

A problem-solution link is a description of a problem with the proposed solution(s) to solve this problem. For each distinct solution to a problem, a new link is created. The problems-solution link can also be seen as a problem-solution pattern. The input form of design patterns illustrated in table 3.2, is the basis for the representation of problem-solution links. For this study, a minimalist link structure is defined. As a consequence, some aspects of the design patterns are left out. The link name is the name of the problem or the name of the solution if the problem is undefined. The intent is transferred to the problem, limited by the form restrictions. The structure is the proposed solution to the problem. Finally, in the parent of, we name the patterns the link is a parent of. In pattern writing, this is defined as leads to [25]. The other elements of the design pattern input form are omitted from the problem-solution link structure. The structure of a problem-solution link is shown in table 4.1.

Table 4.1: Problem-Solution Link

Element	Description
Link Name	The name contains the essence of the link.
Problem	Describes the issue or problem the design addresses.
Solution	Short description of the solution
Ownership	Describes the owner responsible for realising the solution
Parent Of	Links resulting from this link

Problem

We identified two types of problems in a chain that lead to solutions. The first type being following the classical definition of a problem: "a situation, person, or thing that needs attention and needs to be dealt with or solved. The latter type is an obstacle to be dealt with, in order to reach the desired state. [39]" The second type of problem is distinct from the classical archetype of a problem, as it is more in the direction of a requirement. In problem-solution chains, many problems are needs or requirements that are realised by solutions. For that reason, we utilised the naming structure of user requirements in RE (see section 3.5) for the problem names in a chain. We identified two categories within this type of problem and made-up an example for both:

- **Business Problem:** As an Organisation, we want to be more productive.
- **User Problem:** As a User, I want to send emails.

Solution

A solution is an answer to a problem. This can be a developed artefact or anything else that helped to partially or completely resolve a problem. Usually a problem leads to a solution, as a problem creates a certain need to change in order to reach a desired state.

Ownership

The owner of a solution is responsible for realising the solution. This can be a single actor or a whole department, depending on the preference of the organisation.

The Chain

A pattern sequence as described by [24], is in the context of this study a combination of problem-solution links, a specific development path from a high-level business problem to (a) lower-level solution(s). Such a development path we call a problem-solution chain. A chain can be any combination of minimally two links part of the problem-solution link collection.

The Pattern Language

All chains combined form the pattern language of problem-solution chains. Now we have outlined the concepts of a problem-solution chain we go more into detail regarding the syntactic and semantic rules of a problem-solution chain that constitute the structure of the chain.

4.2 The Rules

4.2.1 Syntax

The Cambridge Dictionary defines a chain as [40]: "a set of connected or related things" or "(a length of) rings usually made of metal that are connected together and used for fastening, pulling, supporting, or limiting freedom, or as jewellery." Our main interest goes out to the former definition of a chain. In our view, a chain consists of connected links. A link of a problem-solution chain is called a problem-solution link, certain syntax rules apply to the links.

Syntax rules within a link:

- Exactly one problem must be present
- At least one solution to the problem must be present
- A problem must lead to a solution, not the other way around
- A problem must lead to at least one solution
- A problem may lead to multiple sub-solutions part of the same solution
- A problem may lead to multiple distinct solutions, for which each distinct solution a new link is created
- A solution must be connected to a problem

The links are connected for a chain to exist. The following syntax rules apply to relations between links. It can be the case that during the elicitation of problem-solution links a solution results directly from another solution. In the repository, we shall then add an 'empty' problem to the data structure that links the two solutions.

Syntax rules between links (in Repository):

- Moving down one link, a solution is connected to a problem from another link
- Going up one link, a problem is connected to a solution from another link

4.2.2 Semantics

We also specified semantic rules for the naming of elements in a problem-solution link.
Semantics

- Name of a link is the problem name
- Name of a link is the solution name if the problem is undefined
- The structure of a problem name is: subject + verb + short description
- The structure of a problem name that is a business need is: as an Organisation, we want to + need
- The structure of a problem name that is a user need is: as a Persona, I want to + need
- The structure of a solution name is: verb + subject + short description
- domain-specific language should be reduced to a minimum.

4.3 the Knowledge System

The problem-solution chains are part of a knowledge system. This system entails the representation and elicitation of chains. Three distinct components exist: the repository, visualisations and the FoCons. In the proposed knowledge system it should be possible to modify or create information in all three components. Participants of a collaborative modelling initiate chains through FoCons. While an engineer might apply changes to the chains in the repository or visualisations. As explained in section 3.5.1, an important quality of the KAOS approach is the consistency of information inside the document and models. The consistency is ensured by applying all changes directly to the model that checks input on syntactic and semantic requirements stated in section 4.1. Information consistency is also vital in the problem-solution knowledge system, hence we have shown great interest in the model and document consistency checks in KAOS. The syntax and semantic rules must be applied consistently to all updates and modifications to the information stored in the three components of the system. In our system, the repository should serve as a model that verifies all changes.

4.3.1 Visualisations

The visualisations are the graphical representations of the problem-solution chains. The KAOS methodology is an important source for how we choose to represent the chains in these visualisations. KAOS illustrates how to represent alternative ways for achieving a goal, by adding a circle for every unique approach. This can be translated into problem-solution links. In a chain, for all distinct solutions to a problem, a circle is added to the chain initiating a new link. To emphasise that alternative solutions exist. If two solutions are sub-solution of each other, they merge into a single circle part of the same link. Both options are illustrated in figure 4.1. The same can be applied to two distinct problems that are solved by the same solution, illustrated in figure 4.2. The solutions have a different shape and colour than the problems to clarify the difference.

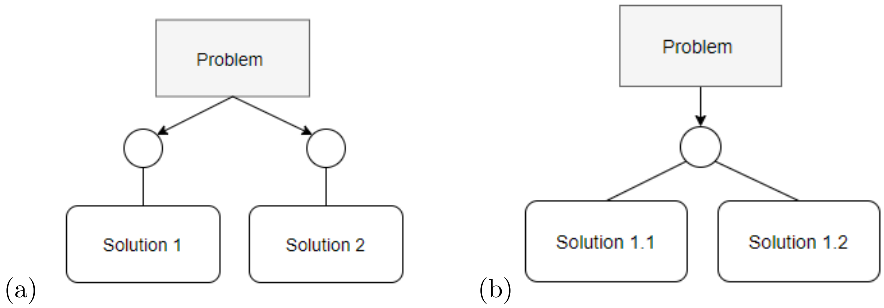


Figure 4.1: Two distinct solutions (a) & two sub-solutions (b)

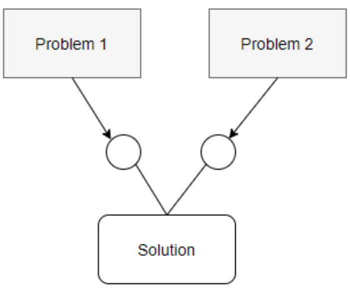


Figure 4.2: Two problems to one solution

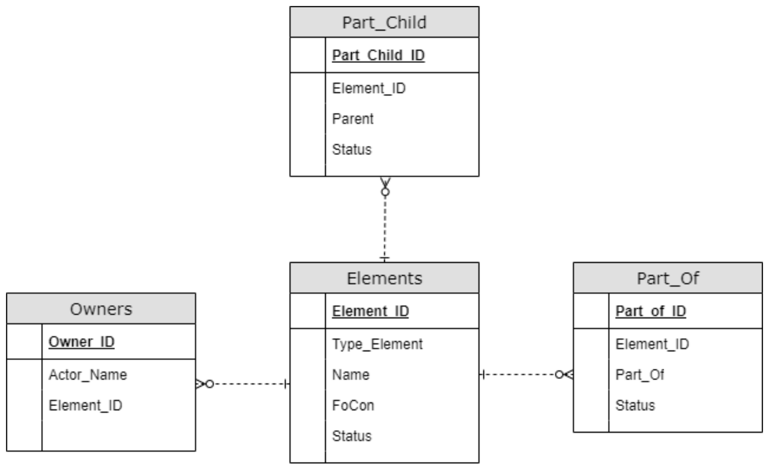
4.3.2 FoCons

The produced Focused Conceptualisations (FoCons) are the outcome of the collaborative modelling sessions aimed at defining problem-solution links part of a chain. To every problem one FoCon is applied, to discover solutions to this problem. If multiple solutions to a problem are proposed in one FoCon, for each distinct solution a new problem-solution link is created. The FoCons describe the discussion and outcome of the session which form the input for the repository and visualisations. We have devoted Chapter 5 to elaborate on the FoCons. In that chapter, we also explain how the individual parts of a FoCon are mapped to a problem-solution link.

4.3.3 Repository

The underlying repository stores all relevant information concerning the elicited problem-solution chains. Similar to the KAOS Model the repository ensures information consistency by applying the semantic and syntactic rules of the problem-solution chain as described in section 4.2. Figure 4.3 illustrates the logical entity-relationship diagram model of the repository. The *Elements* table stores the proposed problems and solutions part of a chain. The table *Owners* records the actor(s) responsible for realising a solution. The *Part_Child* table contains the 'Leads to' relationship of elements to other elements. The Parent leads to the associated element of Element_ID. Finally, table *Part_Of* stores the sub-solution relationship of two solutions part of the same solution. An important motivation for this study is to make explicit modelling choices. For that reason, the repository should record the relevant aspects of a FoCon discussion and make sure that they can be retrieved later conveniently. We decided to store the associated FoCon and ownership of solutions. The discussion itself, regarding specific elements is not stored directly in the repository.

Figure 4.3: Logical ERD Model: Chain Repository



4.4 Problem-Solution Chains as Boundary Objects

The problem-solutions chains help to communicate across departments, high-level business problems to lower-level technical solutions and the underlying motivation for the modelled chain. The elements in a chain, problems and solutions, have different meanings across distinct professional boundaries. A mention of a database solution has multiple local views in an organisation. A database engineer reasons about the technical details of such a system, while a business manager considers the reporting information to monitor their department's performance. The chain explains to the database engineer what business problems his database helps to solve. The business manager is brought to the attention how IT can help reach their department's objectives. In this way, a problem-solution chain is a boundary object that serves as a means of coordination between stakeholders with different knowledge. In section 3.4, we explained the concept of boundary objects. To elaborate on chains as boundary objects we first distinguish the different coordinating roles the chains have, this to draw a clear picture of the problem-solution chain as a boundary object. Then, we shall apply the functional design requirements of boundary objects to the chains and the knowledge system presenting the chains. Following this approach, we ensure that our knowledge system supports the coordination between stakeholders in the most optimal way.

4.4.1 The Coordinating Role of the Chains

The chains support *syntactic coordination* by the structure it creates through its syntax. The structure ensures consistent information across all professional boundaries in an organisation. The chains are constructed from a shared understanding of what the individual nodes and the chain as a whole entails, following a structured discussion between knowledge experts from different backgrounds. Ensuring the *semantic coordination* of the chains as boundary objects. Thirdly, the chains also play the role of *pragmatic coordination* as they enable the acceptance or rejection of any modifications to the existing structure as an intermediary.

4.4.2 Requirements

The functional requirements stated in table 3.3 are to be satisfied by the knowledge system that presents the problem-solution chains. This study does not cover the political requirements of boundary objects.

1. Shared functional representations

The knowledge system ensures the consistency of the concepts through the data model of the repository. The syntax and semantic rules of the model are applied to all visualisations.

2. Knowledge from ambiguous to specific

The technical professional is interested in technical solutions required to realise an abstract problem. The business professional looks at the business goals that a solution aims to reach. By linking business problems and needs to concrete technology solutions, the chains bridge the gap between the business and IT. The business understands what the IT department does to support their core processes, the IT department recognises how their technology solutions directly influence business objectives.

3. Enable Experts to exercise review designs, problem analysis and solution discovery

Experts are enabled to constantly review the chains by analysing the FoCons that motivate the designs. Also, new solutions and problems can be proposed and discussed to create paths within a new or existing chain. Moreover, a modeller can visually analyse the chains and directly apply changes to them.

4. Validate truthfulness and correctness of the boundary object's context

The FoCons demonstrate the validation process of the boundary object's context. They depict the collaboration and negotiation that created the chain. The relevant actors involved can address their concerns or support for individual elements. The result is the acceptance, rejection or postponement of the discussed element. If accepted, the element is added to the chain, in case of a rejection the element is not displayed, when postponed this is also expressed in the visualisation.

Chapter 5

Elicitation of Problem-Solution Chains

Now we defined the representation of problem-solution chains, it becomes relevant to consider how to instantiate this structure. This leads to answering the second research question of this study: *how to elicit problem-solution chains from knowledge experts?*

We have already discussed the theory of collaborative modelling in section 3.6. We discussed collaborative modelling, because the problem-solution chains contain and connect knowledge from different departments notwithstanding non-expert modellers. Hence, we require an approach that supports the conversation of novice modellers that leads to models, the focus of collaborative modelling. Subsequently, we addressed the importance of focused conceptualisations or FoCons that helps to split the modelling process into smaller segments making it more accessible for novice modellers. The section below presents the FoCon designed for the elicitation of problem-solution chains and how the questions are mapped to the problem-solution link. Furthermore, we demonstrate how we applied the Question Asking Framework (QAF) to structure the answers and questions in our FoCon. Finally, we elaborate on another technique to define problem-solution chains.

5.1 FoCon for the Problem-Solution Chain

For the design of the FoCon for the chains, we have taken inspiration from [41]. In this work, a collaborative modelling initiative is analysed and designed using the FoCon approach. Derived from this analysis is a prototype FoCon to be used for the elicitation of problem-solution chains. A single FoCon states the proposed solutions to a problem and the arguments for and against the propositions. Furthermore, it records the relations between different FoCons. In this context, relations between problems and solutions defined in different FoCons. The FoCon can consist of multiple participants of which at least one is the facilitator. Another possibility is that there is solely one participant who takes the role of facilitator and player at the same time. In the latter situation, participants explicitly ask questions to themselves. This is especially helpful to novice modellers and makes the design choices easier to understand for non-participants in a later stage.

From [41], we apply the concept of interaction types to structure actions made within the FoCon. The following interaction types are defined:

- **Questions** - asked by the facilitator to participants (players) to elicit information that results in models.
- **Propositions** - information propositions of players concerning problems, solutions and the relations between those elements.
- **Argue** - arguments for and against propositions made by players.
- **Acceptations** - final agreements or disagreements concerning propositions.

For the formulation of questions and answers of the FoCon we apply the QAF, explained in section 3.6.2.

Goal Questions

The general pragmatic focus of the FoCons in this study is to structure and record modelling decisions about problem-solution patterns in organisations. We do not consider goals of specific modelling tasks. The semantic-syntactic focus of the FoCon is enforced through the focus questions and forms.

Focus Questions

The focus questions are illustrated in the most left column of table A.1. The question formulation is influenced by the syntax of the pattern language. The questions and their order enforce the syntax on the outcome of the collaborative modelling initiative.
The KAOS approach tells us to begin with the identification of intermediate concepts, in our case problem-solution links. From there, they take a mixed top-down and bottom-up approach. With the elicitation of problem-solution chains, we do not always start with intermediate concepts. However, we can learn from the KAOS approach since it helps us to move up and down the chain. We go up by reasoning about the more abstract links, preceding the intermediate links already identified. The question that should be asked is: why is this a problem? In case there is no problem defined in a link: what does this solution solve? Lower-level links are retrieved by analysing how the link is realised. Since the high-level and intermediate links lead to more problem and solution elements that make concrete the higher-level elements in the chain. The question that should be asked is: how is this solution realised?
These questions are rather abstract, we rephrased the questions in our FoCon to fit into the context of problem-solution patterns. Therefore, the FoCons are finalised by the questions: please propose a solution that leads to the problem/solution of this link (go up) and please propose a problem/solution that resulted from a solution of this link (go down). The latter two mentioned questions initiate the creation of a new link. All questions are illustrated in table 5.1 under 'Questions'.

Forms

As explained in section 3.6.1 it is recommended to make use of form restrictions to constrain answers given. We have constrained the answers by mandatory openers stated in table 5.1 under 'Answers'. Moreover, we restricted the naming of problems and solutions. The structure of a problem is: Subject + Verb + short description. If a problem is a need the structure is: Organisation/Persona + Need + Purpose. The structure of a solution is: Verb + Subject + short description.

Sequence of Interactions

A FoCon begins with the definition of a problem or solution, depending on what is most convenient to the particular situation, or what element of the FoCon is already defined in another FoCon as explained in section 5.1. There exists no strict sequence of interactions for the modeller to interact with the FoCons. Players are allowed to move between distinct FoCons to modify or clarify a proposition. The facilitator is also able to ask questions about previously stated propositions.

Examples

To effectively communicate constraints on answers we designed an example case. This also helped us to further redefine our FoCon template to make it more useful in practice. The example is Case A and is presented in section 6.1.

The FoCon Template

The table below is the basic template of our developed FoCon. The first part focuses on a single problem and a solution to this problem. The second part starting from question 8, shows how a modeller can define elements from other links related to elements captured in the first part of the FoCon. The last column indicates to what element of the link structure, as explained in section 4.1, the answer given is mapped to. The arguments and acceptation or rejection of elements are not part of the problem-solution link structure.

Table 5.1: FoCon Template

#	Question	Answer	Mapping to Link (ID Link)
1	Please propose a PROBLEM	PROBLEM	Link Name (1) & Problem (1)
2	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	/
3	Please accept/reject the PROPOSITION	Accept/Reject	/
4	Please propose a SOLUTION that resolved the PROBLEM	SOLUTION	Solution (1)
5	Please propose an actor responsible for realising this solution	ACTOR	Ownership (1)
6	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	/
7	Please accept/reject the PROPOSITION	Accept/Reject	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
8	Please propose a PROBLEM/-SOLUTION that resulted from a SOLUTION of this LINK	SOLUTION (of this LINK): PROBLEM/SOLUTION	Link Name (2) & Problem (2) & Parent Of (1) *Creates new link, child of this link*
9	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	/
10	Please accept/reject the PROPOSITION	Accept/Reject	/
11	Please propose a SOLUTION that leads to a PROBLEM/SOLUTION of this LINK	PROBLEM/SOLUTION (of this LINK): SOLUTION	Solution (3) & Parent Of (3) *Creates new link, parent of this link*
12	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	/
13	Please accept/reject the PROPOSITION	Accept/Reject	/

If alternative or sub-solutions are proposed, the template illustrated in table 5.2 is included. For every additional solution proposed, it must be identified whether this element is an alternative solution or a sub-solution of the first solution proposed. If the solution is an alternative solution, a new link is initiated. In case of a sub-solution, the solution is added to the same link of the first solution (of which it is a sub-solution).

Table 5.2: FoCon Template Multiple Solutions

1	Please propose a SOLUTION that resolved the PROBLEM (2^{nd} Solution)	SOLUTION	If sub-solution: Solution (1) Else: Solution (2) *Creates new link, distinct solution to the problem*
2	Please propose an actor responsible for realising this solution	ACTOR	If sub-solution: Ownership (1) Else: Ownership (2)
3	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	/
4	Please accept/reject the PROPOSITION	Accept/Reject	/
5	Please propose a SOLUTION of which this SOLUTION is part of (for sub-solutions)	SOLUTION	/
6	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	/
7	Please accept/reject the PROPOSITION	Accept/Reject	/

5.2 Visual Analysis

Besides a structured discussion about modelling concepts through FoCons, a modeller might decide to modify or create new chains in the visualisations. The visualisations should allow the modeller to drag and drop elements in the chain to support the modelling process. These actions are to be performed before or after FoCons have been applied. The created elements should adhere to the rules of the language as described in section 4.2. All changes should be applied to the repository directly and to the FoCons at hand or create new FoCons if the element is not yet captured inside one. For instance, a newly created problem element initiates a FoCon. But, an update to an existing problem simply modifies the FoCon in which the problem element already was defined.

Chapter 6

Evaluation of Concepts

In this chapter, we evaluate the proposed concepts presented in Chapter 4 and the elicitation method in Chapter 5. We do this by applying three cases to the concepts. The first case, case example A, is a made-up example and the first chance to evaluate our proof-of-concept knowledge system. Subsequently, we apply a real-life case example selected and represented in collaboration with a knowledge professional. Finally, we apply a workshop about AI innovation to our proposed elements. We show how the concepts evolved based on the observations of the practical examples.

The case examples in this chapter are not extensive instances of our concepts, in reality more elements could be added to the chains. The examples are merely to evaluate the design of the knowledge system's elements that is proposed as a solution in this thesis. The applicability of the chain visualisations is tested, the questions and answer constraints of the FoCon are exercised and the data structure of the repository is examined.

6.1 Case A

This case is the first example used to further redefine the proposed concepts of the knowledge system for problem-solution chains. This is done by first, drawing a problem-solution chain, then in retrospect analyse the modelling decisions made building the chain. The product of this retrospective analysis is the first version of the FoCon template, displayed in Appendix A.1. Finally, the elements are mapped to the repository. This modelling initiative is performed by the author of this study and does not reflect the modelling conversation of an actual project.
The following observations were made:

- We found that it is possible to always include a problem in between two solutions, but not always useful. The problem *train schedule unknown* does not introduce any new information compared to the two solutions it is linked to.

- Moreover, this case example led us to believe that a modeller might return to previously defined elements. The problem *bicycle key is lost* has the alternative solutions: *take train* and *take bus*, solutions defined in a previous FoCon.

- The visualisation of 6.1 introduces another high-level problem *I must go to the supermarket* that is materialised through the same infrastructure as the problem from FoCon 1, *I must go to the University.* If we assume that the two high-level problems that have no parent element both initiate a new chain, the visualisation of figure 6.1 consists of two chains.

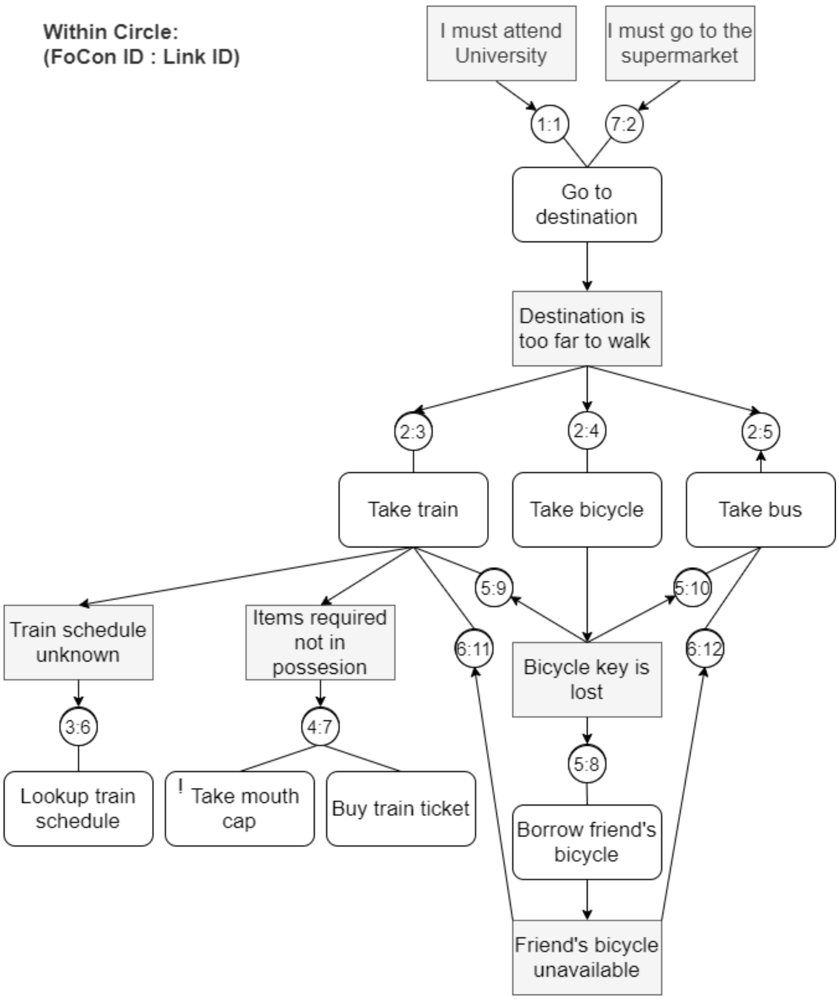
- A modeller might wish to not yet accept a proposition. The modeller can decide to delay this by not answering the question: *Please accept/reject the proposition.* In that case, an exclamation mark is added to the element in the visualisation. This symbol indicates that the judgement regarding this specific element is postponed. The postponed element cannot directly lead to any new elements before it is accepted.

- In defining the problem-solution chain, the modeller made use of visual analysis as described in section 5.2. This appeared to be a very natural way of defining elements of the chain and the mutual relationships.

Visualisation

The case example represented in figure 6.1, is a basic situation of a transportation problem. The number inside the circle of a link indicates to what FoCon the link belongs.

Figure 6.1: Problem-Solution Chain: Example A



FoCons

The FoCon design used in Case A is the first version of our FoCon and is distinct from the final version we presented in the previous chapter. This version of the FoCon is designed to structure the modelling discussion of this case. We redefined the questions and answer format of the FoCon based on this modelling session. In this version of the FoCon the arguments regarding specific elements are mapped to the link structure. In the final version of the FoCon this does not take place, because the arguments are omitted from the link structure. All FoCons are presented in Appendix B.

Repository

The outcome of the FoCons of Example A is stored in the repository tables, shown in Appendix C. The repository design for this case example is different from the final design. This design includes the arguments of the discussion in the repository rather than only storing the associated FoCon number of each element.

6.2 Case B

We applied a second case example to evaluate our developed concepts. This case was built around the 'Patient forum miner' project, a data-mining effort to extract and structure patient experiences posted on online forums regarding a rare form of cancer [42]. We observed the problem-solution analysis of two researchers involved in the project. The researchers were asked to sketch the problems and solutions that occurred during the project to realise the main project's deliverable, the Patient forum miner. From the resulting analysis, we draw a problem-solution chain, completed the FoCons and mapped this to the repository instance.
The following observations were made:

- This case study indicated that it can be challenging for a modeller to strictly follow the syntax structure of relations between different problem-solution links. In some situations it might be preferred to connect a solution directly to another solution in a leads to relationship, without a problem in between. In the data structure, we then would add an 'empty' problem to connect the two solutions. In the chain visualisation, we omit this empty or dummy problem. Because the problem-solution link structure remains intact, it remains an option to later add a problem as an intermediary element between the two solutions. In the chain of example B, seen in 6.2, the solution: *Use system that shares patient experience* directly leads to the solution: *Visualise patient forum posts in system.* In the data structure there is an undefined problem element linking the two solutions. In this case example, we did not observe two problems directly linked to each other by the observed participant.

- Because we abounded the rule that between two solutions there must be a defined problem connecting the two, our FoCon should adopt a more flexible approach. The FoCon accepts statements that propose a leads to relationship between two solutions. To realise this, we modified question 13 and 16 of the FoCon, seen in figure 5.1.

- We observed that problems are often requirements or needs of an organisation or stakeholder involved in a project. We found that we can formulate these problems as user stories: *organisation/persona + need + purpose* [43]. We applied this to the problem names of Example B. Not all problems can be translated to a need and described using the user story format. For those problems we remain in favour of the structure: *subject + verb + description.*

- The modellers also argued that it would be helpful to assign ownership of solutions to specific actors. According to the participant, explicating ownership of solutions increases effective communication between stakeholders. As a response we added the following question to our FoCon: *please propose an actor responsible for realising this solution.* In the data model of the repository we added the table *Owners*, that is to store the actors responsible for realising a solution.

- We also found that the KAOS approach of first defining intermediary elements is not always favourable for problem-solution chains. The knowledge expert in case example B preferred to start with high-level problems and solutions, and subsequently move down on the chain to more technical concepts.

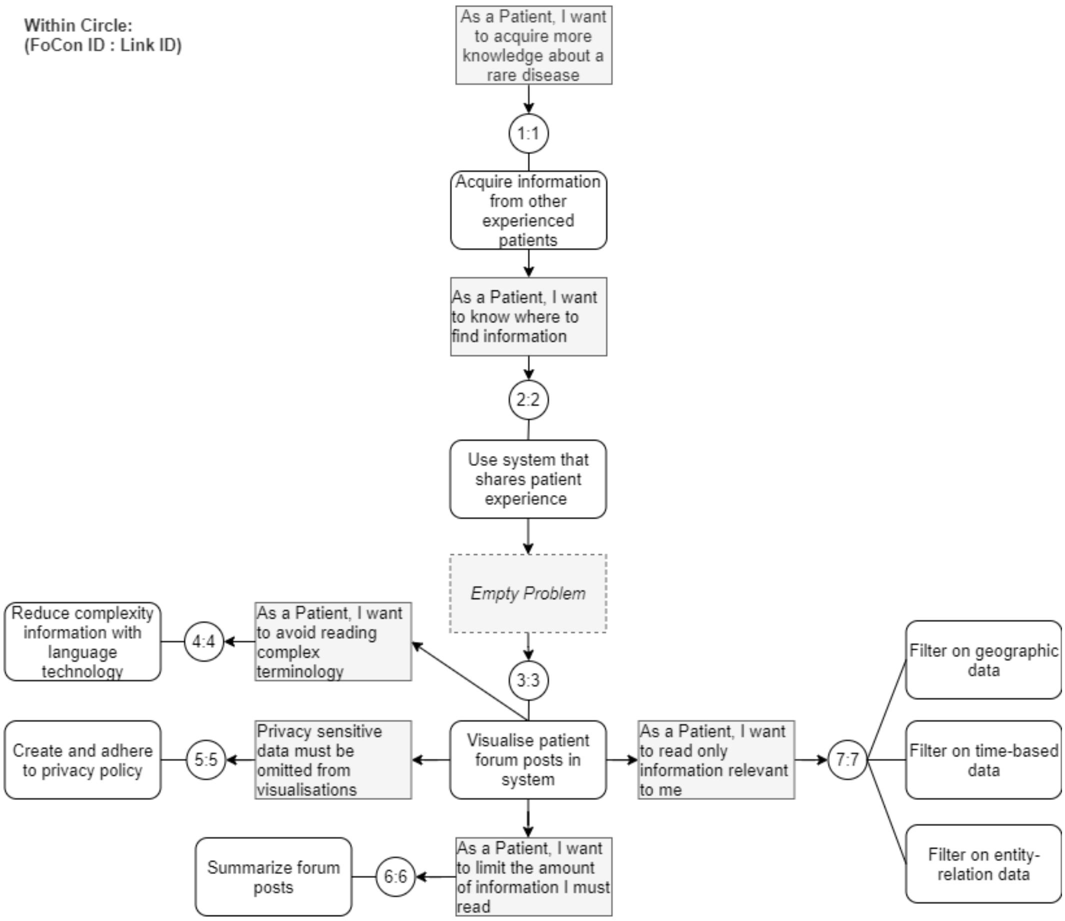
- A modeller argued that certain elements are too technical to be included in the problem-solution chain. Other tools, such as a technical requirements documents or verbal communication, are in that case more convenient to use. The solution *Reduce complexity information with language technology* is sufficiently detailed to be understood by the technical experts who is to implement this solution, and abstract enough for the healthcare professional to realise his need is delivered by IT. For this reason, the solution *Apply Python-Data cleaning* is rejected seen in table D.4.

- The initial name of solution *Reduce complexity information with language technology* was *Reduce complexity information with NLP.* We changed NLP to language technology, to prevent domain-specific terms to hinder effective communication across profession groups.

Visualisation

The visualisation of the chain in this case example is depicted in figure 6.2. In the visualisation, we displayed the empty problem to emphasise its presence in the repository without the explicit definition by the modeller.

Figure 6.2: Problem-Solution Chain: Example B



FoCons

This case helped us to further redefine our FoCon template resulting in the FoCon displayed in table 5.1. The FoCons of this case example that demonstrate the underlying discussion are displayed in Appendix D. The FoCon template that is first redefined and then applied is distinct from the FoCon version used in Case A.

Repository

The repository instance of case example B is displayed in Appendix E. Different from the repository instance of example A, the table Arguments is removed and the table

Owners is added. The Arguments table is removed to further reduce the complexity of the data model since this is not the focus of this study. The FoCon discussion is linked to the element in the Element table. The Owners table is added to the model to satisfy the need derived from this case example for assigned ownership of solutions.

6.3 Case C

The third case example we applied to evaluate our concepts is built around a workshop concerning Artificial Intelligence (AI) innovation in the manufacturing industry [44]. In this workshop, several professionals from the field presented problems they face in the working field, and in what way AI helped them to overcome these obstacles. From the slides of the workshop, we constructed three problem-solution chains. These chains are the final test in our evaluation of our developed concepts. Different from the previous case example, we omitted the empty problems from the visualisations to minimise the elements presented.
The following observations were made:

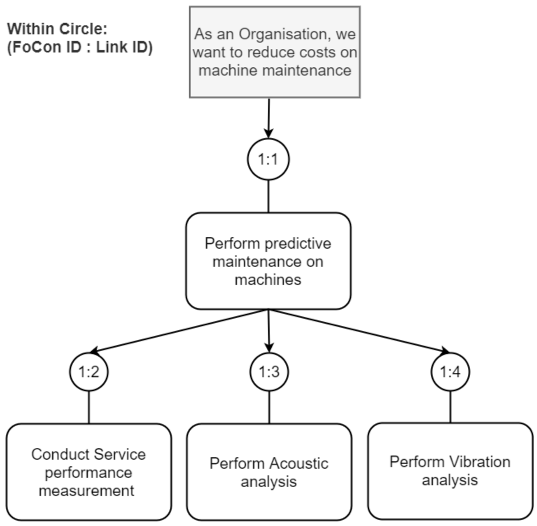
- Similar to case B, it appeared to be convenient to sometimes leave problems empty and directly connect a solution to another solution. This again emphasises the numerical majority of solutions in the chain over problems.
- Chain 3.2 is the only chain of case C that describes problems that are not formulated as a business need: *Install extra sensors, Develop computer model for temperature control* and *Merge sensor data into computer model*. We included these elements because they were explicitly stated in the slides of the workshop.
- In all three chains, the top-level problem is a business need. Indicating that chains are triggered by a certain need from the organisation to achieve objectives. The modeller moved down in the chain, by making the abstract problem more concrete with solutions. The first step was the definition of an abstract solution to the high-level business need. Subsequently, more concrete solutions were established that illustrate specific applications of the abstract solution. The modeller defined these concrete solutions by reasoning about how the abstract solution is realised, which fits well in the KAOS approach as illustrated in figure 3.3.
- The visual analysis of the chains helped the modeller to find that two distinct solutions could be mapped to a single problem. Namely, *Install extra sensors* and *Develop computer model for temperature control* to the problem *Sensor data not in computer model.*

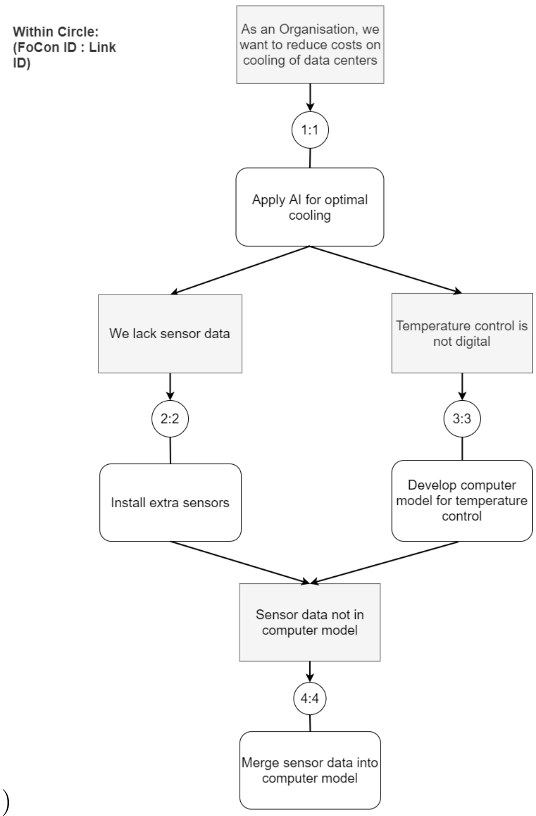
- The final FoCon version defined in case B fits the modelling process of case C well. No changes were applied to the FoCon structure during this case example.

Visualisations

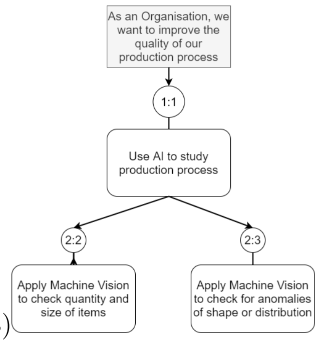
Case C consists of three chains displayed in three distinct visualisations. Chain 3.1 is shown in figure 6.3, chain 3.2 and 3.3 are illustrated in figure 6.4.

Figure 6.3: Problem-Solution Chain: Example C.3.1





(3.2)



(3.3)

Figure 6.4: Problem-Solution Chain: Example C.3.2 & C.3.3

FoCons

The FoCons of chains 3.1, 3.2 and 3.3 are shown in Appendix F. The final version of the FoCon as presented in figure 5.1 is applied to the chains of case C.

Repository

The repository instance of case C is displayed in Appendix G. The data structure of the repository instance is the final data model as displayed in 4.3 and identical to the structure used in case B.

Chapter 7

Discussion

7.1 Key Findings

In this study, we have proposed a possible method to capture and elicit patterns in organisations, patterns we phrased problem-solution chains. We developed the initial representation of problem-solution chains based on theoretical findings presented in chapter 3. The result of this are the proposed concepts the problem-solution chain repository, visualisation and FoCon. We evaluated our concepts by applying three case examples. In this chapter, we reflect on the evaluation of our concepts presented in the previous chapter.

RQ1: How to represent problem-solution chains?

The visualisation of case A demonstrated that a single visualisation can contain two distinct problem-solution chains. The two high-level problems result in the same problems and solutions, and can therefore be shown in a single visualisation. In this way, more information is presented to the user. A modeller might desire to postpone the judgement regarding a specific element in the chain. This demand is expressed by adding an exclamation mark to the element. An element that is not accepted yet cannot lead to any new elements, because these elements would originate from a not accepted element.

The second case indicates that there should be a form of assigned ownership of solutions included in the problem-solution chain. To bridge the communication gap between professionals it must be clear who is responsible for realising certain elements of a chain. Adding ownership of solutions to the chain can improve a project's time efficiency since it reduces the chance of miscommunications concerning deliverables. Our case examples did not indicate the need for assigned ownership of defined problems in the chain. For that reason, we did not include this in our FoCon and repository.

Our case examples showed that a problem can also be a business or user need. The third case example demonstrated that a high-level problem formulated as a business need can often be the only problem required to be present in a chain. An abstract business need can eventually result in concrete solutions without intermediary more concrete problem elements. Intermediary problems do not always introduce new information as is observed in the evaluation of our concepts. This led us to believe that high-level business problems, often formulated as business needs, are more relevant as an information source than lower-level problems. A high-level business need proved to frequently be the initial trigger of a chain. In that case, the business need is the main force behind the solutions part of the chain. Moreover, solutions play a more dominant role in the chain and cannot be omitted as opposed to intermediary problem elements.

Case B proved that not all elements should be included in a chain. Certain elements are too detailed and as a result written in the language of a certain profession group, making it difficult to understand for professionals from other knowledge groups. Elements in the chain should remain abstract enough to be comprehensible to all, and detailed enough to provide meaning. This principle is in line with the chain as a boundary object. The distinct groups in an organisation must have a shared meaning of the elements in the chain but derive different responsibilities from the elements. In case of the solution *Reduce complexity information with language technology*, the software developer understands that different approaches exist and that most fitting is to be realised, the healthcare professional is made aware that a filter has been applied to the information. This helps the healthcare professional to realise the information output of the filter must be verified by a knowledge expert, to ensure the information still provides meaning to the patient using the system.

RQ2: How to elicit problem-solution chains from knowledge experts?

The case Patient forum miner gave us the key insight that a FoCon should adopt a more flexible approach and not strictly adhere to the problem-solution chain syntax and semantic rules. In case of the modeller neglecting certain rules, the system is to step in and correct the input without disturbing the modeller. For instance, the modeller 'forgets' to define a problem in a link. Then, the system adds an 'empty' problem to the link, ensuring that the input complies with the language rules. In this way, the system helps the modeller to define concepts more naturally. The latter is of key concern to the design of a FoCon since it determines the applicability of the method in practice.

The third case example indicates that a chain is often triggered by a business need not a problem. The format of a need is designed according to the user stories format of requirements engineering. This need is then represented as the high-level problem of the chain. The business desires something in order to achieve a certain objective. An abstract solution is defined that is to answer this need, which leads to more concrete problems and solutions. This knowledge is vital for the elicitation of chains as it tells us a specific starting or endpoint of a chain. If starting at the top, a modeller is advised to begin with a high-level business need. When the definition process of a new chain begins with proposing concrete solutions and problems, the modeller is recommended to finish the chain with a high-level business need. We do not include this as a rule in our language, for the reason that there might be cases that require a different approach.

The chains defined in the case examples from the previous chapter are the product of a mixed approach of FoCons and visual analysis. During the visual analysis, elements were dragged and dropped in the visualisations of the chain. The visualisations proved to be useful as a tool to define problem-solution chains. It is recommended that users can also create new or apply changes to the chains in the visualisation views. Any change in the view should lead to a modification in a FoCon or the creation of a new FoCon, depending on the type of update to the chain. Our case examples confirmed our belief that this is a vital technique for the elicitation of problem-solution chains. The chain visualisations help the modeller to relate new links to existing elements of the chain or to map distinct elements to a single element. The latter supports the re-use of elements and prevents the creation of duplicate information in the chain. It also enables experts to exercise review designs, problem analysis and solution discovery, the third functional requirement of boundary objects.

7.2 Limitations

- We are aware of the fact that we omit several elements of design patterns in our problem-solution patterns. In the theory of design patterns, a strong emphasis is placed on the context of a problem affecting possible solutions. The context is described in the shape of forces motivating the reasons for the problem being a problem. The forces are vital for determining the effectiveness of a solution. In the FoCons we have given way for a structured discussion about a proposed problem. This, however, is not nearly as strictly determined as factors in design patterns.

We have decided for a more flexible structure that is easier to adopt for users, evidently impacting the generalisability of the proposed concepts.

- In this thesis research, we have made a start with developing a method to elicit problem-solution chains from knowledge experts, by means of an initial design of a FoCon. It was not feasible in the scope of this study, to execute the proposed FoCon in a real-life situation. We could merely describe the communication situation that led to the problem-solution chains, with the FoCon as an analysis tool. The structure of questions and answers of the initial FoCon design is not yet a feasible approach to directly apply in a modelling conversation. The contents should be further tested and applied in real-life projects to establish a more practical method for collaborative modelling.
- The second case demonstrated that a problem is sometimes better described as a need. In the scope of this study, we have not been able to implement a problem as a need in the questioning format of our FoCon design. We merely added flexibility to our rules constraining the answers. This thesis does not illustrate how the modeller can be supported in thinking about needs rather than problems leading to solutions.

Chapter 8

Conclusions and Future Work

This study aims to conceptualise problem-solution chains in organisations. This is achieved by answering the two sub-research questions. To answer the research questions, a theoretical framework was developed. From this framework, problem-solution chains have been conceptualised, and an initial method to elicit chains from knowledge experts. Three case examples have been applied to the proposed concepts to test their practicability and further redefine the structure.

RQ1: *How to represent problem-solution chains?*

The capability delivery pattern from Capability Thinking connected with design patterns theory forms the basis of the problem-solution link structure. A link has one problem and one or more solutions. If an alternative solution to a problem exists this is stored in another link. We found that a problem in a chain can also be a business or user need and should therefore be formulated as such. Ownership of solutions is determined to effectively communicate the responsibilities of specific elements of the chain. Finally, the element parent of of the link contains all child links. A problem-solution chain is a specific development path from a high-level business problem or need to a lower-level element.

The chains are to be stored in a knowledge system, this system has three distinct components. The repository functions as data storage that also verifies the information consistency by applying the language rules to the information. Secondly, the visualisations demonstrate the chains stored in the repository. The third component, the FoCons, are responsible for information input through the elicitation of problem-solution chains from knowledge experts. Concerning the visualisation of problem-solution chains, we started by analysing the goal-oriented requirements engineering technique KAOS. The KAOS methodology helped us to make visual design choices. The language used in the chain should not be specific domain language.

The chains support steps 2 until 4 of the Business-IT Alignment approach of [7]. The chains enable alignment through an increase of senior executive support for IT, IT understanding of the business and Business/IT partnership. Moreover, we looked at the theory of boundary objects to understand how our proposed solution can support cross-boundary knowledge transfer. The chains satisfy the boundary object's functional requirements for it to be used as a tool to increase effective communication between distinct profession groups. The latter reinforces the way in which the problem-solution chains are represented in this study.

RQ2: *How to elicit problem-solution chains from knowledge experts?*

Concerning the second research question, we proposed the beginnings of what is a method to elicit problem-solution chains from knowledge experts. From the field of collaborative modelling, we looked at Focused Conceptualisations (FoCons). In this study, we developed a FoCon that can be used to model problem-solution links that constitute a chain. The FoCon provides more flexibility in modelling than the problem-solution link syntax rules would allow, this to create a more practical modelling conversation. The FoCon includes a structured discussion about the defined elements of the chain. This discussion can later be used for a better understanding and communication of the specific elements in the chain. Moreover, visual analysis is another technique we identified as a useful tool to define problem-solution chains.
A problem-solution chain is often triggered by a high-level business need. A modeller is recommended to begin with defining a high-level business need from which more concrete problem and solutions will follow.
The Design Science approach helped to identify and structure the activities involved with the creation of our concepts. The five distinct activities are all present in our study, and the iterative working method fits in well with our research process. The proposed concepts of the problem-solution chain are the end products of this thesis.

8.1 Future Work

- In this study, we did not exercise our FoCon in a project with actual stakeholders and a facilitator. The FoCon template should be tested in an actual workshop. The feedback from practitioners could further redefine our concepts. The proposed solution should be evaluated in different types of organisations, to discover whether adaptations should be made.

- The problem-solution knowledge system is a boundary object, such that it is to support effective collaboration across boundaries. The KAOS responsibility model, explained in section 3.5.1, is a tool to increase the awareness of actors concerning their responsibilities in a problem-solution chain. A responsibility diagram can illustrate to individual actors what solutions they should take ownership of or co-create. In future research, we wish to explore more ways in which ownership of solutions is effectively communicated through visualisations.

- We observed that it is practical to connect two solutions without an intermediary problem. In future studies, we would like to explore if two problems can be linked to each other without an intermediary solution, or whether it does not add value to present problems that are not resolved, but result in additional problems.

- Ownership of solutions is included in the FoCons and repository. The chain visualisations do not depict the owners of specific solutions. In a future study, it should be reviewed whether the visualisations should illustrate owners and if so, in what way this should visually be represented. Moreover, problem owners are a common approach in organisations to allocate responsibilities. In this study, problem owners did not come forward as a relevant category in the problem-solution chain. Since we believe problem owners are an informative addition to the chain we would like to explore assigned ownership to problems in a future study.

Bibliography

[1] Sang M Lee, Kihyun Kim, Patrick Paulson, and Hyesung Park. Developing a socio-technical framework for business-it alignment. *Industrial Management & Data Systems*, 2008.

[2] Gideon Mekonnen Jonathan, Lazar Rusu, and Erik Perjons. Business-it alignment in the era of digital transformation: Quo vadis? In *Hawaii International Conference on System Sciences (HICSS), Maui, Hawaii, USA, January 7-10, 2020*, pages 5563–5572, 2020.

[3] Calvin Wang, Elizabeth Walker, and Janice Redmond. Explaining the lack of strategic planning in smes: The importance of owner motivation. *ECU Publications*, 12, 01 2011.

[4] I Martinez-Moyano. Exploring the dynamics of collaboration in interorganizational settings. *Creating a culture of collaboration: The International Association of Facilitators handbook*, 4:69, 2006.

[5] Paul Johannesson and Erik Perjons. *An introduction to design science*. Springer, 2014.

[6] Lerina Aversano, Carmine Grasso, and Maria Tortorella. A literature review of business/it alignment strategies. *Procedia Technology*, 5:462–474, 2012. 4th Conference of Enterprise Information Systems – aligning technology, organizations and people (CENTERIS 2012).

[7] Jerry Luftman and Tom Brier. Achieving and sustaining business-it alignment. *California management review*, 42(1):109–122, 1999.

[8] Janis Stirna. Caas – capability as a service for digital enterprises. `https://caas-project.eu/`, 2015.

[9] Solvita Bērziša, George Bravos, Tania Cardona Gonzalez, Ulrich Czubayko, Sergio España, Jānis Grabis, Martin Henkel, Lauma Jokste, Jânis Kampars, Hasan Koç,

et al. Capability driven development: an approach to designing digital enterprises. *Business & Information Systems Engineering*, 57(1):15–25, 2015.

[10] Kurt Sandkuhl and Janis Stirna. Capability thinking. In *Capability Management in Digital Enterprises*, pages 1–24. Springer, 2018.

[11] Kurt Sandkuhl. It value of capabilities: Experiences from a case in business process outsourcing. In *Capability Management in Digital Enterprises*, pages 355–369. Springer, 2018.

[12] Andrew Swindell. Business capability models: Why you might be missing out on better business outcomes. `https://www.architectureandgovernance.com/strategy-planning/business-capability-models-might-missing-better-business-outcomes/`, 2014.

[13] Janis Stirna, Jelena Zdravkovic, Martin Henkel, and Janis Kampars. Capability patterns as the enablers for model-based development of business context-aware applications. In *17th IEEE International Conference on Business Informatics, Lisbon, Portugal, July 13-16, 2015*. CEUR-WS. org, 2015.

[14] Martin Henkel, Jelena Zdravkovic, Francisco Valverde, and Oscar Pastor. Capability design with cdd. In *Capability Management in Digital Enterprises*, pages 101–116. Springer, 2018.

[15] Hasan Koç and Kurt Sandkuhl. Context modelling in capability management. In *Capability Management in Digital Enterprises*, pages 117–138. Springer, 2018.

[16] Jānis Grabis, Jelena Zdravkovic, and Janis Stirna. Overview of capability-driven development methodology. In *Capability Management in Digital Enterprises*, pages 59–84. Springer, 2018.

[17] Jānis Grabis and Jānis Kampars. Capability management in the cloud. In *Capability Management in Digital Enterprises*, pages 175–188. Springer, 2018.

[18] Andrew Burke, Stuart Fraser, and Francis J Greene. The multiple effects of business planning on new venture performance. *Journal of management studies*, 47(3):391–415, 2010.

[19] Clay Dibrell, Justin B Craig, and Donald O Neubaum. Linking the formal strategic planning process, planning flexibility, and innovativeness to firm performance. *Journal of Business Research*, 67(9):2000–2007, 2014.

[20] Maxime Bernaert, Geert Poels, Monique Snoeck, and Manu Backer. *Enterprise Architecture for Small and Medium-Sized Enterprises: A Starting Point for Bringing EA to SMEs, Based on Adoption Models*. 10 2013.

[21] Daniel Hilpoltsteiner, Stephanie Bäuml, and Christian Seel. Picking process variability in small and medium-sized enterprises: State of the art and knowledge modeling. In *KMIS*, pages 118–125, 2018.

[22] Christopher Alexander. *A pattern language: towns, buildings, construction*. Oxford university press, 1977.

[23] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, and Design Patterns. Elements of reusable object-oriented software. *Design Patterns. massachusetts: Addison-Wesley Publishing Company*, 1995.

[24] Frank Buschmann, Kevlin Henney, and Douglas C Schmidt. *Pattern-oriented software architecture, on patterns and pattern languages*, volume 5. John wiley & sons, 2007.

[25] Doble J Meszaros and Jim Doble. G. a pattern language for pattern writing. In *Proceedings of International Conference on Pattern languages of program design (1997)*, volume 131, page 164, 1997.

[26] Chris Kimble, Corinne Grenier, and Karine Goglio-Primard. Innovation and knowledge sharing across professional boundaries: Political interplay between boundary objects and brokers. *International Journal of Information Management*, 30(5):437–444, 2010.

[27] Paul R Carlile. A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization science*, 13(4):442–455, 2002.

[28] Susan Leigh Star and James R. Griesemer. Institutional ecology, 'translations' and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907-39. *Social Studies of Science*, 19(3):387–420, 1989.

[29] Charlotte Lee. Boundary objects for creativity innovation. PAIR Symposium, 2020.

[30] Geoffrey Bowker and Susan Leigh Star. Sorting things out. *Classification and its consequences*, 4, 1999.

[31] Gloria Mark, Kalle Lyytinen, and Mark Bergman. Boundary objects in design: An ecological view of design artifacts. *Journal of the Association for Information Systems*, 8(11):34, 2007.

[32] Bashar Nuseibeh and Steve Easterbrook. Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, pages 35–46, 2000.

[33] Neil Maiden. User requirements and system requirements. *IEEE Software*, 25(2):90–91, 2008.

[34] Alexei Lapouchnian. Goal-oriented requirements engineering: An overview of the current research. *University of Toronto*, 32, 2005.

[35] Respect-IT. A kaos tutorial. `http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf`, 2007.

[36] George Koliadis and Aditya Ghose. Relating business process models to goal-oriented requirements models in kaos. In *Pacific Rim Knowledge Acquisition Workshop*, pages 25–39. Springer, 2006.

[37] Stijn Hoppenbrouwers and Ilona Wilmont. Focused conceptualisation: framing questioning and answering in model-oriented dialogue games. In *IFIP Working Conference on The Practice of Enterprise Modeling*, pages 190–204. Springer, 2010.

[38] Stijn Hoppenbrouwers. Asking questions about asking questions in collaborative enterprise modelling. In *IFIP Working Conference on The Practice of Enterprise Modeling*, pages 16–30. Springer, 2012.

[39] Cambridge University. Problem. In *Cambridge Dictionary*. Cambridge University Press, 2021.

[40] Cambridge University. Chain. In *Cambridge Dictionary*. Cambridge University Press, 2021.

[41] Stijn Hoppenbrouwers and Etiënne Rouwette. A dialogue game for analysing group model building: framing collaborative modelling and its facilitation. *International Journal of Organisational Design and Engineering*, 2(1):19–40, 2012.

[42] Wessel Kraaij. Patient forum miner (sidn fonds). `https://liacs.leidenuniv.nl/~kraaijw/index.php/sidn-patient-forum-miner/`, 2018.

[43] Robert Galen. Technical user stories – what, when, and, how? `https://rgalen.com/agile-training-news/2013/11/10/technical-user-stories-what-when-and-how/`, 2013.

[44] RCT Gelderland FME, Boost. Zet je bedrijf in pole position met artificial intelligence. `https://boostsmartindustry.nl/agenda/webinar-zet-je-bedrijf-pole-position-met-ai/`, 2020.

Appendices

Appendix A

FoCon Template

Table A.1: FoCon Template Version 2.

#	Question	Answer	Mapping to Link (ID Link)
1	Please propose a PROBLEM	PROBLEM	Link Name (1) & Problem (1)
2	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	Arguments Problem (1)
3	Please accept/reject the PROPOSITION	Accept/Reject	/
4	Please propose a SOLUTION that resolved the PROBLEM	SOLUTION	Solution (1)
5	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	Arguments Solution (1)
6	Please accept/reject the PROPOSITION	Accept/Reject	/
7	Please propose a SOLUTION that resolved the PROBLEM (2^{nd} Solution)	SOLUTION	If sub-solution: Solution (1) Else: Solution (2) *Creates new link, distinct solution to the problem*
8	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	If sub-solution: Arguments Solution (1) Else: Arguments Solution (2)
9	Please accept/reject the PROPOSITION	Accept/Reject	/
10	Please propose a SOLUTION of which this SOLUTION is part of (for sub-solutions)	SOLUTION	/
11	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	/
12	Please accept/reject the PROPOSITION	Accept/Reject	/

Continued Table A.1: FoCon Template Version 1.

#	Question	Answer	Mapping to Link (ID Link)
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
13	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	SOLUTION (of this FoCon): PROBLEM	Link Name (3) & Problem (3) *Creates new link, child of this link*
14	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	Arguments Problem (3)
15	Please accept/reject the PROPOSITION	Accept/Reject	/
16	Please propose a SOLUTION that leads to the PROBLEM of this LINK	SOLUTION	Solution (4) *Creates new link, parent of this link*
17	Please argue for/against the PROPOSITION	Agree/Disagree: (max 100 words each argument)	Arguments Solution (4)
18	Please accept/reject the PROPOSITION	Accept/Reject	/

Appendix B

FoCons Case A

High-Level Problem-Solution

This is the first example of a FoCon applied to the chain of example A. The FoCon shows how the modellers decided for the problem: 'I must attend University' and the solution to this problem: 'Go to destination'. Finally, the modellers concluded that a new problem arises from the proposed solution. This problem: 'Destination is too far to walk' initiates FoCon 2, as does every new proposed problem.

Table B.1: Example A: FoCon (1) - I must attend University

#	**Question**	**Answer**	**Link**
1	Please propose a PROBLEM	I must attend University	Link Name (1) & Problem (1)
2	Please argue for/against the PROPOSITION	Agree: this is an important problem	Arguments Problem (1)
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Go to destination	Solution (1,2)
5	Please argue for/against the PROPOSITION	Agree: this is the initial problem since I am not at the University	Arguments Solution (1, 2)
6	Please accept/reject the PROPOSITION	Accept	/
	The section below initiates a new FoCon for any proposed problem or solution not yet defined		
7	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Go to destination: Destination is too far to walk	Link Name (3, 4, 5) & Problem (3, 4, 5) & Parent Of (1)
8	Please argue for/against the PROPOSITION	Agree: need additional solutions to go to destination thus a problem	Arguments Problem (3, 4, 5)
9	Please accept/reject the PROPOSITION	Accept	/

Distinct Solutions

This FoCon shows how three distinct solutions to a single problem are represented. Although the problem of this FoCon results from the solution: 'Go to destination' we do not answer question 24 accordingly, since we can already derive this fact from FoCon 1.

Table B.2: Example A: FoCon (2) - Destination is too far to walk

#	Question	Answer	Link
1	Please propose a PROBLEM	Destination is too far to walk	Link Name (3, 4, 5) & Problem (3, 4, 5)
2	Please argue for/against the PROPOSITION	Agree: need additional solutions to go to destination thus a problem	Arguments Problem (3, 4, 5)
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Take train	Solution (3, 9, 11)
5	Please argue for/against the PROPOSITION	Agree: the train is quick and the train station is close by	Arguments Solution (3, 9, 11)
6	Please accept/reject the PROPOSITION	Accept	/
7	Please propose a SOLUTION that resolved the PROBLEM (2^{nd} Solution)	Take bicycle	Solution (4)
8	Please argue for/against the PROPOSITION	Agree: destination is within cycling distance	Arguments Solution (4)
9	Please accept/reject the PROPOSITION	Accept	/
10	Please propose a SOLUTION that resolved the PROBLEM (3^{rd} Solution)	Take bus	Solution (5, 10)
11	Please argue for/against the PROPOSITION	Agree: bus drives frequently	Arguments Solution (5, 10)
12	Please accept/reject the PROPOSITION	Accept	/
	The section below initiates a new FoCon for any proposed problem or solution not yet defined		
13	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Take train: Train schedule unknown	Link Name (6) & Problem (6) & Parent Of (3)
14	Please argue for/against the PROPOSITION	Agree: we must know when the train leaves	Arguments Problem (6)
15	Please accept/reject the PROPOSITION	Accept	/
16	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Take train: Items required not in possession	Link Name (7) & Problem (7) & Parent Of (3)
17	Please argue for/against the PROPOSITION	Agree: we still need to acquire the required items for travelling by train	Arguments Problem (7)
18	Please accept/reject the PROPOSITION	Accept	/

Continued Table 6.2: Example A: FoCon (2) - Destination is too far to walk

#	**Question**	**Answer**	**Link**
21	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Take bicycle: bicycle key is lost	Link Name (8) & Problem (8) & Parent Of (4)
22	Please argue for/against the PROPOSITION	Agree: we must have the key to ride a bicycle	Arguments Problem (8)
23	Please accept/reject the PROPOSITION	Accept	/

Lower-Level Problem-Solution

The solution in this FoCon is on the lowest level of the chain and does therefore not result in a new problem.

Table B.3: Example A: FoCon (3) - Train schedule unknown

#	**Question**	**Answer**	**Link**
1	Please propose a PROBLEM	Train schedule unknown	Link Name (6) & Problem (6)
2	Please argue for/against the PROPOSITION	Agree: we must know when the train leaves	Arguments Problem (6)
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Lookup train schedule	Solution (6)
5	Please argue for/against the PROPOSITION	Agree: by doing this we know the train schedule	Arguments Solution (6)
6	Please accept/reject the PROPOSITION	Accept	/

Sub-solutions and Postponement

This FoCon illustrates how sub-solutions are recorded. Also, the answer to question 10 shows that the judgement regarding a solution or problem can be postponed, indicated by the exclamation mark in the chain visualisation. The reason for the delay is that two arguments are brought forward, one in favour and one against the proposition. Hence, the modellers decided to await the judgement. The postponed solution cannot result in a new problem because it is not yet accepted.

Table B.4: Example A: FoCon (4) - Items required not in possession

#	Question	Answer	Link
1	Please propose a PROBLEM	Items required not in possession	Link Name (7) & Problem (7)
2	Please argue for/against the PROPOSITION	Agree: we still need to acquire the required items for travelling by train	Arguments Problem (7)
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM (2nd Solution)	Take mouth cap	Solution (7)
5	Please argue for/against the PROPOSITION	Agree: a mouth cap is mandatory to wear on a train and this solution would solve that issue	Arguments Solution (7)
6	Please argue for/against the PROPOSITION	Disagree: a mouth cap is is no longer mandatory to wear	Arguments Solution (7)
7	Please accept/reject the PROPOSITION	/	/
8	Please propose a SOLUTION that resolved the PROBLEM	Buy train ticket	Solution (7)
9	Please argue for/against the PROPOSITION	Agree: a train ticket is required to board the train and this purchase would solve that issue	Arguments Solution (7)
10	Please accept/reject the PROPOSITION	Accept	/

Return to previous FoCons

This FoCon demonstrates that a FoCon can return to previously defined elements. The problem 'bicycle key is lost' has alternative solutions: 'take train' and 'take bus' which are initially defined in FoCon 2 - 'Destination is too far to walk'. This FoCon routes the chain path to the outcome of another FoCon.

Table B.5: Example A: FoCon (5) - Bicycle key is lost

#	Question	Answer	Link
1	Please propose a PROBLEM	Bicycle key is lost	Link Name (8, 9, 10) & Problem (8, 9, 10)
2	Please argue for/against the PROPOSITION	Agree: we must have the key to ride a bicycle	Arguments Problem (8, 9, 10)
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Borrow friend's bicycle	Solution (8)
5	Please argue for/against the PROPOSITION	Agree: this would be a convenient solution	Arguments Solution (8)
6	Please accept/reject the PROPOSITION	Accept	/
7	Please propose a SOLUTION that resolved the PROBLEM (2nd Solution)	Take train	Solution (3, 9, 11)
8	Please argue for/against the PROPOSITION	Agree: the train is quick and the train station is close by	Arguments Solution (3, 9, 11)
9	Please accept/reject the PROPOSITION	Accept	/
10	Please propose a SOLUTION that resolved the PROBLEM (3rd Solution)	Take bus	Solution (5, 10)
11	Please argue for/against the PROPOSITION	Agree: bus drives frequently	Arguments Solution (5, 10)
12	Please accept/reject the PROPOSITION	Accept	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
13	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Borrow friend's bicycle: Friend's bicycle unavailable	Problem (11, 12) & Parent Of (8)
14	Please argue for/against the PROPOSITION	Agree: if the bicycle is unavailable we cannot ride it	Arguments Problem (11, 12)
15	Please accept/reject the PROPOSITION	Accept	/

No initiation of new FoCons

Since the proposed solutions are part of previously defined FoCons we do not propose any problems that resulted from the solutions. For the reason that the elements are already initiated by FoCon 2.

Table B.6: Example A: FoCon (6) - Friend's bicycle unavailable

#	**Question**	**Answer**	**Link**
1	Please propose a PROBLEM	Friend's bicycle unavailable	Link Name (11, 12) & Problem (11, 12)
2	Please argue for/against the PROPOSITION	Agree: if the bicycle is unavailable we cannot ride it	Arguments Problem (11, 12)
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Take train	Solution (3, 9, 11)
5	Please argue for/against the PROPOSITION	Agree: the train is quick and the train station is close by	Arguments Solution (3, 9, 11)
6	Please accept/reject the PROPOSITION	Accept	/
7	Please propose a SOLUTION that resolved the PROBLEM (2^{nd} Solution)	Take bus	Solution (5, 10)
8	Please argue for/against the PROPOSITION	Agree: bus drives frequently	Arguments Solution (5, 10)
9	Please accept/reject the PROPOSITION	Accept	/

A second Chain

Table B.7: Example A: FoCon (7) - I must go to the supermarket

#	Question	Answer	Link
1	Please propose a PROBLEM	I must go to the supermarket	Link Name (2) & Problem (2)
2	Please argue for/against the PROPOSITION	Agree: this is an important problem	Arguments Problem (2)
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Go to destination	Solution (1, 2)
5	Please argue for/against the PROPOSITION	Agree: this is the initial problem since I am not at the supermarket	Arguments Solution (1, 2)
6	Please accept/reject the PROPOSITION	Accept	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
7	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Go to destination: Destination is too far to walk	Link Name (3, 4, 5) & Problem (3, 4, 5) & Parent Of (2)
8	Please argue for/against the PROPOSITION	Agree: this problem resulted from the solution	Arguments Problem (3, 4, 5)
9	Please accept/reject the PROPOSITION	Accept	/

Appendix C

Repository Case A

Table C.1: Case Example A - Table: Elements

Element_ID	Type_Element	Name	FoCon	Status
1	PROBLEM	I must attend University	1	Accept
2	SOLUTION	Go to destination	1	Accept
3	PROBLEM	Destination is too far to walk	2	Accept
4	SOLUTION	Take Train	2	Accept
5	SOLUTION	Take bicycle	2	Accept
6	SOLUTION	Take bus	2	Accept
7	PROBLEM	Train schedule unknown	3	Accept
8	PROBLEM	Items required not in possession	4	Accept
9	PROBLEM	bicycle key is lost	5	Accept
10	SOLUTION	Lookup train schedule	3	Accept
11	SOLUTION	Buy train ticket	4	Accept
12	SOLUTION	Take mouth cap	4	Undefined
13	SOLUTION	Borrow friend's bicycle	5	Accept
14	PROBLEM	Friend's bicycle unavailable	6	Accept
15	PROBLEM	I must go to the supermarket	7	Accept

Table C.2: Case Example A - Table: Part_Of

Part_Of_ID	Element_ID	Part_Of	Status
1	12	11	Accept

Table C.3: Case Example A - Table: Arguments

Arg_ID	Element_ID	Argument	Type_Arg	FoCon
1	1	this is an important problem	Agree	1
2	2	this is the initial problem since I am not at the University	Agree	1
3	3	need additional solutions to go to destination thus a problem	Agree	2
4	4	the train is quick and the train stationis close by	Agree	2
5	5	destination is within cycling distance	Agree	2
6	6	bus drives frequently	Agree	2
7	7	we must know when the train leaves	Agree	3
8	8	we still need to acquire the required items for travelling by train	Agree	4
9	9	we must have the key to ride a bicycle	Agree	5
10	10	by doing this we know the train schedule	Agree	3
11	11	a train ticket is required to board the train and this purchase would solve that issue	Agree	4
12	12	a mouth cap is mandatory to wear on a train and this solution would solve that issue	Agree	4
13	12	a mouth cap is is no longer mandatory to wear	Disagree	4
14	13	this would be a convenient solution	Agree	5
15	14	if the bicycle is unavailable we cannot ride it	Agree	6
16	15	this is an important problem	Agree	7
17	12	this is part of the same solution, toacquire the required items	Agree	4

Table C.4: Case Example A - Table: Parent_Child

Parent_Child_ID	Element_ID	Parent	Status
1	1		Accept
2	2	1	Accept
3	3	2	Accept
4	4	3	Accept
5	5	3	Accept
6	6	3	Accept
7	7	4	Accept
8	8	4	Accept
9	9	5	Accept
10	10	7	Accept
11	11	8	Accept
12	12	8	Accept
13	13	9	Accept
14	14	13	Accept
15	15		Accept
16	4	9	Accept
17	6	9	Accept
18	4	14	Accept
19	6	14	Accept

Appendix D

FoCons Case B

High-Level Problem

The FoCon displayed in table D.1 indicates the questions and answers involved with defining the top link of the chain of Example B. Different than from the FoCon design from the first case example we added a question-answer structure for the defining of a actor responsible for a solution.

Table D.1: Example B: FoCon (1) - As a Patient, I want to know acquire more knowledge about a rare disease

#	Question	Answer	Link
1	Please propose a PROBLEM	As a Patient, I want to know acquire more knowledge about a rare disease	Link Name (1) & Problem (1)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Acquire information from other experienced patients	Solution (1)
5	Please propose an actor responsible for realising this solution	Software Developer	Ownership (1)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
8	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Acquire information from other experienced patients : As a Patient, I want to know where to find information	Link Name (2) & Problem (2) & Parent Of (1)
9	Please argue for/against the PROPOSITION	/	/
10	Please accept/reject the PROPOSITION	Accept	/

Solution leading to a Solution

The FoCon displayed in table D.2 shows how the second version of our FoCon is able to include a solution that leads to another solution, without an intermediary problem in between. The proposed solution: *Visualise patient forum posts in system* initiates a new FoCon (3) and link (3), in which the problem remains undefined.

Table D.2: Example B: FoCon (2) - As a Patient, I want to know where to find information

#	**Question**	**Answer**	**Link**
1	Please propose a PROBLEM	As a Patient, I want to know where to find information	Link Name (2) & Problem (2)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Use system that shares patient experience	Solution (2)
5	Please propose an actor responsible for realising this solution	Software Developer	Ownership (2)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
8	Please propose a SOLUTION that resulted from a SOLUTION of this FoCon	Use system that shares patient experience : Visualise patient forum posts in system	Link Name (3) & Solution (3) & Parent Of (2)
9	Please argue for/against the PROPOSITION	/	/
10	Please accept/reject the PROPOSITION	Accept	/

Undefined Problem

The FoCon displayed in table D.3 shows how the second version of our FoCon is able to include a solution that leads to another solution, without an intermediary problem in between. Even though the modeller does not include a problem in the FoCon, a dummy problem is added to the link in the repository. In table Elements, seen in Appendix G.3, element 5 is named 'Empty Problem' representing the problem that is not defined by the modeller, but is required to satisfy the syntax rules of the chain. Since the problem is undefined, the FoCon is named after the solution.

Table D.3: Example B: FoCon (3) - Visualise patient forum posts in system

#	**Question**	**Answer**	**Link**
1	Please propose a PROBLEM	/	Problem (3)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	/	/
4	Please propose a SOLUTION that resolved the PROBLEM	Visualise patient forum posts in system	Link Name (3) & Solution (3)
5	Please propose an actor responsible for realising this solution	Software Developer	Ownership (3)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
8	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Visualise patient forum posts in system : As a Patient, I want to avoid reading complex terminology	Link Name (4) & Problem (4) & Parent Of (3)
9	Please argue for/against the PROPOSITION	/	/
10	Please accept/reject the PROPOSITION	Accept	/
11	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Visualise patient forum posts in system : Privacy sensitive data must be omitted from visualisations	Link Name (5) & Problem (5) & Parent Of (3)
12	Please argue for/against the PROPOSITION	/	/
13	Please accept/reject the PROPOSITION	Accept	/

Continued Table 6.10: Example B: FoCon (3) - Visualise patient forum posts in system

#	Question	Answer	Link
14	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Visualise patient forum posts in system : As a Patient, I want to limit the amount of information I must read	Link Name (6) & Problem (6) & Parent Of (3)
15	Please argue for/against the PROPOSITION	/	/
16	Please accept/reject the PROPOSITION	Accept	/
17	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Visualise patient forum posts in system : As a Patient, I want to read only information relevant to me	Link Name (7) & Problem (7) & Parent Of (3)
18	Please argue for/against the PROPOSITION	/	/
19	Please accept/reject the PROPOSITION	Accept	/

Lower-Level Links

The FoCons displayed in tables D.4, D.5 and D.6 show lower-level link examples with one problem and a solution to this problem. The links do not lead to any new links.

Table D.4: Example B: FoCon (4) - As a Patient, I want to avoid reading complex terminology

#	Question	Answer	Link
1	Please propose a PROBLEM	As a Patient, I want to avoid reading complex terminology	Link Name (4) & Problem (4)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Reduce Complexity information with language technology	Solution (4)
5	Please propose an actor responsible for realising this solution	Software Developer	Ownership (4)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
8	Please propose a SOLUTION that resulted from a SOLUTION of this FoCon	Reduce Complexity information with language technology : Apply Python-Data cleaning	/
9	Please argue for/against the PROPOSITION	This is too detailed for a non-technical person to be understood and does not seem to add anything to increase effective communication between different departments.	/
10	Please accept/reject the PROPOSITION	Reject	/

Table D.5: Example B: FoCon (5) - Privacy sensitive data must be omitted from visualisations

#	Question	Answer	Link
1	Please propose a PROBLEM	Privacy sensitive data must be omitted from visualisations	Link Name (5) & Problem (5)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Create and adhere to privacy policy	Solution (5)
5	Please propose an actor responsible for realising this solution	Software Developer	Ownership (5)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/

Table D.6: Example B: FoCon (6) - As a Patient, I want to limit the amount of information I must read

#	Question	Answer	Link
1	Please propose a PROBLEM	As a Patient, I want to limit the amount of information I must read	Link Name (6) & Problem (6)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Summarize forum posts	Solution (6)
5	Please propose an actor responsible for realising this solution	Software Developer	Ownership (6)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/

Sub-Solutions

The FoCon displayed in table D.7 presents an example of multiple sub-solutions to a problem.

Table D.7: Example B: FoCon (7) - As a Patient, I want to read only information relevant to me

#	**Question**	**Answer**	**Link**
1	Please propose a PROBLEM	As a Patient, I want to read only information relevant to me	Link Name (7) & Problem (7)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Filter on Geographic data	Solution (7)
5	Please propose an actor responsible for realising this solution	Software Developer	Ownership (7)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/
8	Please propose a SOLUTION that resolved the PROBLEM (2^{nd} Solution)	Filter on time-based data	Solution (7)
9	Please argue for/against the PROPOSITION	/	/
10	Please accept/reject the PROPOSITION	Accept	/
11	Please propose an actor responsible for realising this solution	Software Developer	Ownership (7)
12	Please argue for/against the PROPOSITION	/	/
13	Please accept/reject the PROPOSITION	Accept	/
14	Please propose a SOLUTION of which this SOLUTION is part of (for sub-solutions)	Filter on Geographic data	/
15	Please argue for/against the PROPOSITION	/	/
16	Please accept/reject the PROPOSITION	Accept	/

Continued Table 6.14: Example B: FoCon (7) - As a Patient, I want to read only information relevant to me

#	Question	Answer	Link
17	Please propose a SOLUTION that resolved the PROBLEM (3^{rd} Solution)	Filter on entity-relation data	Solution (7)
18	Please argue for/against the PROPOSITION	/	/
19	Please accept/reject the PROPOSITION	Accept	/
20	Please propose an actor responsible for realising this solution	Software Developer	Ownership (7)
21	Please argue for/against the PROPOSITION	/	/
22	Please accept/reject the PROPOSITION	Accept	/
23	Please propose a SOLUTION of which this SOLUTION is part of (for sub-solutions)	Filter on Geographic data	
24	Please argue for/against the PROPOSITION	/	/
25	Please accept/reject the PROPOSITION	Accept	/

Appendix E

Repository Case B

Table E.1: Case Example B - Table: Owners

Owner_ID	Actor_Name	Element_ID
1	Software Developer	1
2	Software Developer	2
3	Software Developer	3
4	Software Developer	4
5	Software Developer	5
6	Software Developer	6
7	Software Developer	7
8	Software Developer	8
9	Software Developer	9
10	Software Developer	10
11	Software Developer	11
12	Software Developer	12
13	Software Developer	13
14	Software Developer	14
15	Software Developer	15
16	Software Developer	16

Table E.2: Case Example B - Table: Part_Of

Part_Of_ID	Element_ID	Part_Of	Status
1	13	14	Accept
2	13	15	Accept

Table E.3: Case Example B - Table: Elements

Element_ID	Type_Element	Name	FoCon	Status
1	PROBLEM	As a Patient, I want to acquire knowledge about a rare disease	1	Accept
2	SOLUTION	Acquire information more from other experienced patients	1	Accept
3	PROBLEM	As a Patient, I want to know where to find information	2	Accept
4	SOLUTION	Use system that shares patient experience	2	Accept
5	PROBLEM	Empty Problem	3	Accept
6	SOLUTION	Visualise patient forum posts in system	3	Accept
7	PROBLEM	As a Patient, I want to avoid reading complex terminology	4	Accept
8	PROBLEM	Privacy sensitive data must be omitted from visualisations	5	Accept
9	PROBLEM	As a Patient, I want to limit the amount of information I must read	6	Accept
10	PROBLEM	As a Patient, I want to read only information relevant to me	7	Accept
11	SOLUTION	Reduce complexity information with language technology	4	Accept
12	SOLUTION	Create and adhere to privacy policy	5	Accept
13	SOLUTION	Summarize forum posts	6	Undefined
14	SOLUTION	Filter on geographic data	7	Accept
15	PROBLEM	Filter on time-based data	7	Accept
16	PROBLEM	Filter on geographic data	7	Accept

Table E.4: Case Example B - Table: Parent_Child

Parent_Child_ID	Element_ID	Parent	Status
1	1		Accept
2	2	1	Accept
3	3	2	Accept
4	4	3	Accept
5	5	4	Accept
6	6	5	Accept
7	7	6	Accept
8	8	6	Accept
9	9	6	Accept
10	10	6	Accept
11	11	7	Accept
12	12	8	Accept
13	13	9	Accept
14	14	10	Accept
15	15	10	Accept
16	16	10	Accept

Appendix F

FoCons Case C

Table F.1: Example C: FoCon (1.1) - As an Organisation, we want to reduce costs on machine maintenance

#	Question	Answer	Link
1	Please propose a PROBLEM	As an Organisation, we want to reduce costs on machine maintenance	Link Name (1) & Problem (1)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Perform predictive maintenance on machines	Solution (1)
6	Please propose an actor responsible for realising this solution	AI Developer	Ownership (1)
7	Please argue for/against the PROPOSITION	/	/
8	Please accept/reject the PROPOSITION	Accept	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
9	Please propose a SOLUTION that resulted from a SOLUTION of this FoCon	Perform predictive maintenance on machines : Conduct Service performance measurement	Link Name (2) & Solution (2) & Parent Of (1)
10	Please argue for/against the PROPOSITION	/	/
11	Please accept/reject the PROPOSITION	Accept	/
12	Please propose a SOLUTION that resulted from a SOLUTION of this FoCon	Perform predictive maintenance on machines : Perform Acoustic analysis	Link Name (3) & Solution (3) & Parent Of (1)
13	Please argue for/against the PROPOSITION	/	/
14	Please accept/reject the PROPOSITION	Accept	/
15	Please propose a SOLUTION that resulted from a SOLUTION of this FoCon	Perform predictive maintenance on machines : Perform Vibration analysis	Link Name (4) & Solution (4) & Parent Of (1)
16	Please argue for/against the PROPOSITION	/	/
17	Please accept/reject the PROPOSITION	Accept	/

Table F.2: Example C: FoCon (1.2) - Conduct Service performance measurement

#	**Question**	**Answer**	**Link**
1	Please propose a PROBLEM	/	Problem (2, 3, 4)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	/	/
4	Please propose a SOLUTION that resolved the PROBLEM	Conduct Service performance measurement	Link Name (2) & Solution (2)
5	Please propose an actor responsible for realising this solution	AI Developer	Ownership (2)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/
8	Please propose a SOLUTION that resolved the PROBLEM	Perform Acoustic analysis	Link Name (3) & Solution (3)
9	Please propose an actor responsible for realising this solution	AI Developer	Ownership (3)
10	Please argue for/against the PROPOSITION	/	/
11	Please accept/reject the PROPOSITION	Accept	/
12	Please propose a SOLUTION that resolved the PROBLEM	Perform Vibration analysis	Link Name (4) & Solution (4)
13	Please propose an actor responsible for realising this solution	AI Developer	Ownership (4)
14	Please argue for/against the PROPOSITION	/	/
15	Please accept/reject the PROPOSITION	Accept	/

Table F.3: Example C: FoCon (2.1) - As an Organisation, we want to reduce costs on cooling of data centers

#	Question	Answer	Link
1	Please propose a PROBLEM	As an Organisation, we want to reduce costs on cooling of data centers	Link Name (1) & Problem (1)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Apply AI for optimal cooling	Solution (1)
5	Please propose an actor responsible for realising this solution	AI Developer	Ownership (1)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
8	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Apply AI for optimal cooling : We lack sensor data	Link Name (2) & Problem (2) & Parent Of (1)
9	Please argue for/against the PROPOSITION	/	/
10	Please accept/reject the PROPOSITION	Accept	/
11	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Apply AI for optimal cooling : Temperature control is not digital	Link Name (3) & Problem (3) & Parent Of (1)
12	Please argue for/against the PROPOSITION	/	/
13	Please accept/reject the PROPOSITION	Accept	/

Table F.4: Example C: FoCon (2.2) - We lack sensor data

#	**Question**	**Answer**	**Link**
1	Please propose a PROBLEM	We lack sensor data	Link Name (2) & Problem (2)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Install extra sensors	Solution (2)
5	Please propose an actor responsible for realising this solution	Installation Engineer	Ownership (2)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
8	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Install extra sensors : Sensor data not in computer model	Link Name (4) & Problem (4) & Parent Of (2)
9	Please argue for/against the PROPOSITION	/	/
10	Please accept/reject the PROPOSITION	Accept	/

Table F.5: Example C: FoCon (2.3) - Temperature control is not digital

#	Question	Answer	Link
1	Please propose a PROBLEM	Temperature control is not digital	Link Name (3) & Problem (3)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Develop computer model for temperature control	Solution (3)
5	Please propose an actor responsible for realising this solution	AI Developer	Ownership (3)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
8	Please propose a PROBLEM that resulted from a SOLUTION of this FoCon	Develop computer model for temperature control : Sensor data not in computer model	Link Name (4) & Problem (4) & Parent Of (3)
9	Please argue for/against the PROPOSITION	/	/
10	Please accept/reject the PROPOSITION	Accept	/

Table F.6: Example C: FoCon (2.4) - Sensor data not in computer model

#	Question	Answer	Link
1	Please propose a PROBLEM	Sensor data not in computer model	Link Name (4) & Problem (4)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Merge sensor data into computer model	Solution (4)
5	Please propose an actor responsible for realising this solution	Data Engineer	Ownership (4)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/

Table F.7: Example C: FoCon (3.1) - As an Organisation, we want to improve the quality of our production process

#	Question	Answer	Link
1	Please propose a PROBLEM	As an Organisation, we want to improve the quality of our production process	Link Name (1) & Problem (1)
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	Accept	/
4	Please propose a SOLUTION that resolved the PROBLEM	Use AI to study production process	Solution (1)
5	Please propose an actor responsible for realising this solution	AI Developer	Ownership (1)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/
The section below initiates a new FoCon for any proposed problem or solution not yet defined			
8	Please propose a SOLUTION that resulted from a SOLUTION of this FoCon	Use AI to study production process : Apply Machine Vision to check quantity and size of items	Link Name (2) & Solution (2) & Parent Of (1)
9	Please argue for/against the PROPOSITION	/	/
10	Please accept/reject the PROPOSITION	Accept	/
11	Please propose a SOLUTION that resulted from a SOLUTION of this FoCon	Use AI to study production process : Apply Machine Vision to check for anomalies of shape or distribution	Link Name (3) & Solution (3) & Parent Of (1)
12	Please argue for/against the PROPOSITION	/	/
13	Please accept/reject the PROPOSITION	Accept	/

Table F.8: Example C: FoCon (3.2) - Apply Machine Vision to check quantity and size of items

#	Question	Answer	Link
1	Please propose a PROBLEM	/	/
2	Please argue for/against the PROPOSITION	/	/
3	Please accept/reject the PROPOSITION	/	/
4	Please propose a SOLUTION that resolved the PROBLEM	Apply Machine Vision to check quantity and size of items	Link Name (2) & Solution (2)
5	Please propose an actor responsible for realising this solution	AI Developer	Ownership (2)
6	Please argue for/against the PROPOSITION	/	/
7	Please accept/reject the PROPOSITION	Accept	/
8	Please propose a SOLUTION that resolved the PROBLEM (2nd Solution)	Apply Machine Vision to check for anomalies of shape or distribution	Link Name (3) & Solution (3)
9	Please propose an actor responsible for realising this solution	AI Developer	Ownership (3)
10	Please argue for/against the PROPOSITION	/	/
11	Please accept/reject the PROPOSITION	Accept	/

Appendix G

Repository Case C

Table G.1: Case Example C - Table: Owners

Owner_ID	Actor_Name	Element_ID
1	AI Developer	2
2	AI Developer	3
3	AI Developer	4
4	AI Developer	5
5	AI Developer	7
6	Installation Engineer	9
7	AI Developer	11
8	Data Engineer	13
9	AI Developer	15
10	AI Developer	16
11	AI Developer	17

Table G.2: Case Example C - Table: Part_Of

Part_Of_ID	Element_ID	Part_Of	Status

Table G.3: Case Example C - Table: Elements

Element_ID	Type_Element	Name	FoCon	Status
1	PROBLEM	As an Organisation, we want to reduce costs on machine maintenance	1.1	Accept
2	SOLUTION	Perform predictive maintenance on machines	1.1	Accept
3	SOLUTION	Conduct Service performance measurement	1.2	Accept
4	SOLUTION	Perform Acoustic analysis	1.2	Accept
5	SOLUTION	Perform Vibration analysis	1.2	Accept
6	PROBLEM	As an Organisation, we want to reduce costs on cooling of data centers	2.1	Accept
7	SOLUTION	Apply AI for optimal cooling	2.1	Accept
8	PROBLEM	We lack sensor data	2.2	Accept
9	SOLUTION	Install extra sensors	2.2	Accept
10	PROBLEM	Temperature control is not digital	2.3	Accept
11	SOLUTION	Develop computer model for temperature control	2.3	Accept
12	PROBLEM	Sensor data not in computer model	2.4	Accept
13	SOLUTION	Merge sensor data into computer model	2.4	Accept
14	PROBLEM	As an Organisation, we want to improve the quality of our production process	3.1	Accept
15	SOLUTION	Use AI to study production process	3.1	Accept
16	SOLUTION	Apply Machine Vision to check quantity and size of items	3.2	Accept
17	SOLUTION	Apply Machine Vision to check for anomalies of shape or distribution	3.2	Accept

Table G.4: Case Example C - Table: Parent_Child

Parent_Child_ID	Element_ID	Parent	Status
1	1		Accept
2	2	1	Accept
3	3	2	Accept
4	4	2	Accept
5	5	2	Accept
6	6		Accept
7	7	6	Accept
8	8	7	Accept
9	9	8	Accept
10	10	7	Accept
11	11	10	Accept
12	12	9	Accept
13	12	11	Accept
14	13	12	Accept
15	14		Accept
16	15	14	Accept
17	16	15	Accept
18	17	15	Accept