DESIGNING A MODEL-DRIVEN REPOSITORY FOR BEST PRACTICES ON ETHICAL, SOCIAL AND ENVIRONMENTAL TOPICS

Milo Plomp
5515599
m.plomp2@students.uu.nl

Utrecht University

A thesis presented for the MSc.
Business Informatics
July 8, 2020

Utrecht University
Department of Information and Computing Sciences

First supervisor: Dr. Sergio España
Second supervisor: Dr. Fabiano Dalpiaz

Acknowledgements

It has been over a year since I first contacted Dr. Sergio España about the possibility of writing my master's thesis under his supervision. He compiled a list of projects that he wanted to work on that we discussed shortly thereafter. I left our first meeting with a great sense of optimism and eagerness to start working on the project that I had chosen and that we were both really enthusiastic about. This enthusiasm has not faltered over the course of the 9 months that I have worked on this project and that is for a large part due to Sergio's supervision. Thank you for the enthusiasm and passion for the project that you also bestowed on me, the great ideas that helped me shape the project (even though they sometimes lead to lengthy, but useful, discussions) and the willingness to help me get the project to where I wanted it to be. I would also like to thank Dr. Fabiano Dalpiaz for his guidance. Although it is not necessarily a given that students are able to discuss their thesis project with their second supervisor, you were willing to help me improve the motivation for and validation of my thesis through some very useful discussions. Lastly, I would like to express my gratitude towards everyone else who collaborated with me on this research: Mariëlle Adèr for her collaboration on the interviews and support for any question or discussion, Zoë Coenen for her collaboration on the data model comparison, Stefan van der Pijl for his collaboration on tool development and my interview participants for sharing their expertise.

Abstract

In this research we study how organisations move through a continuous and iterative cycle to shift their practices to becoming increasingly responsible. We call this cycle the Business Ethics Continuous Improvement Cycle (BECIC). One of the phases in the BECIC is called improvement planning for ethical, social and environmental topics (IP4ESET), in which organisations work on improvement actions to improve their performance on previously defined areas. Although tools and other approaches exist already to help organisations with improvement planning, these approaches lack full support for the entire IP4ESET phase and are often considered to be too vague in order to be used for defining *how* improvement actions shoud be carried out. We propose a best practice repository (BPR) that stores best practices (BPs) on ESE topics. We believe that BPs are able to convey how improvement actions should be carried out and by storing them in a BPR to be used by multiple organisations, we improve knowledge sharing on ESE topics.

Other BPRs exist but often have a closed nature and lack the flexibility to be useful for a variety of organisations. We have developed our BPR using the model-driven development (MDD) paradigm in order to allow users to adapt the functionality to their needs. To this end, we have created a domain-specific language (DSL) that describes how users can construct models of BP structures. The DSL consists of a variety of models that have been created based on a domain analysis. We have implemented the DSL in a model editor that is included in our BPR. We have also implemented functionality that allows storage of BPs using the prescribed BP structure and filtering functionality to find relevant BPs.

In order to address the complexity of validating large-scale MDD approaches with long-term goals, we have created a method for the creation of a family of validations. This method allows for the creation of a related set of validation activities that together form an answer to how well the MDD system contributes to its goals. We have applied this method to our BPR and and created a family of validation activities that were carried out using an expert opinion session and single-case mechanism experiment. We have also planned the validation of a future version of the tool.

Keywords: *improvement planning, ethical social and environmental topics, best practice, repository, model-driven development, domain-specific language, conceptual modeling, meta-modeling, family of validations*

Contents

1 | Introduction

If we were to conduct a survey among a group of organisations asking what "sustainability" and "sustainable development" mean, we would likely get inconsistent answers. It seems that many organisations are working on improving their "sustainability performance", but no one is actually able to provide a coherent definition of the terminology. One of the terms that organisations often use to express their efforts towards "sustainability" is Corporate Social Responsibility (CSR), which generally encompasses efforts towards themes such as environmental issues and societal impact. Still, the actual meaning of CSR is undecided on, just like its implementation is undefined and unguided [17]. Furthermore, the motivation for addressing CSR varies between organisations. While factors such as cultural norms shape the business case for CSR in some organisations around moral and ethical themes [88], the motivation might be purely economically driven in other organisations [66; 67; 116]. In this research, we consider *responsible enterprises* as any enterprise/organisation that goes beyond legal obligations to act in accordance with ethical values that aim to benefit the environment and society, regardless of what the motivation is. By taking full advantage of the development of modern ICTs, the field of *responsible software* aims to enable enterprises to transition to becoming (more) responsible enterprises (or organisations) [48].

The process of becoming an increasingly responsible enterprise often involves a continuous improvement cycle, with the outcome of the cycles resulting in organisational reengineering. Adèr [14] proposes the business ethics continuous improvement cycle (BECIC) to structure this improvement cycle by describing four distinct phases. The output of every phase allows us to determine goals for ethical improvement, assess the current ethical performance of an organisation, plan for improvement and eventually follow up on the corrective actions. The BECIC cycle is depicted in figure 1.1. In this research, we focus on the improvement planning for ethical, social and environmental topics (IP4ESET) phase, in which organisations determine and plan the required actions for improving their performance on ethical, social and environmental (ESE) topics. The IP4ESET phase consists of several activities, such as the identification of *goals* and *objectives*), but in this research we are concerned with the *identification of action steps*. To support organisations in this identification, we propose a best practice repository (BPR) that stores best practices (BPs) on action steps successfully taken by other organisations.

1.1 Problem Statement

Although the identification of action steps is crucial in improving ESE performance, the identification alone is not enough. Organisations require not only knowledge about *what* to improve, but

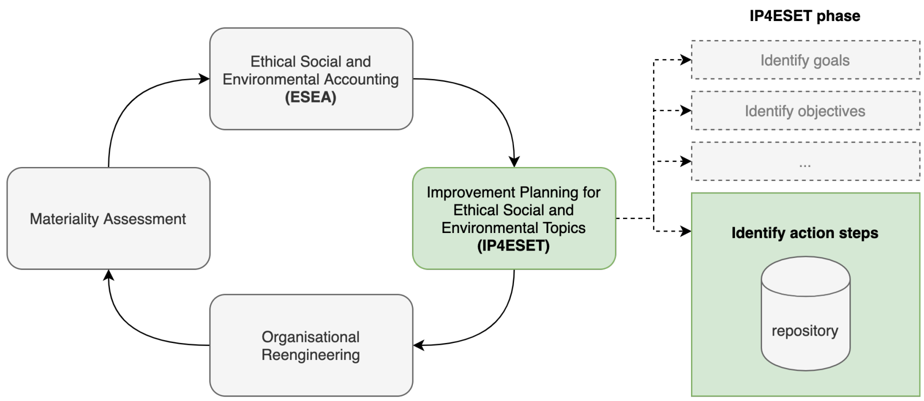

Figure 1.1: The repository in relation to the improvement cycle

also *how* to improve it. This lack of knowledge may lead to organisations only setting goals, but not following up on those goals [106]. We call the gap between intention of improving performance and taking concrete actions the *intention-action gap* [80]. In the field of general improvement planning (IP) and IP4ESET, several approaches have been proposed to support the implementation of action steps, these approaches are limited by a variety of shortcomings that will be addressed in this thesis.

Our proposed solution focuses on BPs as artefacts that allow organisations to get a grip on how to execute improvement actions. BPs for ESE topics are defined as a *"patterns that are proven solutions for problems in the three dimensions of sustainability"* and thus prescribe solutions to organisational problems based on what has worked in other organisational contexts. We propose the development of a repository that stores these BPs in order to promote knowledge sharing between organisations and to improve ESE performance [39; 48].

Other BPRs already exist, but all lack the flexibility to be useful for a wide variety of organisations. These BPRs either include only BPs from a single source, are not tailored to the specific functionality needs of organisations, or both. This rigidity of currently available tools decreases their usefulness and effectiveness.

1.2 Research Aim

In this research, we aim to design and develop a model-driven repository for BPs on ESE topics. This repository aims to contribute to solving the problems we observe in IP4ESET: organisations lacking guidance on *how* to improve their performance, lack of software tool support and the rigidity/closed nature of tools that do exist.

The model-driven aspect of our BPR will allow for functionality to be dependent on desires and needs of organisations. We will develop our BPR following the model-driven development (MDD) paradigm by letting the exact functionality of the BPR be dependent on textual models describing BP structures that are created by users. This approach will require us to research to state of the art and practice of organisational BPs in order to find out what the common structure of organisational BPs is and to design a domain-specific language (DSL) that prescribes rules for how to construct these textual models. We will develop a prototype of a BPR that motivates the possibility of creating a model-driven BPR for organisational BPs that allows organisations to improve their (ESE) performance.

In order to motivate the contribution of our research to the research goals, we will validate our BPR by addressing how well it contributes to the goals of its stakeholders. To address the complexity of stakeholder goals for validation, we will propose a method for decomposing the validation problem into manageable validation activities.

1.3 Research Questions

Based on the problem description above, we pose several research questions. Since the aim of this research is designing a BPR, we are faced with a design science problem [115]. We define the following problem statement, based on [115]: *improve organisations' transition to becoming responsible organisations by designing a repository for best practices so that responsible organisations can learn from other organisations.* It should be noted that we have not included the requirements of the artefact in this problem statement as these requirements are not known yet.

The research questions below are categorized according to the phases of the design cycle [115], which is explained in more detail in the next section.

RQ1 What are the shortcomings of current tools for improvement planning for ethical, social and environmental topics? *(Problem investigation)*

RQ2 What is the state-of-the-art and practice on (domain-independent) best practice repositories? *(Problem investigation)*

RQ3 How can we create a model-driven best practice repository that supports IP4ESET knowledge sharing across organisational contexts? *(Treatment design)*

- **RQ3.1** What are the requirements for a repository for IP4ESET best practices?
- **RQ3.2** How can we determine the applicability of IP4ESET best practices across organisational contexts?
- **RQ3.3** How can we create a domain specific language for best practices that can be interpreted in runtime?

RQ4 How can we validate the contribution of model-driven systems to their high-level goals? *(Treatment validation)*

RQ5 How well does a model-driven best practice repository contribute to its high-level goals? *(Treatment validation)*

2 | Conceptual Framework

2.1 Responsible Enterprises and CSR

2.1.1 Corporate Social Responsibility

Responsible enterprises are enterprises (organisations) that take care in considering the society and the environment in conducting their activities, beyond legal obligations [48]. Responsible organisations[1] pay particular attention to corporate social responsibility (CSR): the responsibility that organisations have for their impact on society and the environment [50; 114]. The actual meaning of CSR has long been contested and consequently, its implementation has been difficult to define [17]. The contested meaning of CSR has been attributed to national differences by [88]. According to them, all corporations ultimately pursue *economic benefit*, but the strategies used to reach those benefits are colored by a larger context: institutional frameworks. For example, the presence or absence of social systems might influence the expectation that corporations take responsibility, and cultural norms might influence social participation and philanthropy. Matten & Moon [88] therefore make a distinction between *implicit CSR* and *explicit CSR*. The former is steered by values, norms, and rules stemming from differences in national systems, while the latter is a voluntary and often strategic choice. Both types of CSR can exist in the same organisation [17], but a rapid shift is seen from implicit CSR to explicit CSR [88]. This form of CSR is based on voluntary actions, but built on a business case that also addresses profitability [66; 67; 116]. Profitability as a result of addressing CSR can stem from the attraction and retention of customers, but also employee motivation as a result of an ethical work climate [61]. We cannot ignore either moral or strategic motives and therefore follow the definition of CSR by [50], which does not distinguish between strategy and ethical beliefs as being the main required driving force behind CSR.

CSR

In addressing Corporate Social Responsibility (CSR), organisations take action towards positive change for society and the environment based on strategic and/or moral beliefs [50].

[1] Although the definition of “responsible enterprises” encompasses several types of organisations, we will refer to responsible “organisations” for the remainder of this thesis [48].

Sustainable Development

According to [45], the notion of CSR is based on sustainable development: the ability of organisations to develop without compromising future generations. The definition of sustainable development has been criticized for lacking clarity on what sustainability actually entails [92], but an attempt to operationalise sustainable development has been made with the definition of the triple bottom line: there is a need for companies to have a business case (economic), natural case (environmental) and societal case (social). Again, we see a combination of moral and economic drivers.

In order to address sustainable development, organisations need to address all three perspectives using strategic management [46]. Strategic management has been mentioned as a requirement for making an actual change: *"if sustainable development is to be a useful and implementable concept, it must enter the field of decision-making and must be considered as a "decision-making strategy"."* [113]. There is no one solution for this strategic decision-making process to address CSR that works for all organisations, and the way organisations should address the elements of the triple bottom line is dependent on various internal and external drivers [46].

Sustainable Development

Responsible organisations work on sustainable development by addressing the economic, social and environmental aspects of the triple bottom line in order to develop without compromising future generations [45]. Actions are operationalised using strategic management.

However, when we talk about sustainable development, we do have to consider the contested meaning of the term "sustainability". Is it actually possible for any organisation to be sustainable? According to Gray [62], we have to look at sustainability as a systems-based concept, meaning that we can never work towards improvement in isolation. For example, considering the environmental perspective of sustainability, we have to measure impact on the level of ecosystems. The complex nature of sustainability does not allow any organisation to become "sustainable"; there are always larger processes at play. For the remainder of this thesis, we will refrain from stating that we are working on BPs for sustainability topics, but rather *ethical, social and environmental* (ESE) topics. Therefore, we also do not aim to create a solution for organisations to become "sustainable", but rather to become "responsible".

Drivers for CSR

Although many theories are skeptical towards the moral drivers for addressing CSR ([57], for example), not all theories presuppose economic purposes to be the starting point for a CSR strategy. By researching drivers for addressing CSR in India, Dhanesh [43] has found both moral (mostly driven by founders' visions) and economic imperatives (creating goodwill and attracting customers) to be driving forces. This finding is supported by Graafland & van de Ven [61]. By researching drivers for CSR among companies in the Netherlands, the authors found that the attention to CSR is driven mostly by intrinsic motivation: the majority of companies saw a moral duty towards society to address CSR. In these researches, the companies have a stronger focus on the societal and environmental aspects of the triple bottom line. Although the motivation to address CSR exists for both small and large companies, the motivation to rely on formal methods to operationalise CSR strategies is less present for smaller companies [61]. Smaller companies sometimes lack the financial resources to adopt practices that larger firms can easily apply [116], but the application of responsible practices does not rely solely on formal methods.

Company Definitions

We mentioned that responsible organisations pay particular attention to addressing CSR. Although that is theoretically true, not all companies explicitly mention "CSR" or "social responsiblity" (SR) as a formal term in their communication on environmental and social responsibility. For example, Fairphone states that they are formally recognized as a B corporation[2] [51], which defines high social and environmental performance standards, but otherwise no formal mention of CSR is given. Similarly, Tony Chocolonely mentions the use of formal systems and frameworks [108], but do not explicitly mention CSR.

2.2 Responsible Software

Responsible software is software that helps organisations in becoming increasingly responsible organisations by supporting responsible work practices [48; 36]. Examples of responsible software are software for sustainability modeling and reporting to aid decision making [15], enterprise resource planning (ERP) systems to support sustainable value chains [36], and software that supports organisational reengineering (transitioning to becoming more responsible) [48]. Boudreau et al. [28] refer to these technologies as *Green information systems (IS)*: systems that aim to contribute to sustainable business processes. They make a distinction with *Green information technology (IT)*, which places a greater focus on energy efficiency of systems. This distinction is described by España & Brinkkemper [48] as the first (Green IT) and second (Green IS) wave of sustainable ICT.

Harmon & Demirkan [67] describe a shift from *green IT* to *sustainable IT*. Where the former is a product-oriented approach defined by a reduction of energy use, the latter is a service-oriented approach that leverages the power of IT to drive sustainable strategies. By using sustainable IT services (SITS) strategies, organisations are able to address four sustainability pillars: ecological environment (e.g. applications for green supply chains), regulatory environment (meeting SITS compliance standards), social responsibility (e.g. systems and strategies for stakeholder engagement), and economic (e.g. sustainability reporting).

Responsible Software

A form of *Green IS*; software that allows organisations to adopt responsible work practices.

[2] https://bcorporation.net/

2.3 Variability in Software Development

In this research we take a flexible approach to the functionality of the BPR that we are developing. We aim to let users decide on the exact functionality of the BPR. This section will highlight approaches to addressing variability in software development.

2.3.1 Software Product Lines

The need for variability in software development is addressed in the paradigm of Software Product Lines (SPLs). SPLs constitute a set of systems that share a common set of assets [65]. Although this enables developers to create systems that tailor to variable market needs, the assets need to be predefined. This decreases the speed of adaptability and increases the complexity of upfront development decisions to be made [65; 111]. By moving the variability capabilities of SPLs to run-time, dynamic SPLs (DSPLs) allow for the configuration of systems by enabling or disabling features along a set of rules [104]. Although this deceases the amount of configuration decisions to be made by developers, there still is the need for upfront development efforts for the various system components. Furthermore, the variability in SPLs manifests itself in the creation of separate (related) systems [65]. Our aim is to tailor to variability within a singular system.

2.3.2 Model-Driven Development

Model-Driven Engineering (MDE) is the use of models as the leading artefacts in the software engineering process. This differs from traditional approaches to software engineering, where models are mainly used to guide conversations between developers and where code is the leading artefact [54]. The use of models on various abstraction levels in the MDE paradigm leads to more abstract reasoning about software development and maintenance [70]. On each abstraction level, there is a model that expresses some aspect of the system specification, including domain-specific information and system requirements. Through model transformations we can move from the highest abstraction level to increasingly specific levels and eventually to code [112]. The benefit of this sequence of models is its adaptability to changes [55]; any changes to aspects on a lower level of abstraction does not influence the higher levels of abstraction.

Model-Driven Development (MDD) is a subset of MDE, in the sense that MDE encompasses more model-based tasks that make up the larger scoped engineering process. MDD focuses on the description of a system using abstract models and the transformation of those models into the system implementation (the code) by a model compiler or the execution of those models by a model interpreter [99]. The use of models and model transformations to influence system implementation and/or execution increases flexibility in the development process [16].

A third view in the model-driven paradigm is Model-Driven Architecture (MDA). MDA is the view of the Object Management Group (OMG) on MDD. It promotes the use of abstract models that are independent of the implementation platform and specified along the MOF language to promote model exchange [16; 55; 76]. The positioning of MDE, MDD and MDA in relation to each other can be seen in figure 2.1.

Many benefits are attributed to MDD over traditional software development, such as increased productivity of developers and reliability of the resulting implementation [99]. In regards to this research, the main benefit of MDD is the high variability it enables. The use of abstract models

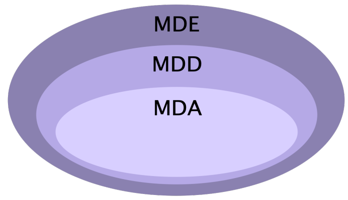

Figure 2.1: Positioning of MDE, MDD and MDA. Adapted by the author from [16]

ensures that the choice of platform and technology does not affect the models and vice versa; i.e. models are not subject to changes on the implementation level [99; 55]. It also ensures that system specification does not have to be fully realised at design-time. Taking an MDD approach to the development of the BPR allows us to create variability in the functionality of the BPR, increasing the chance of the BPR being useful for a large group of organisations and thereby increasing the chance of the BPR contributing to its intended goal of improving knowledge sharing and performance concerning ESE topics.

3 | Research Method

Our research method is based on the design cycle described by Wieringa [115]. The design cycle consists of three phases that are relevant for this study: problem investigation, treatment design, and treatment validation. A full overview of the research method can be seen in figure 3.1. This overview uses the process deliverable diagram (PDD) notation to define activities (left side) and deliverables (right side) [109].

3.1 Research Method Overview

Problem Investigation The goal of the problem investigation phase is to get clear picture of the current state of the research domain in which we are developing our tool. The first step in problem investigation is to define a conceptual framework that is able to define relevant concepts of the domain. A literature study will then proceed to explain what the current state of the research domain on IP4ESET is. This literature study consists of multiple parts and is structured using a literature study protocol. To further detail the state of the domain of BPRs, a set of data models of BPR content will be created in joint research. The comparison results of the various data models is used as input for the creation of a domain-specific language (DSL) in the next phase. The problem investigation phase concludes with interviews with experts in the field of IP4ESET to further define problems and assess shortcomings.

Treatment Design In the treatment design phase, we will develop our BPR. Requirements for the BPR are elicited using the interview and multivocal literature study results of the problem investigation phase. The BPR alters its functionality based on models that are described using a DSL. This DSL will be created based on the domain model that is created as a result of the data model comparison.

Treatment Validation Once the development of the BPR has finished (for this research), the current functionality will be validated to measure its expected contribution to stakeholder goals. As part of the treatment validation phase, a validation method will be created that allows us to construct multiple validation approaches that each validate part of the system. The execution of these validation approaches results in validation results that answer how parts of the BPR will contribute to parts of its overall goal.

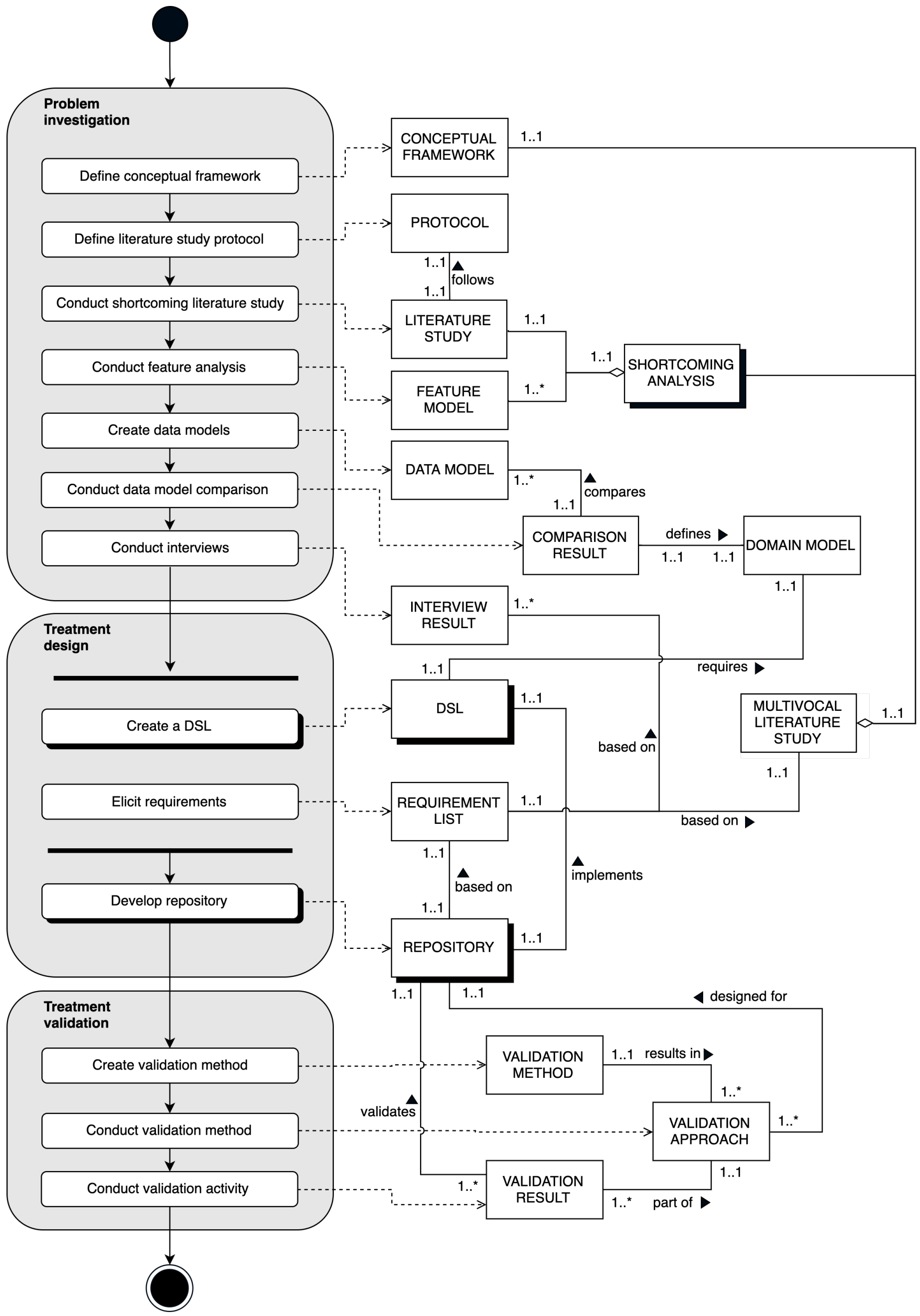

Figure 3.1: Research method

3.2 Literature Review for Problem Investigation

A thorough and fair method for identifying gaps in current literature is a systematic literature review, as it requires the definition of a search strategy and the reporting of results that both do and do not match the preferred hypothesis [75]. However, SLRs have been critiqued for inadequately providing insight into the "state of the practice" in software engineering (SE) [58]. The vast body of knowledge that SE practitioners hold is rarely published academically and can therefore be assessed using a multivocal literature review (MLR): a systematic review in which formal literature (academic and published) is reviewed alongside "grey" literature (non-academic and non-published) [58]. This approach is useful for assessing the shortcomings of current IP4ESET tools (*RQ1*), but an MLR has already been conducted to find IP4ESET tools in other research [14], making it redundant to conduct another MLR. Rather, we will use the findings from this MLR as input for our analysis. The results of the MLR will provide us with a list of IP4ESET tools to be analysed, and a starting point for reviewing IP4ESET tool requirements and shortcomings using the snowballing technique. An overview of the literature review methods used for problem investigation is given in table 3.1.

Table 3.1: Literature review methods

Element	What	Method
Conceptual framework	Concepts definition	*Informal:* a non-structured method using snowballing (non-exhaustive) and serendipitous findings, providing a basic framework for interpreting other results
RQ1	Scientific literature review	*Snowballing*: determining shortcomings of current IP4ESET approaches mentioned in scientific literature using a structured snowballing procedure. The findings of the MLR in [14] will be used as a starting point.
	Feature analysis	*MLR*: doing a feature analysis of tools selected by [14; 103], in which an MLR was done, and comparing to the super method created in [14].
	Grey literature review	*Informal:* using serendipitous findings to compare the state-of-the-practice to problems found in the state-of-the-art.
RQ2	State-of-the-art and practice	*Supervised work:* consolidation of work on this theme supervised by the author

The following sections will elaborate on these review methods, starting with the shortcoming analysis consisting of the three elements for answering RQ1.

3.3 Shortcoming Analysis

Other approaches to IP4ESET currently exist, albeit with some shortcomings that we wish to address in this research. In order to provide support for the shortcomings of current IP4ESET approaches, we will conduct a shortcoming analysis consisting of three parts. First, the method for a previously conducted literature review will be described, after which the three parts of the shortcoming analysis will be explained.

3.3.1 Tools and Methods for IP4ESET

In similar research, an MLR has been conducted to identify tools and methods for IP4ESET and improvement planning in general [14]. In that research, the following search strings were used on Google and Google Scholar:

S1 Improvement planning
AND
(method *OR* tool)

S2 (Sustainability *OR* business ethics *OR* Corporate Social Responsibility *OR* fair)
AND
improvement planning
AND
(method *OR* tool)

This MLR has produced a list of tools and methods for improvement planning in domains related to ESE topics and domains unrelated to ESE topics. Approaches that were unrelated to ESE topics were rejected for the present study. Of the remaining results, one approach only described an outline for an improvement plan and was thus also rejected, resulting in the following set of IP4ESET tools and methods:

Table 3.2: List of IP4ESET approaches selected for analysis

ID	Name	URL
SAPT	Sustainability Action Plan Tool	`http://bit.ly/SAPTforSIP`
PSAT	Program Sustainability Assessment Tool	`https://www.sustaintool.org/psat/`
CEAT	Community Environmental Action Toolkit	`http://bit.ly/CEATforSIP`

The research by Adèr [14] aimed to identify these approaches, whereas we aim to assess these approaches for *shortcomings*. The findings of this MLR will therefore be used as a starting point for our shortcoming analysis. The approaches listed in table 3.2 are used as input for the literature review described in the following section.

3.3.2 Shortcoming Analysis: Scientific Literature

We conduct a literature study to find shortcomings and requirements of/for IP4ESET tools, frameworks and methods described in literature. We use the snowballing technique to select papers for analysis [117]. In this phase of the shortcoming analysis, only scientific sources are included. Other non-scientific sources and serendipitous findings are included in the following two phases.

Selecting a Start Set of Papers

A start set of papers is chosen for the snowballing procedure. This set of papers is selected by searching for specific search terms on Google Scholar and excluding papers based on certain criteria. The search terms are based on the results of [14] and more general findings from the conceptual framework review. The following search strings are used on Google Scholar:

- Based on [14]:
  - *Sustainability Action Plan Tool*
  - *Program Sustainability Assessment Tool*
  - *Community Environmental Action Toolkit*
- Based on findings from the conceptual framework review:
  - *ISO 14001*
  - *ISO 26000 shortcomings*
- General search term to find shortcomings of tools for sustainable development[1]:
  - *Shortcomings tools sustainable development*

Based on the search results for these strings, only the results on the first page of Google Scholar are analysed for inclusion in the start set based on the following *exclusion* criteria:

CF1 Does not mention an IP4ESET approach

CF2 Domain is too specific or unrelated

CF3 Different definition of sustainability

CF4 Not an academic source

CF5 Inaccessible

CF6 Does not mention shortcomings, requirements or proposed solutions for SIP

The resulting set of papers are then read more thoroughly and again excluded if they meet one of the exclusion criteria. The then resulting tentative start set of papers are read in full and included in the final start set of papers if they are included in the shortcoming analysis [117].

Snowballing Procedure

We use the start set of papers to start the snowballing procedure. Based on [117], only papers that are selected and used for analysis are included in the snowballing procedure. Two iteration types are used for finding new papers:

- *Backward snowballing.* The reference list of papers is used to search for new relevant papers to include.
- *Forward snowballing.* Looking at which papers cite the currently assessed paper to find new papers to include.

[1] We use the term “sustainable development” due to its use being more widespread than “improvement planning”

The snowballing procedure is continued until no new papers are found [117].

Analysing Shortcomings and Requirements

The selected papers are analysed for mentions of shortcomings and/or requirements of/for IP4ESET approaches. For each paper, the approaches are tracked using the following metrics:

- *Solution type.* Examples include tools, frameworks and standards.
- *Solution name.*
- *Observed shortcomings.* A description of the shortcomings of the discussed approach, as described by the paper author(s).
- *Observed requirements.* A description of the requirements for IP4ESET approaches. These requirements are either explicitly mentioned by paper authors, or implicitly described through a discussion of shortcomings.
- *Shortcoming/requirement type.* We categorise shortcomings and requirements to be able to give a quantitative analysis of discussed shortcomings and requirements.

The result of this analysis is an in-depth overview of the shortcomings of the state-of-the-art of IP4ESET approaches. The feature analysis described in the following subsection will look at the shortcomings of the state-of-the-practice. The output of the phases of the shortcoming analysis are also depicted in figure 3.2; the result of this phase is “observed shortcomings and proposed requirements”, acting as input for the feature analysis in the next phase.

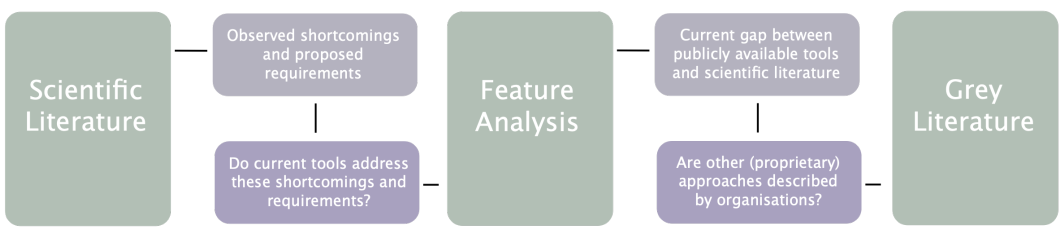

Figure 3.2: Shortcoming analysis flow

3.3.3 Shortcoming Analysis: Feature Analysis

The analysis of scientific literature described in the previous subsection brings to light the state-of-the-art and its shortcomings, but does not necessarily describe the shortcomings of the state-of-the-practice (i.e. tools that are currently being used by organisations). Other tools to address IP4ESET currently exist, albeit with some shortcomings that we wish to address in this research. By conducting a domain analysis on the domain of current IP4ESET tools, we analyse commonality across various existing systems, and are able to reuse successful functionalities of existing systems [74]. Furthermore, by doing a feature-based analysis of current systems, we are able to add empirical evidence to the findings of the previously described literature review on the shortcomings of current tools for SIP.

Tool Selection

We conduct a feature analysis on several IP4ESET tools to analyse the functionality (and shortcomings) of these tools. The tools that are selected for this analysis are based on previous research by Adèr [14], who has conducted an MLR using two search strings. From this list of tools and methods, only PSAT remained to be analysed in the present study. Please note that we have also selected the B Impact Assessment tool for this analysis, despite it not showing in the results of the MLR. The full list of selected IP4ESET tools can be seen in table 3.3.

Table 3.3: List of selected IP4ESET tools

ID	Name	URL
BIA	B Impact Assessment Tool	`https://bimpactassessment.net/`
PSAT	Program Sustainability Assessment Tool	`https://www.sustaintool.org/psat/`

Feature Analysis

Feature models are created for both tools and compared to desired functionality. The desired functionality is based on the activities and deliverables of the IP4ESET super method [14]. Please refer to appendix A for the process deliverable diagram (PDD) depicting the activities and deliverables of the IP4ESET phase.

We define a *feature* as follows:

Feature

"A prominent or distinctive user-visible aspect, quality, or characteristic of a software system or systems." [74]

This definition of a feature focuses on the problem space (user-centered requirements) rather than the solution space (system-centered requirements) [38]. Focusing on the role of users is crucial for domain analysis, which can subsequently be conducted using feature models [74]. Feature models are created using the definition of a feature by Kang et al. [74], but extended by Pelle [100] who distinguishes between regular features, action-based features (which requires the user to perform an action based on the result of an activity), and attributes (which indicate options for features).

Feature Mapping Based on the IP4ESET super method [14], we use the inclusion criteria in table 3.4 to map features identified in current IP4ESET tools to the IP4ESET activity they contribute to.

Table 3.4: Inclusion criteria for mapping features to the SIP super method

Activity name	ID	Criterium
Identify target area of improvement	C1.1	Feature should allow user to *assess* area(s) for improvement
		OR
	C1.2	Feature should allow user to *select* area(s) of improvement
Identify goals	C2.1	Feature should allow user to *identify* a goal
Identify objectives	C3.1	Feature should allow user to *set an aim* for a goal
		OR
	C3.2	Feature should allow user to *set a purpose* for a goal
Identify action steps	C4.1.1	Feature should require user to *set resources* for an action step
		AND
	C4.1.2	Feature should require user to *assign a responsibility* for an action step
Identify staff responsibilities	C5.1	Feature should allow user to *identify responsibilities* for action steps
Identify resources	C6.1	Feature should allow user to *select resources* required to realise an action step
		OR
	C6.2	Feature should allow user to *identify resources* required to realise an action step
Document improvement plan	C7.1.1	Feature should allow users to document an improvement plan, including action steps
		AND
	C7.1.2	Feature should allow users to document an improvement plan, including objectives
		AND
	C7.1.3	Feature should allow users to document an improvement plan, including goals

Any feature that is *supportive* for meeting an inclusion criterium is also included. For activities in which several criteria are distinguished using an *OR* indicator, only one criterium has to be met for the tool to be mapped to that activity. For activities in which criteria are distinguished using an *AND* indicator, the tool needs to meet all criteria in order to be mapped. An ideal IP4ESET tool would meet all mapping criteria.

3.3.4 Shortcoming Analysis: Grey Literature

To conclude the shortcoming analysis, grey literature describing IP4ESET approaches will be analysed for shortcomings. Unlike the selection of scientific literature, no formal approach to the search and selection of grey literature will be used, due to the lack of results in a previous (multivocal) literature review [14]. Grey literature will be selected as serendipitous findings, primarily based on interviews that will be conducted. The structure of these interviews will be explained in a later section.

The goal of the analysis of grey literature is to find if the state-of-the-practice (approaches described by the industry) addresses the shortcomings of the state-of-the-art. Grey literature may be any (online) source, such as guidelines, whitepapers, solution papers and case descriptions. We define the following inclusion criteria for grey literature:

CG1 Source is freely accessible.

CG2 Source describes an approach to IP4ESET (guidance on defining and executing action steps)

CG3 Source describes a replicable solution.

CG4 Source does not describe an approach to one single, predefined sustainability issue.

3.4 State-of-the-art on Best Practice Repositories

In research supervised by the author [39], the state-of-the-art and practice on best practices and best practice repositories has been researched. To this end, a multivocal literature review, feature analysis of best practice repositories, content analysis of best practices, and interviews have been conducted. This section will briefly elaborate on the method for this research. Please refer to [39] for a more extensive description.

3.4.1 Multivocal Literature Review

Sources for analysis were collected using a snowballing procedure by searching for the following search strings:

S1 Digital repository

S2 Digital repository for (best practices *OR* patterns)

S3 (Best practice *OR* pattern) repository

S4 (Best practices *OR* patterns) on (sustainability *OR* business ethics)

Based on the resulting set of papers, an analysis has been conducted to create a conceptual framework (both textual and as a UML clas diagram). No analysis structure has been used for the conceptual framework, but some descriptive statistics have been applied on the findings, describing the concepts defined in the various papers.

3.4.2 Feature and Content Analysis

In [39], 8 BPRs have been analysed based on their features (by creating feature diagrams) and the contents of the BPs they are storing (by creating UML data models). The feature analysis takes a similar approach and notation to the present research. Feature models are created for each selected repository (*"case-based"* feature models) and compared to a feature model created based on descriptions of BPRs found in scientific literature (*"literature-based"* feature model). The content of these repositories (consisting of BPs) are modeled using the UML class diagram notation. Again, these models are compared to a literature-based UML model.

The features in all case-based feature models are mapped to the features found in the literature-based feature model. The same approach is used to map elements of the case-based data models to the literature-based data model. This approach highlights gaps between case-based findings and literature-based findings.

The data models are created in collaboration with the author of the present research and are used as input for the domain analysis described in the following section.

3.4.3 Interviews

Three interviews have been conducted with creators of BPRs to discuss shortcomings and requirements for other BPRs. The interviews took a structured approach (the interview protocol and transcripts can be found in [39]) and were aimed at the the design and the use of the BPRs. The findings of these interviews can be used to shape the development of our repository.

3.5 Defining a Language for Best Practices

In section 3.4.2 we described how research supervised by the author has produced an analysis of the contents of BPRs. By comparing the data structure of BPs in these BPRs, we are able to indicate the level of variability in BPs. We aim to create a BPR that allows for sharing BPs across domains and organisational boundaries and therefore need to address this variability. In this section, we elaborate on the creation of a language for defining the structure of BPs. This language is an attempt to provide a common structure for defining BPs, while taking the variability of the domain into consideration.

3.5.1 Creation of a DSL

A domain-specific language (DSL) is a language that, different from general purpose languages (GPL), increases expressiveness in a limited domain [90]. Examples of GPLs are Java and UML, of which the latter is a general-purpose modeling language. Benefits of implementing a DSL include the expression of domain-specific constructs [31], which allows DSL developers to limit the concepts in the language to concepts that are specific to the domain in question. This is useful in this research, as we are aiming to tailor our DSL to the concepts that we have observed in the analysed domain of BPs; e.g. *which concepts do we need to construct a BP and how are these concepts* *related?*. Following España et al. [49], we apply the first five DSL development phases defined by Mernik et al. [90]: decision, domain analysis, design, implementation and testing. The first step in developing a DSL is the decision phase, in which we provide the following rationale for deciding on the development of a DSL.

Choosing to develop a DSL

Based on the decision patterns defined in [90], we base our choice for the development of a DSL on the following decision patterns:

- *System front-end:* the DSL facilitates system configuration.
- *GUI construction:* the DSL facilitates GUI construction.

The BPR that we create in this research takes a flexible approach to its functionality, as this functionality can be adapted to user needs. The functionality is dependent on the structure of BPs that users want to store and this structure can be described by users in models. The DSL we develop will be able to specify how these models can be structured. Therefore, the DSL is used for system configuration. Based on the structure of BPs, the GUI of the tool is changed to include relevant features for storing BPs along the specified structure.

3.5.2 Domain Analysis

Since a DSL increases expressiveness in a domain, that domain needs to be analysed. In the analysis phase, we gather a variety of sources to identify the problem domain and to gather knowledge about the domain in question [90]. No limitations or requirements for domain analysis input exists, so the input can take several forms. In this research, the *formal analysis pattern* specified by Mernik et al. [90] is used, which results in the creation of a domain model that specifies terminology and commonality of domain concepts. The same approach has been used by España et al. [49], where the domain analysis resulted in the development of a conceptual metamodel. To this end, the authors collected, among other things, process-deliverable diagrams (PDDs) describing ESEA methods with their activities and deliverables. In the present research, we do not aim to describe methods. Since the DSL will define the structure of repository instances that store BPs, our domain analysis should result in the construction of a domain model based on UML models describing the *structure* of these BPs.

Defining Best Practice Structures

For the domain analysis, we take input in the form of UML class diagrams defining the structure of BPs. These UML models have been created in other research, supervised by the author [39]. The methodology of using UML (a general-purpose modeling language) to develop a DSL has been described by Brucker & Doser [31]. We create UML models of BPs found in BPRs analysed by Coenen [39]. Each model is created by both authors separately and then discussed and updated. This approach is iterated until an agreement is reached on the structure of the models.

Model Construction

The result of the domain analysis phase is a domain model that expresses commonalities and variations in the domain [90]. This domain model allows us to describe the *abstract syntax* of the DSL by specifying how concepts related to a BP are related to each other. We will also specify a metamodel of the domain model. This metamodel is also part of the abstract syntax of our DSL, as it further specifies the relationships between BP-related concepts and defines how the concepts in the domain model may be extended. We will also specify the *concrete* syntax of the DSL, which specifies the notation of the language.

Concrete and Abstract Syntax

We define the abstract and concrete syntax as follows:

- *Abstract syntax.* Describes the structure of concepts in the language. In this research, the DSL is used to create textual models that reflect BP structures. The abstract syntax therefore defines how the concepts in BP structures are related to each other. The abstract syntax is created as a result of the domain analysis.
- *Concrete syntax.* Users of the BPR need to create textual models of BP structures that are used by the BPR to adapt its functionality. The concrete syntax of the DSL defines the notation of these models.

In defining the domain model, we take an approach inspired by [14; 49], in which we analyse the UML models on commonality. For the sake of addressing variability between the models, we will split the analysis in three parts:

- *Concepts.* We will analyse the commonality of concepts in the models inform our decision for inclusion.
- *Attributes.* Similar concepts may have different attributes in the various models. Our decision for included attributes will be based on a commonality analysis.
- *Relationships.* Cardinality of relationships influence whether a concept should be considered as mandatory or recommended. Including this analysis is required for the construction of the domain model, which indicates mandatory and recommended concepts for the textual models.

3.6 Stakeholder Interviews

In order to elicit requirements for the BPR, we will conduct interviews with experts in the field of IP4ESET. Experts are working for organisations that address CSR in their strategies and aim to improve performance on ESE topics. We conduct these interviews as part of *problem investigation* and *treatment design* and its goals are therefore twofold:

- *Assessing IP4ESET practices.* We wish to relate the findings of the literature review to practices that are actually carried out by organisations. This is related to *problem investigation* as it might bring to light (further) shortcomings of IP4ESET approaches or information on the use of BPRs.

- *Collecting requirements for a BPR.* Experts are able to express requirements for the repository that is to be developed by indicating what their goals would be for using the system.

The interview materials (questions and informed consent forms) are listed in appendix B. We take a structured approach to these interviews by specifying all interview questions beforehand, although we will leave room for further discussion when desired. These interviews are conducted in collaboration with another researcher and therefore also used in other research [14].

3.6.1 Transcribing and Coding

Interviews will recorded with permission of the participants and later transcribed. Interview transcripts will be coded in NVivo[2] along a predefined coding scheme. The coding scheme is listed in appendix C. Three main topics will be analysed: *improvement planning general*, *improvement planning tools* and *best practice repositories*. Coding utterances in the interviews along these nodes helps in understanding both the methods and shortcomings for current IP approaches and the sentiment towards BPs and BPRs.

[2] `https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home`

4 | Problem Investigation

This chapter will elaborate on the results of activities described in the previous chapter. We will present the findings of the shortcoming analysis (consisting of the scientific literature review, feature analysis and grey literature review), discuss the state-of-the-art and practice on BPRs, present interview results and discuss the inclusion of contextual factors in BPs.

4.1 Shortcoming Analysis

4.1.1 Snowballing Procedure

The use of the search strings listed in section 4.1.2 resulted in an initial set of 55 papers (we excluded quotes) that were judged for inclusion. After an inspection based on the exclusion criteria, 13 papers remained that formed the start set. This start set has been consequently used to start the snowballing procedure. Figure 4.1 shows the instance count of exclusion criteria present in the analysed papers (multiple exclusion criteria could be present in each paper):

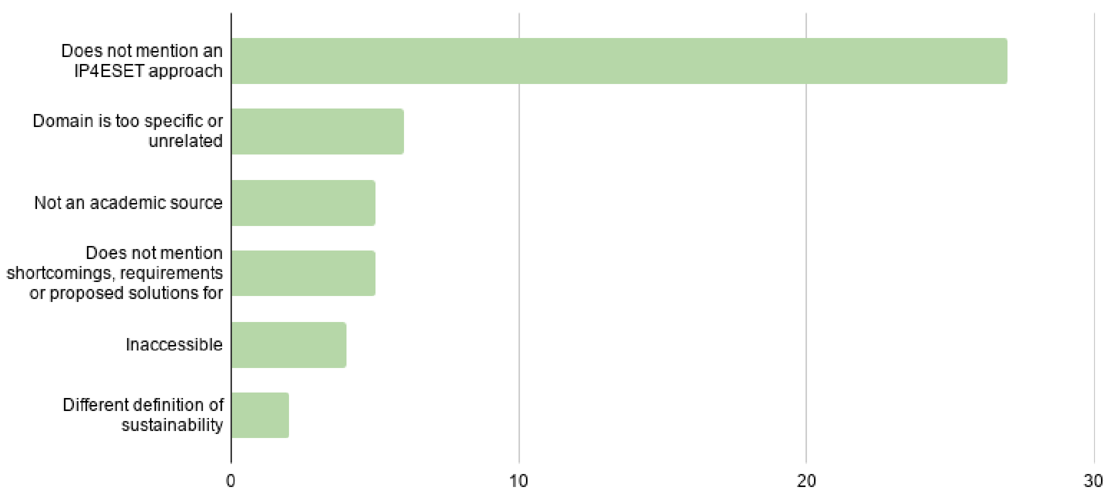

Figure 4.1: Instance count of exclusion criteria

The snowballing procedure was continued until no new and relevant papers were found. The amount of papers analysed in each round of the snowballing procedure is represented in figure 4.2. This does not indicate all papers judged for inclusion, but rather the papers that were eventually determined to be relevant and therefore included in the analysis. The results of the snowballing procedure, including the full list of analysed papers, can be accessed in an online spreadsheet[1].

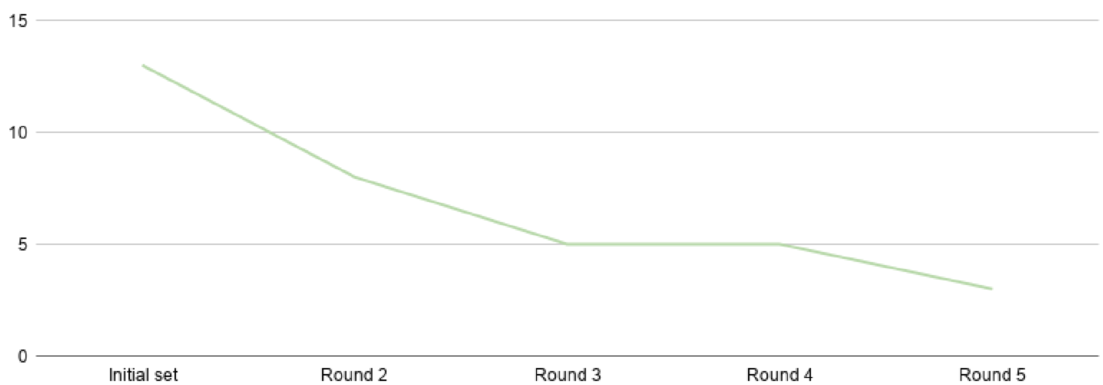

Figure 4.2: Amount of papers analysed in the snowballing procedure

4.1.2 Scientific Literature

34 papers have been analysed for mentioning shortcomings and/or requirements/solutions for current IP4ESET tools. Since many approaches to addressing IP4ESET mentioned in literature do not necessarily take the form of an IT tool, we also include other (possibly non-IT supported) approaches, such as methods and frameworks. Shortcomings are included if they are explicitly mentioned in the analysed papers, whereas requirements may be mentioned implicitly by authors stating how an approach addresses a certain problem in SIP. Table D.1 in appendix D shows the types and instances of IP4ESET approaches discussed in the selected papers. Table D.2 shows all discussed approaches. For a full overview of the papers included in analysis, the approaches and shortcomings/requirements, please refer to table D.6.

The distinction between the ESEA and IP4ESET phases depicted in figure 1.1 is less clear in literature. We view performance assessment and improvement planning as separate activities in the BECIC, but many sources view assessment data as a defining element of the improvement plan, stating that assessment results are enough to know how to plan for improvement. We argue that knowing *what* is going wrong is not enough to know *how* to solve the problem. Still, the IP4ESET phase is dependent on assessment data (which can also be seen in the IP4ESET super method in figure A.1). Therefore, approaches with a heavy focus on assessment are also included in the analysis if the authors also mention a clear link to improvement planning. An example can be found in [22], where indicators and metrics are mentioned as being required for both performance measurement and strategy formulation.

[1] `https://bit.ly/SnowballingSIP`

When looking at the problem types related to IP4ESET approaches, the literature mainly mentions problems related to strategy, vagueness and resources (see figure 4.3[2]), accounting for 25.5%, 23.4% and 19.1% of the discussed problems respectively. Half of the proposed solutions and requirements are also related to strategy (50% in total).

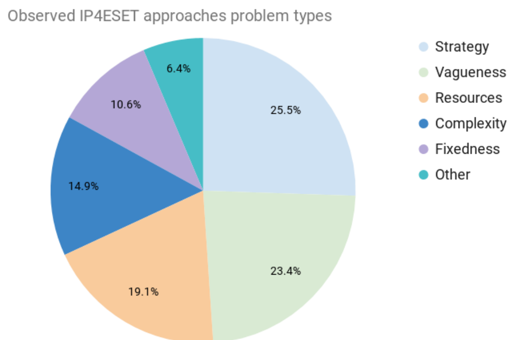

Figure 4.3: Observed IP4ESET problem types

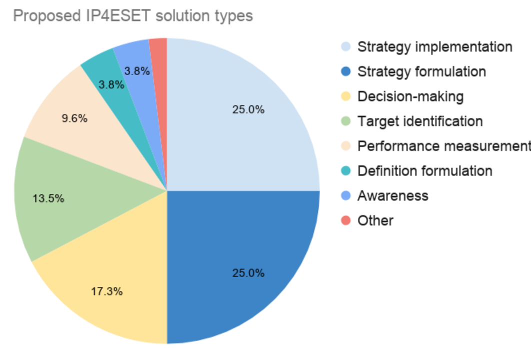

Figure 4.4: Proposed IP4ESET solution types

Strategy

Problems concerning strategy arise due to organisations not knowing how to either formulate or implement a CSR strategy, leading to a gap between intent of addressing CSR and implementation of CSR strategies [106]. Another difficulty is the alignment of a CSR strategy with the traditional business strategy [23; 45; 46; 83]. A clear distinction between strategy formulation (defining goals and objectives) and strategy implementation (defining steps for achieving CSR) is given by Han et al. [63], indicating that more is required than just formulating a strategy.

Problems concerning strategy are mainly rooted in the general field of strategic management, but are also present for specific methods. For example, the Balanced Scorecard (BSC), a method for strategy design, is critiqued for placing too much focus on the operational level, thereby not adequately taking strategic views into consideration [32]. Furthermore, the traditional BSC does not involve CSR as a strategic dimension [53].

Intention-Action Gap

Many organisations have the intent to define and implement CSR strategies, but do not know how. We call this an *intention-action gap* (see [80], for example). This is a problem rooted in strategic management, which consequently hinders sustainable development as defined in the conceptual framework.

[2] A larger version can be found in figure D.1 and figure D.2

The majority of proposed solutions and indicated needs for IP4ESET approaches cover the problem of strategy; either in formulation or implementation. For example, concerning strategy formulation, [63; 92] mention the need for organisations to be able to define their own goals and strategic guidelines and [26] advocates the Sustainability Balanced Scorecard (SBSC) as a framework that helps in developing a sustainability strategy that matches corporate strategy. Baumgartner [21] addresses the Framework for Strategic Sustainable Development (FSSD) to be useful in strategy formulation. The most often mentioned solutions to the strategy problem are the frameworks of FSSD [30] and SBSC [47].

Although these approaches go some way in supporting the IP4ESET phase, they are not fully adequate. These approaches have gotten critiques of their own ([52]), but they are also mainly unsupported by software tools, thereby leaving room for shortcomings that are also observed in other approaches, such as vagueness due to own interpretations of how to apply methods [98; 101]. A solution that is in line with the goal of the present study is indicated by Sroufe [106]: in a study where interviews with CSR professionals from multinational organisations were conducted, the need for a global information repository was expressed to link strategic actions with knowledge management [106]. The need for knowledge sharing to aid decision-making is also mentioned by [64]. Knowledge-sharing approaches like a repository would aid the translation of strategy into action.

ISO standards

The International Organization for Standardization (ISO) issues several standards related to sustainability, including ISO 14001 (prescribing requirements for environmental management systems (EMSs)) and ISO 26000 (defining the concept of social responsibility). An EMS described by ISO 14001 is not necessarily a software tool, but a systematic and iterative set of procedures, processes and responsibilities that describes how an organisation will manage its environmental impacts [18; 19; 72]. Other certifications for EMSs also exist.

Although EMSs theoretically have the ability to improve an organisation's environmental performance, the standard prescribing its functionality is criticized for being too vague. For example, Boiral [27] calls the standard a "myth" due to its room for interpretation. Furthermore, the costliness of certifying an EMS as being ISO compliant has been coined as another great downside [102].

The ISO 26000 standard aims to give a definition to social responsibility and is not certifiable. Since it does not describes harsh requirements for organisations to adhere to, it has been critiqued for being too vague. The scope of definitions in ISO 26000 is too broad [68; 101] and therefore not only vague, but also complex. Missimer et al. [92] call ISO 26000 a "check the boxes exercise", stating that the standard should be paired with a more strategic framework.

Takeaway Scientific Literature

The scientific literature analysis shows an *intention-action gap*. Organisations have a hard time formulating and implementing CSR strategies due to vagueness of approaches, and a lack of know-how, tool support and resources. One approach to structure the IP4ESET phase using a tool is proposed by Adèr [14]. The feature analysis will assess the support given by other tools for the IP4ESET phase.

4.1.3 Feature Analysis

The tools listed in table 3.3 were chosen due to their usefulness for IP4ESET. The BECIC in figure 1.1 indicates a clear distinction between the IP4ESET phase and the preceding ESEA phase, with the ESEA phase outputting assessment results that can be used as input for improvement planning. However, both IP4ESET and ESEA approaches may have some elements that are also related to the other phase. With that in mind, ESEA methods with tool support that have been analysed in [103] were also considered for the present analysis if those methods supported (one of) the following activities: *"discover ideas for action"* or *"prepare improvement report"*. Reasons for including or excluding tools from this feature analysis are listed in table 4.1.

Table 4.1: Included and excluded tools

ID	Name	Include	Reason
BIA	B Impact Assessment	Yes	Supports the creation of an improvement report and includes best practices.
PSAT	Program Sustainability Assessment tool	Yes	Provides guidelines for improvement planning based on assessment results. Planning is not tool-supported.
S-CORE	Sustainability - Competency, Opportunity, Reporting & Evaluation	No	Tool is inaccessible for this research.
SMETA	Sedex Members Ethical Trade Audit	No	Tool is inaccessible for this research.
GIT	Green IT Data Centre Assessment	No	Tool is heavily focused on assessment. Ideas for action are included in a separate tool, which also does not support the creation of an improvement plan.

A description of the included tools is given in table 4.2.

Table 4.2: SIP tool descriptions

Tool 1

Name	B Impact Assessment (BIA) tool
Organisation	B Lab
Description	The BIA tool is a free tool, used by 50,000+ organisations. It allows organisations to assess their socio-environmental impact using questionnaires, compare their scores to other organisations, and it points out areas for improvement. The questionnaires, as well as best practices for improvement, are tailored towards B Lab's certification. As such, only B Lab's own best practices and assessment areas are present.

Tool 2

Name	Program Sustainability Assessment Tool (PSAT)
Organisation	Washington University
Description	PSAT is a free tool that allows users to measure the sustainability of their (public health) program. It uses the Sustainability Framework for assessment and gives guidelines (but no functionalities) for the development of an improvement plan. Although this tool uses the term "sustainability" to define the "longevity" of programs, we include it for its ability to construct improvement plans that may as well be used for ethical, social and environmental topics.

It should be noted that the definition of "sustainability" in the PSAT tool differs from the definition of sustainability in relation to ESE topic that we discuss in the present research. PSAT focuses on sustainability as the "continuation" of a program, rather than sustainability as a socio-environmental topic. However, since the tool aids the creation of an improvement plan (which may or may not be applied in a socio-environmental setting), the tool is included in this analysis.

Feature Models

For both tools, feature models are created using the notation described in chapter 2. An excerpt of BIA's feature model can be seen in figure 4.5. The full feature models for both tools can be seen in appendix E.

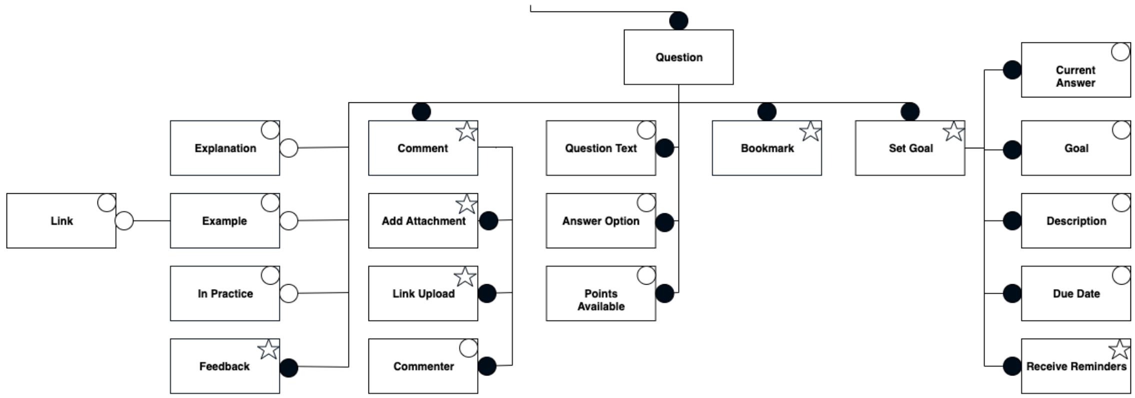

Figure 4.5: BIA feature model excerpt

The features present in these feature models have been mapped to the IP4ESET super method [14] based on the mapping criteria discussed in section 3.3.3. The result of this mapping can be seen in table 4.3. For both tools, rows are coloured green if features are present that match the criteria for that IP4ESET method activity. It should be noted that the colourisation depends on the *AND* and *OR* operators defined in section 3.3.3. For example, in order to be mapped to the first activity (*identify target area for improvement*), the features need only to meet criterium C1.1 *OR* C1.2.

Table 4.3: Mapping criteria results

Criterium	**ID**	**BIA**		**PSAT**	
		N	*%*	*N*	*%*
Identify target area of improvement	C1.1	17	29.82%	22	81.48%
	C1.2	5	8.77%	0	0
Identify goals	C2.1	13	22.81%	4	14.81%
Identify objectives	C3.1	7	12.28%	0	0%
	C3.2	1	1.75%	0	0
Identify action steps	C4.1.1	0	0%	0	0%
	C4.1.2	0	0%	0	0
Identify staff resp.	C5.1	0	0%	0	0%
Identify resources	C6.1	0	0%	0	0%
	C6.2	3	5.26%	0	0
Document improvement plan	C6.1	0	0%	0	0%
	C6.2	0	0%	0	0
	C6.2	1	1.75%	0	0

The results in table 4.3 indicate that neither tool supports all activities in the IP4ESET method.

Both tools have a strong focus on their usefulness for assessment, which is reflected in their support for the first activity, but organisations using these IP4ESET tools are mostly unsupported after the first two activities in the IP4ESET method.

Documenting the Improvement Plan

In order to better support all phases of the IP4ESET super method, tools are required that allow users to document an improvement plan in which goals, objectives, action steps, responsibilities and resources are indicated. Adèr [14] has proposed a concept of a tool that supports the IP4ESET phase in its entirety, called openESEIP. This tool takes input from ESEA methods and allows users to consequently set goals/objectives, prioritise actions, assign responsibilities, assign resources and view a visualisation of the improvement plan. The concept version of the tool is currently implemented using Google Sheets.

This tool allows organisations get a grip on their assessment data and translate those results into improvement plans. However, as we observed in section 4.1.2, defining a strategy alone is not enough. Since organisations need support in the implementation of their strategy, they have to be able to determine *how* to execute improvement actions. We propose the use of BPs for that aspect. An ideal tool for the IP4ESET phase would allow organisations to construct an improvement plan, but also to discover how to work on the actions in their improvement plan.

4.1.4 Grey Literature

In this section, we will elaborate on the analysis of grey literature on IP(4ESET) approaches to find whether any approaches described in grey literature (such as proprietary systems) address the shortcomings that we found in the analysis of scientific literature and the feature analysis. The grey literature sources we have analysed are largely based on findings from interviews that we have conducted in parallel with the shortcoming analysis (these interviews will be described in the following section).

Based on findings from interviews and responses to interview request, 16 sources were considered for inclusion in the shortcoming analysis (see table F.2). Of these sources, 6 met the previously described inclusion criteria (also used for scientific literature) and were included in the analysis. Table F.1 shows to what extent each source answers the *what* and *how* questions (indicated using -, +/- or +). As can be seen in table F.1, none of the approaches described in grey literature fully answers both the *what* and *how* questions related to IP4ESET. This indicates again that no approaches are found that allow users to identify *what* needs to be done but also *how* to execute it. This is similar to the findings of the scientific literature review and the feature analysis.

All approaches described in these sources follow an iterative cycle, and for each of those phases guidelines are given for how to work on that phase. Many of the sources describe examples or inspiration for possible points of action, but no concrete action steps are given. Many of the sources also link to additional resources, such as websites and reports.

Shortcoming Analysis Takeaway

Formal literature indicates the need for a structured approach to IP4ESET, asking *what* needs to be done to improve and *how* it should be done. This approach is supported by the tool developed by Adèr [14]. As indicated in the feature analysis, other tools for IP4ESET currently do not support the entire IP4ESET phase, primarily lacking support for the identification of action steps and improvement plan documentation. The grey literature analysis concludes with no approaches answering both questions. The tool developed by Adèr [14] aids in the development of an improvement plan, mainly answering the *what question*. We propose the development of a best practice repository that may be used to inspire which improvement actions will be carried out (*what*), but mainly to define *how* appropriate actions should be carried out.

4.2 Interviews

Interviews have been conducted with various stakeholders (CSR experts at organisations working on IP4ESET) in order to gain insight in the process of IP4ESET in organisations and how that relates to the findings of the shortcoming analysis. The interviews further served as a means of eliciting requirements for the repository. The interviews have been conducted with 3 possible end-users from various organisations:

- *Tony's Chocolonely*. Dutch manufacturer of responsibly sourced and produced slave-free chocolate. Interview conducted with the Impact Navigator.
- *Verstegen*. Dutch spice manufacturer, actively focusing on responsible business conduct and manufacturing. Interview with the Director of Sustainability.
- *Utrecht University*. Actively looks at sustainability issues in its strategic plan. Interview with the Project Manager Sustainability.

A further 2 interview have been conducted with the following organisations:

- *Cultivating Capital*. Consultancy firm helping organisations with obtaining B Lab certification.
- *Xomnia*. Organisation creating artificial intelligence solutions and just started looking at more sustainable work practices.

Interviews have been transcribed and coded using the coding scheme in table C.1. This table also lists the amount of references in the transcript that were coded under the various nodes. Some interviews were conducted and transcribed in Dutch. These references were later translated to English.

4.2.1 Difficulties in IP4ESET

An overarching theme in difficulties related to IP4ESET is a lack of clarity. For example, Tony's Chocolonely expresses a lack of clarity on internal processes for IP4ESET and is looking for a more structured approach to storing and sharing information (such as a wiki). This lack of clarity of processes is also expressed by Verstegen, but specifically framed as a problem that is primarily present for smaller organisations. This unclear view on processes for IP4ESET as a result of less resources fits the findings of scientific literature study in the shortcoming analysis. Xomnia is one of the smaller organisations with less resources dedicated specifically towards IP4ESET, and is mainly struggling with defining priorities.

For Utrecht University, the lack of clarity results in a vision with unclear targets and the issue of not knowing how to tackle an observed problem. Again, we see an intention-action gap. This may be a result of not connecting the target to concrete actions, as expressed by Utrecht University. This difficulty of strategy implementation may be a result of a lack of tools that support the identification of action steps, as has been brought to attention in the shortcoming analysis. Another problem of the lack of tools used to structure the IP4ESET process is the lack of documentation on past improvement actions; Utrecht University only tracks what has been done and by whom in an ad hoc fashion.

Of course, the desired course of action is specific to any organisation. Cultivating Capital expresses that B Impact's improvement reports often lack the specificity that is required to give organisations clarity on improvement actions.

4.2.2 Tools for IP4ESET

Several tools are used, but mostly for the identification of problem areas. An example of a proprietary software tool is Tony's Chocolonely's Bean Tracker for tracking the origin of cocoa beans. Other non-proprietary tools include Ecovadis[3], a tool that gives sustainability ratings to organisations.

Some tools are used to structure the process of IP4ESET, such as Microsoft Teams and Sharepoint, but these tools are mostly used to structure communication flows and are not specifically tailored to IP4ESET. Another problem that we observe in tool use is that the application of these tools is predominantly focused on measuring and reporting, as opposed the identification/documentation of action steps. This would attribute to the intention-action gap and explains the difficulties expressed by the organisations.

4.2.3 Use of a Best Practice Repository

Organisations find it hard to define best practices in their organisation. This could be due to organisations not specifically using the terminology of "best practices". Verstegen does define their own best practices and shares that on its intranet, which holds much more information than just these best practices. Verstegen's best practices are defined in a textual format. Other organisations have processes that might be called a best practice, but these "best practices" are not stored and shared in structural fashion. Utrecht University exchanges experiences in taking IP4ESET approaches with other universities in a sort of community of practice.

The usefulness of a best practice repository is acknowledged by the interview participants; there is relevancy in seeing approaches by other organisations. Cultivating Capital expresses that most improvements made by their clients are based on best practices. However, not all best practices are relevant for each organisation. The importance of addressing context in best practices (in

[3] https://ecovadis.com/

which context applying the best practice would be successful) is also expressed by Xomnia, Utrecht University and Tony's Chocolonely.

Interviews Takeaway

Organisations again express a lack of clarity on processes for IP4ESET, contributing to the intention-action gap. Tool support is somewhat existing, but not focused on the identification of action steps and their execution. Although organisations have a hard time defining what "best practices" are defined, they do express the usefulness of a BPR to be used for guiding improvement actions.

4.2.4 Requirements for a BPR

Primarily based on these interviews, as well as some other sources, a list of requirements for our BPR has been drafted[4]. The requirements are written in the form of user stories and checked for quality using the *Quality User Stories* framework [84]. User stories express requirements that stakeholders have for the system and therefore indicate stakeholder goals. We have written our user stories with the following template: *As a [role], I want to [goal], so that [motivation].* The final clause is optional. The requirements are based on 4 sources:

- *Interviews:* both explicitly and implicitly expressed desired functionality for the BPR has been used as input for requirements.
- *Shortcoming Analysis:* the template used for tracking shortcomings of current IP4ESET approaches in the shortcoming analysis was also used to track requirements.
- *EU Project:* the BPR created in this research has been included in a European project proposal as a means of storing and structuring research findings [59]. This project aims to increase the success of refugee and migrant initiatives in various parts of Europe and would be able to use the BPR to share findings on the result of these initiatives. We attended two meetings during the preparation of the proposal during which the project partners expressed several requirements.
- *Serendipitous:* several requirements were drafted based on serendipitous findings during the execution of this research.

In the linked Google Sheets file we also link the source for each user story by indicating the sheet and row number where the source is located. For example, source "A2" indicates that we have based this requirement based on a goal expressed on sheet A on row 2. The list of requirements, without sources, are also included in appendix H.

[4]The full overview of requirements, including their sources and prioritization can be found in this Google Sheets file: `https://bit.ly/OpenBPreqs`

Requirements Prioritization

A requirements prioritization survey has been filled in by 3 possible end-users of the BPR (all are project partners of the EU project) and 3 researchers with experience in the IP4ESET domain. The survey asked to rate the necessity of each requirement for the BPR on a 5-point Likert scale (ranging from *not necessary at all* to *very necessary*). The requirements have been rephrased to questions about features that are better understandable for non-technical stakeholders. In the previously linked Google Sheets file, the average results of the requirements prioritization survey are listed. A higher average indicates a higher necessity of addressing this requirement with our BPR.

4.3 Use of Best Practice Repositories

In research by Coenen [39], the use of BPRs for both ESE-related and ESE-unrelated domains has been researched. In this research, BPs (or patterns) are seen as easily adaptable and repeatable organisational design proposals. The following definition has been given for BPs for sustainability[5]:

Sustainability Best Practice

"Patterns that are proven solutions for problems in the three dimensions sustainability." [39]

The idea to store (ESE) BPs in a repository has been proposed previously [39; 48], mainly to promote knowledge sharing and reuse of information. We define a repository as *"a shared database of information on engineered artifacts which are produced or used by an enterprise"* [37]. In this case, the artefacts would be (ESE) BPs.

4.3.1 State-of-the-art and practice

In [39], the use of 12 BPRs has been researched. The researcher has conducted an analysis of the feature structure of the BPRs and the content structure of the BPs that they store. Based on papers analysed in a literature review, a feature model for an "ideal" BPR has been created. The feature models of 8 of the analysed BPRs (feature models could not be created for all BPRs) have been compared to the ideal BPR feature set. The table in figure 4.6 shows the mapping of features in the analysed BPRs to the features in the feature model created based on literature. It shows some common features that are present in all BPRs (such as storing and categorising BPs), but some variability can also be seen.

The main takeaway from this comparison is that there is a lack of common ground on how to construct a feature set for a BPR. There is no BPR that is able to address all features described by the ideal feature set, and some BPRs are missing crucial features that would allow for organisational knowledge sharing (such as the possibility of submitting a BP to the system).

[5]The term "sustainability" has been used by Coenen [39] so we also use that terminology here, even though we usually refer to *ESE topics*.

	Literature	Déroche	BIA	GIS	CaaS	FAO	PHBPR	COE	OBPS
Store (RF1)	x	x	x	x	x	x	x	x	x
Display (RF2)	x	x		x	x	x	x	x	x
Alter (RF3)	x								
Search (RF4)	x		x		browse		browse	*filter**	browse
Categorise (RF5)	x	x	x	x	x	x	x	x	x
User Management (RF6)	x								
Link (RF7)	x		x						
Submit (RF8)			*submit a ticket*				x	x	x
Download		x	x			x	x		x
List of Best Practices (RF9	*from quality feature 'content'*	List of patterns	x	List of measures	List of patterns	List of Good Practices	x	List of Good Practices	x

Figure 4.6: Mapping of case-based repository feature models to a literature-based repository feature model [39]

A similar comparison has been made based on the content structure of analysed BPRs in relation to the ideal content structure described by literature. Figures G.2 and G.3 show the variability in BP content for the analysed BPRs. The content analysis is subject to even greater variability than the feature analysis. The analysis of the state-of-the-practice on BPs and BPRs shows the lack of a common ground for how BPs and BPRs should be structured and designed [39].

4.3.2 Extensibility of BPRs

Quality requirements for BPRs have been determined based on other research. The *flexibility* requirement described by Coenen [39] aims to improve extensibility of BPs (as it defines flexibility in the application of BPs), but the analysed BPRs do not score well on this quality requirement. One possible explanation for low flexibility is an ill-described context for best practices: less information on application context decreases extensibility to other domains.

The lack of extensibility (consolidating BPs from various sources to extend their applicability to multiple organisations) is mentioned frequently in [39]. This problem has been mentioned in both literature and interviews with repository creators, but no solutions have been proposed. We propose to increase the extensibility of BPs in our repository by storing BPs from various sources. To allow for flexibility in storing BPs (and subsequently the functionality of the repository), the functionality will be model-driven. This development approach will be described in the following chapter.

BPR Takeaway

There is a lack of common ground on the structure of BPR feature sets, as well as the contents of BPs. BPRs score low on flexibility and extensibility, indicating that it is hard to share BPs across organisational boundaries). This decreases the possibility of knowledge sharing.

4.4 Context of Best Practices

One of the goals of our BPR is to let user improve their organisation's performance on ESE topics by identifying which action steps can be taken towards improvement. This identification of action steps and corresponding BPs that current tools for IP4ESET lack can be provided passively (by giving user full responsibility over the identification of action steps) or actively (by recommending BPs that would be relevant for the user/organisation and improvement action in question). Either way, there is one important aspect to consider: *what is the context in which a BP is applied*? During the previously mentioned interviews and EU project meetings, we observed that *context* is an important aspect to consider for determining BP implementation success; a BP that has been successfully applied by one organisation, may not automatically be successfully applied by another. If we aim to actively recommend BPs based on contextual factors, we have to determine which contextual factors to compare. For example, would a BP that has been successfully applied by a technology start-up in the Netherlands also be likely to be successful for a technology start-up in Italy? Do we need more information, such as the company size or the industry type? The selection and quantity of contextual factors need to be able to accurately attribute BP implementation success to these factors.

4.4.1 Theory of Change

The context of application for a BP can be made up of a wide variety of variables. We can draw some inspiration from the BIA tool, where the results of the assessment are dependent on three contextual organisational factors: *size* (in number of employees), *industry* and *geographic location*. These factors are not sufficient if we want to attribute factors of successful BP application, however. BPs are applied by organisations, but always in a larger context. This could be one of the many networks of responsible enterprises as researched by Adèr [14] or an institutional, political, social or cultural context [59]. The improvement plan itself is another context in which BPs are applied. In any of these contexts, a multitude of variables will be influential to implementation success. More accurate recommendations of BPs to be applied by a particular organisation in a particular improvement plan requires calculating in as many of these variables as possible.

In defining an improvement plan, organisations set a goal and define actions to achieve that goal. This approach is similar to defining a theory of change. Theory of Change is a method that aims to provide structure to planning and implementing various types of programmes/initiatives [89]. This method is used by various organisations, such as non-profits defining which actions to take in a programme towards a large-scale goal. It is therefore similar to the idea of improvement plans in IP4ESET, where concrete improvement actions need to be defined to reach a large-scale, diffuse goal. The result of the Theory of Change method is a theory of change (in lowercase, further referred to as ToC) that explains *how* and *why* a programme works [40]. ToCs are useful to construct improvement plans, but also to externally communicate why a programme is thought to be successful.

Structure of ToCs

There is no overall agreement on how to structure ToCs [89]. However, often used elements include: stakeholders affecting or affected by the programme, inputs required for interventions, activities in the intervention, direct output, long-term outcomes and impact [59]. Based on Mayne [89], we view a ToC as consisting of a goal, several sub-goals and *impact pathways* between these goals. The impact pathways describe what needs to happen in order to achieve impact and to move further towards the overall goal of the programme. In defining the impact pathways, we also describe *why* we assume that our activities will lead to impact.

Let us assume that we define a ToC depicted in figure 4.7. This model is a very abstract depiction of a ToC, but it is able to highlight the structure of a ToC. When we define an overall goal for the programme, we also determine what smaller sub-goals there are to achieve as a precedence for the overall goal. Sub-goals may also have further sub-goals. In order to move from the current situation to achieving the sub-goals (*making impact*), we have to define the actions we are going to take. These actions are defined with an *intervention* and for each impact pathway (arrows) in the model, we define what the intervention is going to be and why we assume that this intervention will help to achieve the (sub-)goal.

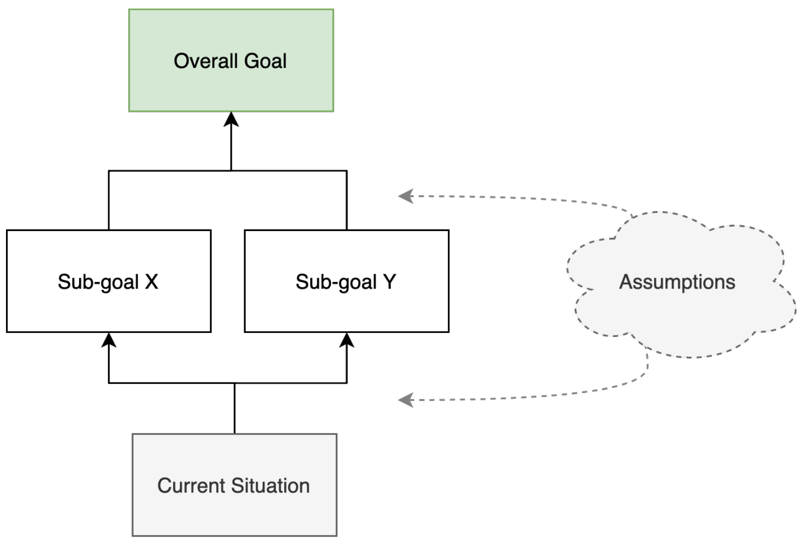

Figure 4.7: Basic ToC model

Similar to this approach, one of the interview participants expressed that they define small sub-plans to which actions and responsibilities are linked. This approach helps in making a large and complex problem more manageable. Considering the context of BPs, by defining a ToC we create a large source of contextual information. As we have expressed, BPs help to define *how* action steps can be carried out and in that sense, they are useful for shaping the interventions that are defined in a ToC. When we create a ToC, we not only define which BPs are used and how successfully, but we also define by which organisation they are applied, in which project, with which intent and assumption, and together with which other BPs. Consolidating all this information in an improvement planning tool is a promising approach to recommending ideas for conducting interventions (BPs) based on the contextual factors of programmes in which they are and have been applied.

A Model of ToCs

The model in figure 4.8 shows a concept of how a ToC might be defined in an improvement planning tool such as our BPR. It reflects how an *organisation* defines a *programme* with multiple *sub-goals*. For each *sub-goal*, we need to define on what hierarchy level it is located, what its current status is and which threshold needs to be reached. An *impact pathway* links two sub-goals and includes an assumption on the effectiveness of applying an *intervention* for these goals. An impact pathway is influenced (or executed) by *internal stakeholders* and has influence on *external stakeholders*. Interventions are evaluated with an *evaluation*. Interventions use *best practices* in order to shape their action steps. We have not elaborated on the structure of a best practice in this model, as this will be done in the next chapter.

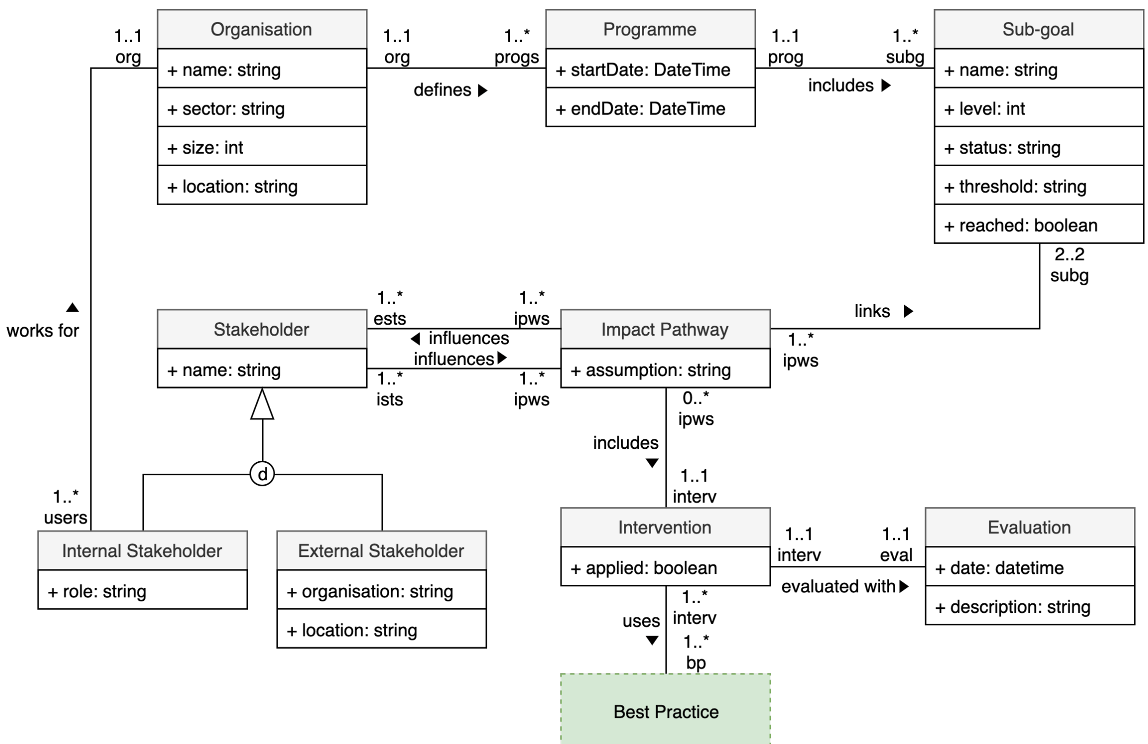

Figure 4.8: Structure of ToCs in an improvement planning tool

5 | Design Implementation

This chapter will detail the development process for the BPR created in this research. The BPR we have developed is called OpenBest. We will address the creation of a DSL for the model-driven approach, as well as the implementation of system components (using the DSL).

5.1 Creating an MDD Repository

So far, we have observed a multitude of obstacles in the use of software tools and BPs for IP4ESET. On one side of the problem, there is a general lack of software tools to support the IP4ESET process. On the other side the tools that do exist impose some restrictions, mainly due to a lack of flexibility. Our domain analysis of BPRs show variability in their feature sets and the BPs that they store, but the contents of those BPs are either restricted by the rigidity of the BPR architecture, or restricted by the closed nature of the BPRs.

The closed nature of BPRs and similar tools leads to a smaller variety of sources for stored BPs. For example, the intranet used by Verstegen to share BPs only takes input from within the organisation's boundaries. Similarly, the BIA tool only stores BPs defined by B Lab, as does the FAO repository with BPs defined by the FAO.

The rigid architecture of BPRs in turn restricts the usefulness for diverse user groups. Users are limited to the predefined feature set (such as predefined querying capabilities), which increases the difficulty of finding useful BPs. Furthermore, these rigid architectures impose restrictions on the content of BPs to be submitted to the BPRs: any user wishing to submit a BP has to comply with filling in information prompted by the BPR.

This rigidity does not relate well to the variability observed in the domain analysis and prompts the question: how can a BPR support this need for variability? We have elaborated on the usefulness of MDD in the conceptual framework. This chapter will detail the development of a MDD BPR to address the restrictions of currently available BPRs.

5.2 Creating a Model-Driven Tool

In its simplest form, all that is required in MDD is a conceptual model of (part of) the system, an interpreter or compiler to transform the model and a system to map the code or execution files to [99]. MDA does not specify the modeling language that should be used for the conceptual model (and neither does MDD place this restriction); MDA only requires the use of models that are specified along the MOF rules [16].

In chapter 3, the rationale for developing a DSL was listed as a twofold benefit: the DSL allows for system configuration and GUI construction. It does this by specifying how organisational BP models can be created, which can then be transformed into textual models to be interpreted by the tool. This approach reflects the variability in the domain and allows users to construct their models along a set of rules. The following sections will elaborate on the process of developing the DSL for this research.

5.2.1 Developing a DSL

By developing a DSL we are able to address the complexity and variation observed in our domain, due to its specialization to our problem domain and the restricted use of concepts for the problem domain only [78]. The use of general-purpose modeling languages (like UML) would lack the specificity we need for addressing our domain elements [93]. Another benefit stated by Kosar et al. [78] is that DSLs are typically easy to use for front-end users; i.e. the simplicity of a DSL allows us to move the task of adding tool variability from the developer to the user. We thus aim to use models to configure the functionality of the repository. In similar research, a DSL was developed based on ESEA methods and used to configure the functionality of a ESEA tool [49]. In [49], five DSL development phases are applied based on [90]: decision, domain analysis, design, implementation and testing. The rationale of the decision phase has been mentioned in chapter 3. The following sections will elaborate on the other development phases.

Domain Analysis

Various formal approaches to domain analysis exist. One such method is Feature Oriented Domain Analysis (FODA), in which variability in the domain is researched by assessing commonality and variability among feature models [90]. The aim of our DSL is not to serve as a means for constructing feature-oriented models, however. Rather, we aim to let our DSL describe the commonality and variability among data models of organisational BPs. Inspired by FODA, we therefore take a slightly different approach to domain analysis.

For the domain analysis, 12 UML models have been created to describe the structure of BPs in both ESE-related and ESE-unrelated domains (appendix I). These UML models are created in research supervised by the author [39] and updated when necessary using a UML comparison approach. Appendix I shows three different model versions:

- *Version 1* shows the initial models created by Coenen [39].
- *Version 2* shows the models created by the author. This version only includes updated v1 models that were deemed to be incorrect.
- *Version 3* shows the models that required further changes based on discussions between the two researchers.

Models were updated after discussions between the two authors. The final models used for domain analysis may be v1 models if no changes were required, and otherwise v2 or v3 models. Changes across different model instances were tracked and can be seen in table I.1.

For each analysed model, an overview of its concepts has been created with a short explanation of its meaning. This overview has been used to compare the concepts of the various models. Section I.4.1 shows an overview of the most commonly present elements in the analysed models, as well as some concepts with a high perceived usefulness. This comparison resulted in the following concepts:

- Best Practice
- Solution
- Result
- Problem
- Example
- Effort
- Rating
- Author
- Category

This list of concepts indicates which elements are likely to be required in order to construct a model of a BP structure.

DSL Models

Mernik et al. [90] state that the output of a domain analysis for DSL development is a domain model consisting of various elements: a domain description, domain terminology, description of domain concepts and a description of the commonalities and variabilities of domain concepts. This domain model then is able to derive a DSL (although there are no formal guidelines for this process) [90]. However, this assumption is made on the basis that the developer of the DSL specifies all concepts and dependencies and that users of the DSL are restricted to these elements. As stated before, the aim of our DSL is to allow users more freedom in the creation of models in order to cater to the high variability observed in the domain.

We have therefore opted to specify a DSL that abstracts the concepts, attributes and relationships observed in the domain analysis. Our DSL is called OBL (OpenBest language). The language is expressed with several models that will be detailed in this section. The OBL metamodel is one of these models and expresses (part of) the abstract syntax of the language. To guide the model creation process to some extent, we have decided to create an instantiation of this metamodel that includes both mandatory and recommended concepts, relationships and attributes. We call this the OBL "core model". The core model is based on our observation of these elements in the domain analysis explained in the previous section. The mandatory concepts are considered to be what is minimally required to specify organisational BPs. The recommended concepts are not required to specify organisational BPs, but recommended based on our experience in analysing the domain. The concepts that were considered for the core model are listed in table I.2 and are based on the results of the model comparison in the domain analysis.

The overview in table 5.1 shows the various models related to OBL. A distinction is made between the abstract syntax models and the concrete syntax models. The abstract syntax defines the "vocabulary" of OBL (i.e. the concepts in the language and their structural context). The concrete syntax expresses the notation of OBL and is therefore influenced by the choice of technology. The output of using the DSL is a textual model that can be interpreted by OpenBest to determine functionality. These models have to be in JSON format, and so the concrete syntax of OBL is influenced by the choice of JSON as the model notation.

Table 5.1: OBL models

Abstract Syntax	
Metamodel	The OBL metamodel is part of the abstract syntax [55; 31] as it expresses how conceptual organisational BP models are structured. The metamodel resides on the conceptual level, rather than the implementation level, as it does not define the notation required for the creation of textual models. Any implementation technology (i.e. regardless of notation) can build on top of this metamodel [55].
Core model	The OBL core model shows what the core of every organisational BP should be according to our domain analysis. It also recommends useful elements. The elements in the core model are based on the most common and useful elements listed in section I.4.1. The core model is an instantiation of the metamodel and can therefore also be *extended* by adding concepts, attributes and relationships in accordance with the metamodel.
Concrete Syntax	
Textual grammar	The textual grammar of a DSL defines its concrete syntax (another example is given in [49]). The models that can be read by OpenBP are JSON models and so that notation shapes the definition of the concrete syntax. The textual grammar defines rules for the creation of these textual models and is written in EBNF.
Core JSON model	The core JSON model is a JSON representation of the core model. It adheres to (and is defined in) the EBNF. This model is the minimal textual model required by the tool.

The relationships between the models in figure 5.1 are depicted in figure 5.1. It indicates the abstract syntax models in green and concrete syntax models in gray. Through the use of a model editor, a textual model can be produced that adheres to the rules in the textual grammar. The model editor is constructed by taking the metamodel into consideration. The JSON core model is a translation of the core model (also adhering to the textual grammar rules) and is at least partly (the mandatory elements) included in any textual model. OBL specifies how *textual models* can be created that specify the structure of BPs. These textual models are written in JSON format and include the various concepts, attributes and relationships that are present in each BP. This textual model is interpreted by OpenBest and used to instantiate an instance of the repository for a domain.

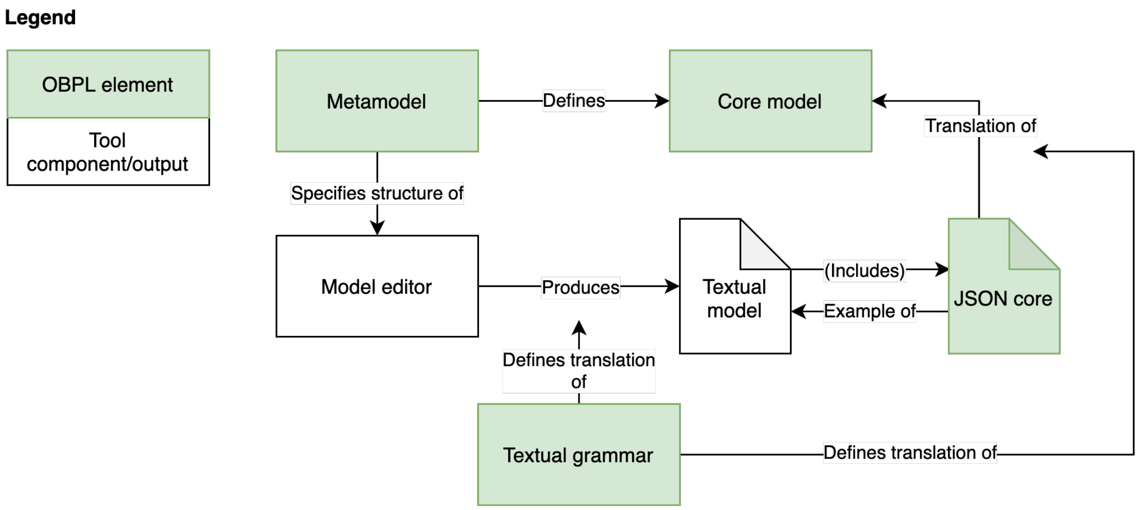

Figure 5.1: Overview of the relationships between the OBL models

5.2.2 Abstract Syntax

The abstract syntax of OBL is reflected in the core model and the metamodel. The OBL metamodel is shown in figure 5.2. The core model in figure 5.3 shows the mandatory concepts (in green) and recommended concepts (in yellow).

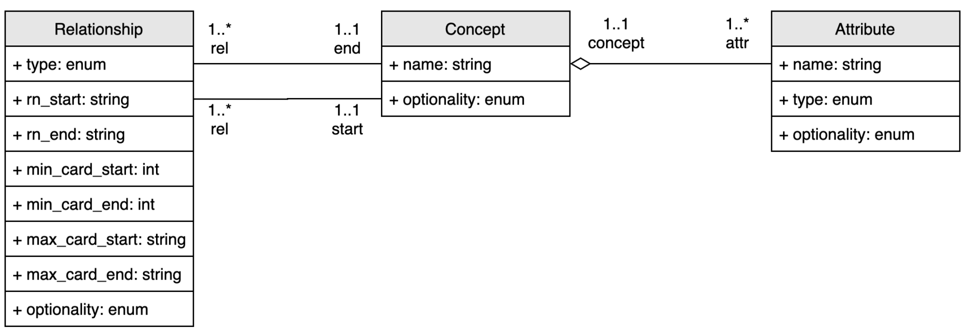

Figure 5.2: OBL metamodel

Table J.1 in appendix J lists an overview with explanations of the various concepts, attributes and relationships present in the core model and metamodel. The metamodel indicates rules for how the core model is structured and can be extended. For example, a concept should have at least one (and possibly many) attribute(s). Although the metamodel is able to express a base-level of rules and restrictions, there are some restrictions that we cannot express in this model. We therefore express further restrictions by annotating the metamodel with constraints. We take inspiration from work by Le et al. [81], who use "meta-attributes" (attributes of attributes) to express additional information on how domain concepts can be realised.

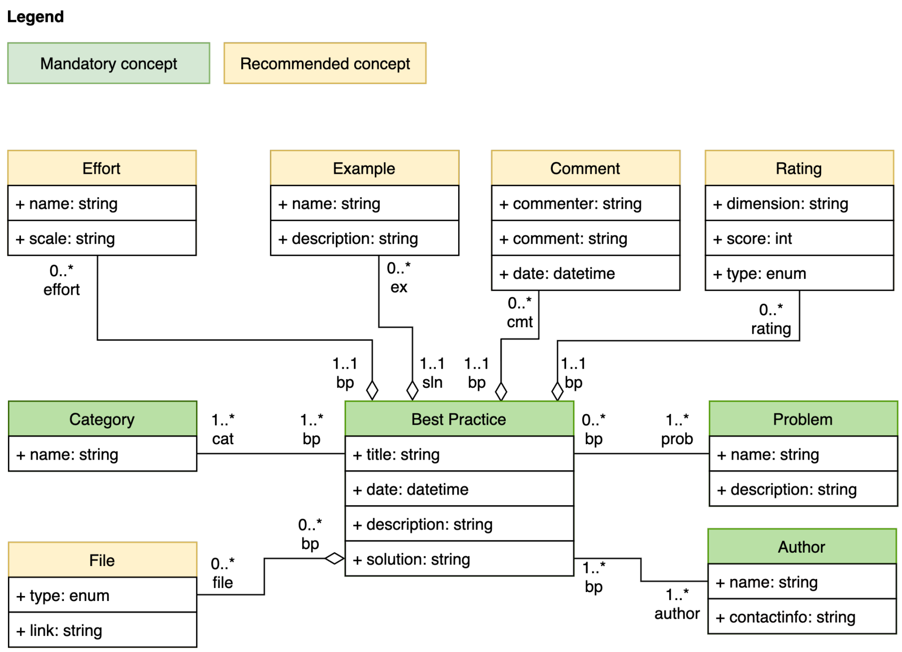

Figure 5.3: OBL core model

Metamodel Constraints

Figure 5.4 shows our metamodel with added constraints (please refer to appendix J for a full-size figure). We use this model to express additional (meta-)information on classes (*MetaClass*) and attributes (*MetaAttribute*) in the metamodel. Expressions of constraints use the following notation: [***relationship***].[***Class* OR *attribute***].[***meta-attribute***].

- The relationship component may be specified along a relationship link (such as *end* or *start*) to indicate to which class we are pointing from the starting point of the expression, or it may be *self* to refer to the class itself.
- The constraint may point to a Class (capital) or attribute. In the former case, the constraint holds for the entire class. In the latter case, the constraint only holds for the attribute.
- The meta-attribute defines the meta-information on which we are placing a constraint, such as the *value* or *mutability*.

We provide further explanation on the restrictions below:

- Possible values of attributes are listed. For example, the *optionality* attribute may take the values *mandatory*, *recommended* or *userDefined*, indicating if the elements are present in the core model or defined by the user.
- *unique* = *true* indicates that only one instance of the class can be created.
- The *mutable* meta-attribute indicates whether or not the object is mutable. *Relationship* has

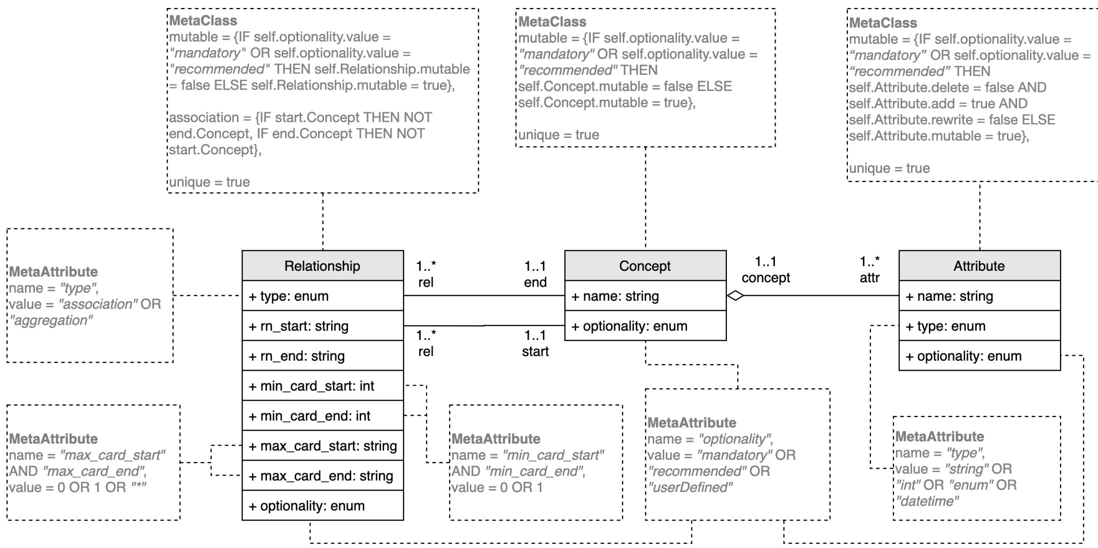

Figure 5.4: OBL metamodel with constraints

the following mutability: *IF self.optionality.value = "mandatory" OR self.optionality.value = "recommended" THEN self.Relationship.mutable = false ELSE self.Relationship.mutable = true*. This indicates that the class *Relationship* is only mutable (meta-attribute *mutable* of the class) if its optionality is set to "userDefined".

- The *mutable* meta-attribute has three sub-attributes: *add*, *delete* and *rewrite*. For *Attribute*, we specify that for recommended and mandatory attributes, no attributes can be deleted and rewritten (i.e. all attributes in recommended and mandatory concepts must be left as-is). Attributes can be added for these concepts.
- For *Relationship*, we specify constraints on the double association with *Concept*. We indicate that the concept at the start point of a relationship may not be the same concept at the end point of the relationship (i.e. no self-referential relationships are allowed).

5.2.3 Concrete Syntax

The concrete syntax of OBL is defined in the JSON core model and textual grammar. The abstract syntax of OBL defines (on a conceptual level) how to structure BPs. Our goal is to develop a tool that stores these BPs. OpenBest interprets the structure of a *textual model* and uses it to define the correct database structure and functionality. The structure of BPs as defined in the abstract syntax is therefore reflected in the *textual models* defined using the concrete syntax. In order to create textual models that adhere to the abstract syntax and are written in the correct notation (JSON), we require the textual grammar to specify rules for textual model creation.

Hiding the Concrete Syntax

The concrete syntax of OBL specifies how to create textual models. These models are less easily read by humans and our concrete syntax may therefore seem less usable than a graphical alternative. It would seem that the specification of a textual model is quite difficult for humans to achieve. However, textual models are created in OpenBest through the use of a model editor that "hides" the concrete syntax from the user. Models are constructed with an easily understood form-based wizard. The concrete syntax is hidden away in the back-end to create the textual models. Later in this chapter further explanation on the model editor will be provided.

Textual Grammar

The textual grammar defines rules for creating these textual models based on how BPs should be constructed according to the abstract syntax, restrictions of the JSON notation and restrictions as a result from the choice of implementation platform. The textual grammar is written in Extended Backus-Naur form (EBNF). An excerpt is shown in listing 5.1. The full textual grammar is shown in listing J.2 in appendix J.

```
(* JSON requires the correct number of closing braces to be valid *)
BP model = opening brace, domain name, colon, opening brace, domain
   state, best practice, comma, [ comment, 2 * closing brace ],
   comma, [ rating, 2 * closing brace ], [ example, 2 * closing
   brace ], [ effort, 2 * closing brace ], 2 * closing brace, comma,
    user, 2 * closing brace, comma, author, 2 * closing brace, comma
   , problem, 2 * closing brace, { collection }, 3 * closing brace,
   ';';
domain name = quotes, text, quotes;
(* Elements of the core model. Since these elements are mostly fixed
   , we just include the JSON code here *)
domain state = '"domainstate": { "displayfeature": false, "name": ',
    text, ' , "administrator": ', text, ',';
```

Listing 5.1: Excerpt of the textual grammar

Core JSON Model

The concrete syntax is tied to the JSON notation, since that is the notation required for our textual models. The core JSON model is such a textual model. It is a reflection of how concepts are structured in the abstract syntax core model, but specified according to the rules in the textual grammar. Listing 5.2 shows the core JSON model, only including the *mandatory* concepts. A full core JSON model (also including *recommended* concepts) is listed in appendix J.

```
{
```

```
    "ADomain": {
        "domainstate": {
            "displayfeature": false,
            "model": "string",
            "name": "A Domain",
            "administrator": "person@email.com",
            "bestpractices": {
                "bpdocument": {
                    "01grouptitle": "Best Practices",
                    "02groupdesc": "Enter basic best practice info here.",
                    "1displayfeature": true,
                    "2title": "string",
                    "3description": "text",
                    "4author": [{"name" : "Written by", "self": "document reference", "related": "document reference"}],
                    "problems": [{"name" : "Solves", "self": "document reference", "related": "document reference"}],
                    "5solution": "text",
                    "6categories": ["string"],
                    "7date": "string"
                }
            },
            "users": {
                "userdocument": {
                    "1displayfeature": false,
                    "2email": "string",
                    "3name": "string",
                    "4role": "string"
                }
            },
            "authors": {
                "authordocument": {
                    "1displayfeature": false,
                    "2contactinfo": "string",
                    "3internal": "boolean",
                    "4name": "string"
                }
            },
            "problems": {
                "problemdocument": {
                "01grouptitle": "Problem",
                "02groupdesc": "Describe the problem here.",
                "1displayfeature": true,
```

```
                    "2name": "string",
                    "3description": "text",
                    "bestpractices": [{"name" : "Solved by", "self": "
                        document reference", "related": "document
                        reference"}]
                    }
                }
            }
        }
}
```

Listing 5.2: Brief JSON core model

The core JSON model shows similarities with the core model in the direct translation of some attributes to JSON code. Concept names in the core model are also present in the core JSON model (in plural). Every concept includes a "document" in which the attributes are listed. The following sections will detail the data structure used for the JSON models.

Implementation Platform

We have chosen to develop and deploy OpenBest on Google's Firebase[1]. Firebase is a development platform that provides hosting and database services, as well as several other services that we have used for development. We require OpenBest to read and interpret textual models, structure the database according to the structure of the textual model and store BPs according to the specified database structure. We therefore require the use of Firebase's database called Firestore.

Restrictions Imposed by Firestore Firestore is a NoSQL database. NoSQL databases use no schema to describe how data should be structured. Instead, they provide full flexibility over how and where data is stored in the database. This is useful functionality, as the structure of BPs (and therefore of the database) differs for every domain and changes every time the domain administrator alters the model. The textual models that are uploaded to OpenBest reflect the required data structure (how the elements of a BP relate to each other). Since NoSQL database structures can be described using JSON objects, we have opted to use JSON as the notation for textual models.

NoSQL databases are represented in a hierarchical tree structure. This structure is reflected in listing 5.2: all concepts in this model (such as *bestpractices*), and therefore their structure, are defined under the hierarchy level of this domain (*"ADomain"*). We can summarise the aspects of Firestore's data model that influence how we construct our JSON models as follows:

- Data is stored in documents. For example, every attribute related to a single BP in the core model, is listed under *"bpdocument"* in the core JSON model. A document is a single instance of a concept in the core model; i.e. every Problem would be defined in a separate document.
- Documents must be part of collections. Every concept in the core model, such as *BP*, has its own collection in the core JSON model. All BP documents will be stored in the *"bestpractices"* collection in the database.

[1] `https://firebase.google.com/`

- Documents can have subcollections. We can move further down the hierarchy by defining subcollections for documents. This is reflected in the full core JSON model in appendix **??**, where *"comments"* is an example of a subcollection of *"bpdocument"*: every comment falls under the hierarchy of exactly 1 bpdocument.
- Data is retrieved from Firestore alphabetically, so we prepend the attributes with a number based on the order in which we want to display features; e.g. we want to ask a user for a BP title before they define the BP description.
- We can store a variety of data types in Firestore, of which we use the following:
  - Strings. Both *string* and *text* values in the core JSON model are string values in Firestore, but result in slightly different features in OpenBest.
  - Booleans
  - Integers
  - Arrays. The core model shows an array of strings for categories, for example. The possibility of adding arrays enables us to include *categories* as an attribute, rather than a separate collection (like we have done for other concepts in the core model).
  - Maps. JSON-type objects called maps can be stored, which is useful for storing more complex attributes. We use maps to store association relationships between concepts, as we require three values for every relationship to be stored.

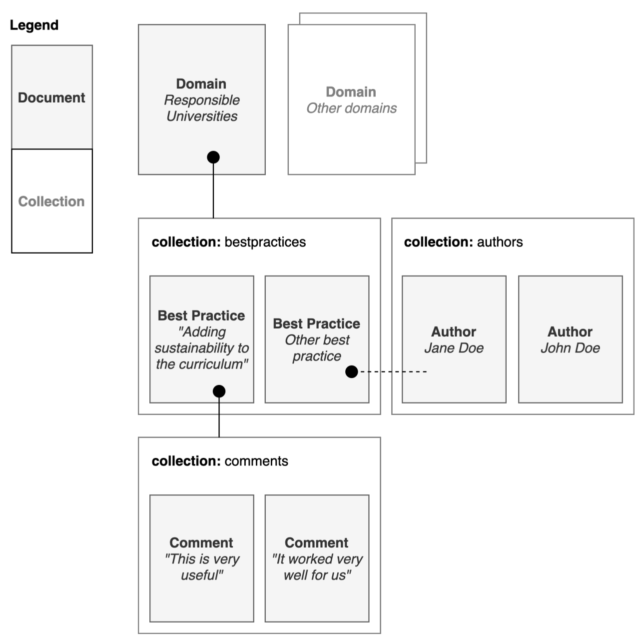

Figure 5.5: Overview of the Firestore data model

Figure 5.5 shows how the structure of the core model is related to Firestore's data model. It shows how documents are ordered in collections (in this case we only show *bestpractices*, *comments* and *authors*). The collections also reflect the concepts in the OBL core model. Documents can also point to other documents: e.g. a document in *bestpractices* always stores a reference to the document in which the author is specified. This reflects the association relationships in the OBL core model.

Graphical Respresentation of the JSON Core Model

The choice to use Firebase influences how we construct the textual models. It also limits how accurately we can create a textual model that reflects the conceptual OBL core model. The graphical representation of the JSON core model in figure 5.6 shows these restrictions. This model is not part of the OBL language definition.

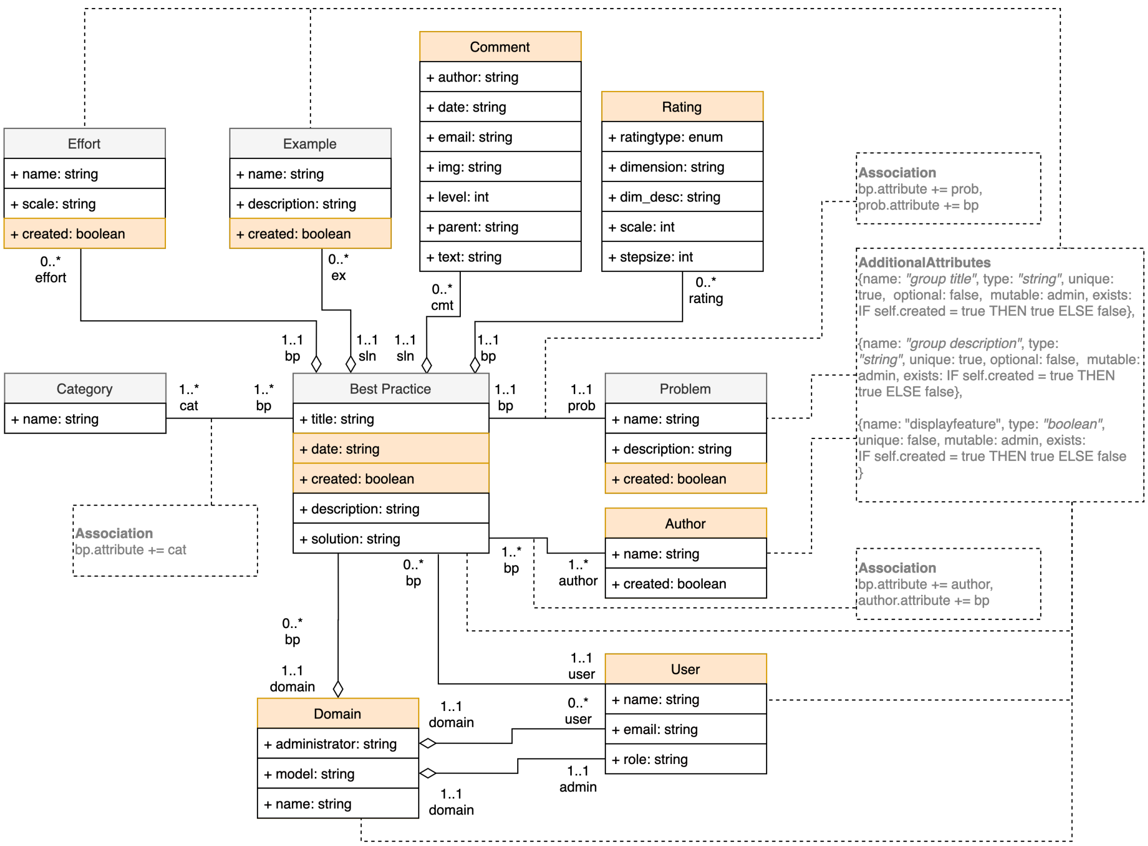

Figure 5.6: *Graphical representation of the JSON core model*

Model Distinctions Differences in concepts and attributes between the core model and graphical representation of the JSON core model are indicated in orange. The following differences can be observed:

- The possibility of adding files to BPs has not been implemented yet, so the *File* concept is removed.

- The *Domain* concept is added to reflect that BPs are stored under the hierarchy of a domain in OpenBest (as can be seen in figure 5.5). Each domain has an *administrator*, a *name* and stores the textual *model*.
- The *User* concept is added to reflect the (types of) users using OpenBest. There are two possible user *roles*: administrators and regular users. Users write the BPs, although they can enter any author name (hence the relationship between *BP* and *Author* as well).
- In most concepts, the *created* attribute is added. This indicates whether the document is the "base document" (created by the model interpreter) or a document created by the user (when storing a BP). More explanation on this distinction will be provided in section 5.2.4.
- The attributes for the *Comment* and *Rating* concepts have been altered. The tool functionality for these concepts has been created in other research [110] and the attributes in the graphical representation of the JSON core model are required for correct tool functionality.
- The cardinality of some relationships has been changed. For example, the core model shows that a *Problem* may have no *BPs* defined to solve it. This makes sense conceptually, but not on an implementation level. In OpenBest, a *Problem* is only defined when a *BP* has been defined to solve it.

Model Annotations Similar to the metamodel constraints in figure 5.4, we have annotated the graphical representation of the JSON core model with additional information.

- *Association*. Association relationships are stored in the database explicitly as attributes. For example, a *BP* has an attribute in which the relationship with *Author* is defined. This relationship is also depicted in figure 5.5. The association *bp.attribute += author* indicates that the *author* attribute is added to *BP* due to the association relationship.
- *AdditionalAttributes*. Depending on whether or not a document is a base document or a user-created document (explanation on this distinction is given in section 5.2.4), 3 additional attributes are present. Since these attributes are present for some documents of these concepts but not all, we add them as additional attributes.
  - For every document in the textual model, the model interpreter creates a set of features. To provide an explanation about the features to the user entering a BP, a *group title* and *group description* is displayed. These attributes are only present in the base document which is used to create the set of features, since they do not contain valuable information for the BP itself.
  - For every document in the textual model, the *displayfeature* attribute determines if a set of features should be created in the form for entering a BP. For example, *displayfeature* is set to false for the User concept, since users are not required to define another user when creating a BP.

5.2.4 System Components Overview

Figure 5.7 shows an overview of the system components that are responsible for storing BPs. An administrator uses the model editor to generate a JSON model. This model is interpreted by the model interpreter. The interpreter instantiates the domain that is described in the JSON model (or updates the domain if it already exists) by creating the correct database structure. Based on the structure of the JSON model, the hierarchy structure in the database is determined. In each collection, a "base document" is created. These base documents contain the previously described attributes *group title*, *group description* and *displayfeature*, as well as other attributes with their name (e.g. *title*) and value (e.g. *string*). These attributes determine which features need to be instantiated when a user uses the BP entry form.

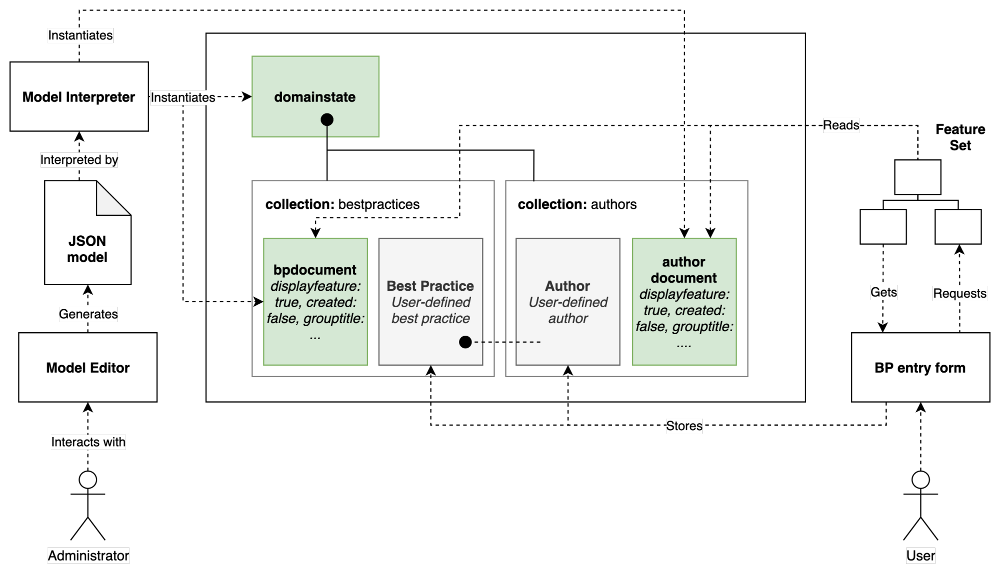

Figure 5.7: Overview of system components

5.3 Model Editor

We have chosen to implement the model editor in the OpenBest tool, so that OBL models can be created in the same tool that interprets them. We have also considered defining our DSL in Eclipse's Xtext[2]. Based on the DSL definition, Eclipse spins up a text editor that checks textual models against the specified language syntax. The benefit of this approach is that changes in the DSL definition lead to changes in the text editor without any additional effort by the DSL developer. In our case, the DSL and model editor are both embedded in OpenBest and the model editor uses the DSL to generate textual models. The tight coupling of these elements to the technology has the drawback that changes in the DSL also require changes to the model editor, but we have chosen to accept this drawback as a trade-off over the increased ease-of-use for users in not having to use an external model editor.

The model editor is a form-based wizard for the creation of textual models. Users create textual models by entering information in a form and this information is translated to a textual model using transformation rules that adhere to the textual grammar. The textual grammar, as well as

[2]`https://www.eclipse.org/Xtext/`

the JSON models, are hidden from users. Users therefore are not required to have knowledge of the concrete syntax, making the tool more usable for non-technical users.

The model editor is only accessible to the *domain administrator*: the user in charge of creating an OBL model that accurately reflects the structure of BPs required by that particular *domain*. A domain is any group of organisations and users that use the same BP structure and are grouped on any other factor. For example, organisations in the same network of responsible enterprises may form a domain. The domain administrator uses the model editor to define the data structure of BPs. This structure is reflected in a JSON model, which is used by the model interpreter to generate the corresponding feature set.

Domain and Domain Administrator

A **domain** in is any group of organisations. Each domain has its own instance of OpenBest with a dedicated database section. The **domain administrator** is in charge of the domain model, in which the structure of BPs for that domain is specified. The structure of the database and the feature set of the tool instance are determined by this domain model. All organisations in one domain use the same instantiation of OpenBest and are therefore restricted to the same BP structure. This is necessary to be able to compare the various BPs in one domain. We assume that organisations within the same domain are similar enough to be able to use the same BP structure.

The domain administrator specifies concepts, attributes and (implicitly) relationships in the model editor. Users are later asked to specify values for the attributes when submitting a BP. Every group of features for a concept is accompanied by a group title and group description (informing the user what is expected to be filled in). This group title and description is also specified by the domain administrator. Figure 5.8 shows part of the model editor, where the domain administrator is asked to define the name of the domain and a group title and description for the best practices concept.

The model editor can also be used to specify BP structures for BPs that are not related to IP4ESET, but in this research we primarily study the influence of OpenBest on specifying and sharing BPs related to IP4ESET.

5.3.1 Relation to the Core Model

The core model contains 4 mandatory concepts: Best Practice, Category, Problem and Author. Category is modeled as a separate concept in the core model due to the 1..* cardinality with Best Practice, but this cardinality can be captured in the tool by storing arrays of categories in best practice documents. The model editor therefore only includes elements for the other three concepts. For each concept, the domain administrator is asked to provide a group title and group description. Recommended concepts from the core model can be added by checking a checkbox. Although the OBL metamodel specifies that all concepts of the core model can be extended with new attributes and relationships, we have not yet implemented this functionality in the model editor.

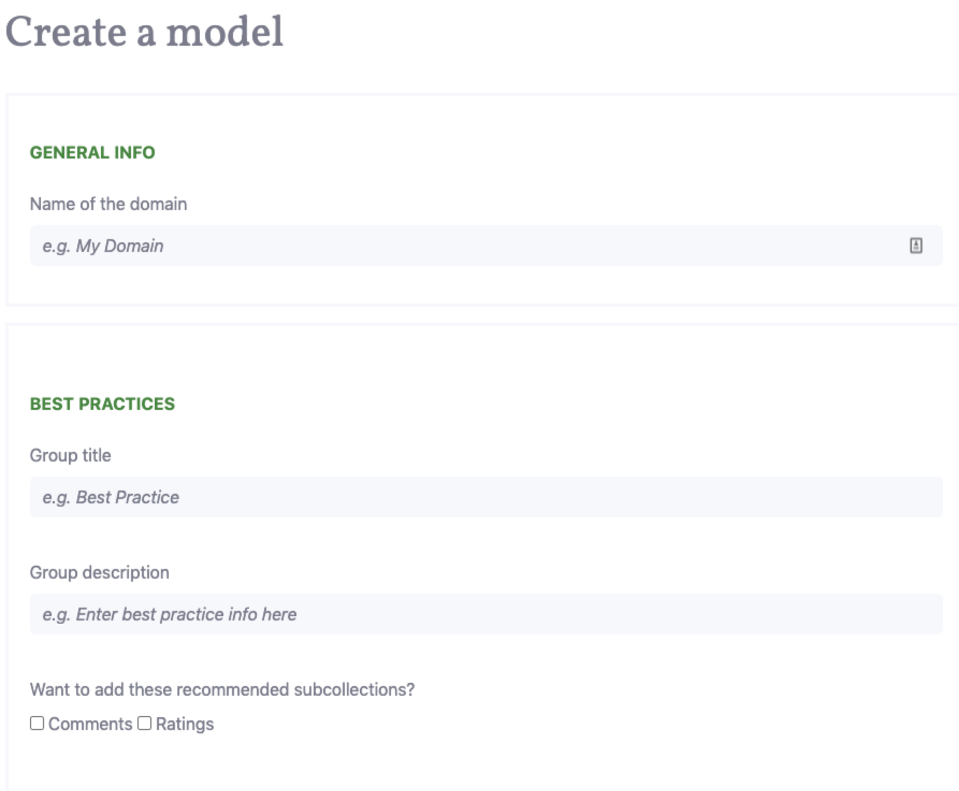

Figure 5.8: Screenshot of the OpenBest model editor

5.3.2 Modeling Relationships

The abstract syntax defines the structure of BPs on a conceptual level, while the concrete syntax defines notation and is therefore tied to our choice of technology. When using the model editor, the domain administrator creates models that adhere to the OBL abstract syntax (although the domain administrator may not be aware of this): any model created by the domain administrator may be represented as a graphical OBL model. The output of the model editor, however, is a textual OBL model. The model editor makes this translation using the textual grammar.

In order to textually model the relationships that are present in the abstract syntax models, we have to determine how to model relationships in JSON. The excerpt of the JSON core model in listing 5.3 shows all relationship types, which we will explain in more detail.

```
{
    "ADomain": {
        "domainstate": {
```

```
            ...
            "bestpractices": {
                "bpdocument": {
                    ...
                    "4author": [{"name" : "Written by", "self": "
                       document reference", "related": "document
                       reference"}],
                    ... ,
                    "comments": {
                                "commentdocument": {
                                        "author": "string",
                                        "date": "string",
                                        ...
```

Listing 5.3: Brief JSON core model

Association

JSON:API is an attempt to structure JSON files according to a common structure, which the schemaless language generally lacks. We define relationships in our JSON files by drawing inspiration from the JSON:API specification on relationships[3]. For each association relationship, we define a relationship *name*, the concept that is the start point of the relationship, and the end point concept.

An example of how relationships are stored in the database is given in figure 5.9. In this example, two different relationship types are defined between the authors collection and the bestpractices collection. The domain administrator defines which relationship names to establish between two collections; users are able to define the exact document in each collection to reference to. These different relationship names indicate separate association links between *Best Practice* and *Author*. The *"4author"* attribute in listing 5.3 shows this relationship in JSON.

Figure 5.9: Example relationship stored in the database

[3] https://jsonapi.org/format/#document-resource-object-relationships

When using the model editor, the domain administrator has to define the subcollection that is referenced, along with a name for the relationship. Firgure 5.10 shows a concept “Lessons Learned” being specified. This concept has a “name” attribute and one relationship to a concept/subcollection called “implementation”. The relationship (“Learned from”) is a one-directional relationship from Lessons Learned to Implementation. For a bi-directional relationship, the reference also has to be defined when creating the Implementation relationship.

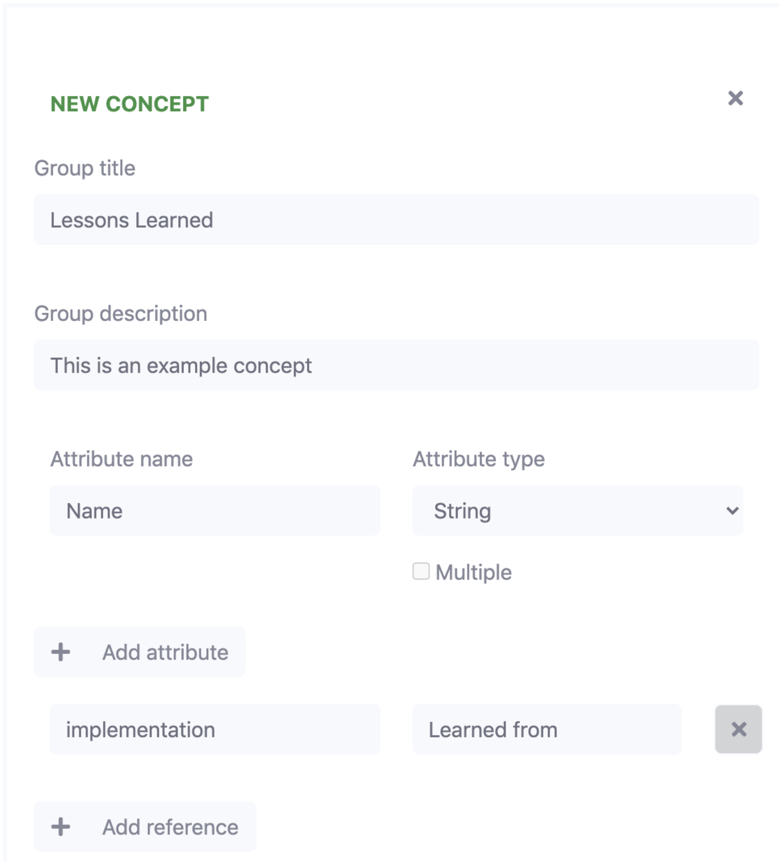

Figure 5.10: Example association relationship specification in the model editor

Aggregation

Aggregation relationships define a *has-a* relationship. For example, a best practice has comments. These comments only belong to one specific best practice and do not exist on their own (i.e. without a best practice to be attached to). The hierarchical structure in figure 5.5 depicts this kind of relationship, which can also be seen by looking at the positioning of “comments” within the “bestpracticedocument” in the full JSON core model in appendix J. Aggregation relationships are therefore reflected in the JSON models as subcollections within documents. Figure 5.11 shows an example of an aggregation relationship being specified.

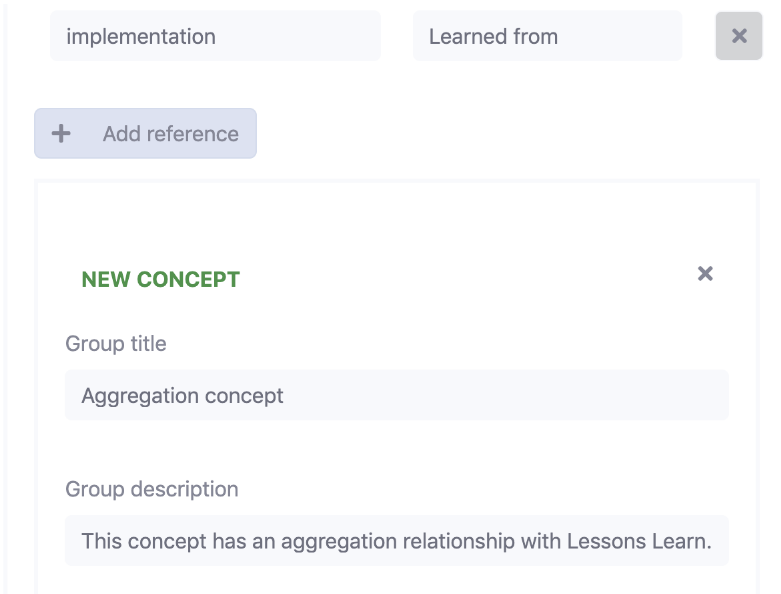

Figure 5.11: Example aggregation relationship specification in the model editor

5.3.3 Creating a JSON Model

The model editor creates a JSON model as output based on the textual grammar, which specifies how textual OBL models should be structured. On line 20 of the textual grammar, we see that within the specification of a document a new collection may be specified. In this collection a document is specified, in which a new collection may be specified. This process can go on for quite a while. Since any document may or may not get a subcollection, we can end up with subcollections on various hierarchy levels. Figure 5.12 shows a document 0 with two subcollections 101 and 102. The document in 101 also has two subcollections 203 and 204, while the document in 102 has no subcollections.

When constructing the JSON model, the model editor has to iterate over these documents in the correct order in order to include all documents and their subcollections on the right level in the JSON hierarchy. Figure 5.12 shows how the model editor iterates over the concepts defined by the domain administrator.

1. Concept 1 is added first as a collection with one document. This collection is located on level 0 of the JSON hierarchy, which is the same level as the Best Practice collection.
2. After the attributes for this concept are added to the document, it is checked for subcollections. The model editor finds subcollection 101 and add that collection with a document including all attributes.
3. Subcollection 203 is found for collection 101 and is added.
4. 203 has no subcollections, so 101 is checked again for further subcollections, after which 204 will be added.

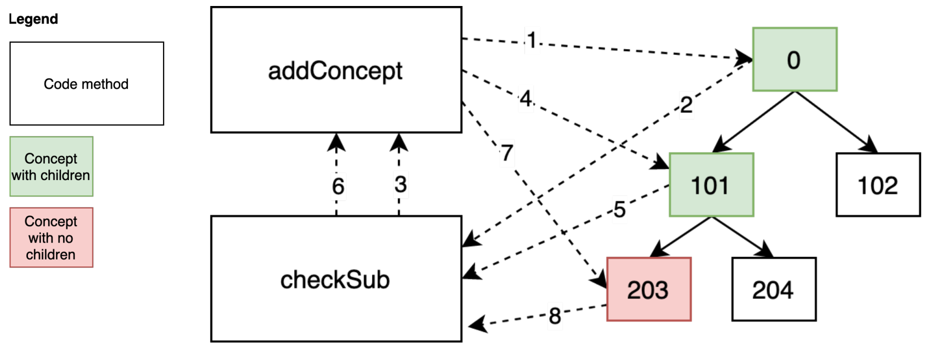

Figure 5.12: Iterating over concepts to construct a JSON model

When iterating over the collections, it is important that the model editor first checks if there are subcollections before adding collections on the same level. Collections are closed off with curly braces in the JSON model, so checking on an increasingly deep level is important to not prematurely close off a collection.

5.3.4 Limitations of the Model Editor

The model editor currently has the following limitations:

- We cannot extend elements of the core model yet. Domain administrators are able to decide which recommended elements to include, but no changes can be made to the attributes of the mandatory and recommended concepts.
- The domain administrator has to correctly fill in the name of the related subcollection for every relationship. For bi-directional relationships, this has to be done in the collections on both sides of the relationship. This approach is sensitive to mistakes.

5.4 Model Interpreter

MDD software tools typically have either a model compiler or a model interpreter. The former takes a model and generates source code based on this model and is therefore more aimed towards software development. The purpose of an interpreter is not to generate code, but to interpret the model and adapt functionality based on it. Model interpreters are used in approaches where the result is aimed at users. OpenBest makes use of a model interpreter.

5.4.1 Creating a Database Instance

When the domain administrator stores a model in OpenBest, the structure of the database is instantiated in accordance to this model. This is also depicted in figure 5.7. The hierarchy in the JSON model translates to a hierarchy in the database and so the various collections (such as *bestpractices*, *authors* and *problems*) can be created. Within each of these collections, a base document is created that stores the attributes and their values.

5.4.2 Determining the Feature Set

Based on the base documents in each of the collections, the feature set of the BP entry form (which regular users use to store a BP) is adapted.

1. Does the base document have a *true* value for the *displayfeature* attribute? If so, features are displayed in the BP entry form for this concept.
    - This attribute determines if features should be displayed in the BP entry form. This means that when it is set to *true*, regular users are able to enter information about that concept.
    - This attribute is always true, except for some of the elements of the JSON core model. For example, the *comments* concept has a *false* value for *displayfeature*.
    - OpenBest does display features for the *comments* concept, but not in the BP entry form.
2. The *grouptitle* and *groupdescription* attributes determine the title and description of the group of elements displayed in the BP entry form.
3. The value for every attribute is read and the corresponding feature is added to the BP entry form.

The contents of the base documents are direct translations of what is described in the textual model. Figure 5.13 shows how the contents of the base documents are translated to form features.

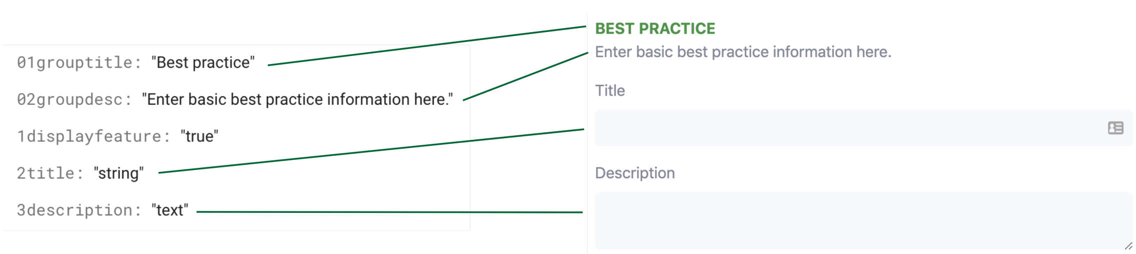

Figure 5.13: Translation of the base document content to form features

The group title and description show the user what kind of information they should fill in. The attribute values for *title* and *description* determine the type of form feature to be instantiated. In general, only two form feature types are created: smaller text boxes for *string* values and larger text areas for *text* values. For any array value, users are able to instantiate multiple text boxes. An example is shown in figure 5.14: users are able to define multiple categories, as *categories* has *[“string”]* defined as its value. The square brackets indicate an array and thus multiple values can be provided.

Figure 5.14 shows special form features for the BP concept. For this concept, an exception is made on the restriction to regular text boxes and larger text areas. When entering a BP, users have to be able to select an existing author as the author of the BP. Any document in the *bestpractices* collection stores a reference to the document in the *authors* collection that stores the author information. Although every concept may have a relationship with another concept (depending on what the

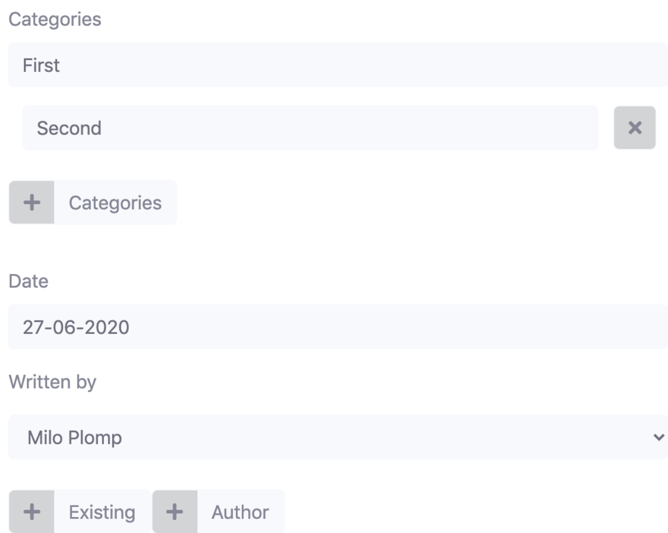

Figure 5.14: Special form features for the BP concept

domain administrator defines), the authors relationship is currently the only one that requires a feature to be instantiated. This way, multiple BPs can be written by the same author.

Any other relationship between collections is stored automatically when the user submits the BP. When a BP is submitted, a new document is placed in every collection storing the corresponding information (for example, a document in the *problems* collection that stores the problem defined by the user). When the domain administrator has specified that there is a relationship between two concepts (such as the *bestpractices* collection and the *problems* collection), references to the other documents are stored in the documents that are stored in these collections. That way, a reference to a BP document is stored in a problem document, without the user having to select a document for this relationship.

5.4.3 Limitations of the Model Interpreter

At the moment, the model interpreter and the resulting feature set in the BP entry form have the following limitations:

- We currently are not able to select pre-existing documents for any relationships other than authors. This goes against the possible 1..* cardinality of the relationship between any concept and BP.
- When entering a BP, users define information for only one document of every concept. For example, only one document is stored in the *problems* collection.

5.5 Searching and Viewing BPs

Once a BP has been stored by an organisation, it should be accessible to other organisations within the domain. This section will discuss OpenBests functionality for searching and viewing BPs.

5.5.1 Filtering Options

Figure 5.15 shows the current filtering options for OpenBest. This functionality allows users to search through the repository to find relevant BPs. We have currently implemented the following filtering options:

1. *Searching BPs based on categories.* The categories that users have assigned to the BP in the entry form can be used to filter results in the overview table. Presently, users are only able to search on one category at a time.
2. *Searching BPs using a search string.* A search string may be used to find BPs based on the information that is visible in the table. It is not possible to search through the entire BP content.
3. *Ordering the columns.* Information in the columns can be ordered alphabetically.

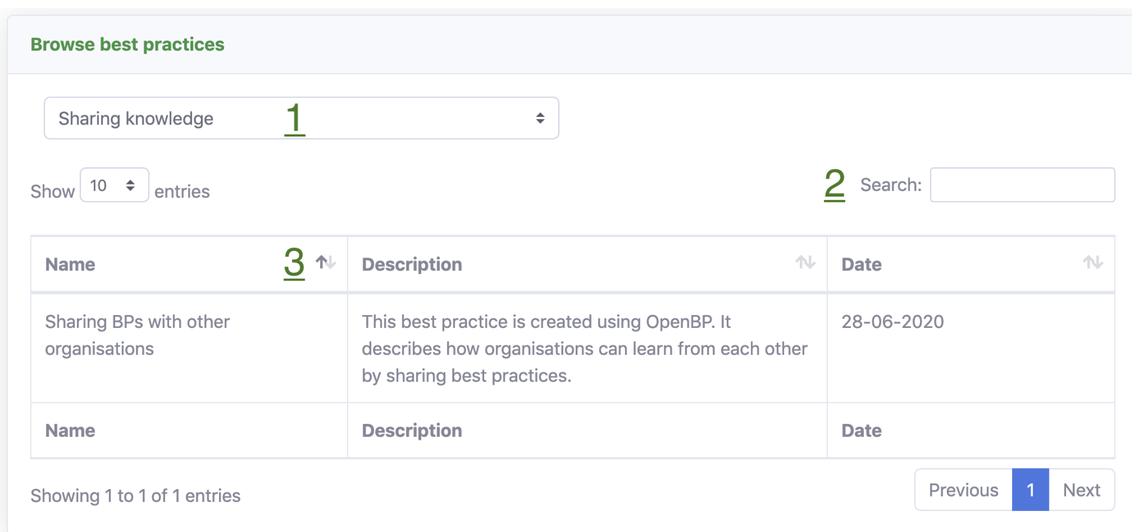

Figure 5.15: Filtering options in OpenBest

5.5.2 Viewing BPs

When a BP has been clicked in the overview table, its contents are displayed on a pop-up window. An example can be seen in figure 5.18. This BP includes information for the mandatory concepts *Best Practice* and *Problems* and the recommended concepts *Effort* and *Example*.

BP Retrieval

In order to display the contents of a BP in a logical order, we have to consider the structure of BPs. The textual models that are uploaded to OpenBest describe how concepts of a BP are related to each other. For example, the core (JSON) model shows that there is an aggregation relationship between *Best Practice* and *Example*, indicating that an Example is always *part of* a Best Practice. In the JSON models and the database, this relationship is expressed as a subcollection. Association relationships are specified between two concepts (such as *Best Practice* and *Problem*) in the BP entry form by specifying the related concept and the name of the relationship. The relationship name for the association relationship between *Best Practice* and *Problem* is shown in figure 5.18. It shows a box with the contents of the *Problem* concept with the relationship name (*solves*), the concept name (*problem*), the title and the description.

The placement of concept information in the screen in figure 5.18 is influential to how the user perceives the structure of the BP and how the concepts are related to each other. We therefore have to consider how we retrieve the contents of a BP and where we place the contents on the screen. Figure 5.16 shows an example of how a screen showing a BP (such as the example in figure 5.18) is structured.

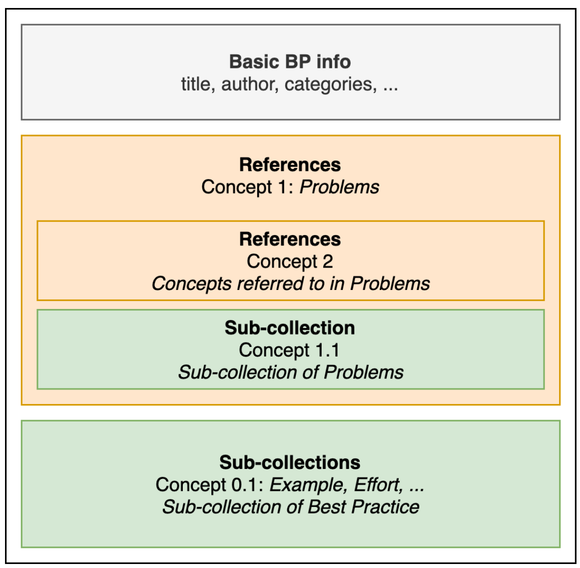

Figure 5.16: Structuring the BP retrieval

The structure in figure 5.16 can be explained as follows:

1. Information related to this concept is retrieved and placed on the screen. In the case of the *Best Practice* concept, this is the basic BP info.
2. The concept is checked for association relationships.
   (a) If a relationship is found (such as *Concept 1* in figure 5.16), the contents of this concept are retrieved. The information related to this concept are placed within the context of the *Best Practice* concept.

   (b) The process is repeated from step 2. Information related to any other related concepts is placed within the context of the *Problem* concept.

3. The concept is checked for sub-collections.

   (a) If a sub-collection is found (such as *Concept 0.1* in figure 5.16), the contents of this concept are retrieved.

   (b) The process is repeated from step 2.

Comments and Ratings

At the end of every BP, a comments and ratings section allows users to discuss the BP. This functionality is developed in work supervised by the author [110] and allows for the addition of contextual information to BPs by defining experiences in applying the BP by various organisations. The comments and ratings section are shown in figure 5.17. A full explanation on the workings of these features can be found in [110].

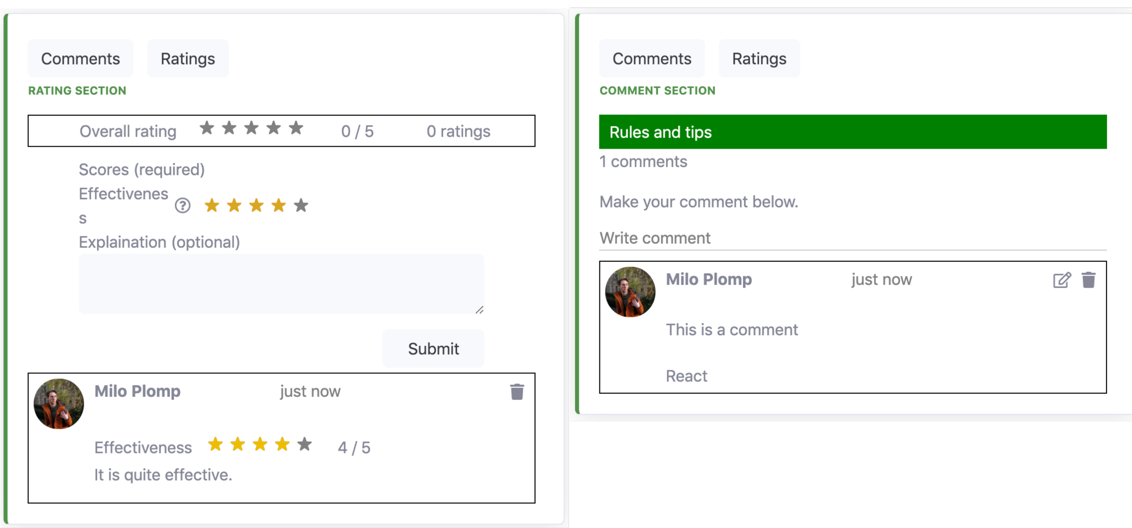

Figure 5.17: Comment and ratings sections of a BP

5.5.3 Limitations of Search and View Functionality

Currently, OpenBest only provides basic filtering functionality in order to find relevant BPs. In section 4.4.1, we have elaborated on the necessity of the inclusion of contextual factors (for example, based on the construction of a ToC) in BPs in order to recommend relevant BPs to users. We have not been able to implement functionality related to the construction of ToCs within the current research scope, but this would be a very relevant option to explore in future research. By extending OpenBest's functionality towards becoming a tool for the construction of an improvement plan in the form of a ToC, we are able to capture contextual information on how BPs have been applied. This information can be used to recommend BPs to other users constructing their improvement plan. With the current functionality, the possibility of finding relevant BPs is dependent on users only: the quality of stored BPs needs to be good enough to be considered relevant and users are dependent on their own search techniques for finding these BPs.

A different approach to IP4ESET is presented by Adèr [14], who has proposed a concept for an IP4ESET tool based on the IP4ESET super method. This tool aims to help in the identification and prioritization of improvement actions, assignment of responsibilities and resources and the documentation of an improvement plan. Future work for the platform should look into the possible connection between this tool and OpenBest, so that improvement plans can be created using the IP4ESET super method structure and BPs can be assigned to improvement actions. Similarly to the ToC approach, this would allow for storing more contextual information on the application of BPs.

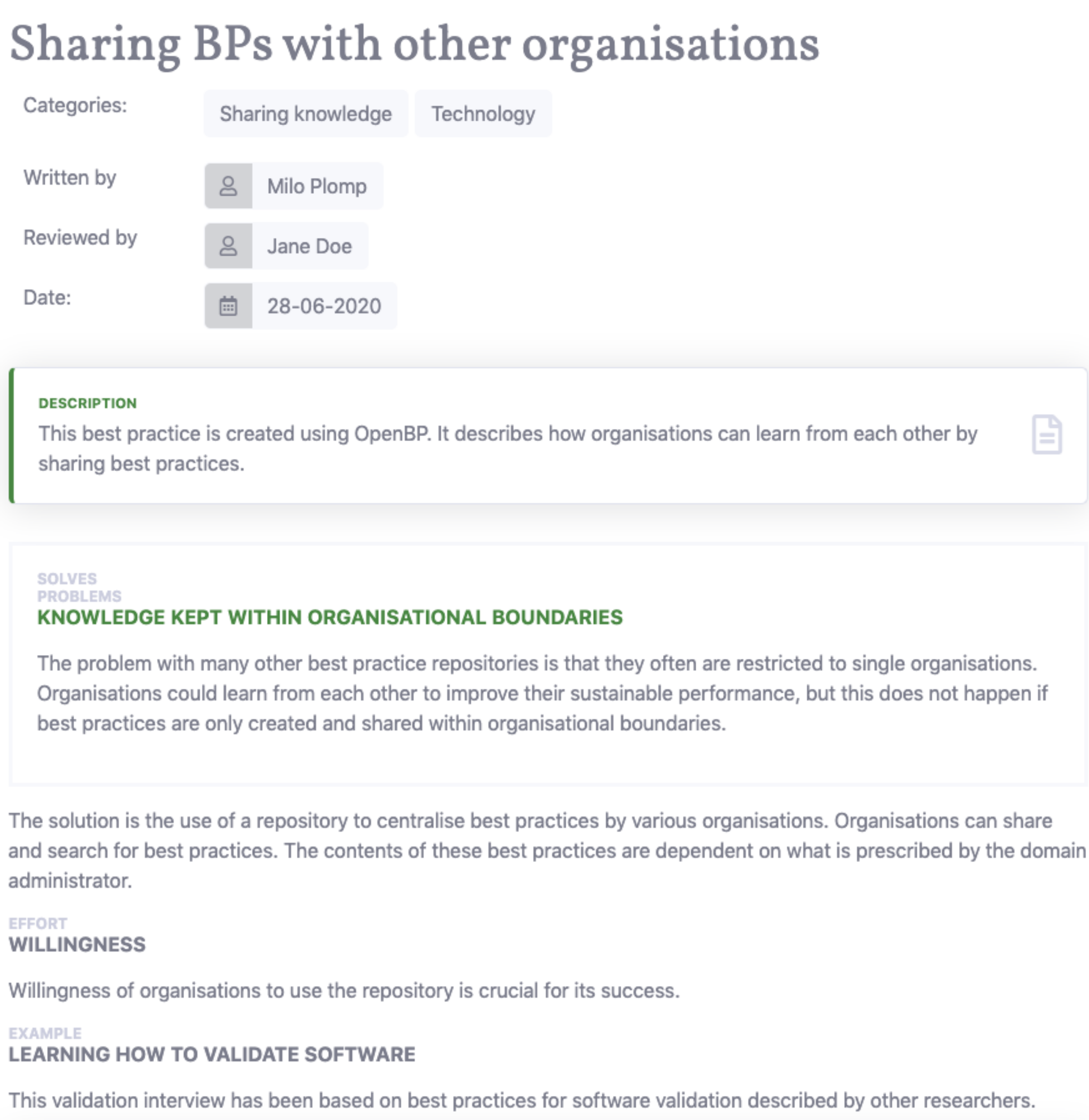

Figure 5.18: Screen showing the contents of a BP

6 | Validation

This chapter will elaborate on the validation of our research. We present a framework for the creation of a family of validations for MDD system and use this framework to validate OpenBest.

6.1 What is Validation?

According to Wieringa [115], the aim of validation is to justify if a treatment (in our case, OpenBest) would attribute to stakeholder goals in the case of implementation. The author further emphasizes that validation is about a prediction of an artifact's interaction with its intended implementation context in that the artifact is not implemented yet. The current status of OpenBest is a proof of concept, so we do not have a realistic implementation that can be transferred to the intended context yet. In order to estimate how OpenBest would behave in the intended context, we require a validation model. A validation model consists of a model of the artifact (the OpenBest proof of concept) interacting with a model of the intended context (a simulation of the intended context in a laboratory setting). Our validation model is depicted in figure 6.1.

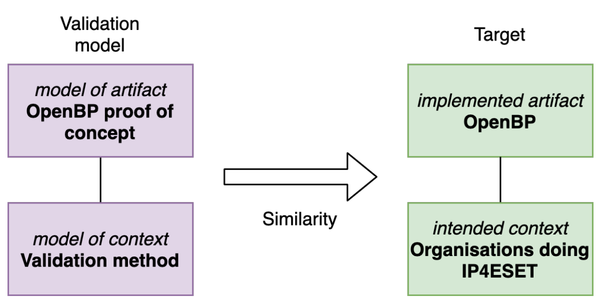

Figure 6.1: Validation model, adapted by the author from [115]

Using this validation model, we can make estimations about the effects of implementing OpenBest in the context of organisations doing IP4ESET activities. Any validation method will be conducted in a laboratory setting and aimed at estimating (rather than evaluating) effects.

6.2 Validating MDD Tools

Many validation methods exist for software engineering in general and MDD in particular (examples are given in [115; 118] for example). However, many validation methods focus only on a single system component. Specifically for MDD approaches, many validation methods focus on the system's MDD components, such as the domain model ([70]), model transformations ([54; 112]), the development process ([78; 87]) or the user interaction ([25]). Although these methods are useful for the validation of a certain aspect of the system, they do not provide enough data to validate the entire system; e.g. when we only validate the quality of models we do not take into account if the usability aspect of the system contributes to the high-level hypothesis and vice versa.

6.2.1 Family of Validations

The difficulty of empirical work in SE is highlighted by Basili et al. [20], who state that software artifacts require a large data set in order to reject or accept hypotheses. This problem of empirical work is also expressed by Panach et al. [99], who state that there are often too many aspects to consider for validation to be addressed in a single method. The lack of ability to validate all system aspects often leads to researchers aiming to validate a single aspect. Basili et al. [20] propose a method for the creation of a "family of experiments": a set of related empirical studies that together can form an answer to a high-level hypothesis. In this method, a high-level hypothesis about the system is decomposed through choice points that represent decisions on how various experiments would address the high-level hypothesis. Through this decomposition a thorough understanding of the system can be achieved by organizing a related set of experiments that address various system components, as well as various contextual factors. A Google Scholar search on *"family of experiments"* reveals that families of experiments have been created by many other researchers (an example is given in [78]).

These families of experiments may be useful for validation, but experiments only make up a part of the possible validation methods. Furthermore, experiments may have other uses than validation. We are concerned with the design cycle, rather than the empirical cycle, and so we are not mainly concerned with scientific knowledge questions [115]. Although experiments may be used in both cycles, they are used in validation (in the design cycle) in order to study the effects of changes in the interaction between artifact and context. The estimation of (mechanisms leading to) effects can be studied with more validation methods than just experiments.

A Google Scholar search for *"family of validations"* only yields one result: Elser & Richmond [44] describe the use of "Validation Master Plans" in the pharmaceutical industry. To the best of our knowledge, no guidelines exist for the creation of a related set of validation methods in SE. The guarantee of alignment of validation methods is not possible when independently defining several validation methods. With our method, we do not prescribe how to set up a validation method. Rather, by defining a method for defining a family of validations, we provide guidelines on how to plan, motivate and align multiple validation methods that together provide a thorough validation of the system and its contribution to stakeholder goals.

This resulting set of validation methods may not be executed in full within one research scope. Since we often do not have the resources to validate the entire system and its contribution to stakeholder goals at once, we want to be able to motivate why only a part of the system components and goals have been validated within the current research scope. By looking at the available resources for validation, we are able to distinguish *planned* validation methods from *executed* validation methods. This distinction is depicted in figure 6.2.

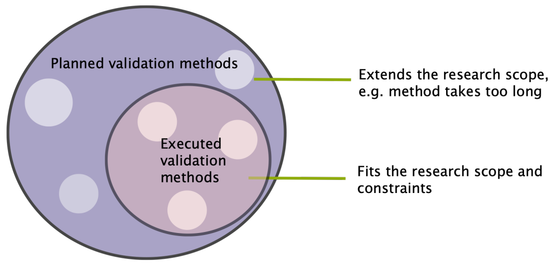

Figure 6.2: Planned validation methods and executed validation methods

6.3 Creating a Validation Method

Wieringa [115] lists four kinds of knowledge questions that are relevant for validation research. For each of these knowledge questions, we will give an example of a validation method that aims to answer the question in the context of OpenBest (in italics):

1. Effect questions: what are the effects produced by the interaction between artifact and context?
   - *What is the quality of a model created by a student interacting with the model editor in a laboratory setting?*
2. Trade-off questions: what are the effects of alternative artifacts interacting with the context?
   - *How does model quality differ between the use of OpenBest's model editor and a model editor instantiated by Xtext?*
3. Sensitivity questions: how do different contexts influence the produced effects?
   - *What is the perceived usefulness of OpenBest to stakeholders from different domains?*
4. Requirements satisfaction questions: do the effects satisfy the requirements?
   - *Do study participants think that OpenBest adequately adapts to the organisation's domain?*

By studying enough validation models that answer a variety of these knowledge questions, we can build up a picture of the interaction between the implemented artifact and its intended context. We therefore have to take this range of knowledge questions into consideration when creating a family of validations. The variety of knowledge questions and contexts (and therefore validation models) that we are required to study is indicative of the multi-faceted nature of the artifact that we are validating. We cannot expect to ask a one-sided, singular knowledge question about the system as a whole and expect to validate that the system would contribute stakeholder goals. We have to take a variety of contextual factors into account, such as different users, with different goals, using different system components.

To make the complex and extensive nature of validation more manageable, we propose a method for constructing a family of validations by decomposing the validation problem along a set of increasingly specific choice points into distinct validation methods. This method allows us to establish clear trace links from validation methods to the high-level goals that they aim to validate. Another benefit is that a decomposition results in a clear overview of the amount of coverage of the *planned* versus *executed* validation methods: the portion of goals that are validated within the current research scope as well as goals that require validation in future work [20].

6.3.1 Aspects for Validation

Validation is about estimating the system's contribution to stakeholder goals. We have to specify how we measure this contribution. This section will elaborate on the validation aspects that we use to validate elements of the system.

Quality Frameworks

Software validation is closely tied to software quality [13]: higher quality software has a higher chance of contributing to stakeholder goals. In order to validate the quality of software, we use a quality framework. A quality framework is a set of quality characteristics and their interdependent relationships that we wish to validate [93].

In MDD, quality can and should be assessed at various stages, as the successfulness of the approach is dependent on the quality of the models used in the design phase and during runtime, as well as the quality of the model produced by the tool. MDD is about the use and transformation of models [112], so many quality frameworks used for validating MDD approaches are aimed at the validation of these models and their transformations. Furthermore, just like any other software artifact, we also require validation of the interaction between the artifact's components and the context. Table 6.1 shows an overview studies looking at quality aspects for MDD and MDE approaches, as well as quality frameworks for assessing model quality.

Table 6.1: MDD and MDE quality aspects for validation in literature

Source	Quality Aspects
Solheim & Neple [105]	Proposes a quality framework for MDD specifically. Looks at the **quality of models** by addressing *organizational quality* (the goal of the modeling activity, *syntactic quality*, *semantic quality* and *technical pragmatic quality*.
Vanderose & Habra [112]	Proposes a framework for empirical studies in MDE. Focuses on the **quality of models and model transformations**. Distinguishes functional aspects from non-functional aspects (maintainability, understandability, modifiability).
Krogstie [79]	Proposes SEQUAL: a framework that addresses the **quality of models** and the **goals of modeling**. Distinguishes quality aspects and ways to achieve them. Includes aspects like *goals of modeling* and *interpretation*.
Giraldo et al. [60]	A list of 16 quality categories for MDE is presented. These quality categories are a set of procedures defining the **quality of models**.

Table 6.1 – continued from previous page

Moody [94]	Proposes the *Method Evaluation Model* to validate **methods in IS design**. The use of the artifact can be validated by looking at the *perceived ease of use*, *perceived usefulness* and *intention to use*
Fleurey et al. [54]	Looks at validation of correctness and reliability of **model transformations** in MDE. Semantic quality is primarily important.
Kosar et al. [78]	Studies **developer experience** of using a DSL by looking at *efficiency*, *time* and *simplicity of use*.
Martínez et al. [87]	Looks at **developers' subjective experience** in using MDD over traditional approaches. Perceived usefulness, perceived ease of use and intention to adopt are among the studied variables.
Bevan [25]	Proposes **"quality in use"** to extend quality frameworks by including a usability aspect (e.g. effectiveness, satisfaction) for validation.
Mohagheghi & Dehlen [93]	Propose a quality framework for MDE by looking at the relevant objects for MDE approaches. They focus on the **transformation of models** in MDE and validate the following aspects: transformation engine, transformation language, transformation rules, models. Considered aspects include consistency (of syntactic and semantic quality) and traceability.
Lindland et al. [82]	Proposes a **quality framework for conceptual models** that include syntactic quality, semantic quality and pragmatic quality.
Panach et al. [99]	No quality framework proposes, but an overview of studied variables in MDD empirical studies is presented. Variables include: model size and complexity, effort, quality, productivity, intention to use and intention to adopt.

The studies listen in table 6.1 show the relevant aspects for validation of an MDD approach. The listed quality aspects can be categorised as follows:

- Functional quality aspects of models and model transformations, including aspects such as:
  - Syntactic quality
  - Semantic quality (and consistency)
  - Pragmatic quality
- Non-functional quality aspects of the modeling process and artifact-context interaction, including aspects such as:
  - Perceived ease of use
  - Perceived usefulness
  - Intention to adopt
- Quality aspects related to the goal of using an MDD approach, including aspects such as:

- Effectiveness
- Flexibility

Syntactic quality assesses if a model adheres to its grammar specification (for example, does a model adhere to its metamodel) [82]. Studies on quality in MDE often also consider syntactic consistency (an example is given in [93]). This is a relevant aspect for MDE to consider, as there are models of different view that should be syntactically consistent to each other; e.g. they are all described using the same metamodel. In our approach, however, we do not have models for different views. Rather, we have a single abstract syntax model (the core model) that should adhere to its metamodel and we have a single concrete syntax model (the core JSON model) that should adhere to its grammar specification (the textual grammar). We will therefore only consider syntactic correctness.

Unlike many of the studies in table 6.1, we do not aim to validate a singular aspect of our MDD approach (such as model quality or user satisfaction). Rather, we aim to (plan to) validate an entire system and therefore aim to make a well-supported estimation on the contribution of this system to stakeholder goals based on validation methods [115]. The previously mentioned knowledge questions for validation study the artifact model and context model (and their interaction), the result of this interaction and the satisfaction of requirements. Based on these knowledge questions, the validation model in figure 6.1 and the quality aspects in table 6.1, we propose the quality framework in figure 6.3.

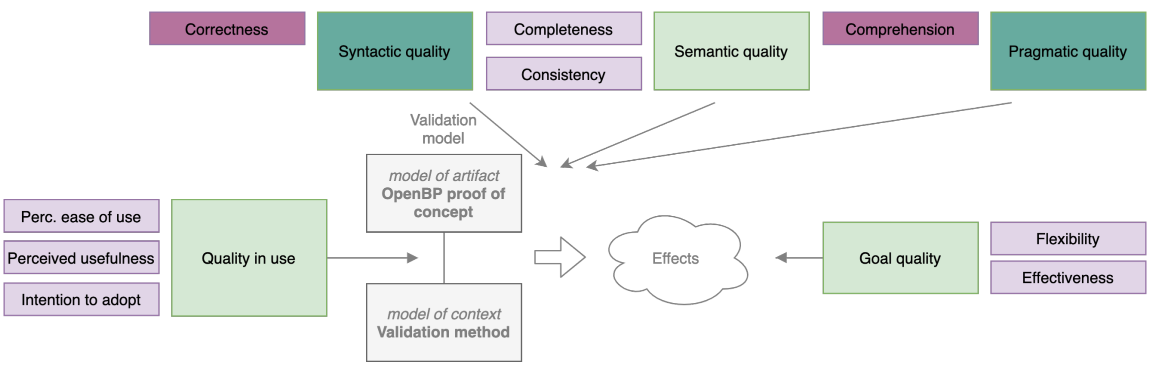

Figure 6.3: Our proposed quality framework for validating MDD approaches

By including the five quality aspect categories, we are able to (plan to) validate an entire MDD artifact. The quality categories (in green) are the mandatory elements of our quality framework to be used when defining a family of validations. The sub-aspects (in purple) for each quality aspect in figure 6.3 are examples of aspects that are relevant to OpenBest, but these may be adapted to the system in question. For example, while the flexibility to adapt to different domains is important to validate for OpenBest, this sub-aspect of goal quality may be swapped for a different sub-aspect in a different system.

Assessing the Quality Framework

When deciding on the quality aspects in our quality framework, we assess the validity of our quality framework by looking at the validity analysis questions by Moody et al. [95].

- *Completeness:* does the set of quality aspects include all necessary aspects for validation? The use of the four categories in our method creates a set of quality aspects that includes relevant categories for artifact validation. This is crucial in planning a family of validations for an entire system, as we would otherwise only be able to validate a single aspect of the artifact. Based on the overview of validation studies, we believe that these four categories will cover all relevant quality (validation) aspects.
- *Parsimony:* are all quality aspects necessary for validation? They are necessary in order to validate the entire artifact. We have motivated this necessity with the brief literature study presented above, as well as the indication of knowledge questions that are crucial to validation.
- *Independence:* are the quality aspects independent of each other? They are, as they all look at different elements of artifact-context interaction: the artifact, the interaction and the contribution of effects to goals.

6.4 Creating a Family of Validations

Our decomposition approach to defining a family of validations is inspired by Basili et al. [20], who propose a tree structure for empirical software evaluation that decomposes the problem space into a set of experiments. In this section, we will elaborate on the method for constructing a family of validations for MDD approaches. Three phases make up this method: *system description*, *decomposition* and *protocols*. These phases are explained in more detail in the following subsections.

We use a process-deliverable diagram (PDD) to express our method. A PDD consists of an "activity side" and a "deliverable side", being adaptations of the UML activity diagram and UML concept diagram respectively [109]. The PDD shows the activities in our method, their output and the relationships between the outputs. The full PDD with its explanatory activity and concept tables can be found in appendix K.

6.4.1 System Description

The goal of this phase is to accurately define the artifact that is being validated. The results of this activity will provide a basis for the decomposition in the next phase. This phase consists of three main activities: *describe system*, *describe context* and *define requirements*. Figure 6.4 shows a bird's eye PDD for our validation method. The closed activities and deliverables are elaborated on in the fine-grained PDD listed in appendix K.

In the sections below, we will highlight mandatory, recommended and optional techniques. These are externally-defined techniques (not created by the author) to be applied within the method activities. For example, the use of *user stories* is a technique to be used for the method activity "defining requirements".

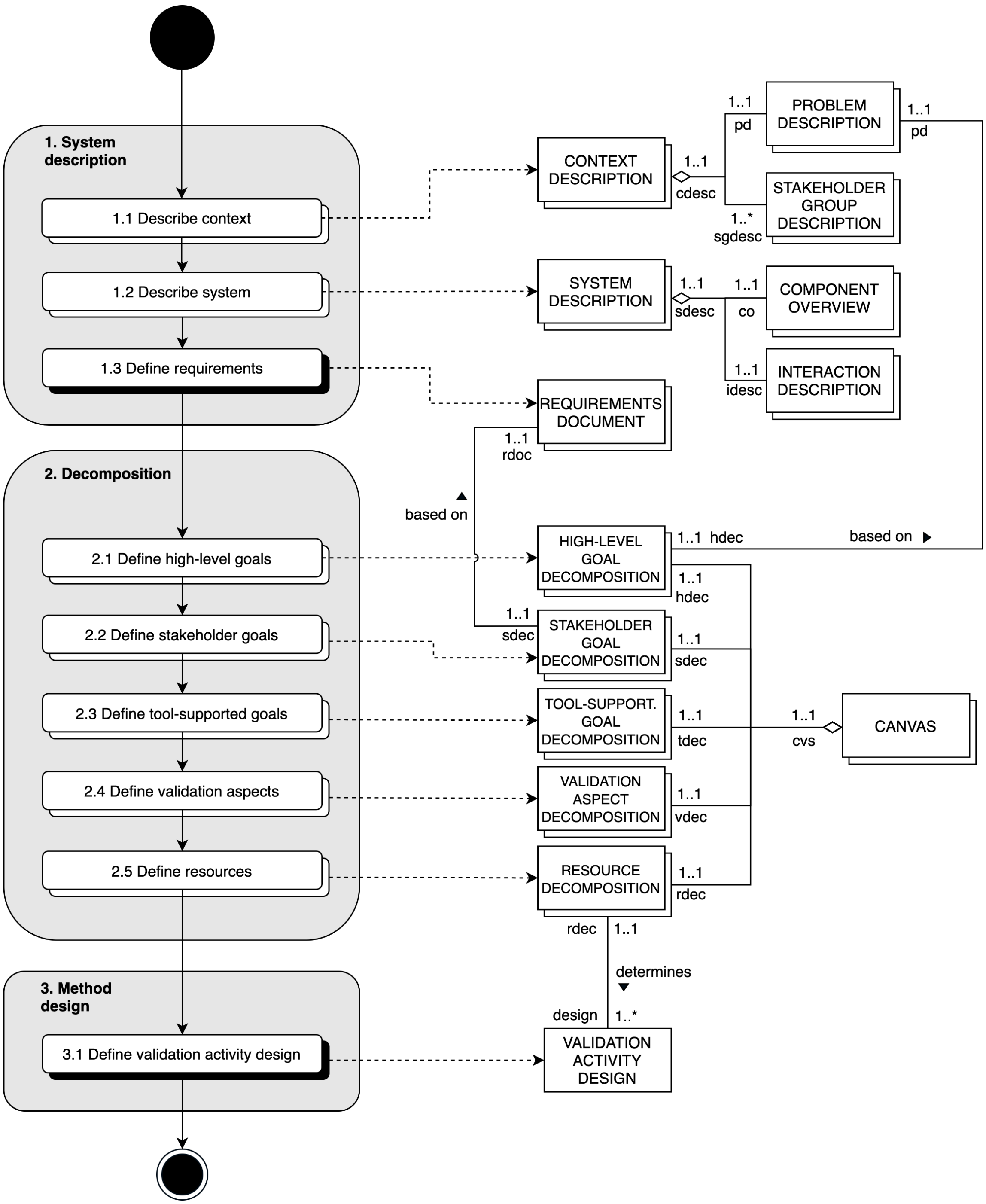

Figure 6.4: Bird's Eye PDD of the validation method

Step 1.1: Describe Context

In this activity, the context of the system will be described. The context description consists of a problem description indicating the need for system development (created in step 1.1.1) and a stakeholder description (created in step 1.1.2).

The first deliverable of this activity is a problem description that defines the problem leading to system development. To validate if the system would contribute to stakeholder goals, we have to know with which intention it has developed. We do not define any restrictions or recommendations for writing a problem description, but the goals of the system should be apparent. A basic description of the stakeholder groups suffices, although a more rigorous approach (such as personas [33]) would be an option too.

Step 1.1

Mandatory technique: problem description
Recommended technique: basic stakeholder description
Optional technique: personas

Step 1.2: Describe System

In this activity, a high-level overview of the system and its components should be created. This overview should be able to express which components constitute the system and should therefore be validated.

The following components are generally present in MDD systems (see [49; 70; 112] for examples):

- A conceptual domain model
- A model compiler or interpreter
- A DSL that specifies rules for model creation
- A model editor
- The system itself (including MDD-unrelated components)

We aim to also plan for validations of components that may not have been implemented yet. It is therefore important to create a component overview showing all envisioned system components; either planned or implemented. For the creation of a component overview (step 1.2.1), we recommend the creation of a functional architecture model (FAM) [29]. This type of model decomposes the system into functional modules and provides an easy notation for defining how these modules communicate with each other. It provides enough abstraction to create a comprehensive overview of system components and their interaction. If so desired, the FAM may also be replaced by a description of the components in natural language. We also define how stakeholders are interacting with the system. A basic description of the interactions (created in step 1.2.2) is sufficient.

Step 1.2

Mandatory technique: basic interaction description
Recommended techniques: FAM
Optional techniques: basic component description

Step 1.3: Define Requirements

The requirements for the system express goals that users have when using the system. This is crucial information, as it expresses what the system is aiming to achieve. We assume that the requirements have been defined before validation commences, but a useful overview of requirements elicitation techniques is given by Zowghi & Coulin [119].

Several techniques exist for defining requirements (step 1.3). Although any technique that is able to express the goals of the various stakeholder groups would suffice, we recommend the use of *user stories*. User stories are an often used format for expressing requirements that define *who* a requirement is about, *what* is expected of the system and (optionally) what the *benefit* of this requirement is [84]. The benefit of this structure is its clear link between user, system and goal; something that is crucial for establishing the correct trace links in the decomposition. We recommend checking the structure of user stories with the Quality User Story framework by Lucassen et al. [84].

The resulting requirements need to be grouped in the following method activity. In order to allow for this grouping to be manageable, attention should be given to the quantity and level of detail of the requirements. For this method, it is not necessary to define detailed requirements. An example of creating requirements on a manageable level of detail is the creation of *epic stories* when defining requirements using user stories. We do not prescribe the level of detail for requirements that are defined in our method. The goal of this activity is the creation of a list of requirements that is able to convey the goals that stakeholders have when using the system.

Step 1.3

Recommended techniques: user stories
Optional techniques: goal model [42]

6.4.2 Decomposition

The system description of the previous phase serves as input and support for the decomposition. In the decomposition phase, we decompose the high-level goals of the system into validation methods. The decomposition includes 5 choice points that increasingly break down the validation.

Figure 6.5 shows the structure of the decomposition. The decomposition is made on a *decomposition canvas* consisting of 11 lanes (indicated with the gray and colored boxes on the right). The colored lanes indicate *choice points*, while the gray lanes indicate the *decomposition results* of a choice point. In the following sections, we will refer to this figure to show how the result of an activity should be modeled. The overview in figure 6.5 shows the notation elements of this decomposition tree and therefore reflects the concrete syntax. The abstract syntax of the decomposition tree is part of the deliverable side of the method PDD. A full explanation of these elements and their relation to the abstract syntax will be provided in a later section.

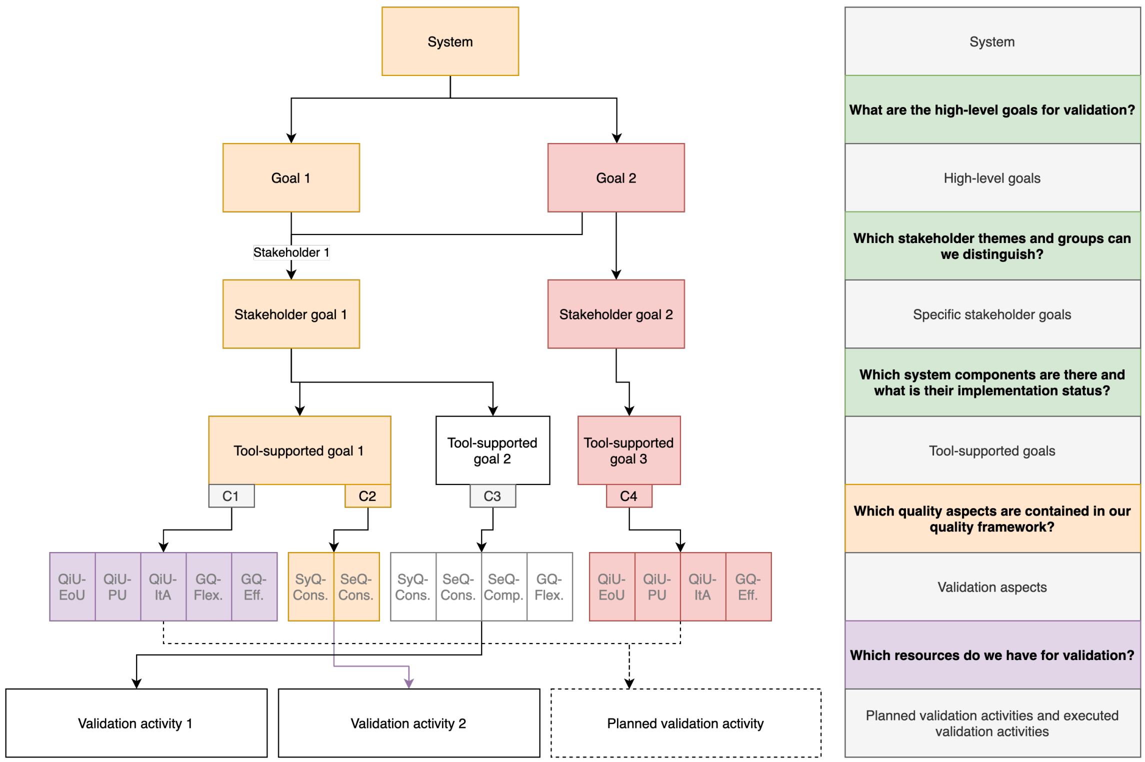

Figure 6.5: Structure of the decomposition

We use colors in the decomposition tree to indicate if an element/goal is currently ready for validation. Figure 6.6 shows the legend of the decomposition tree concrete syntax. Any element that is colored red *can not be validated at all* at the present moment. Examples include tool-supported goals for which no component is implemented yet, or validation activities that can not be conducted with the currently available resources. Any element that is colored orange can only be *partly* validated. This happens when either a parent element or child element of this element in the decomposition tree is colored red or orange. For example, "tool-supported goal 1" in figure 6.5 can only be partly validated as one of its contributing components ("C2") can only be partly validated with the currently available resources. As a result, all parent elements of "tool-supported goal 1" can also only be partly validated. Validation aspects can also be colored purple, indicating that we have resources available to validate these aspects for these components with one method, but that another validation method is also desired for future work.

Figure K.1 shows the PDD for our validation method. We use the PDD to show the structure of our method, but also to reflect the *abstract syntax* of the decomposition tree; e.g. the elements that make up the structure of a decomposition tree. These abstract syntax elements are colored

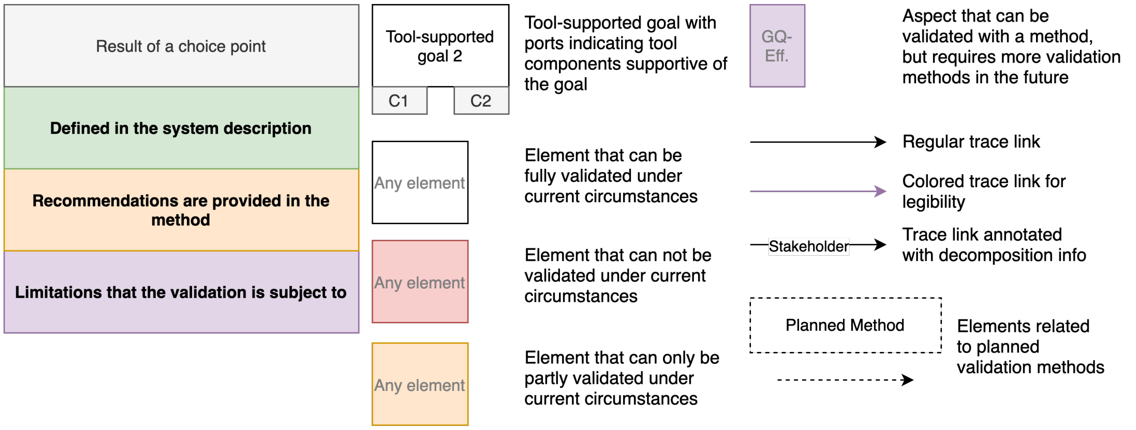

Figure 6.6: Legend of the decomposition tree concrete syntax

orange. Not all deliverables in the PDD have a concrete syntax notation, since we want to minimize the amount of elements on the decomposition canvas. As a result the deliverable side of the PDD includes more than just the abstract syntax elements.

Step 2.1: Defining High-Level Goals

The first step in the decomposition activity is to define the high-level goals (step 2.1.1). The high-level goals describe what users ultimately aim to achieve by using the system. The system aims to solve the problem phenomena by contributing to the high-level goals. The high-level goals are described in the problem description in step 1.1. High-level goals are extracted from this problem description, placed on the canvas and linked to the system element using trace links (step 2.1.2).

We may also define what the impact level of the high-level goals is. Hilty & Aebischer [69] define the *LES model* to indicate that ICT can have impact on 3 levels: life-cycle impact, enabling impact and structural impact. Using this model to determine on which impact level a high-level goal resides is a useful approach to determining the possibility of validating the goal in the current research. At the beginning of this thesis, we have explained why we do not use the term “organisational sustainability”, as sustainability is a concept that is always placed in an institutional context. It is therefore impossible for any organisation alone to be sustainable. Similarly, any high-level goal that is aimed at structural impact is harder to validate than goals aimed at enabling impact. Figure 6.7 shows how the impact levels of the LES model can be related to OpenBest’s goals.

If we find that a high-level goal is located on the level of structural impact, it would indicate that we need much more resources to validate this goal and it may not be able to be validated at all. This should be taken into consideration for the decomposition of high-level goals.

Step 2.2: Defining Stakeholder Goals

Stakeholder goals (which we define in step 2.2.1) are based on the requirements defined in step 1.3. We group requirements based on what they aim to achieve. In the case of user stories, the grouping is based on the second clause (the *what* statement). Grouping requirements is similar to

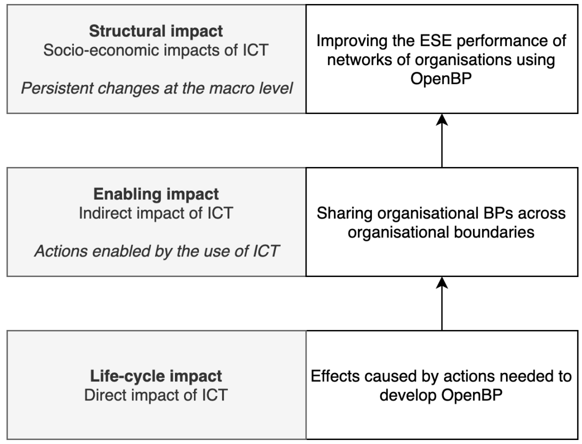

Figure 6.7: Impact levels of the LES model. Adapted by the author from [69]

the use of *epics* for user stories. Stakeholder goals are focused on goals related to the real-world problem space and are not influenced by how the system might solve the problems. Refrain from defining stakeholder goals by indicating what the solution would be. Similarly, the QUS framework prescribes defining user stories that are problem-oriented rather than solution-oriented [84].

The stakeholder goals are placed on the decomposition canvas in the corresponding lane and connected to high-level goals with trace links annotated with the stakeholder group that aim to achieve this goal (step 2.2.2). By constructing the trace link we indicate to which high-level goal a stakeholder goal contributes. For example, in figure 6.5 “stakeholder goal 1” indicates that if that stakeholder goal is achieved, it contributes to “goal 1” and “goal 2”. A stakeholder goal may thus be connected to multiple high-level goals.

Step 2.3: Defining Tool-supported Goals

Stakeholders use the system to achieve their goals. The system may or may not provide the functionality to achieve these goals, indicating what the *tool-supported goals* are. By defining the tool-supported goals, we move from *problem space* to *solution space*. The system components have been defined in the component overview in step 1.1. The component overview may also contain system components that have not been implemented yet.

Implementation Status In order to establish which tool-supported goals may be validated and which may only be planned, it is important to first establish the implementation status of the tool components (step 2.3.1). We then define how stakeholder goals may be supported by system components (step 2.3.2), place these tool-supported goals on the canvas and establish trace links with the corresponding stakeholder goal(s) (step 2.3.3). The implementation status of tool components specifies whether the tool-supported goal should be coloured red/orange or not. A tool-supported goal for which *no components at all* are implemented is coloured red, while a tool-supported goal for which *part of the components* is implemented is coloured orange. Ports are similarly coloured red (the component is not implemented), orange (the component is partly implemented) or not (the component is implemented).

Scoping Status

The scoping status of goals and other decomposition elements are defined in various activities. In each scoping activity, the colourisation of an element also influences its parent elements. A parent element of a coloured element should always be coloured as well, as it may not be validated in full in the current research scope.

Ports Multiple system components may contribute to the same tool-supported goal. To distinguish which system components need to be assessed for each tool-supported goal, we use *ports*. Ports indicate the components that contribute to a tool-supported goal. For example, referring to figure 6.5, two system components contribute to tool-supported goal 1: C1 and C2. These ports are located on the outgoing side of tool-supported goals, so that we are able to distinguish to which system component a validation aspect is related.

The need for this distinction becomes clear when we assume that C1 indicates the model editor and C2 indicates the domain model. If we were to establish trace links to validation aspects without ports, we would not know if the connected validation aspects should be considered for the model editor or the domain model.

Step 2.4: Defining Validation Aspects

Once we know which tool-supported goals to validate, we have to determine which aspects of these goals we can and should validate (step 2.4.1). The validation aspects are based on a quality framework. We have proposed a quality framework in section 6.3.1. We recommend the use of this quality framework due to its specificity for MDD and its ability to validate an entire system. Although we do not recommend it, another quality framework may be chosen as well, as long as the quality framework is able to validate all aspects of the system. Another approach would be to extend or alter the proposed quality framework with quality aspects found in other quality frameworks. For example, the quality framework in figure 6.3 may be extended by including further details on quality categories as highlighted by Krogstie [79] or any other of the studies listed in table 6.1.

Step 2.4.2: Assessing a Quality Framework Our proposed quality framework has already been assessed using the guidelines by [95]. If another quality framework is chosen, the choice of quality aspects should be motivated by looking at these guidelines. If the quality framework is not sufficiently able to validate the entire system, it should be adapted (step 2.4.3).

Step 2.4.4: Mapping the Quality Framework All validation aspects in the quality framework should be placed on the canvas and linked to the relevant combination of tool-supported goal and system component. For example, the *perceived ease of use* quality aspect may be considered for the combination of component *model editor* and the defined tool-supported goal. To increase the brevity of the decomposition tree, we use abbreviations in the example in figure 6.5. Any validation aspect that is to be validated for a non-implemented system component should be colored red. The scoping status of other quality aspects will be determined in the next step.

Estimating Effects With validation, we have to make predictions about whether or not the estimated effects meet requirements. Through this decomposition tree, we can now motivate a more specific approach to estimating these effects: by studying the validation aspects related to tool-supported goals, stakeholder goals and high-level goals. These goals are the representation of requirements for the system and the quality aspects are a way to look at the effect. For example, does the validation reveal that the perceived usefulness of a system component is estimated to be low? And what does that mean for the requirements? Should an assessment of the validation aspects not comply with the requirements, then a redesign of the system should be considered.

Step 2.5: Defining Resources

The resources we have available for validation are determined in step 2.5.1 and determine which validations are executed and which are only planned. Examples of resources for validation are:

- Time
- Available subjects
- Location factors (e.g. *can the validation method be carried out in real life if necessary?*)

We also define which validation methods may be used to create validation activities for validating the validation aspects. We distinguish between validation *methods* (a collective name for a technique that specifies how a validation may be constructed) and validation *activities* (the specific activity constructed using a validation method that we will carry out for our system). Table 6.2 lists some examples of validation methods, but any other validation method may also be used. The chosen validation methods are placed on the canvas and connected to the validation aspects that are studied using these methods. Groups of validation aspects for different system components may be validated using the same method.

Table 6.2: Examples of validation methods

Method	Explanation
Expert opinion	In expert opinion sessions, experts are asked to imagine the validation model of the artifact and to make predictions on the effects it would produce. It is important that experts understand the artifact and that they are able to express their reasoning on how interactions would produce results [115]. It is not necessary to let experts interact with the artifact. Expert opinion sessions are similar to focus groups in requirements engineering. Both suffer from downsides such as the influence of group dynamics on results, but it is a relatively easy method to set up.
Single-case mechanism experiments	In this type of experiment, we apply stimuli to the validation model to study the effects [115]. For example, we build a prototype, set up an experiment and feed the prototype test scenarios. This method allows us to study the mechanisms that produce an effect in more detail, where expert opinion sessions try to explain these mechanisms through expert's expectations. An example is given in [54], where tests were run to validate model transformations.

Table 6.2 – continued from previous page

Technical action research (TAR)	In TAR, the artifact is applied to a problem in a real-world context. This differs from single-case mechanism experiments, where the artifact is only exposed to a model of the context. Before doing TAR, you have to determine if your artifact is mature enough to be exposed to a real-world context. An example of TAR being conducted to validate an IP4EST tool is given in [14].

Scoping Status Available resources determine which validation aspects can be considered using validation activities within the current research scope. In step 2.5.2 we therefore colour validation aspects based on the available resources. We do no prescribe thresholds of resources that are required for various validation aspects and validation activities. The amounts of resources that are required are dependent on the nature of the system under validation, as well as the goals of validation.

If the expectation is that a validation activity that validates certain validation aspects may not be carried out with the currently available resources, the colourisation of the decomposition tree is influenced. For example, if it has been determined that we have 2 weeks available for validation and we aim to conduct an expert opinion session, we might determine that we do not have enough time available to sample and interview experts. The resources may lead to the following colours being assigned to validation aspects:

- None: all validation aspects for the related tool component and tool-supported goal may be validated in the current research scope with a validation activity.
- Purple: the validation aspects for the tool component and tool-supported goal may be validated, but a different validation activity is also desired for future work. For example, a more extensive activity or a more mature component.
- Orange: the validation aspects are to be validated for a tool-supported goal for which multiple tool components are validated with the same validation aspects, but not all tool components are implemented.
- Red: the validation aspects are to be validated (in future research) for a non-implemented tool component.

6.4.3 Method Design

After the decomposition results in validation methods to be planned and/or executed, protocols for the validation methods need to be created in step 3. We create protocols for *all planned* methods on various levels of detail: detailed protocols are created for the *executed* validation methods, while only basic protocols are created for the *planned but non-executed* validation methods. How the protocols are structured is dependent on the entire decomposition graph; i.e. what is the aspect of validation, for which system component and which goals should it contribute to? Our method does not prescribe how validation method protocols should be created. In this section, we will briefly highlight some protocols for various validation methods as examples.

Types of Protocols

A protocol for a validation method describes how the method is set up and executed, i.e. it describes guidelines for the validation method, but also descriptions of factors such as context, subjects and instrumentation. This information is already available as a result of the previous phases, but should be synthesized to provide a coherent overview of the intent of and method for the various validation methods.

Based on guidelines for constructing a protocol by Wohlin et al. [118], we recommend to include the following information in the validation method protocol:

- *Context description*: what is the goal and scope of this validation method? This information is known as a result of the decomposition, but should be explicitly mentioned.
- *Variable selection*: the variables are also known as a result of the decomposition.
- *Subject selection*: if applicable for methods such as TAR and expert opinion.
- *Instrumentation*: which instruments are required for the validation method? Examples include questionnaires, consent forms, a system prototype and instructions.
- *Data collection procedure*: how is the method structured in order to collect data? For example, when is a questionnaire provided to subjects or how much time do they have for completing a task?
- *Validity evaluation*: how do we ensure that we mitigate threats to validity?

We will list some recommendations for the creation of validation method protocols for expert opinion and experiments below.

Expert Opinion Expert opinion can be collected using various types of instrumentation, including focus group sessions. Guidelines for setting up a focus group for SE are provided in [77]. Other methods for collecting expert opinions are surveys (an example is given by [97]) and interviews (which differ from focus groups that focus more on the interaction between experts).

The protocol should define the data collection procedure by specifying a schedule, order of activities and other decisions on how to conduct the session (e.g. *do we let all participants first express their opinions individually and do we interrupt?*).

Experiments Single-case mechanism experiments in validation research study the effects of applying a stimulus to the validation model. We aim to estimate the mechanisms in the validation model that produce these effects. This differs from experiments where we aim to compare treatments; for example, an experiment for studying whether a development method is easier to use than another alternative.

Fleurey et al. [54] use testing as a validation method for model transformations. In a testing scenario like this, the variables described in the protocol could be semantic completeness and consistency, as well as syntactic well-formedness and consistency. No subjects need to be selected and instrumentation is made up of artifact components. For the data collection procedure, the authors describe a method for determining meta-model coverage using "partition analysis". Lindland et al. [82] also propose means for checking syntactic, semantic and pragmatic model quality. The data collection procedure for validating model quality can include activities such as syntax checking, model evaluation and consistency checking.

Relation to Resources

The protocols are dependent on the resources we have available. As a result of the decomposition, we already know that certain aspects of system components may not be validated within the current research scope, but the executed validation methods are also shaped by the available resources. For example, experiments can be conducted with either actual end-users or students as subjects. The actual protocol is therefore dependent on the available resources.

Levels of Detail

Since we plan also plan for validation methods that are outside of the current research scope, we need to define validation method protocols on various levels of detail. The executed validation methods require fully detailed protocols that are usable for the validation method that is to be executed. These protocols contain full descriptions of all previously mentioned elements. The non-executed validation methods require less information. In many cases, such as planned validation methods for system components that are not implemented yet, it will not be possible to create fully detailed protocols for these validation methods. In the most basic form, the protocol requires at least an *expectation* of the context, variables, required resources and instrumentation.

Family of Validations

After we have decomposed to the level of validation activities and we have defined protocols for all validation activities, we have created a *family of validations* for the system that we are validating. The family of validations consists of planned and executed validation methods.

6.5 Method Case

To motivate the usefulness of our validation method for validating MDD approaches, we will use our method to construct a family of validations for OpenBest. By applying this method to OpenBest, we also aim to provide an initial validation of the method by showing how well the method is able to define a family of validations. The following sections will elaborate on the method activities. Deliverables of the method can be found in appendix L.

6.5.1 Describing OpenBest

Describing the Context

The problem description in appendix L highlights the need for developing OpenBest. The problem description indicates that OpenBest is developed with the goal of helping organisations get a grip on *how* they can improve their ethical, social and environmental performance. To be usable for a large variety of organisations and to promote knowledge sharing across organisational boundaries, OpenBest needs to have flexible functionality. These two main goals (helping to improve performance and promoting knowledge sharing across organisational boundaries) are indicative of how the rest of this validation is shaped.

We also describe the stakeholders of OpenBest, since we wish to validate the contribution of OpenBest to their goals. We can distinguish two stakeholder groups: *users* and *domain administrators*. We have opted for a basic stakeholder description.

Describing the System

The system description is a high-level overview of the system components and their functionality. In accordance to the method, we have created a FAM which is repeated in figure 6.8. We can distinguish 5 types of functionality:

- Editor module, so that textual models can be created and updated by domain administrators.
- Interpreter module, that interprets the textual models created in the editor module, stores the results and handles requests for feature sets.
- BP entry module, which is used by users to store best practices. The feature set of the interface in this module is dependent on the interpreter module.
- BP viewing module, in which users can view best practices.
- Improvement plan module, which is used by users to construct improvement plans for their organisation. Best practices are used in improvement plans and evaluated.

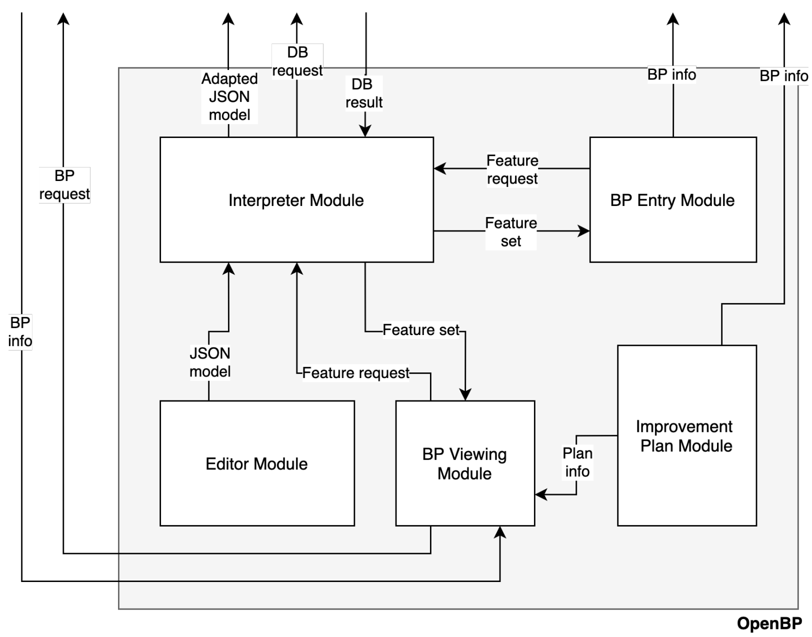

Figure 6.8: High-level FAM of OpenBest

Of these components, only the model editor is used solely by the domain administrator. All other components are used by all users. The interaction descriptions in appendix L provide more information on how the system is used. The descriptions of these components and interactions are crucial for the decomposition, as they explain which stakeholder needs need to be validated for which components.

Requirements are also drafted and are listed in appendix H. For most requirements, no tool functionality has been implemented yet. These requirements are used as input to determine the stakeholder goals in the decomposition.

6.5.2 Decomposing the Problem

This section will list the result of the decomposition phase of this method. The resulting decomposition tree of OpenBest is depicted in figure 6.9. We have excluded the canvas lanes and elements related to non-implemented system components in this version for legibility. A full version can be found in figure L.4.

Defining High-Level Goals

The first step in the decomposition is to define the high-level goals of OpenBest. In the problem description, we have identified two main goals:

- *Sharing organisational BPs across domains.* OpenBest aims to promote knowledge sharing among organisations and domains (groups of similar organisations).
- *Improving organisational performance with IP(4ESET) BPs.* OpenBest has been developed with the idea of using BPs to improve organisational performance. These BPs can be aimed at ethical, social and environmental topics, but can also be general organisational BPs.

We place these high-level goals in their corresponding lane on the canvas and connect them to the system using trace links. We already find that the first goal is located on the enabling impact level, while the second goal is located on the structural impact level. This would indicate that the second goal is much harder to validate and requires a more extensive (and prolonged) validation method. We can therefore already state that this goal will not be validated in its entirety in the current research scope and color this goal orange.

Defining Stakeholder Goals

Based on the requirements in appendix H, we can define 4 requirement themes. Of these themes, 3 are specified with *user* as the role, while the other theme is specified for the *domain administrator*. We define the following stakeholder goals based on these themes:

- *Prescribing organisational BP structure.* This requirement theme was specified for the *domain administrator*.
- *Storing a BP.* This requirement theme was specified for the *user*.
- *Searching for a BP.* This requirement theme was specified for the *user*.
- *Planning improvement actions.* This requirement theme was specified for the *user*.

We place these stakeholder goals on the canvas and link them to the high-level goals they contribute to using trace links. The trace links are annotated with the identified stakeholder group.

Defining Tool-Supported Goals

In this step, we first identify the implementation status of the various tool components based on the requirements. As has been indicated in the requirements list in appendix H, no functionality has been implemented yet for planning improvement actions. We therefore color this stakeholder goal red. Since this goal can not be validated within the current research scope, we also can not *fully* validate the connecting high-level goal, which gets colored orange as a result. The same is also true for the system element.

We then define the tool-supported goals based on the previously defined tool components. Using the tool components we have identified (both implemented and non-implemented) we can define 8 tool-supported goals that contribute to the stakeholder goals:

- *Editing the core model* with tool components: core model, textual grammar, model editor
- *Creating a textual model* with tool components: model editor, textual grammar
- *Creating a useful feature set* with tool component: model interpreter
- *Creating BPs* with tool component: entry form
- *Finding relevant BPs* with tool component: viewing interface
- *Providing contextual BP info* with tool components: comments and ratings (included in the viewing interface), improvement planning functionality
- *Constructing an improvement plan* with tool component: improvement planning functionality
- *Evaluating an intervention* with tool component: improvement planning functionality

The tool-supported goals are placed on the canvas and linked to their respective stakeholder goals. Every tool-supported goal also has one or several ports that indicate the previously listed components that contribute to the goal. We color the tool-supported goals that are descendants of *planning improvement actions* red, since these components are not implemented yet. An exception is visible for *providing contextual BP info*. This goal has two parent elements and one component that contributes to this goal that has been implemented. This goal is therefore colored orange, while the port for the improvement planning functionality is colored red.

Defining Validation Aspects

The first step in this activity is to pick a quality framework and assess it for fitness. We have elaborated on the quality framework for this research in section 6.3.1. We therefore choose this quality framework to be mapped to the decomposition tree with one exception. The core model and textual grammar are part of the concrete syntax of OBL. We have explained earlier that these aspects of the language are hidden from users and we will therefore not validate their quality using *pragmatic quality* aspects; the concrete syntax does not have to be comprehensive to users in our case. Furthermore, we will also not validate *quality in use* and *goal quality* aspects for these system components.

Syntactic, semantic and *pragmatic* aspects are related to model quality and cannot be validated for the other system components. These components will therefore be validated using the *quality in use* and *goal quality* aspects only. We will validate the following aspects for the system components:

1. Editing the *core model*
    - Syntactic correctness of the *resulting* core model: does it adhere to the textual grammar?
    - Semantic completeness of the *resulting* core model: does it contain all defined elements?
2. Using the *textual grammar* to edit the core model
    - Semantic consistency of the translation of model editor info to JSON models: does the same input always result in the same output?
3. Using the *model editor* to edit the core model
    - Ease of use, usefulness and intention to adopt the model editor
    - Flexibility of prescribing BP structure using the model editor
4. Using the *model editor* to create a textual model
    - Same validation approach as item 3, with the addition:
    - Effectiveness using the model editor to prescribe BP structure
5. Using the *textual grammar* to edit the core model
    - Same validation approach as item 2
6. Using the *model interpreter* to create a useful feature set
    - Flexibility of defining various feature types using textual models
    - Semantic completeness of the feature set: are features instantiated for all elements of the textual model?
    - Semantic consistency of the feature set: does the same input in the textual model always result in the same feature?
7. Using the *entry form* to create BPs
    - Ease of use, usefulness and intention to adopt the entry form
    - Effectiveness of using the entry form to create BPs
8. Using the *viewing interface* to find relevant BPs
    - Ease of use, usefulness and intention to adopt the entry form
    - Effectiveness of using the viewing interface to find relevant BPs
9. Using *comments and ratings* to provide contextual BP info
    - Ease of use, usefulness and intention to adopt comments and ratings functionality
    - Effectiveness of using comments and ratings functionality to provide contextual BP info
10. Using *improvement planning functionality* to provide contextual BP info

- Ease of use, usefulness and intention to adopt improvement planning functionality
- Effectiveness of using improvement planning functionality to provide contextual BP info

11. Using *improvement planning functionality* to construct an improvement plan
    - Ease of use, usefulness and intention to adopt improvement planning functionality
    - Effectiveness of using improvement planning functionality to construct an improvement plan
12. Using *improvement planning functionality* to evaluate an intervention
    - Ease of use, usefulness and intention to adopt improvement planning functionality
    - Effectiveness of using improvement planning functionality to evaluate an intervention

These validation aspects are placed on the canvas and are grouped according to the *tool-supported goal and system component* combination they belong to. Trace links are established between the groups of validation aspects and their corresponding port.

Defining Resources

We define the following resources for our validation:

- *Time:* we have three weeks for conducting the validation
- *Available subjects:* we have access to domain experts (representative end-users), researchers (experts on the IP4ESET process) and students
- *Other factors:* due to the Corona virus pandemic, we do not have access to a physical location to conduct experiments. This limits the types of methods that we can conduct.

Without regarding the validation aspects for non-implemented components, we find that these resources are sufficient for validation methods for all *tool-supported goal and system component* combinations. We conduct single-case mechanism experiments to validate *syntactic* and *semantic* quality aspects, as these can be validated in experiments without humans. For *goal quality* and *quality in use* aspects, we conduct expert opinion sessions. This method is useful as it does not require subjects to interact with the system; only to estimate effects. Since we do not have a fully mature tool and cannot control experimental settings when conducting a virtual validation, a (single-case mechanism) experiment with humans is deemed to be an ineffective method for validating *goal quality* and *quality in use*.

However, we do want to validate these quality aspects with different validation methods in future research. We therefore color these elements purple to indicate possible but partial validation. As a result, we color all connected elements on higher levels orange to indicate partial validation. The methods are placed on the canvas, as well as an element for planned validation methods and connected to the validation aspects. Arrows might be colored or annotated to distinguish them and promote legibility.

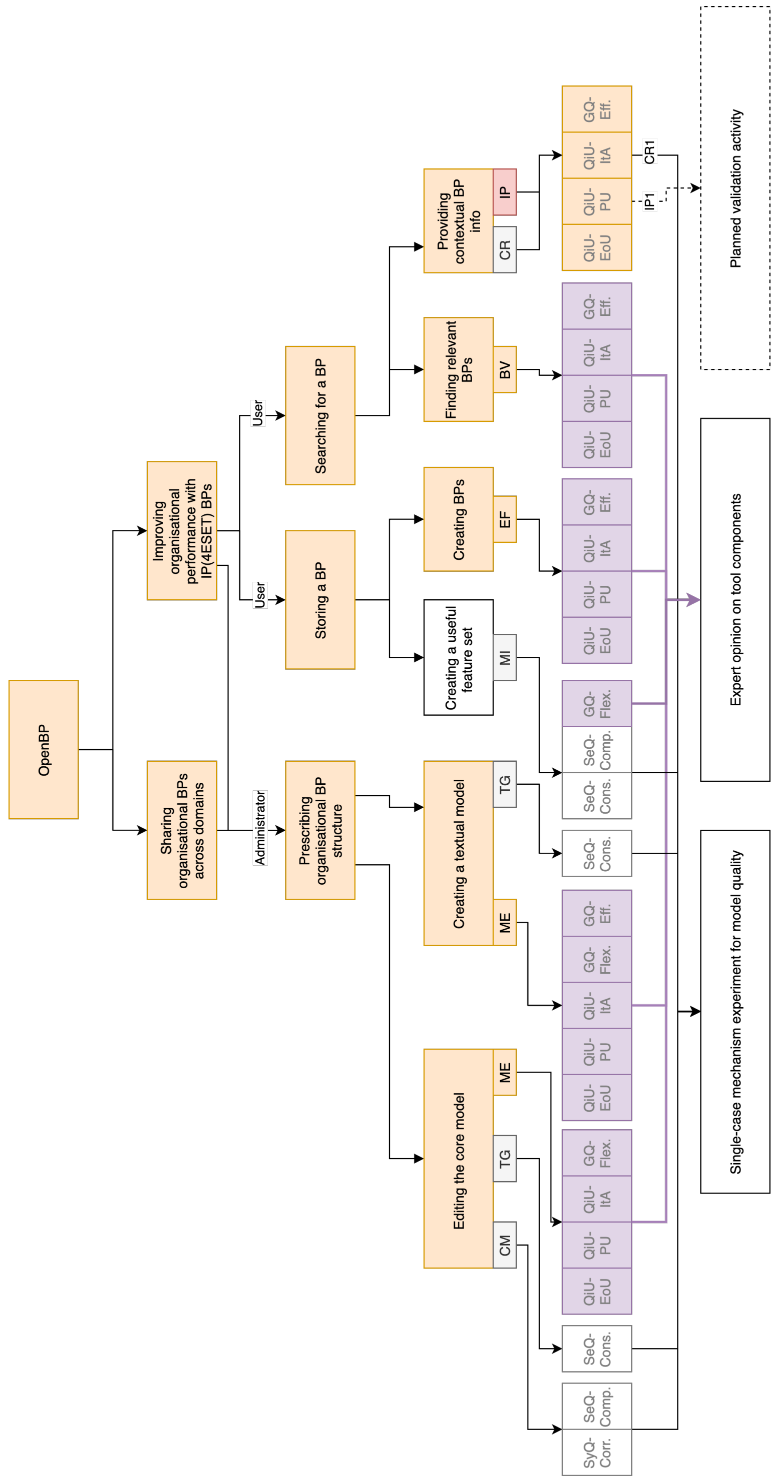

Figure 6.9: Cropped decomposition tree for OpenBest

6.5.3 Defining Protocols

The decomposition has resulted in two validation methods to be executed: a single-case mechanism experiment and expert opinion. In this section, we will elaborate on the protocols created for these methods, as well as a basic protocol created for a *planned but non-executed* validation method. The protocols are listed in appendix L and are constructed using the previously listed guidelines.

Expert Opinion

The expert opinion protocol is listed in section L.5. A context is defined stating that the method is aimed at validating goal quality and quality in use aspects of a non-mature version of the tool. Subjects of the expert opinion sessions are practitioners and researchers. Since the expert opinion session takes the form of an interview, the protocol includes interview questions.

Based on the decomposition tree, we know the validation aspects we are validating for each system component and with which intention: the combination of using a system component for a tool-supported goal is done to contribute to a stakeholder goal. This relation shapes the interview questions. The interview is structured based on the tool components so that the researcher can demonstrate one system component and then ask questions to discover the expert's estimation of the effects produced by system components by addressing the various validation aspects. The interview questions are partly based on Moody [94] for the quality in use aspects. In order to ensure that the results of the interviews are comparable, we follow the same procedure for each interview. This procedure details what will be demonstrated and in what order.

The validation aspects that are assessed with the expert opinion method are colored purple in the decomposition tree to indicate that we wish to validate them using different methods in the future. For example, a more mature version of the tool should be validated for goal quality and quality in use using experiments when the available resources allow it.

Testing

We do a single-case mechanism experiment through testing. With this method, we aim to validate the *syntactic* and *semantic quality* of the models that drive OpenBests functionality. The procedure in section L.6 elaborates on the steps and deliverables for creating and validating test data. The measures for these validation aspects are largely based on Lindland et al. [82].

Case Study

A basic protocol is created for a case study. It is presently estimated that a case study will be a useful method for validating the non-implemented system components and further validate implemented components than what is possible using current resources. A case study could be used as a method for the creation of new test data, as well as a method for validating expert opinion.

When OpenBest is mature enough and resources allow a more extensive validation, we should validate OpenBest with an extended case study. By letting organisations work with OpenBest over an extended period of time to construct and evaluate improvement plans and share their BPs, we are able to more clearly define its impact on the high-level goals.

6.5.4 Conclusions and Discussion

With this method we have created a family of validations for validating OpenBest. The family of validations consists of an expert opinion method and a single-case mechanism experiment, as well as a planned case study. We are able to motivate the usefulness of these validation methods by indicating how their validation aspects allow us to look at the use of system components to contribute to increasingly high-level goals. With the resulting decomposition tree, we are able to express the amount of coverage of the validation we have so far conducted. We find that we cannot make any claims about the contribution of OpenBest to high-level goals at the moment. We have only been able to validate the quality of the core model, model interpreter and textual grammar at this point.

Although we have been able to successfully apply the method for defining a family of validations to OpenBest, the method requires further validation. The method has resulted in a set of validation activities for OpenBest and we expect this set of activities to be more suited for estimating OpenBest's contribution to high-level goals than a set of validation activities that have been defined without this method, but this assumption needs to be tested. This assumption may be tested by validation using opinions of other researchers applying this method. By applying our method to other MDD systems further allows us to validate how well our method is able to define families of validations for various MDD systems.

6.6 Validation Results

This section will elaborate on the results of the executed validation activities that we defined as part of the family of validations.

6.6.1 Expert Opinion

Validation Set-up

We have interviewed 2 experts working for various organisations that are working on IP4ESET in some way. We have also interviewed 3 researchers with knowledge of the IP4ESET domain. The interviews were conducted virtually via Microsoft Teams and all lasted approximately 30 minutes.

As specified in the protocol in section L.5, the interview consisted of a demonstration of 4 components of OpenBest and questions about these components concerning *goal quality* and *quality in use*. This section will address the findings of these interviews for the individual components. We annotate expressions by one of the subjects with an identifier (R1, R2 and R3 for researchers and E1 and E2 for experts).

Model Editor

Table 6.3 shows the findings of the expert opinion interviews concerning the model editor.

Table 6.3: Interview findings for the model editor

Perceived ease of use	
Component is perceived as easy to use	All subjects express they find the component easy to use and intuitive.
There are some misconceptions about	There is some doubt about what the mandatory and recommended concepts mean and what their internal contents are (the attributes and such) (R1, R2, E2). It is also unclear how the recommended elements are hierarchically related to the mandatory elements (R1).
Points for improvement	Some would require further explanation on what the recommended concepts mean (R1, R2, E2) or further explanation on how to construct relationships (R3).
Perceived usefulness	
Component is perceived as useful	The form is a useful approach to prescribing BP structure and that in itself is a useful thing to do (R1, E1, R2, R3, E2). It is useful to address BP structure variability (R1, E1, R2, R3) and useful to have mandatory and recommended elements (R1, E1, R3). The flexibility has the added benefit to give users a feeling of control and to allow them to specify their own meaning of such a diffuse term (E2).
Points for improvement	It would be useful if the domain administrator would be able to add attributes and relationships to mandatory and recommended concepts (R3).
Intention to adopt	
Experts have the intention to adopt	Experts want to be able to define which pieces of information are shared with others and which are not (E1, E2). One expert (E2) shared concern about the motivation of using such a tool. Based on experience, it was found that such motivation often slacks after a while.
Effectiveness	
Component is an effective way to improve organisational performance	R2 expresses that this is a very difficult goal, but the tool may help to some extent. E2 expresses that it would be important that the quality of the BPs is really high, so that you do not have to work through unnecessary BPs.

Model Interpreter and Entry Form

Table 6.4 shows the findings of the expert opinion interviews concerning the model interpreter and BP entry form.

Table 6.4: Interview findings for the model editor

Flexibility - *Model Interpreter*	
The instantiated features provide enough flexibility	All subjects expected that the current feature variability is sufficient. R2 indicates that too much complexity would be a bad thing.
Perceived ease of use - *Entry Form*	
Component is perceived as easy to use	All participants thought the entry form seemed easy to use, although R2 indicated that too much text is not very visually appealing. R3 was unsure about what some features do and requires further explanation.
Perceived usefulness - *Entry Form*	
Component is perceived as useful	R3 indicates that in order for it to be useful, people need to see the added benefit of taking the effort to enter BPs. All subjects thought the component seemed useful. An approach like this is more useful than storing PDF files somewhere, since this allows you to more easily search the database (E2).
Points for improvement	Having control over the amount of information you share with other organisations. Being able to define contextual factors for solutions.
Intention to adopt - *Entry Form*	
Experts have the intention to adopt	If you would be able to have control over the amount of information to share. E2 indicated that it is not a question about functionality of the tool, but motivation of a group to use it.
Flexibility - *Entry Form*	
Component is able to address the required flexibility	Using this form to prescribe BP structure provides enough flexibility (R1, R2), but a useful addition might be to add an attribute that allows for uploading files (E2).
Effectiveness - *Entry Form*	
Component is seen as effective	The component is perceived as effective for prescribing BP structures as it allows the addition of all required information for BP structures.

Search and View

Table 6.5 shows the findings of the expert opinion interviews concerning OpenBest's functionality for finding and viewing BPs.

Table 6.5: Interview findings for the model editor

Perceived ease of use	
Component is perceived as easy to use	All subjects thought the component seemed easy to use.
Perceived usefulness	
Component is perceived as useful	Seems useful at the moment (E1). Users have freedom in choosing categories and BP titles, but this can get chaotic (R1, R3).
Points for improvement	Having other ways of searching, for when another user has used a synonym for what you are looking for (R1). Being able to see what is happening in other domains (R1). Having a predetermined naming format for BPs (R3).
Intention to adopt	
Experts have the intention to adopt	E1 indicated that they have searched for a tool like this to be used by the industry, and this seems like a nice solution. They are looking for tools that are able to find underlying connections between problems and solutions. E2 also indicates they would use it. R3 addresses that it would need to gain momentum first in order to be useful.
Effectiveness	
Component is seen as effective	All subjects thought the tool would be effective to some extent to improve organisational performance.

Conclusions

In general, the interview subjects were positive about the *quality in use* and *goal quality* aspects of OpenBest. Concerning the ease of use of the tool, subjects generally think that the use of the tool is clear and easy. Some minor errors could be fixed by adding explanations. The current tool functionality is found to be useful and experts all express the intention to adopt the tool.

Subjects express no major drawbacks of OpenBest and the contribution to its goals. It is expressed that the tool seems to be a useful approach to improving organisational performance, but it is crucial that users have the motivation to keep on using the tool. To this end, the quality of BPs that are entered needs to be high and users need to be able to see the added value of *sharing* BPs (rather than only using the tool to *find* BPs).

In future research, it would be useful to repeat the validation of *quality in use* and *goal quality* aspects with experts after letting expert interact with the system themselves. This could be done through a single-case mechanism experiment or a case study.

6.6.2 Testing

Creating Textual Models

We have created 5 models using OpenBests model editor. For each model, we have made sure to include a different set of concepts. We have also ensured that all options of the model editor have been used at least once in any of the models so that we can test all functionality. The resulting models are available in a Google Drive folder[1].

We have checked all models for syntactical errors using the textual grammar, but no syntactical errors have been found. The validity of the models as being JSON models has been checked with an online JSON checker[2]. An overview of the information entered in the model editor has been created for all models. We have found no missing elements that have been entered in the model editor but were not present in the textual models. We therefore state that the *edited core model*, when validated using the *textual grammar*, is *syntactically correct* and *semantically complete*.

We now also have to validate the semantic consistency of the translation of form input to textual models. Following the validation protocol, we look at the consistency of translating the following types of elements:

1. Names and values of attributes
    - All attribute names are consistently modeled in their input order
    - All attribute names include a number
    - All attribute values are exactly translated
    - Attributes with the “multiple” checkbox checked are consistently included as arrays
2. Structure of relationships and inclusion of relationship names
    - Relationships consistently follow the structure *name, self, related*
    - Relationship names and concept names are consistently translated
3. Placement of concepts in the hierarchy
    - Concepts and sub-concepts are consistently placed in their correct order in the hierarchy
4. Ordering of attributes
    - Attributes are consistently ordered according to their placement in the model editor

We find no inconsistencies in the translation of model editor information to textual models.

[1] `https://bit.ly/OpenBestvalidationmodels`
[2] `https://jsonlint.com/`

Quality of Models

We find no syntactic or semantic errors, as well as no semantic inconsistencies. We therefore make the following conclusions:

- Edited core models created with the model editor are *syntactically correct*
- The model editor and its transformation rules create *semantically consistent* models

We also state that the *core model* and *textual grammar* contribute to *editing the core model* and *creating a textual model*.

Creating Feature Sets

For all created models, we determine which features we expect to be instantiated for all concepts. An excerpt of this overview for model 1 is visible in table 6.6; the full overviews are available in the previously mentioned Google Drive folder. The overview lists the attributes that are added for three concepts and the information that has been entered in the model editor. Based on this information we can establish which features are required to be instantiated and with which information. For example, we expect a feature to be instantiated for *New Concept* called *Type* and we expect this feature to be a small text box (due to the *String* value) with add functionality to add multiple values (due to the *Array* value). We also see a relationship called *solves* that points from *New Concept* to *Best Practice*. We do not expect a feature for this value, since relationships are automatically stored without asking user input.

Table 6.6: Overview of instantiated features

Element	Form info	Feature required	Instantiated
Best Practice			
Group title	This is the group title for best practices	Group title	Group title
Group description	This is the group description for best practices	Group description	Group description
Problems			
Group title	This is the group title for problems	Group title	Group title
Group description	This is the group description for problems	Group description	Group description
New Concept			
Group title	This is the group title for New Concept	Group title	Group title

Table 6.6 – continued from previous page

Group description	This is the group description for New Concept	Group description	Group description
Name	String	Small text box with label	Small text box with label
Description	Text, Array	Large text box with label and add functionality	Large text box with label and add functionality
Type	String, Array	Small text box with label and add functionality	Small text box with label and add functionality
Solves	*bestpractices*	None	None

This overview has been created for all models. We find that all required features are instantiated for all models after uploading the models to OpenBest (*semantic completeness)*. Since we base the required features on the information entered in the model editor and there are no inconsistencies between required features and instantiated features, we can also conclude that the model interpreter makes semantically consistent translations between textual models and feature models.

Quality of the Model Interpreter

We find no inconsistencies between textual models and instantiated features. We also find no inconsistencies between required features and instantiated features. We therefore conclude that the *model interpreter* makes *semantically complete* and *semantically consistent* translations from textual models to features and therefore contributes to *creating a useful feature set*.

7 | Discussion

7.1 Implications

In this research we have studied the contribution of a BPR to the IP4ESET domain. We defined the following problem statement: *improve organisations' transition to becoming responsible organisations by designing a repository for best practices so that responsible organisations can learn from other organisations.* The validation of the BPR we have developed shows that it is possible to create a DSL that specifies the description of BP structures and to create a model-driven BPR that adapts its functionality to user needs. Interview subjects have expressed positive views on the expected contribution of the BPR to the goals of knowledge sharing across organisational boundaries and improving organisational performance using IP(4ESET) BPs. We are therefore confident that a more mature version of the BPR may eventually be transferred to a real-world context and produce satisfying effects.

This research build on the existing body of research related to IP4ESET by proposing the use of a software tool to structure (part of) the IP4ESET process. This helps to decrease the vagueness inherent in other existing approaches, decreasing the intention-action gap observed in the shortcoming analysis. Furthermore, we contribute to the growing body of knowledge around the BECIC. We have proposed a tool for the IP4ESET phase, that may be used by organisations together with the ecosystem of responsible software tools for other phases created at Utrecht University for this research line.

By using the MDD paradigm to develop our BPR, we have contributed to the field of MDD and DSL creation. We have been able to show that an MDD approach may be used to provide flexibility in system functionality and database structures during runtime and that the responsibility for this flexibility may be given to users. By specifying a method for the creation of a family of validations for MDD approaches, we have also contributed to the body of knowledge on large-scale validation of MDD approaches with long-term and high-level goals. This approach may also inspire future work on the creation of families of validations for other SE approaches.

7.2 Limitations

This research has several limitations that we need to address. First of all, some limitations are present as a result of the development. The components currently implemented in OpenBest are not fully mature and can therefore not be fully validated. An example is the model editor that does not allow for full flexibility in adapting the core model. We also do not have any functionality implemented for the creation of improvement plans, which is a crucial element for describing the context of BP application. As long as OpenBest is not able to recommend BPs to organisations based on contextual factors, it is less likely to help organisations to find relevant BPs and thereby improve their organisational performance. Lastly, OpenBest currently operates in isolation and is not connected to any of the other tools in the BECIC software ecosystem. This connection would help organisations to improve their performance by supporting all phases of the BECIC.

Another limitation of the current functionality is the limitation of only being able to share BPs with organisations within the same domain. Our initial goal for OpenBest was to also allow knowledge sharing across domains, so that organisations may learn from BPs applied by non-related organisations. This is not possible with the current functionality.

The validation of this research suffers from some limitations as well. First of all, we have only had a limited number of subjects available for the validation interviews, decreasing the generalizability of our validation results. Due to other resource conflicts and maturity levels of the system components, we have not been able to validate the interaction between users and system in a real-world context. We have only been able to make estimations about the effects that would be produced by the use of OpenBest in a real-world context, but the reality might be different. Lastly, concerning validation, we have not been able to study if the system actually contributes to its goals. Again, we have only been able to make estimations. Studying the actual contribution to high-level goals would require a much more mature version of OpenBest that would be applied in a real-world context for a prolonged period of time.

7.3 Future Work

Future work on OpenBest should aim to work on the previously mentioned limitations. First of all, the functionality of OpenBest should be extended by including functionality for creating improvement plans. These plans can be constructed as the ToCs elaborated on in this research, but may also be linked to the work by Adèr [14]. The information present in improvement plans that include BPs should then be used to define contextual information for BPs. This would allow for active recommendation of relevant BPs to organisations. Other options for the inclusion of contextual information may also be researched, such as the inclusion and specification of organisational models (e.g. organisational charts or process models) in OpenBest.

Current functionality should be further developed to improve its usefulness. The model editor requires the ability to further adapt elements of the core model and easier specification of relationships. Furthermore, OpenBest currently relies on completely textual BPs. In future work, functionality should be implemented that allows users to specify BPs using images, models and external files. The functionality for finding BPs should be extended with more filtering options. Lastly, OpenBest should be connected to other tools in the BECIC software ecosystem.

In order to promote knowledge sharing across domains, we should extend OpenBests functionality to allow users to search through repository instances of other domains. This functionality would need to take privacy-related issues into account; some organisations may not want to share knowledge with organisations outside of their own domain.

The validation of OpenBest in future research should focus on the interaction between users and system in real-world context. Ideally, this validation should be done over a longer period of time in order to more accurately say whether OpenBest contributes to its high-level goals. An approach to this validation method could be to apply OpenBest in the context of networks of responsible organisations that together will be able to use the repository and to share BPs.

8 | Conclusion

In this research we have aimed to improve organisational performance and knowledge sharing on ESE topics by creating a model-driven BPR that stores BPs on these topics. Although our BPR is developed with a focus on ESE BPs, it may be used by any type of organisational BP. The model-driven approach to functionality of the BPR allows for more flexibility and therefore a higher chance of successfully allowing organisations to use the BPR and to share and find BPs.

For RQ1, we asked: *What are the shortcomings of current tools for improvement planning for ethical, social and environmental topics?* We have conducted a shortcoming analysis consisting of three parts: a scientific literature review, a feature analysis of existing IP4ESET tools and a grey literature review. All elements of the shortcoming analysis highlight the *intention-action gap* in IP4ESET: organisations are motivated to improve their ESE performance, but do not know how to. To further investigate shortcomings of IP4ESET approaches, we have interviewed various experts that work on IP4ESET in their organisations. All experts again highlight the existence of an intention-action gap, but are generally positive on the usefulness of a BPR to be used to guide their improvement actions.

To research RQ2 (*What is the state-of-the-art and practice on (domain-independent) best practice repositories?*), we have collaborated with another researcher by studying the feature models of BPRs and the data models of the BPs that they store. By comparing these models, we have found a lack of common ground on how to structure BPRs and BPs. It has also been highlighted that BPRs lack the flexibility required to be able to improve knowledge sharing across organisational boundaries. The BPRs were also found to be non-extensible, indicating that they rely on BPs from single sources, rather than allowing for collaboration.

RQ3, *How can we create a model-driven best practice repository that supports IP4ESET knowledge sharing across organisational contexts?*, consists of several sub-questions. First of all, we have answered RQ3.1 (*What are the requirements for a repository for IP4ESET best practices?*). Based on interviews and a variety of other sources, a list of requirements has been created. The requirements in this list were based around themes related to planning (for which no functionality has been implemented in this research scope), evaluation, best practices and context. The requirements have been prioritized using surveys with various stakeholders. For RQ3.2 (*How can we determine the applicability of IP4ESET best practices across organisational contexts?*), we have described how the method of Theory of Change and the construction of ToC models can be used to include contextual information in BPs. By placing BPs in the larger context of improvement plans created by organisations, we would be able to actively recommend BPs to users. We have not been able to implement this type of functionality within this research scope. The third sub-question RQ3.3 (*How can we create a domain specific language for best practices that can be interpreted in runtime?*) has been answered through the creation of a DSL for constructing BP models, called OBL. We have elaborated on the approach of creating OBL, including the domain analysis and the creation of various models that make up the language. By also addressing how other components of the tool have been implemented, we have formed a coherent answer to RQ3.

In order to answer RQ4 (*How can we validate the contribution of model-driven systems to their high-level goals?*), we have created a method for defining a family of validations for MDD systems. This method deals with the complexity of validating MDD approaches that require validation of more aspect than a single study allows. This validation method encompasses activities such as the definition of the validation problem and the decomposition of the problem into manageable validation activities. By applying this method to OpenBP, we have been able to answer RQ5 (*How well does a model-driven best practice repository contribute to its high-level goals?*). By executing the validation activities that resulted from the decomposition of the validation problem, we have found that the models used and created by OpenBP are *syntactically* and *semantically correct*. We have also found that models are *semantically consistently* transformed. We have also studied *quality in use* and *goal quality* aspects of OpenBP with expert opinion interviews. We have found that experts and researchers are positive about the possibility of OpenBP contributing to its goals (sharing organisational BPs across organisational boundaries and improving organisational performance with IP(4ESET) BPs). We have not been able to validate quality aspects related to planning improvement actions since no functionality has been created for those goals. We have planned for future validation of these aspects.

Bibliography

[1] Due Diligence Toolkit for Responsible Business Conduct. https://www.specerijenvereniging.nl/imvo/toolkit/. Accessed: 2020-02-15.

[2] Using technology to prevent child labour. https://info.chainpoint.com/requestchildlaboursolutionpaper. Accessed: 2020-02-17.

[3] IDH Gender Toolkit and Guide. https://www.idhsustainabletrade.com/publication/idh-gender-toolkit-guide/, 2014. Accessed: 2020-02-16.

[4] NPR 9036:2015 en. https://www.nen.nl/NEN-Shop/Norm/NPR-90362015-en.htm, 2015. Accessed: 2020-02-17.

[5] SDG Compass Guide. https://sdgcompass.org/, 2015. Accessed: 2020-02-17.

[6] Blueprint for Business Leadership on the SDGs. https://www.unglobalcompact.org/library/5461, 2017. Accessed: 2020-02-17.

[7] IMVO-convenanten. https://www.rijksoverheid.nl/onderwerpen/internationaal-maatschappelijk-verantwoord-ondernemen-imvo/imvo-convenanten, 2018. Accessed: 2020-02-17.

[8] OECD Due Diligence Guidance for Responsible Business Conduct. https://www.oecd.org/investment/due-diligence-guidance-for-responsible-business-conduct.htm, 2018. Accessed: 2020-02-15.

[9] Circularity Gap Report Austria. https://www.circle-economy.com/insights/circularity-gap-report-austria, 2019. Accessed: 2020-02-17.

[10] Ending child labour, forced labour and human trafficking in global supply chains. https://www.unglobalcompact.org/library/5730, 2019. Accessed: 2020-02-17.

[11] MVO Prestatieladder. http://www.mvoprestatieladder.nl/download-norm/, 2020. Accessed: 2020-02-16.

[12] Tracking Value. https://www.circle-economy.com/insights/tracking-value, 2020. Accessed: 2020-02-17.

[13] Adrion, W. R., Branstad, M. A., and Cherniavsky, J. C. Validation, verification, and testing of computer software. *ACM Computing Surveys (CSUR) 14*, 2 (1982), 159–192.

[14] Adèr, M. The state of the art and practice in sustainable improvement planning for ethics, social and environmental topics. Master's thesis, Utrecht University, 2020. Not published.

[15] Ahmed, M. D., and Sundaram, D. Sustainability modelling and reporting: From roadmap to implementation. *Decision Support Systems 53*, 3 (2012), 611–624.

[16] Ameller, D. Considering Non-Functional Requirements in Model-Driven Engineering. Master's thesis, Llenguatges i Sistemes Informàtics (LSI), June 2009.

[17] Angus-Leppan, T., Metcalf, L., and Benn, S. Leadership styles and CSR practice: An examination of sensemaking, institutional drivers and CSR leadership. *Journal of Business Ethics 93*, 2 (2010), 189–213.

[18] Arimura, T. H., Darnall, N., Ganguli, R., and Katayama, H. The effect of ISO 14001 on environmental performance: Resolving equivocal findings. *Journal of Environmental Management, Forthcoming* (2015).

[19] Bansal, P., and Bogner, W. C. Deciding on iso 14001: economics, institutions, and context. *Long Range Planning 35*, 3 (2002), 269–290.

[20] Basili, V., Shull, F., and Lanubile, F. Building knowledge through families of software studies: An experience report1. *IEEE Transactions on Software Engineering, 5 ()* (1999), 56–73.

[21] Baumgartner, R. J. Managing corporate sustainability and csr: A conceptual framework combining values, strategies and instruments contributing to sustainable development. *Corporate Social Responsibility and Environmental Management 21*, 5 (2014), 258–271.

[22] Baumgartner, R. J., and Korhonen, J. Strategic thinking for sustainable development. *Sustainable Development 18*, 2 (2010), 71–75.

[23] Baumgartner, R. J., and Rauter, R. Strategic perspectives of corporate sustainability management to develop a sustainable organization. *Journal of Cleaner Production 140* (2017), 81–92.

[24] Berzosa, A., Bernaldo, M., and Fernández-Sanchez, G. Sustainability assessment tools for higher education: An empirical comparative analysis. *Journal of Cleaner Production 161* (2017), 812–820.

[25] Bevan, N. Extending quality in use to provide a framework for usability measurement. In *International Conference on Human Centered Design* (2009), Springer, pp. 13–22.

[26] Bieker, T., et al. Sustainability management with the balanced scorecard. *International summer academy on technology studies-corporate sustainability* (2003), 1–17.

[27] Boiral, O. Corporate greening through iso 14001: a rational myth? *Organization Science 18*, 1 (2007), 127–146.

[28] Boudreau, M.-C., Chen, A., and Huber, M. Green IS: Building sustainable business practices. *Information systems: A global text* (2008), 1–17.

[29] Brinkkemper, S., and Pachidi, S. Functional architecture modeling for the software product industry. In *European Conference on Software Architecture* (2010), Springer, pp. 198–213.

[30] Broman, G. I., and Robèrt, K.-H. A framework for strategic sustainable development. *Journal of Cleaner Production 140* (2017), 17–31.

[31] Brucker, A. D., and Doser, J. Metamodel-based uml notations for domain-specific languages. In *4th International Workshop on Software Language Engineering (ATEM 2007)* (2007).

[32] Bull, M. "balance": the development of a social enterprise business performance analysis tool. *Social Enterprise Journal 3*, 1 (2007), 49–66.

[33] Castro, J. W., Acuña, S. T., and Juristo, N. Integrating the personas technique into the requirements analysis activity. In *2008 Mexican International Conference on Computer Science* (2008), IEEE, pp. 104–112.

[34] Chen, D., Thiede, S., Schudeleit, T., and Herrmann, C. A holistic and rapid sustainability assessment tool for manufacturing smes. *CIRP annals 63*, 1 (2014), 437–440.

[35] Chofreh, A. G., Goni, F. A., and Klemeš, J. J. Sustainable enterprise resource planning systems implementation: A framework development. *Journal of cleaner production 198* (2018), 1345–1354.

[36] Chofreh, A. G., Goni, F. A., Shaharoun, A. M., Ismail, S., and Klemeš, J. J. Sustainable enterprise resource planning: imperatives and research directions. *Journal of Cleaner Production 71* (2014), 139–147.

[37] Choi, I., Kim, K., and Jang, M. An XML-based process repository and process query language for integrated process management. *Knowledge and Process Management 14*, 4 (2007), 303–316.

[38] Classen, A., Heymans, P., and Schobbens, P.-Y. What's in a feature: A requirements engineering perspective. In *International Conference on Fundamental Approaches to Software Engineering* (2008), Springer, pp. 16–30.

[39] Coenen, Z. An informational analysis of best practice repositories for responsible enterprises. Bachelor's thesis, Utrecht University, 2020. Not published.

[40] Connell, J. P., and Kubisch, A. C. Applying a theory of change approach to the evaluation of comprehensive community initiatives: progress, prospects, and problems. *New approaches to evaluating community initiatives 2*, 15-44 (1998), 1–16.

[41] Côté, R., Booth, A., and Louis, B. Eco-efficiency and smes in nova scotia, canada. *Journal of cleaner Production 14*, 6-7 (2006), 542–550.

[42] Dalpiaz, F., Franch, X., and Horkoff, J. istar 2.0 language guide. *arXiv preprint arXiv:1605.07767* (2016).

[43] Dhanesh, G. S. Why corporate social responsibility? An analysis of drivers of CSR in India. *Management Communication Quarterly 29*, 1 (2015), 114–129.

[44] Elser, C., and Richmond, F. J. Validation master plans: Progress of implementation in the pharmaceutical industry. *Therapeutic innovation & regulatory science 53*, 3 (2019), 354–363.

[45] Engert, S., and Baumgartner, R. J. Corporate sustainability strategy–bridging the gap between formulation and implementation. *Journal of cleaner production 113* (2016), 822–834.

[46] Engert, S., Rauter, R., and Baumgartner, R. J. Exploring the integration of corporate sustainability into strategic management: a literature review. *Journal of cleaner production 112* (2016), 2833–2850.

[47] Epstein, M. J., and Wisner, P. S. Using a balanced scorecard to implement sustainability. *Environmental quality management 11*, 2 (2001), 1–10.

[48] España, S., and Brinkkemper, S. Responsible software: A research agenda to help enterprises become more sustainable. In *ICT for Sustainability 2016* (2016), Atlantis Press.

[49] España, S., Bik, N., and Overbeek, S. Model-driven engineering support for social and environmental accounting. In *2019 13th International Conference on Research Challenges in Information Science (RCIS)* (2019), IEEE, pp. 1–12.

[50] European Commission. A renewed EU strategy 2011-14 for Corporate Social Responsibility. *Brussels, European Commission* (2011).

[51] Fairphone. Impact Report Vol. 1 - Fairphone, 2018.

[52] Falle, S., Rauter, R., Engert, S., and Baumgartner, R. Sustainability management with the sustainability balanced scorecard in smes: Findings from an austrian case study. *Sustainability 8*, 6 (2016), 545.

[53] Figge, F., Hahn, T., Schaltegger, S., and Wagner, M. The sustainability balanced scorecard–linking sustainability management to business strategy. *Business strategy and the Environment 11*, 5 (2002), 269–284.

[54] Fleurey, F., Steel, J., and Baudry, B. Validation in model-driven engineering: testing model transformations. In *Proceedings. 2004 First International Workshop on Model, Design and Validation, 2004.* (2004), IEEE, pp. 29–40.

[55] Fondement, F., and Silaghi, R. Defining model driven engineering processes. In *Third International Workshop in Software Model Engineering (WiSME), held at the 7th International Conference on the Unified Modeling Language (UML)* (2004).

[56] França, C. L., Broman, G., Robèrt, K.-H., Basile, G., and Trygg, L. An approach to business model innovation and design for strategic sustainable development. *Journal of Cleaner Production 140* (2017), 155–166.

[57] Friedman, M. The social responsibility of business is to increase its profits. In *Corporate ethics and corporate governance*. Springer, 2007, pp. 173–178.

[58] Garousi, V., Felderer, M., and Mäntylä, M. V. The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In *Proceedings of the 20th international conference on evaluation and assessment in software engineering* (2016), ACM, p. 26.

[59] Geuijen, K. SUPER: Scaling Urban Project to Enhance Results, 2020. European Commission Proposal, Topic: H2020-SC6-MIGRATION-2018-2019-2020, Proposal Number: SEP-210640765.

[60] Giraldo, F. D., España, S., Pastor, O., and Giraldo, W. J. Considerations about quality in model-driven engineering. *Software Quality Journal 26*, 2 (2018), 685–750.

[61] Graafland, J., and Van de Ven, B. Strategic and moral motivation for corporate social responsibility. *Journal of Corporate Citizenship*, 22 (2006), 111–123.

[62] Gray, R. Is accounting for sustainability actually accounting for sustainability… and how would we know? an exploration of narratives of organisations and the planet. *Accounting, organizations and society 35*, 1 (2010), 47–62.

[63] Hahn, R. Iso 26000 and the standardization of strategic management processes for sustainability and corporate social responsibility. *Business Strategy and the Environment 22*, 7 (2013), 442–455.

[64] Hallstedt, S. I. Sustainability criteria and sustainability compliance index for decision support in product development. *Journal of Cleaner production 140* (2017), 251–266.

[65] Hallsteinsen, S., Hinchey, M., Park, S., and Schmid, K. Dynamic software product lines. *Computer 41*, 4 (2008), 93–95.

[66] Halme, M., and Korpela, M. Responsible innovation toward sustainable development in small and medium-sized enterprises: a resource perspective. *Business Strategy and the Environment 23*, 8 (2014), 547–566.

[67] Harmon, R., and Demirkan, H. The next wave of sustainable IT. *IT professional 13*, 1 (2011), 19–25.

[68] Hemphill, T. The iso 26000 guidance on social responsibility international standard: what are the business governance implications? *Corporate Governance: The international journal of business in society 13*, 3 (2013), 305–317.

[69] Hilty, L. M., and Aebischer, B. *ICT innovations for sustainability*, vol. 310. Springer, 2015.

[70] Hutchinson, J., Rouncefield, M., and Whittle, J. Model-driven engineering practices in industry. In *Proceedings of the 33rd International Conference on Software Engineering* (2011), pp. 633–642.

[71] James, P., and Donaldson, S. Action for sustainability: Northwest england's tool for regional strategic sustainability appraisal. *Journal of Environmental Assessment Policy and Management 3*, 03 (2001), 413–430.

[72] Jiang, R. J., and Bansal, P. Seeing the need for ISO 14001. *Journal of Management Studies 40*, 4 (2003), 1047–1067.

[73] Johnson, M. P., and Schaltegger, S. Two decades of sustainability management tools for smes: how far have we come? *Journal of Small Business Management 54*, 2 (2016), 481–505.

[74] Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., and Peterson, A. S. Feature-oriented domain analysis (foda) feasibility study. Tech. rep., Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.

[75] Kitchenham, B. Procedures for performing systematic reviews. *Keele, UK, Keele University 33*, 2004 (2004), 1–26.

[76] KLEPPE, A., WARMER, J., and BAST, W. The model driven architecture™: Practice and promise.

[77] Kontio, J., Lehtola, L., and Bragge, J. Using the focus group method in software engineering: obtaining practitioner and user experiences. In *Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE'04.* (2004), IEEE, pp. 271–280.

[78] Kosar, T., Mernik, M., and Carver, J. C. Program comprehension of domain-specific and general-purpose languages: comparison using a family of experiments. *Empirical software engineering 17*, 3 (2012), 276–304.

[79] Krogstie, J. *Model-based development and evolution of information systems: A Quality Approach.* Springer Science & Business Media, 2012.

[80] Kuo, F.-Y., and Young, M.-L. A study of the intention–action gap in knowledge sharing practices. *Journal of the American Society for Information Science and Technology 59*, 8 (2008), 1224–1237.

[81] Le, D. M., Dang, D.-H., and Nguyen, V.-H. Domain-driven design using meta-attributes: a dsl-based approach. In *2016 Eighth International Conference on Knowledge and Systems Engineering (KSE)* (2016), IEEE, pp. 67–72.

[82] Lindland, O. I., Sindre, G., and Solvberg, A. Understanding quality in conceptual modeling. *IEEE software 11*, 2 (1994), 42–49.

[83] Loeser, F., Erek, K., Schmidt, N.-H., Zarnekow, R., and Kolbe, L. M. Aligning green it with environmental strategies: Development of a conceptual framework that leverages sustainability and firm competitiveness. In *AMCIS* (2011).

[84] Lucassen, G., Dalpiaz, F., van der Werf, J. M. E., and Brinkkemper, S. Improving agile requirements: the quality user story framework and tool. *Requirements Engineering 21*, 3 (2016), 383–403.

[85] Luke, D. A., Calhoun, A., Robichaux, C. B., Elliott, M. B., and Moreland-Russell, S. Peer reviewed: the program sustainability assessment tool: a new instrument for public health programs. *Preventing Chronic Disease 11* (2014).

[86] Maas, S., and Reniers, G. Development of a csr model for practice: connecting five inherent areas of sustainable business. *Journal of Cleaner Production 64* (2014), 104–114.

[87] Martínez, Y., Cachero, C., and Meliá, S. Mdd vs. traditional software development: A practitioner's subjective perspective. *Information and Software Technology 55*, 2 (2013), 189–200.

[88] Matten, D., and Moon, J. "Implicit" and "explicit" CSR: A conceptual framework for a comparative understanding of corporate social responsibility. *Academy of management Review 33*, 2 (2008), 404–424.

[89] Mayne, J. Theory of change analysis: Building robust theories of change. *Canadian Journal of Program Evaluation 32*, 2 (2017).

[90] Mernik, M., Heering, J., and Sloane, A. M. When and how to develop domain-specific languages. *ACM computing surveys (CSUR) 37*, 4 (2005), 316–344.

[91] Missimer, M. *Social sustainability within the framework for strategic sustainable development*. PhD thesis, Blekinge Tekniska Högskola, 2015.

[92] Missimer, M., Robèrt, K.-H., and Broman, G. A systems perspective on iso 26000. In *2nd International Symposium "systems thinking for a sustainable economy. Advancements in Economic and Managerial Theory and Practice* (2014), Universitas Mercatorum.

[93] Mohagheghi, P., and Dehlen, V. A metamodel for specifying quality models in model-driven engineering. In *Proceedings of the Nordic Workshop on Model Driven Engineering* (2008).

[94] Moody, D. L. The method evaluation model: a theoretical model for validating information systems design methods. In *European Conference on Information Systems 2003* (2003), Department of Information Systems, London School of Economics, pp. 1–17.

[95] Moody, D. L., Sindre, G., Brasethvik, T., and Solvberg, A. Evaluating the quality of information models: empirical testing of a conceptual model quality framework. In *25th International Conference on Software Engineering, 2003. Proceedings.* (2003), IEEE, pp. 295–305.

[96] Moratis, L. Out of the ordinary? appraising iso 26000's csr definition. *International Journal of Law and Management 58*, 1 (2016), 26–47.

[97] Müller, V. C., and Bostrom, N. Future progress in artificial intelligence: A survey of expert opinion. In *Fundamental issues of artificial intelligence*. Springer, 2016, pp. 555–572.

[98] Nikolaou, I. E., and Tsalis, T. A. Development of a sustainable balanced scorecard framework. *Ecological Indicators 34* (2013), 76–86.

[99] Panach, J. I., España, S., Dieste, O., Pastor, O., and Juristo, N. In search of evidence for model-driven development claims: An experiment on quality, effort, productivity and satisfaction. *Information and software technology 62* (2015), 164–186.

[100] Pelle, T. A quest for requirements. Bachelor's thesis, Utrecht University, 2017. Not published.

[101] Perera, O. Smes, iso 26000 and social responsibility. *ISO management systems 9* (2009), 13–19.

[102] Prakash, A., and Potoski, M. Racing to the bottom? trade, environmental governance, and iso 14001. *American journal of political science 50*, 2 (2006), 350–364.

[103] Ramautar, V. The state of the art and practice on social and environmental accounting methods and tools. Master's thesis, Utrecht University, 2019. Not published.

[104] Sharifloo, A. M., Metzger, A., Quinton, C., Baresi, L., and Pohl, K. Learning and evolution in dynamic software product lines. In *2016 IEEE/ACM 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)* (2016), IEEE, pp. 158–164.

[105] Solheim, I., and Neple, T. Model quality in the context of model-driven development. *MDEIS 6* (2006), 27–35.

[106] Sroufe, R. Integration and organizational change towards sustainability. *Journal of Cleaner Production 162* (2017), 315–329.

[107] Telesford, J. N., and Strachan, P. A. Strategic sustainability procedures: Focusing business strategic planning on the socio-ecological system in an island context. *Sustainable Development 25*, 1 (2017), 35–49.

[108] Tony Chocolonely. annual FAIR report 2017 - 2018, 2018.

[109] van de Weerd, I., and Brinkkemper, S. Meta-modeling for situational analysis and design methods. In *Handbook of research on modern systems analysis and design technologies and applications*. IGI Global, 2009, pp. 35–54.

[110] van der Pijl, S. Designing discussion and rating mechanisms to foster community based feedback for a repository of sustainability best practices. Bachelor's thesis, Utrecht University, 2020. Not published.

[111] Van Gurp, J., Bosch, J., and Svahnberg, M. On the notion of variability in software product lines. In *Proceedings Working IEEE/IFIP Conference on Software Architecture* (2001), IEEE, pp. 45–54.

[112] Vanderose, B., and Habra, N. Towards a generic framework for empirical studies of model-driven engineering. In *Proceedings of the First Workshop on Empirical Studies of Model-Driven Engineering Toulouse, France* (2008), p. 90.

[113] Waas, T., Hugé, J., Block, T., Wright, T., Benitez-Capistros, F., and Verbruggen, A. Sustainability assessment and indicators: Tools in a decision-making strategy for sustainable development. *Sustainability 6*, 9 (2014), 5512–5534.

[114] Whelan, T., and Fink, C. The comprehensive business case for sustainability. *Harvard Business Review 21* (2016), 2012.

[115] Wieringa, R. J. *Design science methodology for information systems and software engineering*. Springer, 2014.

[116] Williamson, D., Lynch-Wood, G., and Ramsay, J. Drivers of environmental behaviour in manufacturing SMEs and the implications for CSR. *Journal of Business Ethics 67*, 3 (2006), 317–330.

[117] Wohlin, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering* (2014), pp. 1–10.

[118] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. *Experimentation in software engineering.* Springer Science & Business Media, 2012.

[119] Zowghi, D., and Coulin, C. Requirements elicitation: A survey of techniques, approaches, and tools. In *Engineering and managing software requirements*. Springer, 2005, pp. 19–46.

A | IP4ESET Super Method

Figure A.1 shows a PDD depicting the IP4ESET super method created by [14]. It shows activities that need to be conducted in IP4ESET, as well as the deliverables for each activity. This model has been based on a literature review.

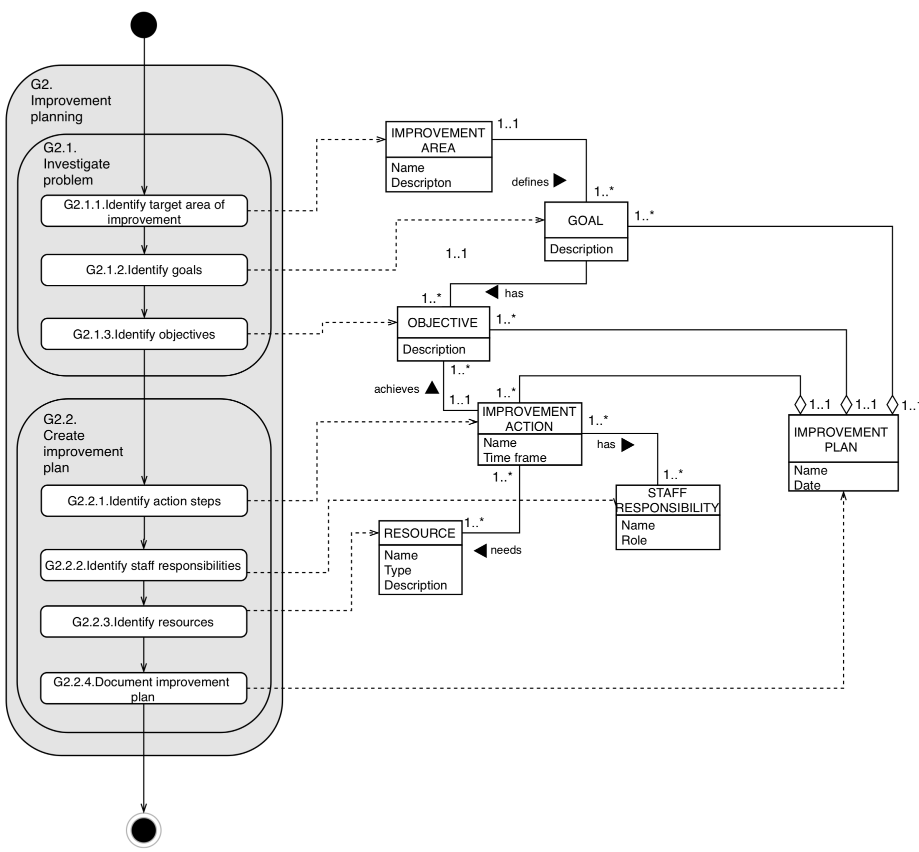

Figure A.1: PDD model of the IP4ESET phase [14]

B | Interview Materials

This appendix includes the consent form and interview questions used for the interviews with CSR experts as part of the problem investigation and treatment design.

B.1 Informed Consent

INFORMED CONSENT
Study working titles:
Designing a Model-Driven Repository for Sustainability Best Practices
&
The state of the art and practice in sustainable improvement planning

Introduction
You are asked to participate in a case study as part of an ongoing research at the Utrecht University on ICT for sustainability. We seek to understand how organisations are working on sustainability improvement planning: setting goals and making improvement plans based on sustainability performance information. To this end, an interview of approximately 1 hour will be conducted where we ask you questions about sustainability improvement planning activities in your organisation.

Your personal name will not be processed or shared as part of this research. When indicated below, the name of your organisation may be used in the reporting on the research results.

By sharing your experience and opinion, you are contributing to this body of knowledge. If you want, we will share the results with you. You are able to drop out of this research at any time. This consent form is necessary for us to ensure that you understand the purpose of your involvement and that you agree to the conditions of your participation.

Responsible for this research are Milo Plomp, Mariëlle Adèr and Sergio España. You can contact us at:

- Main researcher: Milo Plomp (m.plomp2@students.uu.nl)
- Main researcher: Mariëlle Adèr (m.j.ader@students.uu.nl)
- Supervisor: Sergio España (s.espana@uu.nl)

Please indicate your choice for the following questions:

☐ **The name of my organisation may be used in reporting on the research results.**

☐ **I give permission for the researchers to undertake audio recording during the interview. The audio files are only accessible to the main researchers and will be destroyed after transcribing.**

Please tick the following boxes for agreement:

☐ **I know that participating is completely voluntary. I know that at any moment I can decide not to participate anyway. I do not have to give a reason for that.**

☐ **I understand that the research data, without any personal information that could identify me, may be shared with others.**

☐ **I give permission to keep the collected data for at least 10 years after the end of this investigation.**

PARTICIPANT SIGNATURE

Name ______________________________

Signature ______________________________

Date ______________________________

RESEARCHER SIGNATURE

Name ______________________________

Signature ______________________________

Date ______________________________

B.2 Interview Questions

Improvement Planning - General

1.1 What is your focus related to sustainability performance?
1.2 How do you measure sustainability performance in your organisation? Do you use any standard assessment?
  1.2.1 *If yes,* which one?
  1.2.2 *If no,* how do you determine where/what to improve?
1.3 What are typical measuring/reporting/planning activities that are performed and how are they related?
1.4 Who is responsible for the improvement planning process and its activities?
  1.4.1 What about the assignment of responsibilities to specific steps/activities?
1.5 How do you determine which subjects are important to your organisation? *Assessment, vision, etc.*
  1.5.1 How do you analyse the results of an assessment?
1.6 In how much detail is the improvement plan (i.e. actions) documented? *Vision, goals, points of action?*

Improvement Planning - Tools

2.1 Do you use any (software) tools during this process?
  2.1.1 *If yes,* which one and why?
  2.1.2 *If yes,* what does the tool do?
  2.1.3 *If no,* why? Would you like to use one?
2.2 For which part of this phase would you like to (possibly) use a tool? *(e.g. goal setting, prioritising actions, process determination)*
2.3 Which problems do you face concerning improvement planning? *(e.g. strategy alignment, vagueness)*
  2.3.1 In general
  2.3.2 Related to tools or methods

Repository of Best Practices

3.1 Do you use best practices in your organisation?
- **3.1.1** *If yes,* do you find it useful?
- **3.1.2** *If yes,* where do you get these best practices from?
- **3.1.3** *If yes,* do you have place to collect these best practices?
- **3.1.4** *If not,* why not?

3.2 Do you think a repository that stores a large collection of sustainability best practices (by various organisations) would be useful for your organisation?
- **3.2.1** *If no,* is there something that would make such a tool useful?

3.3 What functionalities would you expect from a best practice repository? *(e.g. crowd-sourcing, voting)*

C | Interview Coding Scheme

The table in this appendix lists the coding scheme used for analysing the results of the interviews. We display each node with the amount of references that were coded in the interview transcripts.

Table C.1: Interview coding scheme and results

Node	Name	Contents	References
1	**Improvement planning general**	**On the process of IP.**	**91**
1.1	Assessment	How areas for improvement are assessed.	12
1.2	Difficulties	Observed difficulties in IP.	18
1.3	Improvement plan documentation	How the improvement plan is documented.	7
1.4	Networks	On operating in larger networks.	8
1.5	Responsibilities	Responsibilities for IP.	9
1.5.1	Responsibility assignment	How responsibilities are assigned.	2
1.6	SIP activities	Activities part of the IP process.	10
1.7	Sustainability focus	Focus point of strategy.	27
1.7.1	Focus determination	How the focus is determined.	16
1.7.2	Targets	Targets for the strategy.	6
2	**Improvement planning tools**	**On the use of tools for IP.**	**47**
2.1	Tool desires	Expressed desired functionality for IP tools.	7
2.2	Use of tools	Current use of tools.	39
2.2.1	Application	How tools are applied in the organisation.	22

Table C.1 – continued from previous page

Node	Name	Contents	References
2.1.2	Motivation	Why tools are used.	8
2.1.3	Shortcomings	Shortcomings of currently used tools.	7
3	**Best practice repositories**	**On the use of a BPR.**	**30**
3.1	Best practices	On the use of BPs.	15
3.1.1	Storage	How and where BPs are stored.	2
3.1.2	Use	How BPs are used.	12
3.2	Desires	Expressed desires for a possible BPR.	7
3.3	Sentiment	Expressed sentiment on the idea of BPRs.	7

D | Shortcoming Analysis

This appendix shows the results of the scientific literature review for the shortcoming analysis.

Table D.1: Discussed types of SIP approaches

Type	Instances
Standard	18
(Strategic) management	12
General	5
Method	4
Tool	3
Framework	2
Performance measurement	1

Table D.2: Criticized SIP approaches

Name	Type	Instances
ISO 26000	Standard	13
ISO 14001		5
Strategic management	(Strategic) management	12
BSC	Method	4
SBSC		2
Tools	Tool	2
SATs		1
Performance measurement	Performance measurement	1

Table D.3: Proposed SIP solutions

Name	Type	Instances
FSSD	Framework	10
SBSC		7
GRI Framework		1
Action for Sustainability		1
BSC		1
PSAT	Tool	8
Tools		2
S-ERP		2
ISO 26000	Standard	6
Strategic management	Strategic management	3

Table D.4: Problem instance count

Problem	Instances
Strategy	12
Vagueness	11
Resources	9
Complexity	7
Fixedness	5
Other	3

Table D.5: Solution instance count

Solution	Instances
Strategy formulation	13
Strategy implementation	13
Decision-making	9
Target identification	7
Performance measurement	5
Definition formulation	2
Awareness	2
Other	1

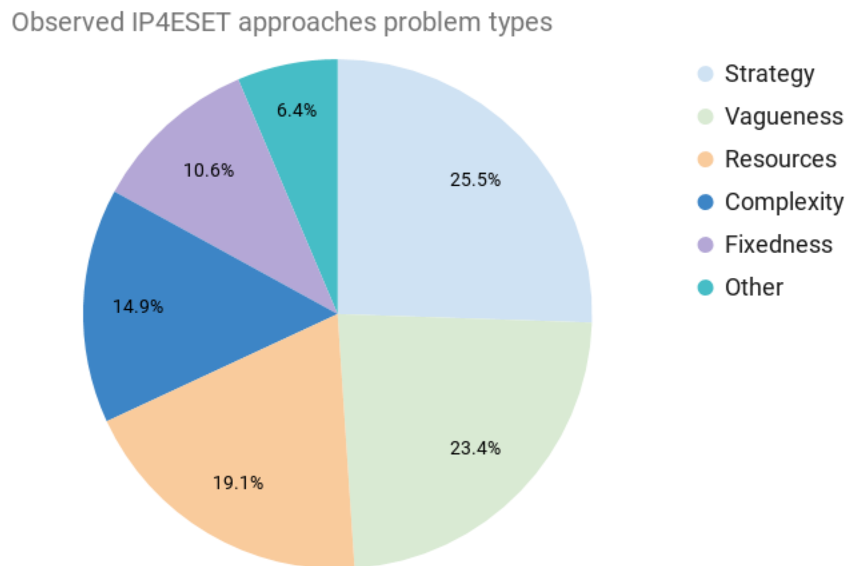

Figure D.1: Observed SIP problem types

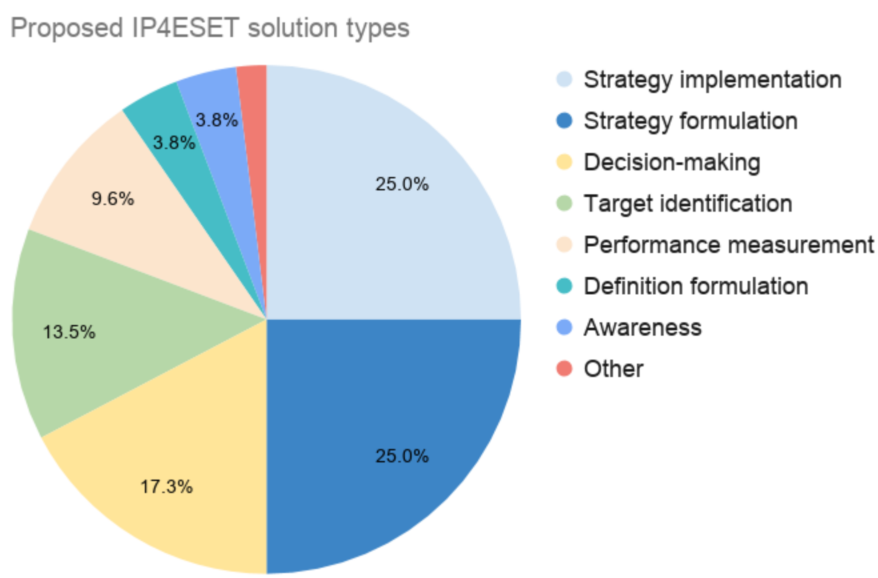

Figure D.2: Observed SIP solution types

Table D.6: Findings of the literature shortcoming analysis (shortcomings indicated with "S", requirements with "R")

Ref	*Type*	*Name*	*S/R*	*Description*	*S/R type*
[21]	Framework	FSSD	R	Backcasting based on FSSD principles.	Strategy formulation
			R	Measurements, activities and goals should be determined.	Target identification
[22]	Framework	FSSD	R	Inclusion of strategic principles to guide efforts to practical solutions and measures.	Target identification
			R	Indicators and metrics measure success of actions.	Performance measurement
		SBSC	R	Tackling product displacement by dividing strategy into four perspectives.	Strategy formulation
	General	General	S	Approaches to sustainable development are reductionist: problems are part of a larger system and should be treated as such.	Other
[23]	(Strategic) mgmt.	Strategic mgmt.	S	Lack of strategic orientation towards sustainability practices and goals.	Strategy
			S	Identification and attainment of goals is difficult.	Complexity
	Framework	FSSD	R	Operationalization of sustainable improvement needs to be testable.	Target identification
[24]	General	General	R	Reports need to be integrated in top-tier management for decision-making.	Decision-making
	Tool	SATs	S	Standards are too informal to follow.	Vagueness

Table D.6 – continued from previous page

Ref	*Type*	*Name*	*S/R*	*Description*	*S/R type*
[26]	Method	SBSC	R	Sustainable strategy must match corporate policy.	Strategy formulation
			R	Systems should be aligned with strategy.	Strategy implementation
			R	SBSC includes TBL principles in development of strategy.	Strategy formulation
[27]	Standard	ISO 14001	S	The standard has loose rules. The room for interpretation makes the standard a "myth".	Vagueness
			S	Employees have only a vague understanding of the ISO 14001 prescriptions.	Vagueness
[30]	Framework	FSSD	R	Support for decision-making.	Decision-making
			R	Support for planning.	Strategy formulation
[32]	Performance measurement	Performance measurement	S	Costly to implement.	Resources
	Method	BSC	S	Time-consuming and complex.	Complexity
			S	Decision-making influencing factors are not influenced by environment.	Fixedness
			R	Dynamic decision-making.	Decision-making
			S	Does not take strategic views into consideration.	Strategy

Table D.6 – continued from previous page

Ref	*Type*	*Name*	*S/R*	*Description*	*S/R type*
[34]	General	General	S	Lack of one of the elements of the triple bottom line in sustainability approaches.	Other
			S	SMEs lack knowledge of most promising ways for improvement.	Strategy
			S	SMEs lack resources in terms of capital, personnel and time.	Resources
[35]	Tool	S-ERP	R	Sustainability data and processes need to be centralised in S-ERP systems.	Other
			R	Decision-making processes are important to reach goals.	Decision-making
[41]	(Strategic) mgmt.	Strategic mgmt.	S	Lack of relevant info for SMEs.	Vagueness
			S	Lack of resources to build networks.	Resources
			S	Lack of employee training.	Resources
[45]	(Strategic) mgmt.	Strategic mgmt.	S	Companies have a better understanding of strategy formulation than implementation	Strategy
[46]	(Strategic) mgmt.	Strategic mgmt.	S	Diverse set of aspects to CSR makes strategy formulation and implementation more difficult.	Complexity
			S	Objectives and environment are ever-changing.	Strategy

Table D.6 – continued from previous page

Ref	*Type*	*Name*	*S/R*	*Description*	*S/R type*
[47]	(Strategic) mgmt.	Strategic mgmt.	S	Managers do now know how to translate strategy into action.	Strategy
	Framework	SBSC	R	Using SBSC, performance metrics for sustainability are tied to sustainability strategy.	Performance measurement
[52]	Framework	SBSC	S	Lack of know-how for SBSC implementation.	Complexity
[53]	Method	BSC	S	BSC does not include social and environmental perspectives in defining strategies.	Strategy
[56]	Framework	FSSD	R	The FSSD uses BMC to embed sustainability in value-creating process.	Strategy implementation
[63]	General	General	R	CSR strategies need to be implemented in core strategies.	Strategy implementation
			R	Mission should be operationalized by setting objectives.	Target identification
			R	Action plans should include definite measures with the help of programs, policies, procedures and processes.	Strategy implementation
	Standard	ISO 26000	S	ISO 26000 is too broad for SMEs.	Vagueness
			S	ISO 26000 tries to give one definition of the management standard to all types of organisations.	Fixedness
			S	ISO gives little guidance on generating CSR strategies.	Strategy

Table D.6 – continued from previous page

Ref	*Type*	*Name*	*S/R*	*Description*	*S/R type*
			R	ISO 26000 gives several good practice examples that help in the implementation of strategy.	Strategy implementation
	(Strategic) mgmt.	Strategic mgmt.	R	Organisations should look at strategy formulation.	Strategy formulation
			R	Organisations should look at strategy implementation.	Strategy implementation
			R	Organisations should look at strategy evaluation.	Performance measurement
			S	Sustainability-related strategies are linked to long term, uncertainty, and ambiguous cause-effect. Impacts are more difficult to observe.	Complexity
	Framework	GRI Framework	R	Aids the operationalization of objectives and goal setting.	Strategy formulation
[64]	Tool	Tools	R	Tools should support the sharing of knowledge to make new decisions.	Decision-making
			R	Tools are primarily aimed at assessment and should be more proactively used to get ideas.	Decision-making
[68]	Standard	ISO 26000	S	Scope of ISO 26000 is too broad.	Vagueness
			S	No detailed guidance provided to SMEs.	Vagueness
			S	Steep learning curve for SMEs.	Complexity
			S	Costly and time-consuming to implement.	Resources
			S	Non-certifiable.	Other

Table D.6 – continued from previous page

Ref	*Type*	*Name*	*S/R*	*Description*	*S/R type*
[71]	Framework	Action for Sustainability	R	Challenging and monitorable targets are required for success within short and long term.	Target identification
[72]	Standard	ISO 14001	S	Certification costs are high.	Resources
[73]	Tool	Tools	S	Tools are costly to implement.	Resources
			S	Tools are complex to implement.	Complexity
			R	Awareness for issues is prerequisite for tool implementation.	Awareness
[83]	(Strategic) mgmt.	Strategic mgmt.	S	Companies struggle to align sustainability and strategy.	Strategy
			S	Companies struggle to align sustainability and IT strategies.	Strategy
[85]	Tool	PSAT	R	Selection of people who are in leadership roles.	Decision-making
			R	Selection of people who can commit to sustainability planning.	Strategy implementation
			R	Review short-, medium- and long-term objectives to include in action plan.	Target identification
			R	Programs must be able to adapt to changes in the environment.	Strategy formulation
			R	Prioritize points of action for domains and indicators.	Strategy formulation
			R	Write an action plan.	Strategy formulation
			R	Update stakeholders and continually brainstorm.	Strategy implementation

Table D.6 – continued from previous page

Ref	*Type*	*Name*	*S/R*	*Description*	*S/R type*
			R	Annual reassessment.	Performance measurement
[86]	General	General	R	Awareness needs to be created.	Awareness
			R	Clear direction and selection of strategy pillars needs to be set/made.	Target identification
			R	Procedures and actions need to be puzzled out.	Strategy formulation
[91]	Framework	FSSD		Support for creating definition of sustainability	Definition formulation
[92]	Standard	ISO 26000	S	Relies on goals that are seen as norms.	Fixedness
			R	Organisations should establish own goals, guidelines and tools.	Strategy formulation
			S	Goals do not change with society.	Fixedness
			R	Organisations should be able to plan and improve systematically.	Strategy formulation
			S	It has no strategic element.	Fixedness
[96]	Standard	ISO 26000	R	ISO 26000 should give guidelines for the implementation of best practices for addressing CSR.	Strategy implementation
			R	ISO 26000 gives organisations the ability to create own meaning of CSR.	Definition formulation

Table D.6 – continued from previous page

Ref	*Type*	*Name*	*S/R*	*Description*	*S/R type*
[98]	Framework	SBSC	R	SBSC can aid in implementation of sustainability strategy.	Strategy implementation
			R	SBSC can aid in measurement and disclosure of sustainability performance info.	Performance measurement
	Framework	BSC	S	No standard instruction on how to integrate sustainability in BSC perspectives.	Vagueness
[101]	Standard	ISO 26000	S	The scope of ISO 26000 is too broad for SMEs.	Vagueness
			R	Performance monitoring should be done by SMEs following implementation.	Performance measurement
			S	Lacks practical guidance on implementation for SMEs.	Resources
[102]	Standard	ISO 14001	S	Implementation requires high-cost employee training and investments.	Resources
			S	Large costs for ISO 14001 certification.	Resources
[106]	General	General	S	Various terms for sustainaiblity are used.	Vaguenss
			R	Sustainability needs to be incorporated into decision-making.	Decision-making
			R	A global integrated repository is needed to link knowledge management and actions.	Decision-making

Table D.6 – continued from previous page

Ref	*Type*	*Name*	*S/R*	*Description*	*S/R type*
[107]	Framework	FSSD	R	In aligning internal mission, vision, goals and strategies with the vision for sustainability, frameworks need to be applied.	Strategy formulation

E | Feature Models

In this appendix, the feature models that are created as part of the feature analysis for the shortcoming analysis are shown.

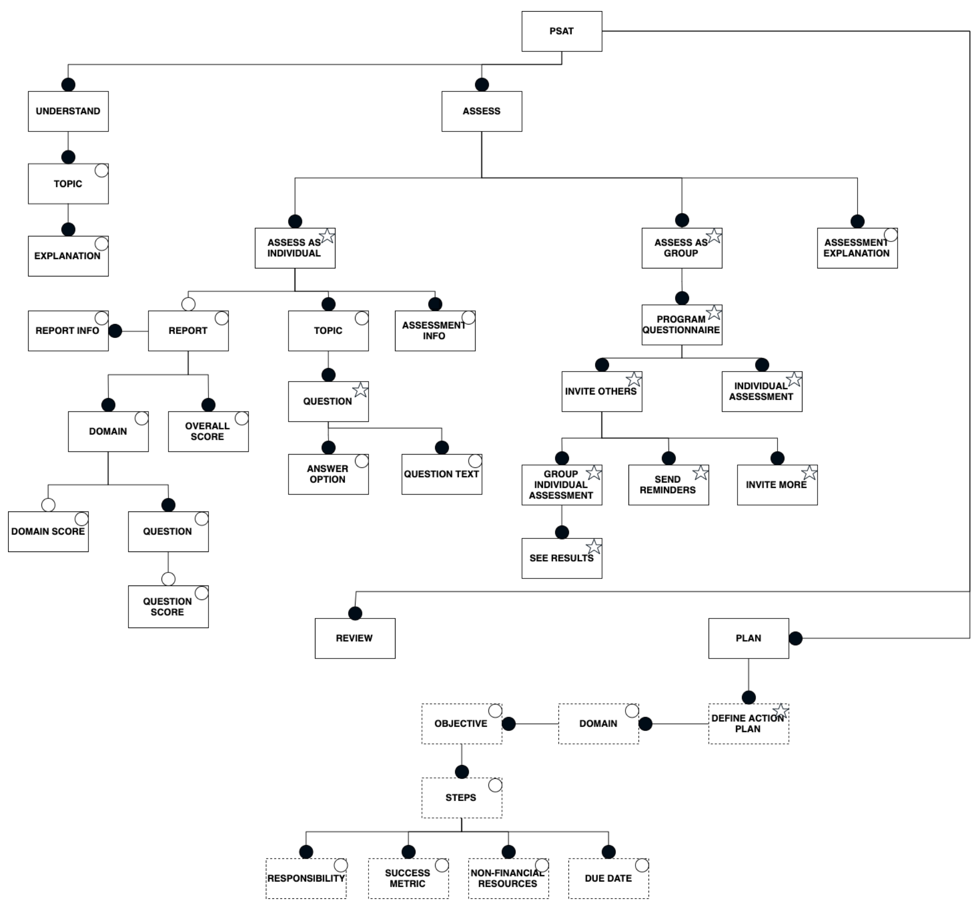

Figure E.1: PSAT feature model

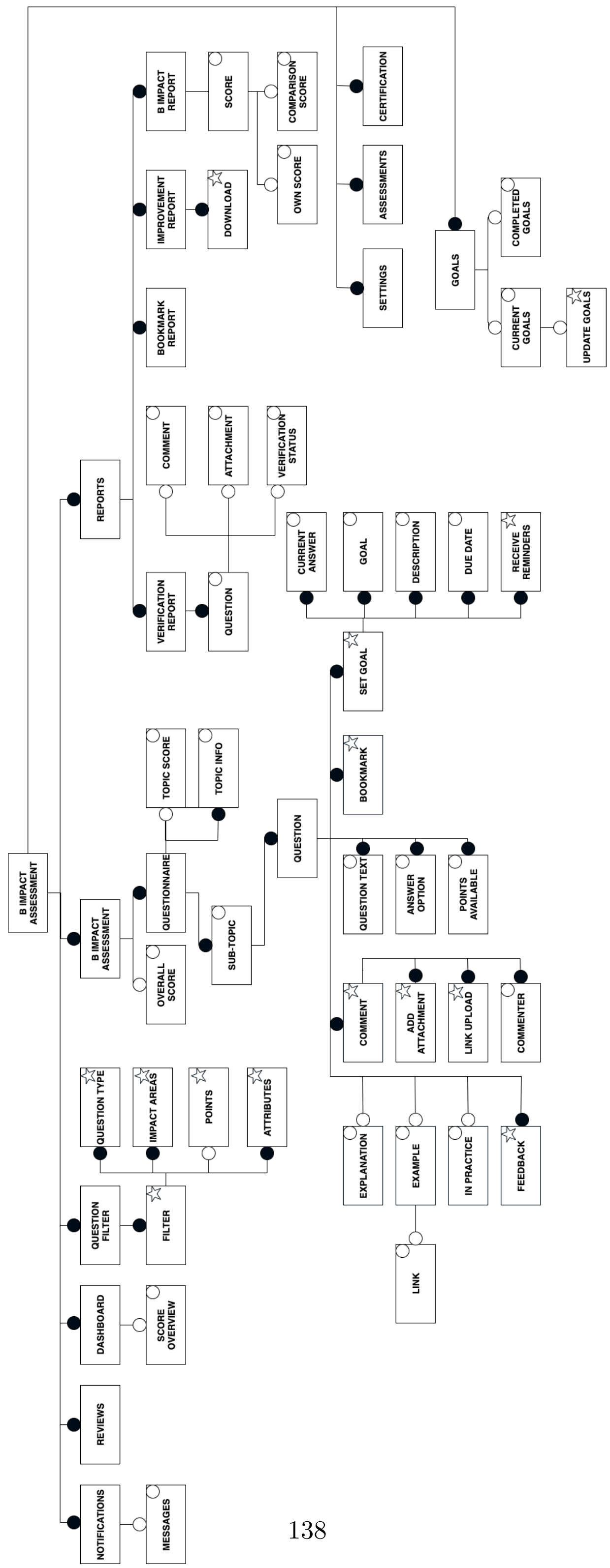

Figure E.2: BIA feature model

F | Grey Literature

In this appendix, we show the results of the grey literature analysis for the shortcoming analysis. In the table below, the *what* question is concerned with the following aspects:

W1 Includes guidelines on action steps identification

W2 Lists action steps

The *how* question is concerned with the following question:

H1 Specifies how to perform action steps

Table F.1: Relation of grey literature to the what and how question of SIP

Source			Remarks
OECD Due Diligence Guidance for Responsible Business Conduct [8]	W1	+/-	This guideline provides an abstract overview of due diligence: its definition, iterative addressal, and practical actions. These action steps don't provide enough detail to be implementable. The guideline is aimed at setting a starting point for building out a due diligence strategy using additional resources.
	W2	+	Lists "practical actions" with no further detailed specification.
	H1	-	The guideline mentions many of the steps in the SIP super method (e.g. the importance of responsibility assignment), but provides no detailed support on how to perform a step.
Due Diligence Toolkit for Responsible Business Conduct [1]	W1	+	The self-assessment questionnaire helps in the identification of action steps. External assessments for target identification are mentioned.

Table F.1 – continued from previous page

Source			Remarks
	W2	+	Defines required steps that help in the implementation of RBC policies in management systems.
	H1	-	Additional resources are required to specify how action steps are to be carried out. The how question is only partly answered in this documentation.
MVO Prestatieladder [11]	W1	+	Provides an overview of the PDCA cycle that is required to be followed in order to certify your MVO management system. Each step has guidelines. For each level of certification, expected indicator status is mentioned. Links are given to ISO 26000.
	W2	+/-	Steps are given to define further concrete action steps.
	H1	+/-	Some links are given to additional resources. The chapter on planning for the MVO management system gives no further details that explain the action steps.
IDH Gender Toolkit and Guide [3]	W1	+/-	The guide presents an abstract overview of how to define action steps for each of the 6 programming steps. The gender tool (which is being tested) should highlight opportunities.
	W2	+/-	No concrete action steps are given in the guide, but should be provided by the tool.
	H1	+/-	The tool should highlight potential interventions which are not provided in the guide.
Blueprint for Business Leadership on the SDGs [6]	W1	+/-	Guiding questions, targets and considerations are posed to inspire action steps identification. No guidelines are given.
	W2	+	Action steps are given as “business actions” for each SDG.
	H1	-	Gives references to additional resources that give further explanation on action steps, but no detailed guidelines are given in the source.

Table F.1 – continued from previous page

Source			Remarks
SDG Compass Guide [5]	W1	+/-	References are given to the SDG Compass website, which includes an inventory of business indicators for each SDG. These indicators are linked to sources from GRI, UN Global Compact and more, and explain how to assess impact.
	W2	+/-	The guidelines gives an iterative approach to targeting SDGs and explains each phase. For the selection of concrete steps, additional resources are required.
	H1	+/-	Each phase is explained in more detail using text and models. Only examples are given, and no clear-cut solutions.

The following table shows an overview of all sources that have been considered for inclusion in the grey literature analysis.

Table F.2: Sources considered for grey literature analysis

Source	Included	Remarks
Chocolonely FAIR Report [108]	No	Does not describe an SIP approach.
Tony's Open Chain[1]	No	Describes tools for specific sustainability issues.
Circle Economy: Austria GAP report [9]	No	Not an SIP approach or solutions are given.
Circle Economy: Tracking Value [12]	No	Does not describe an SIP approach.
UN Global Compact: Ending Child Labour [10]	No	Does not describe an SIP approach.
Blueprint for Business Leadership on the SDGs [6]	Yes	Meets inclusion criteria.
SDG Compass [5]	Yes	Meets inclusion criteria.
Child labour solution paper [2]	No	Describes tool for specific sustainability issue.
Due Diligence Toolkit for Responsible Business Conduct [1]	Yes	Meets inclusion criteria.
OECD Due Diligence Guidance for Responsible Business Conduct [8]	Yes	Meets inclusion criteria.
NPR 9036:2015 Guidance for due diligence integration [4]	No	Inaccessible.
MVO Prestatieladder [11]	Yes	Meets inclusion criteria.
IVMO Convenant [7]	No	Does not describe an SIP approach.
IDH Gender Toolkit and Guide [3]	Yes	Meets inclusion criteria.
SIFAZ Equivalency Tool[2]	No	Not an SIP approach and no guidelines available.

[1] `https://www.tonysopenchain.com/`
[2] `http://www.standardsmap.org/sifav/`

Table F.2 – continued from previous page

Source	Included	Remarks
Ecovadis tool[3]	No	Tool and documentation inaccessible.

[3] https://www.ecovadis.com/

G | State-of-the-art and practice on BPRs

The figures in this appendix describe the state-of-the-art and practice on BPRs and BPs and are created in other research supervised by the author [39].

	Literature	Déroche	BIA	GIS	CaaS	FAO	PHBPR	COE	OBPS
Store (RF1)	x	x	x	x	x	x	x	x	x
Display (RF2)	x	x		x	x	x	x	x	x
Alter (RF3)	x								
Search (RF4)	x		x		browse		browse	*filter**	browse
Categorise (RF5)	x	x	x	x	x	x	x	x	x
User Management (RF6)	x								
Link (RF7)	x		x						
Submit (RF8)			*submit a ticket*				x	x	x
Download		x	x			x	x		x
List of Best Practices (RF9	*from quality feature 'content'*	List of patterns	x	List of measures	List of patterns	List of Good Practices	x	List of Good Practices	x

Figure G.1: Mapping of case-based repository feature models to a literature-based repository feature model [39]

	Bravos et al. (2014)	Dani et al. (2006)	Stirna and Persson (2009)	Alwazae et al. (2019)	Déroche	BIA
BP	pattern	x	x	x	pattern	x
Problem (DR3)	x	enablers	x		in description	x
Solution (DR5)	guidelines	implementation	x	x	in description	x
Model (DR15)						
Context (DR4)	x	process description			in description	x
Keyword (DR12)	x					
Link (DR13)	x					in BPR
Related info (DR7)	x				problematic	x
Feedback (DR14)	x					in BPR
Example (DR8)	x	x			x	BP is example
Consequence (DR6)	forces	cause and effect	x	x	x	x
Author (DR9)	x			x		
Type	x					

Figure G.2: Mapping of case-based repository feature models to a literature-based repository feature model [39]. Image is cropped for legibility.

GIS	SAP	CaaS	CEES	FAO	PHBPR	COE	OBPS
measure	x	pattern	x	good practice	x	x	x
in description	pain point	x	in description	x	x	in description	in description
in description	x	x	measurement	approach	x	in description	in description
		x					
	prerequisites	x		x	x	in description	in description
		x			x	x	
x							x
	x		x	x			
		x					
					BP is example		BP is example
x	benefit		success	impact			x
		x				origin	x
						x	

Figure G.3: Mapping of case-based repository feature models to a literature-based repository feature model [39]. Image is cropped for legibility.

H | Requirements

Table H.1 lists the requirements, in the form of user stories, drafted for OpenBest based on a variety of sources. This overview is not prioritized. User stories in **bold** have been implemented. Other user stories have not been implemented yet.

Table H.1: Requirements

Planning
As a user, I want to link short-term interventions to long-term goals, so that I can see which interventions are required to reach a goal.
As a user, I want to be able to divide long-term goals into short-term project states, so that I know in which order to work on interventions.
As a user, I want to see an overview desired future project states, so that I can see why I am applying an intervention.
As a user, I want to link interventions to project states, so that I know which interventions are required to move between project states.
As a user, I want to assign themes to interventions, so that I can group interventions.
As a user, I want to relate interventions to the company mission, so that my intervention is consistent with the company mission.
As a user, I want to define the motivation for applying an intervention, so that I can track the reasoning within a project.
As a user, I want to know the structure of interventions, so that I store interventions according to the correct structure.
As a user, I want to see the current state of the project, so that I know which preconditions still need to be met.
As a user, I want to view my improvement plan as a graphical theory of change, so that I can communicate the improvement plan.
As a user, I want to know the structure of the planning process, so that I adhere to the correct structure.

Table H.1 – continued from previous page

As a user, I want to be able to define all resources required for interventions, so that I know which resources are required for successful application.
As a user, I want to be able to identify indicators for project states, so that I know when a new project state can be reached.
As a user, I want to be able to identify thresholds for indicators, so that I know when an indicator is reached.
Evaluation
As a user, I want to monitor intervention implementation, so that I know how best practices have been applied.
As a user, I want to identify the result of an intervention, so that I know what its outcome is.
As a user, I want to view the results of interventions in various contexts, so that I know the degree of success in previous applications.
Best practices
As a user, I want to view best practices on how to apply interventions, so that I can learn from other organisations.
As a user, I want to be able to rate a best practice on its perceived usefulness, so that others can learn from my opinion.
As a user, I want to be able to discuss the application of a best practice in my intervention, so that others can learn from my implementation.
As a user, I want to be able to search for best practices using search strings, so that I can get inspiration for interventions to conduct.
As a user, I want to be able to search for best practices using filters, so that I can get inspiration for interventions to conduct.
Context
As a user, I want to view recommended interventions based on my organisation's size, so that I know what has worked for similar organisations.
As a user, I want to view recommended interventions based on my organisation's sector, so that I know what has worked for similar organisations.
As a user, I want to view recommended interventions based on my organisation's geographic location, so that I know what has worked for similar organisations.
As a user, I want to identify stakeholders, so that I know who impacts the result of an intervention.
As a user, I want to be able to identify exclusion criteria for interventions, so that I can describe which criteria lead to a failure of intervention implementation.
Other

Table H.1 – continued from previous page

As a user, I want to include stakeholders of the intervention within my organisation, so that I can raise awareness.
As a domain administrator, I want to be able to upload textual models describing the project's structure, so that the system functionality can adapt to the current project.
As a user, I want to be able to upload additional project documents, so that these documents can be reused in future project iterations.

I Best Practice UML Class Diagrams

This appendix shows the UML class diagrams created to depict the content structure of BPs. This has been done in collaboration with another researcher [39]. We show 3 versions of models, depending on how many modifications are needed to agree on a model.

I.1 Version 1 models

These UML models are the first version models proposed by [39].

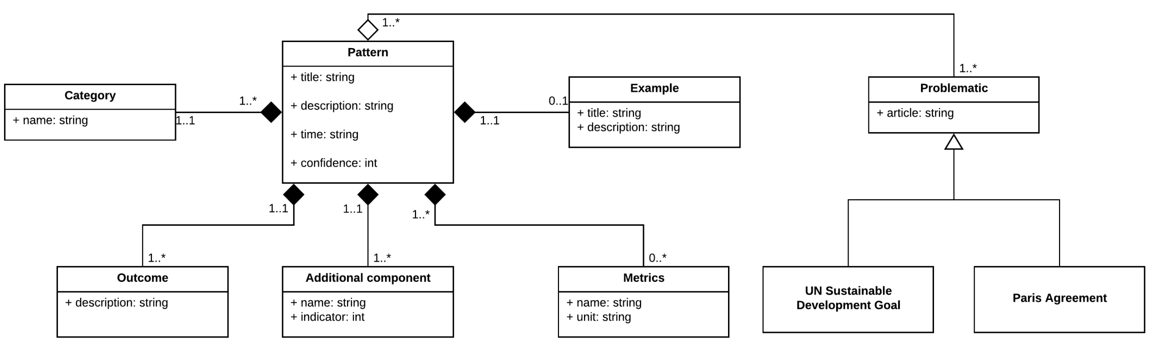

Figure I.1: UML of Déroche best practice [39]

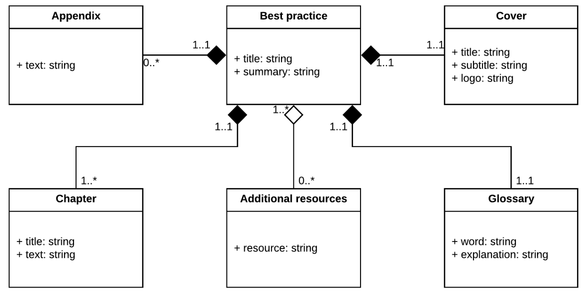

Figure I.2: UML of BIA best practice [39]

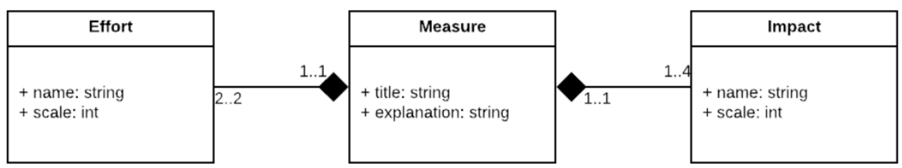

Figure I.3: UML of GIS best practice [39]

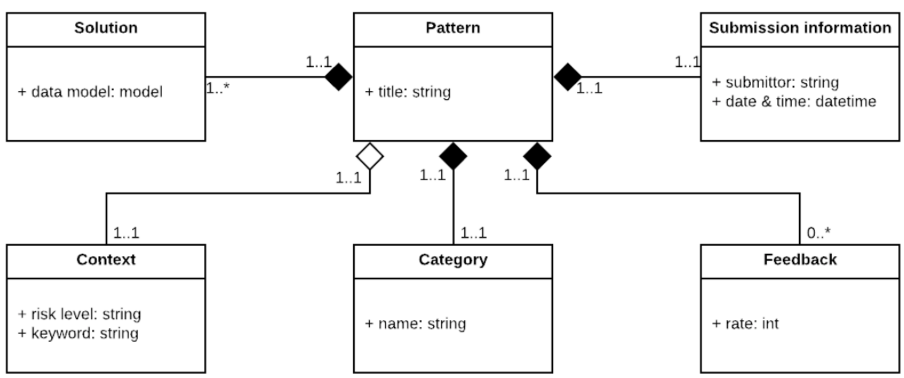

Figure I.4: UML of CAAS best practice [39]

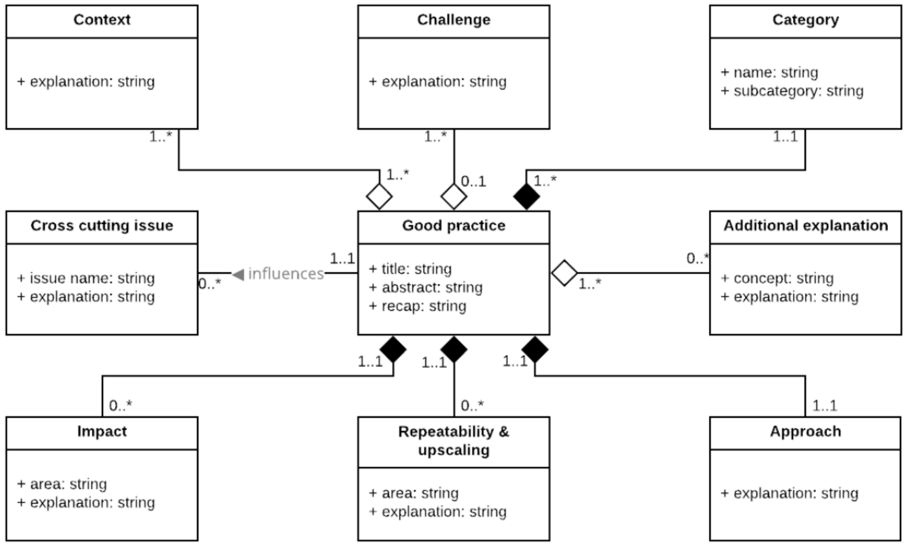

Figure I.5: UML of FAO best practice [39]

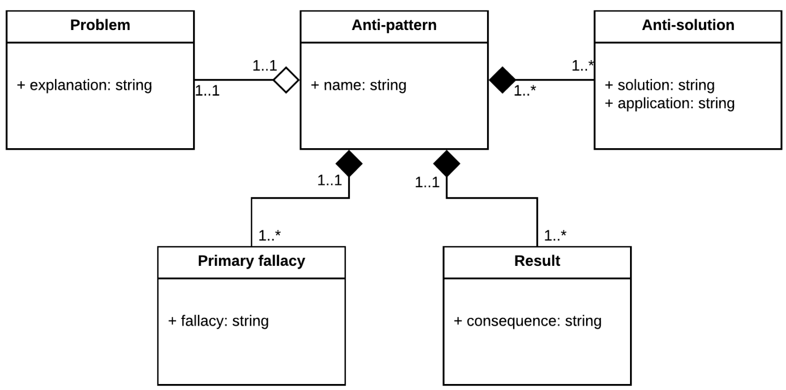

Figure I.6: UML of anti-pattern [39]

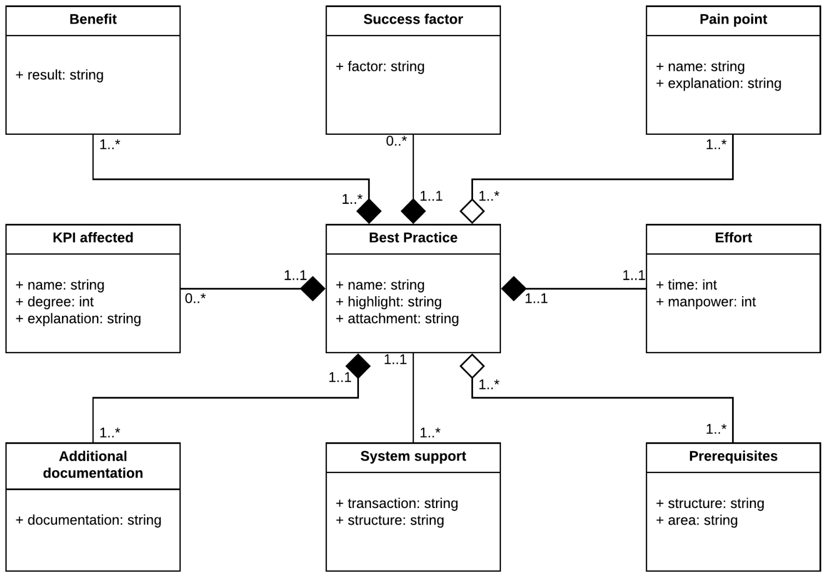

Figure I.7: UML of SAP best practice [39]

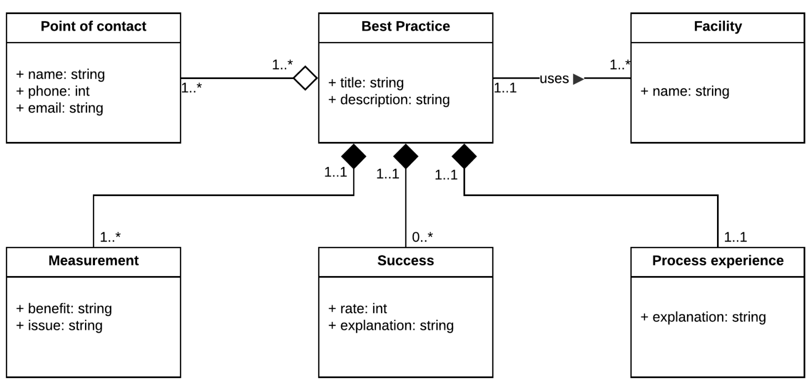

Figure I.8: UML of CEES best practice [39]

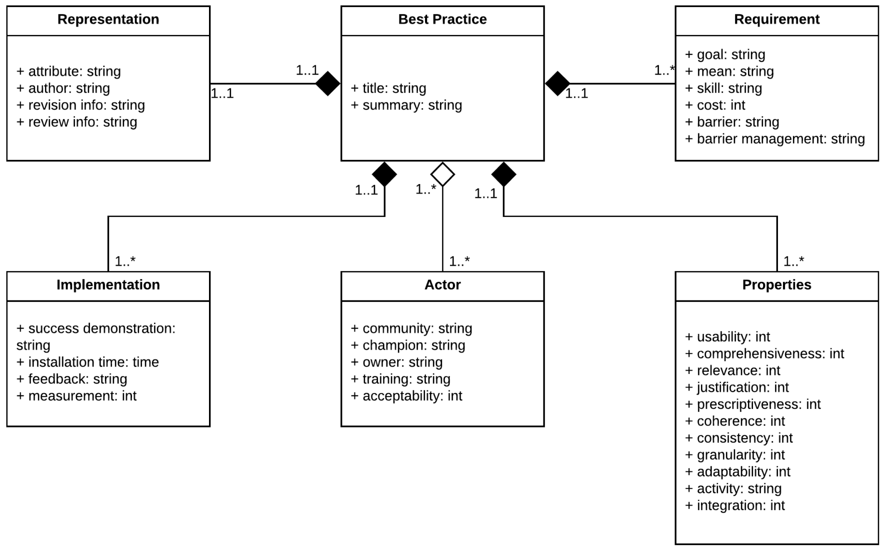

Figure I.9: UML of TDBPD best practice [39]

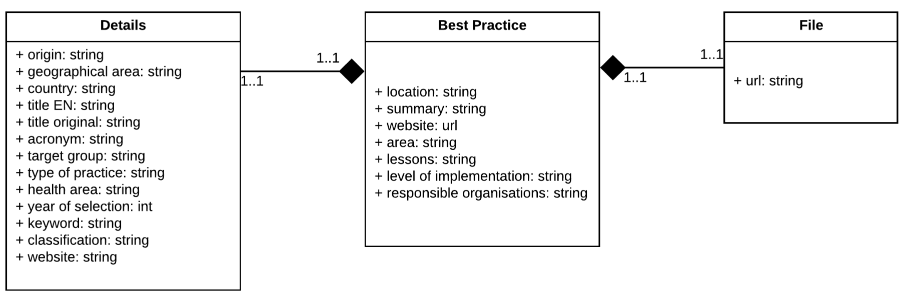

Figure I.10: UML of PHBPR best practice [39]

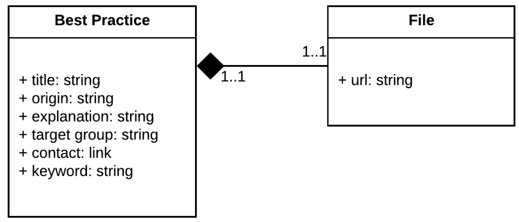

Figure I.11: UML of COE best practice [39]

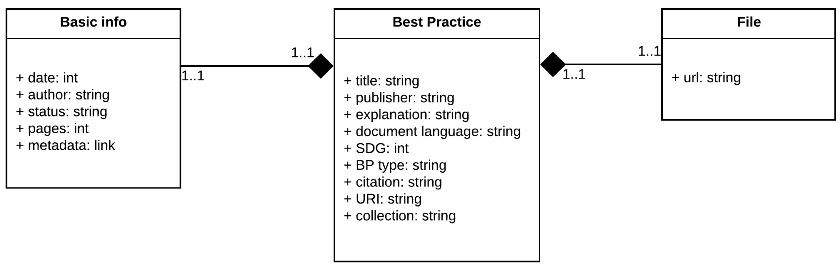

Figure I.12: UML of OBPS best practice [39]

I.2 Version 2 models

These UML models indicate the proposed changes by the author to some of the version 1 models.

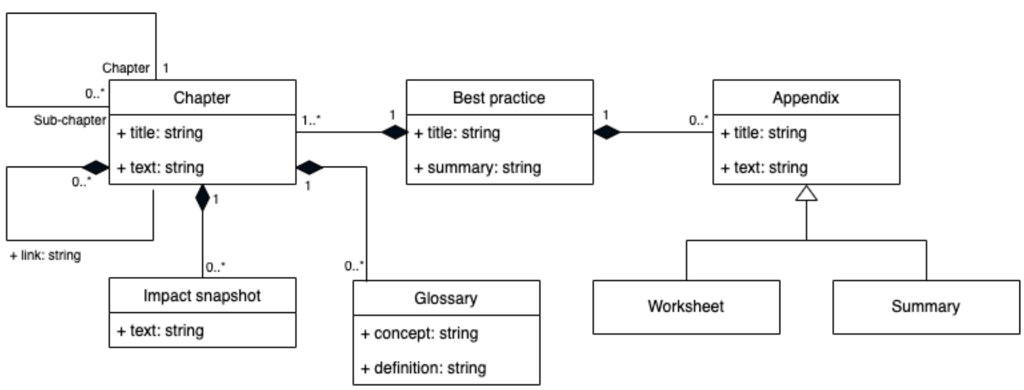

Figure I.13: UML of BIA best practice v2

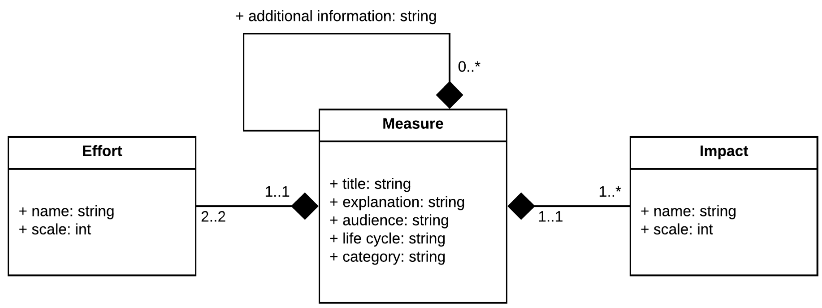

Figure I.14: UML of GIS best practice v2

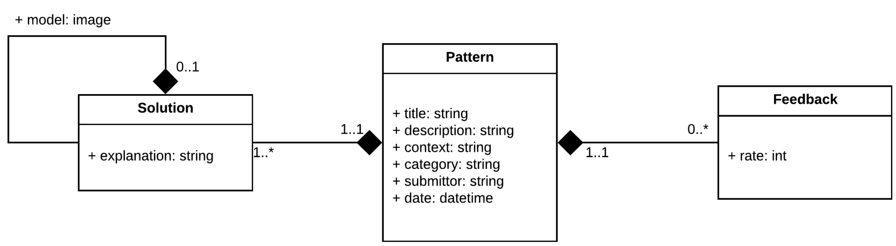

Figure I.15: UML of CAAS best practice v2

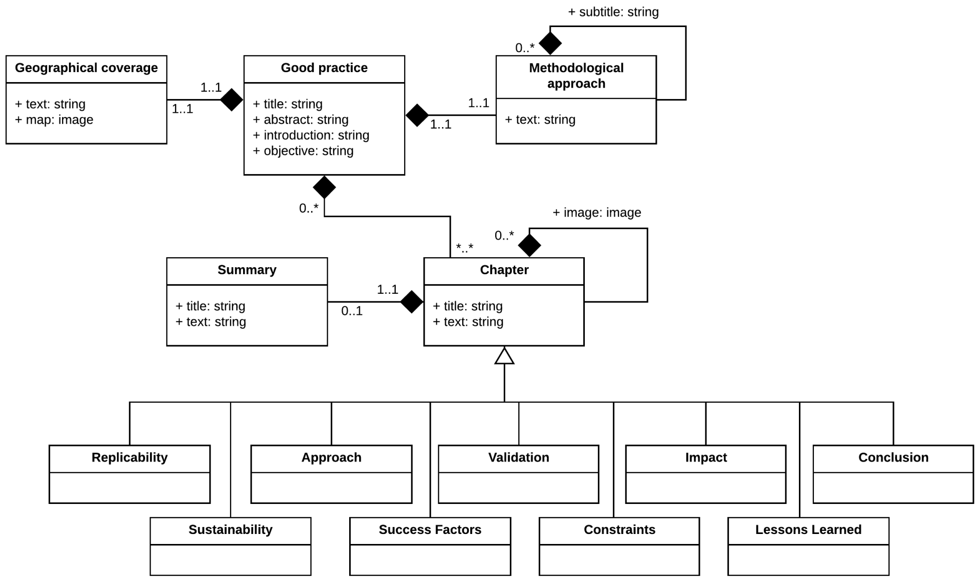

Figure I.16: UML of FAO best practice v2

I.3 Version 3 models

This section shows the version 3 models that were created after consolidation of version 1 and version 2 models. This was only required for models that were not correct at version 2.

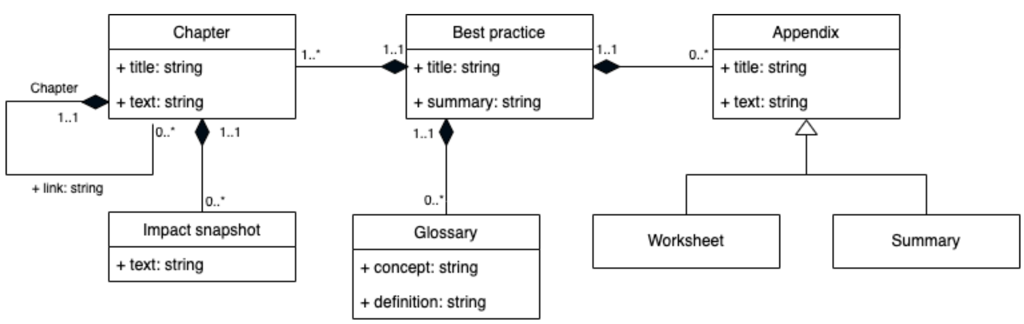

Figure I.17: UML of BIA best practice v3

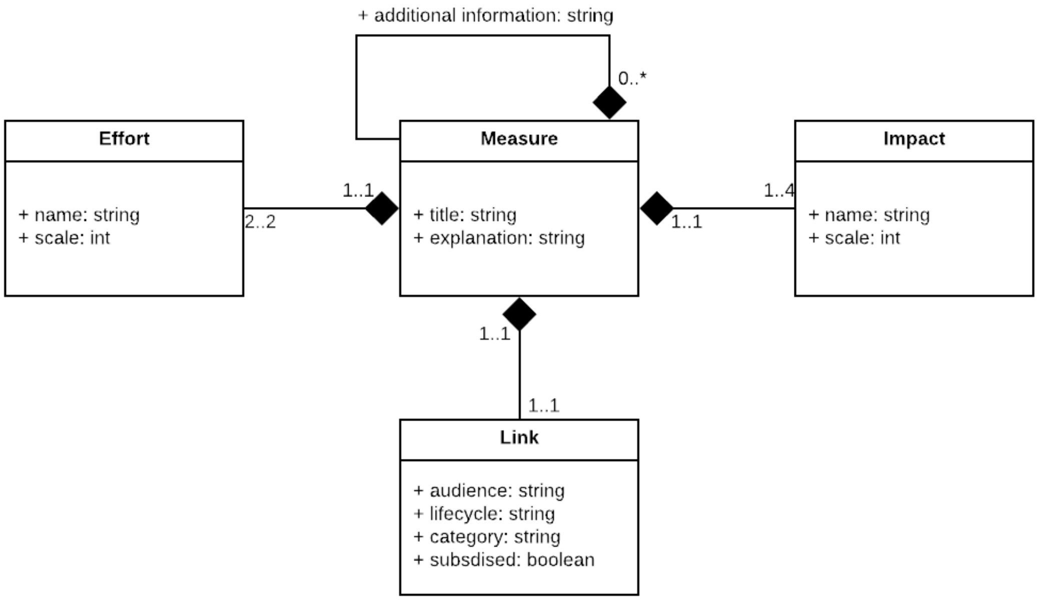

Figure I.18: UML of GIS best practice v3 [39]

I.4 UML Comparison Results

The following table shows the instance count of concepts and attributes across model instances. Column *M1* shows the instance count of the initial models created by the author, *M2* shows the instance count of the initial models created by [39]. Column *Final* shows the instance count of the final version models. The *change* column shows what changes are made across model instances.

- *M1* = (or *M2* =) indicates that no changes were made to the instances listed in column *M2* (*M1*).
- *M2* - indicates that the final version model required removal of one or more instances from the *M2* column.
- *M2* + indicates that the final version model required an addition of one or more instances to the first *M2* column.

The final column *version* shows the model version that was used for domain analysis (please refer to the previous subsections).

Table I.1: UML model comparison results

ID		M1	M2	Final	Change	Version
DER	Concepts	9	9	9	M2 =	v1
	Attributes	13	13	13	M2 =	
BIA	Concepts	7	6	7	M1 =	v3
	Attributes	11	11	10	M1 -	
GIS	Concepts	3	4	4	M2 +	v3
	Attributes	10	10	11	M2 +	
CAAS	Concepts	3	6	3	M1 =	v2
	Attributes	9	8	9	M1 =	
FAO	Concepts	14	9	14	M1 =	v2
	Attributes	13	16	13	M2 =	
ANTI	Concepts	5	5	5	M2 =	v1
	Attributes	6	6	6	M2 =	
SAP	Concepts	9	9	9	M2 =	v1
	Attributes	17	17	17	M2 =	
CEES	Concepts	6	6	6	M2 =	v1
	Attributes	11	11	11	M2 =	
TDBPD	Concepts	6	6	6	M2 =	v1
	Attributes	32	32	32	M2 =	

Table I.1 – continued from previous page

ID		M1	M2	Final	Change	Version
PHBPR	Concepts	3	3	3	M2 =	v1
	Attributes	21	21	21	M2 =	
COE	Concepts	2	2	2	M2 =	v1
	Attributes	7	7	7	M2 =	
OBPS	Concepts	3	3	3	M2 =	v1
	Attributes	15	15	15	M2 =	

I.4.1 Common Elements

The following table lists an overview of the concepts extracted from the analysed UML models to be considered for the core model. It also lists the instance count of these concepts in the analysed UML models (similar concepts are grouped under a synonymous concept name). These concepts were considered for inclusion based on their instance count and perceived usefulness.

Table I.2: Concepts considered for the core model

Concept	Count	Reasoning
Best practice	12	Central to what we are storing in the repository. Concerns a best practice *document* that stores various pieces of information.
Solution	8	Describes what needs to be applied in the best practice to target the problem(s).
Result	7	Describes what the solution achieves. This is influential to both the decision to apply the best practice and the assessment of its successfulness.
Problem	3	Describes what is targeted by the best practice.
Example	3	Gives contextual info on how a best practice has been applied in previous settings.
Effort	2	Describes what is needed for successful implementation (success factors). Gives contextual info and helps in decision-making.
Rating	2	Provides feedback on the best practice based on earlier implementations.
Author	2	The author of the best practice.

Table I.2 – continued from previous page

Concept	Count	Reasoning
Category	1	Best practices can be grouped along various categories.

J | OBL models

This appendix shows an overview of the (documentation for) models that make up the OBL language. We define abstract syntax models and concrete syntax models.

J.0.1 Abstract Syntax Models

Table J.1 lists an explanation of the concepts and relationships in the OBL core model and metamodel. Attribute names are written in italics and association relationship names are written in capital letters.

Table J.1: Explanation of concepts and relationships in the OBPL core model and metamodel

Core Model	
Element	**Explanation**
Best practice	A best practice defines a solution to 1 or more problems. It defines *exactly 1 solution* (therefore as an attribute) to these problem(s). Any other solutions to (the same) problems are reflected in a separate best practice. The attributes *title*, *date* and *description* form the basic info of the BP.
1..* prob	A single best practice may SOLVE multiple problems.
1..* author	A best practice may be WRITTEN BY 1 or more authors.
1..* cat	A best practice may be CATEGORISED BY 1 or more categories.
0..* effort	A BP may have several efforts required for successful implementation.
0.. ex	A BP may be described by examples.
0..* cmt	Multiple comments may be made on BPs.
0..* rating	Multiple ratings may be given on various aspects of the BP.
0..* file	Multiple files may be linked to the BP to provide further explanation.
Category	Describes to which category/categories a BP belongs. A category is only described by a *name*.

Table J.1 – continued from previous page

1..* bp	The same category may be used for many best practices, but does not exist without a minimum of 1 BP defined for it.
Problem	Describes the problem that is solved by a BP. The problem is a textual instance and has a *name* and *description*.
0..* bp	Multiple BPs may solve the same problem. A problem might have no BPs defined to solve it yet.
Author	The author of the BP. Has a *name* and *contact info* (e.g. an e-mail address).
1..* bp	Multiple BPs may be written by the same author, but an author is only defined in the scope of a BP (hence the minimum cardinality of 1).
File	A file that is attached to the BP, such as an image or an external website. Has a *type* (such as image or survey results) and a *link* to the external source.
0..* bp	An external file can be linked to many best practices, but it might also be used in no BP at all.
Effort	The effort describes what is required for successful BP implementation. This could be a resource, such as "20 trainers".
1..1 bp	A specific effort is specified for only one BP, although many BPs may require similar efforts.
Example	An example implementation of the BP may be defined.
1..1 sln	The example is specified for the solution description in the BP.
Comment	Comments may be provided to BPs, that indicate the *commenter*, the *comment* and the *date* of commenting. Comments foster discussion and knowledge sharing.
1..1 bp	A comment is provided to only 1 BP.
Rating	Ratings may be given to BPs on various *dimensions*. A dimension example is "effectiveness". Ratings provide quick information on several BP aspects.
1..1 bp	A rating is provided to only 1 BP.
Metamodel	
Element	**Explanation**
Concept	Describes any concept, such as "best practice". Each concept has a *name* and *optionality* describing if the concept is mandatory or recommended (displayed in the core model) or user-defined.
1..* attr	A concept has at least 1 attribute, but possibly many.

Table J.1 – continued from previous page

1..* rel	A concept may be linked by either the start or end of a relationship (hence the double association). A concept has to be linked to another concept.
Attribute	Every concept has attributes. These have a *name* and *type* (e.g. "name: string") and an *optionality*.
1..1 concept	A specific attribute only exists in one concept.
Relationship	Links two concepts. Has a *type* (association or aggregation), role names for the start and end points (*rn_start* and *rn_end*), minimum and maximum cardinalities and an *optionality*.
1..1 start	A specific relationship only connects to one starting concept.
1..1 end	A specific relationship only connects to one end concept.

Figure J.1 shows the OBL metamodel with added constraints.

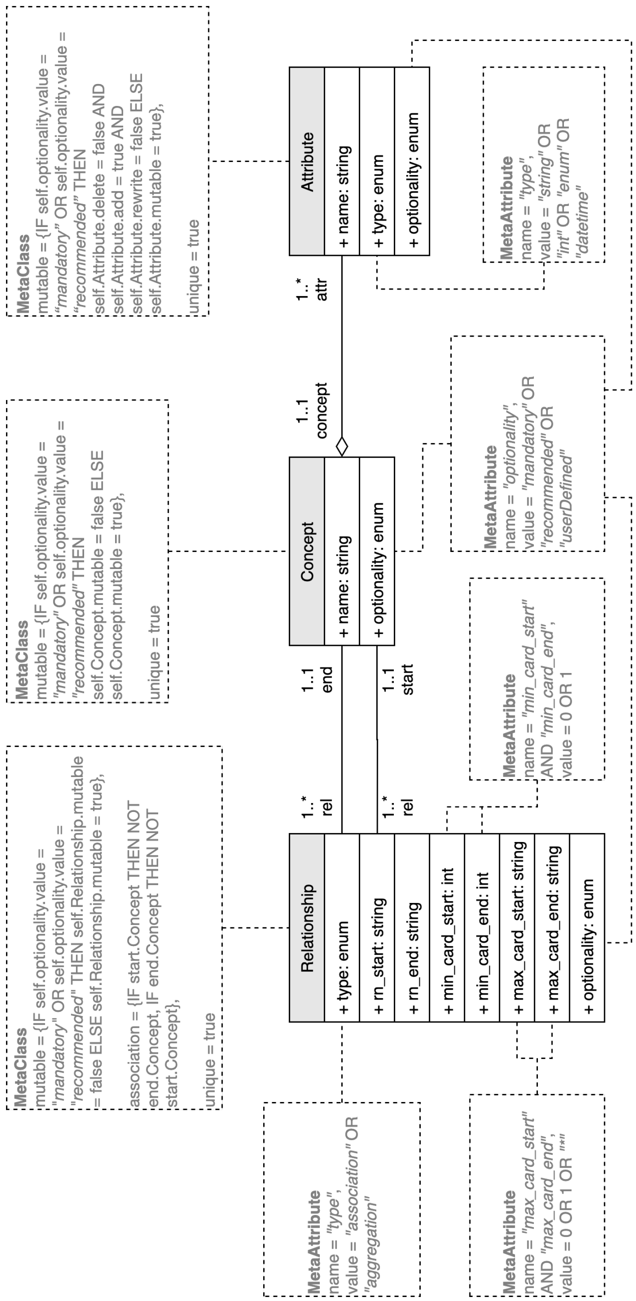

Figure J.1: OBL metamodel with constraints

J.0.2 Concrete Syntax Models

Full Core JSON Model

Listing J.1 shows the full core JSON model, in which all recommended concepts are also included. The indentation is decreased for legibility reasons.

```
{
"ADomain": {
"domainstate": {
        "displayfeature": false,
        "model": "string",
        "name": "A Domain",
        "administrator": "plomp1996@gmail.com",
        "bestpractices": {
                "bpdocument": {
                        "01grouptitle": "Best Practices",
                        "02groupdesc": "Enter basic best practice
                           info here.",
                        "1displayfeature": true,
                        "2title": "string",
                        "3description": "text",
                        "4author": [{
                                "name": "Written by",
                                "self": "document reference",
                                "related": "document reference"
                        }],
                        "problems": [{
                                "name": "Solves",
                                "self": "document reference",
                                "related": "document reference"
                        }],
                        "5solution": "text",
                        "6categories": ["string"],
                        "7date": "string",
                        "comments": {
                                "commentdocument": {
                                        "author": "string",
                                        "date": "string",
                                        "email": "string",
                                        "img": "string",
                                        "level": "int",
                                        "parent": "string",
                                        "text": "string"
                                }
                        },
                        "ratings": {
```

```
                "ratingdocument": {
                    "01grouptitle": "ratings title",
                    "02groupdesc": "ratings description",
                    "1displayfeature": false,
                    "2ratingtype": ["Stars"],
                    "3dimension": ["string"],
                    "4dimension description": ["string"],
                    "5scale": ["5"],
                    "6stepsize": ["1"]
                }
            },
            "examples": {
                "exampledocument": {
                    "01grouptitle": "Example",
                    "02groupdesc": "Describe an example here.",
                    "1displayfeature": false,
                    "2title": "string",
                    "3description": "text"
                }
            },
            "efforts": {
                "effortdocument": {
                    "01grouptitle": "Efforts",
                    "02groupdesc": "Define the effort required for this best practice here.",
                    "1displayfeature": false,
                    "2name": "string",
                    "3scale": "string"
                }
            }
        }
    },
    "users": {
        "userdocument": {
            "1displayfeature": false,
            "2email": "string",
            "3name": "string",
            "4role": "string"
        }
    },
    "authors": {
```

```
                "authordocument": {
                        "1displayfeature": false,
                        "2contactinfo": "string",
                        "3internal": "boolean",
                        "4name": "string"
                }
        },
        "problems": {
                "problemdocument": {
                        "01grouptitle": "Problem",
                        "02groupdesc": "Describe the problem here.",
                        "1displayfeature": true,
                        "2name": "string",
                        "3description": "text",
                        "bestpractices": [{
                                "name": "Solved by",
                                "self": "document reference",
                                "related": "document reference"
                        }]
                }
        }
}
}
}
```

Listing J.1: Full JSON core model

Full Textual Grammar

Listing J.2 shows the full textual grammar in EBNF.

```
(* JSON requires the correct number of closing braces to be valid *)
BP model = opening brace, domain name, colon, opening brace, domain state, best practice, comma, [ comment, 2 * closing brace ], comma, [ rating, 2 * closing brace ], [ example, 2 * closing brace ], [ effort, 2 * closing brace ], 2 * closing brace, comma, user, 2 * closing brace, comma, author, 2 * closing brace, comma, problem, 2 * closing brace, { collection }, 3 * closing brace, ‘;’;
domain name = quotes, text, quotes;
(* Elements of the core model. Since these elements are mostly fixed, we just include the JSON code here *)
domain state = ‘"domainstate": { "displayfeature": false, "name": ’, text, ‘ , "administrator": ’, text, ‘,’;
best practice = ‘"bestpractices": { "bpdocument": {’, group title, group description, ‘"1displayfeature": true, "2title": "string", "3description": "text", "4author": [{"name" : "Written by", "self
```

```
": "document reference", "related": "document reference"}], "
problems": [{"name" : "Solves", "self": "document reference", "
related": "document reference"}], "5solution": "text", "6
categories": ["string"], "7date": "string"';
problem = ‘"problems": { "problemdocument": {’, group title, group
description, ‘"1displayfeature": true, "2name": "string", "3
description": "text", "bestpractices": [{"name" : "Solved by", "
self": "document reference"}]’;
comment = ‘"comments": { "commentdocument": { "displayfeature":
false, "author": "string", "date": "string", "email": "string", "
img": "string", "level": "int", "parent": "string", "text": "
string"’;
rating = ‘"ratings": { "ratingdocument" : { "01grouptitle": "Ratings
", "02groupdesc": "Enter rating information here", "1
displayfeature": true, "2ratingtype": ["’, ratingtype, ‘"], "3
dimension": ["string"], "4dimension description": ["string"], "5
scale": [’, digit, ‘], "6stepsize": [’, digit, ‘]’;
example = ‘"examples": { "exampledocument": { "01grouptitle": "
Example", "02groupdesc": "Describe an example here.", "1
displayfeature": true, "2name": "string", "3description": "text";
effort = "effort": { "effortdocument": { "01grouptitle": "Effort",
"02groupdesc": "Define the effort required for this best practice
 here.", "1displayfeature": true, "2name": "string", "3scale": "
text";
author = "authors": { "authordocument": { "1displayfeature": false,
"2name": "string";
user = "users": { "userdocument": { "1displayfeature": false, "2
email": "string", "3name": "string", "4role": "string";
(* Can be filled in by the administrator, also for fixed elements *)
group title = quotes, ‘01grouptitle’, quotes, colon, quotes, text,
quotes, comma;
group description = quotes, ‘02groupdescription’, quotes, colon,
quotes, text, quotes, comma;
displayfeature = quotes, ‘1displayfeature’, ‘true’ | ‘false’, comma;
(* Documents always have a collection as a parent *)
collection = quotes, text, quotes, colon, opening brace, document, {
 document }, closing brace, [ comma ];
document = quotes, text, quotes, colon, opening brace, group title,
group description, displayfeature, attribute, { attribute |
collection }, closing brace, [ comma ];
(* We define two attribute types *)
attribute = quotes, text, quotes, colon, (quotes, string | text,
quotes) | docref, [ comma ];
string = ‘"string"’;
text = ‘"text"’;
docref = ‘[{"name": ’, quotes, text, quotes, ‘"self": "document
```

```
    reference", "related": "document reference"}]';
ratingtype = 'Stars' | 'Slider' | 'eBay' | 'binStars' | 'DislikeLike
    ' | 'Like';
opening brace = '{';
closing brace = '}';
quotes = '"';
colon = ':';
comma = ', ';
text = ( letter | digit ), { letter | digit | space };
letter = 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' |
     'K' | 'L' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U'
    |  'V' | 'W' | 'X' | 'Y' | 'Z' | 'a' | 'b' | 'c' | 'd' | 'e' | 'f
    ' | 'g' | 'h' | 'i' | 'j' | 'k' | 'l' | 'm' | 'n' | 'o' | 'p' | '
    q' | 'r' | 's' | 't' | 'u' | 'v' | 'w' | 'x' | 'y' | 'z';
digit = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9';
space = ' ';
```

Listing J.2: Full textual grammar

J.0.3 Graphical Representation of JSON Core

Figure J.2 shows the graphical representation of the JSON core model.

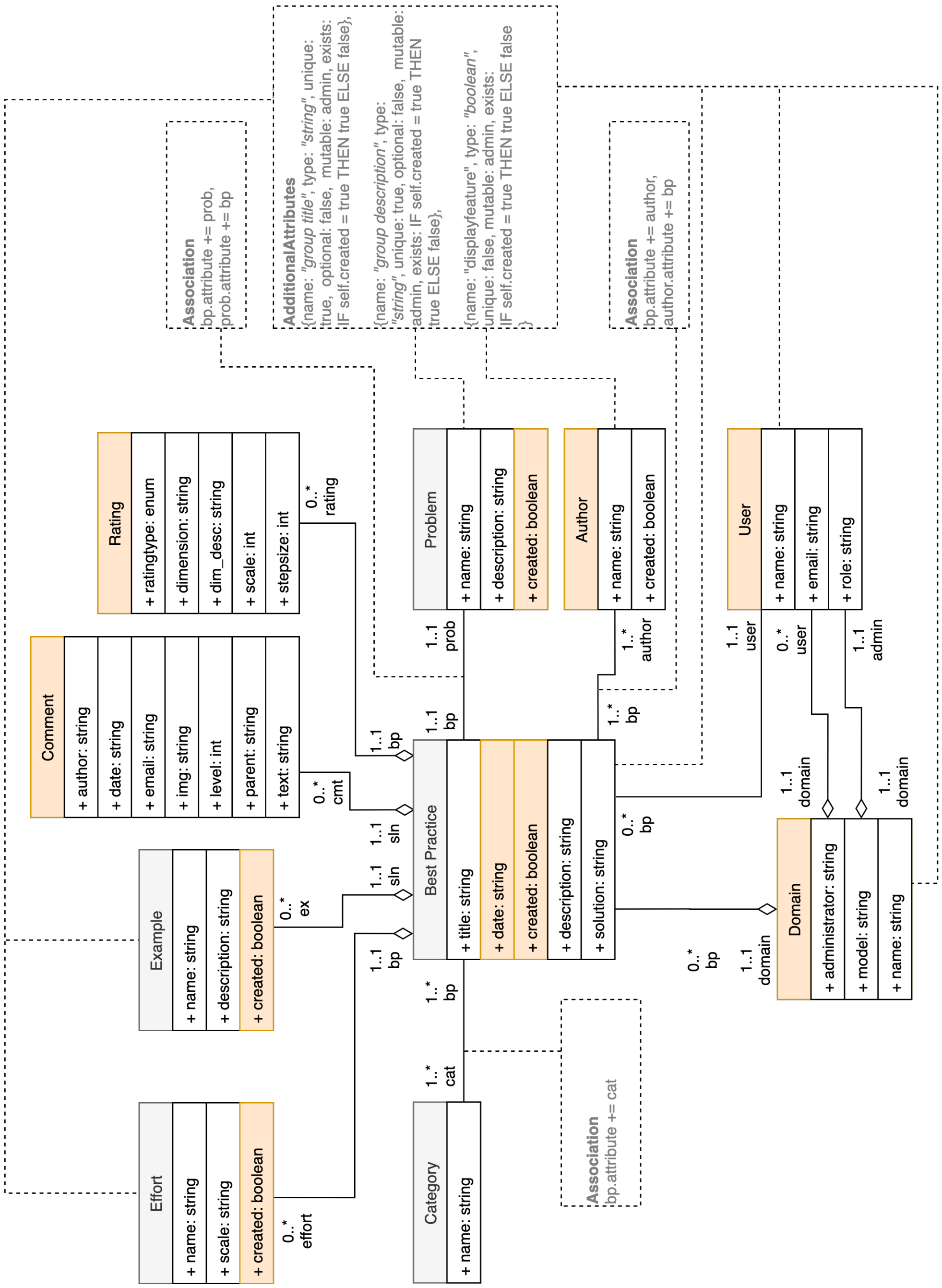

Figure J.2: Graphical representation of the JSON core model

K | Validation Method

This appendix lists the PDD describing the method for creating a family of validations for MDD approaches. We will explain the PDD with the following activity and concept tables.

Table K.1: Validation method PDD activities

Activities	
System description	**Before the decomposition of validation goals can start, we need to describe the system and its problem area. This information is the basis for some of the decompositions.**
Describe context	*The system operates in a certain context that influences the validation. This context will be defined.*
Define system problem description	The problem description defines what the system is intended to solve; i.e. the reason for development.
Define stakeholders	Stakeholders use the system and/or are influenced by the system. Their needs, goals and desired need to be validated.
Describe system	*We describe the problem the system is intended to solve and the system's components.*
Describe system components	Give an overview of the various components that make up the tool.
Define system interactions	Stakeholders interact with the system. These interactions define how the system is used and influence what needs to be validated.
Define requirements	Requirements can be defined through various complex processes, but we do not provide guidelines for these processes in our method. Hence the closed complex activity.
Decomposition	**Once the basis of the validation has been established, we can decompose high-level goals to validation methods.**
Define high-level goals	High-level goals are goals that the system aims to achieve. They indicate solutions to problems observed in the real world.

Table K.1 – continued from previous page

Map high-level goals	High-level goals are placed on the canvas and linked to the system element.
Define stakeholder goals	We need to define how stakeholders are affected by these high-level goals.
Define stakeholder goals	Stakeholder goal themes are groupings of goals for a stakeholder based on a common overarching goal. They are based on requirements
Map stakeholder goals	Stakeholder goals are placed on the canvas and linked to the respective high-level goals that they contribute to.
Define tool-supported goals	*Tool-supported goals indicate how the tool components can help stakeholders achieve their goals.*
Define implementation status	If the tool is still (partly) in development, not all components may have been implemented yet. We plan validations for all (planned) components, but we wish to distinguish between currently possible validations and other planned validations. Defining the implementation status is required to make this distinction. It is a a closed complex activity since it also requires changes to the stakeholder goal decomposition to be made.
Define tool-supported goals	The goals that the system components contribute to should be determined.
Map system components	Mapping all (planned) system components to the stakeholder goal decomposition to further decompose to tool-supported goals.
Define validation aspects	*Defining which aspects to validate for the tool-supported goals and system components.*
Pick quality framework	A quality framework must be picked based on our recommendations or any other source.
Assess quality framework	The quality framework must be assessed for fitness for purpose based on criteria we provide.
Adapt quality framework	If the quality framework is not sufficient, the quality framework needs to be adapted. We provide guidelines for the construction/adaption of quality frameworks.
Map quality framework	The aspects in the quality framework need to be mapped to the tool-supported goals so that we can further decompose. The result will be the validation aspects for the various tool-supported goals and system components.
Define resources	*Resources need to be defined so that we can motivate why components and tool-supported goals are (not) validated.*
Define resources	We need to define the resources we have, such as time and available study participants.

Table K.1 – continued from previous page

Map resources to validation aspects	We decompose the tool-supported goals based on whether or not we have the resources available for validation.
Protocols	**Validation method protocols are made for all validation methods, whether they are planned for execution or future execution.**
Define validation method design	The protocols for validation activities need to be defined. This process differs based on the goal and aspect of validation.

Table K.2: Validation method PDD concepts

Concepts	
PROBLEM DESCRIPTION	Describes the problems observed in the real world that the system aims to solve.
STAKEHOLDER DESCRIPTION	Describes the stakeholders of the system.
BASIC DESCRIPTION	A short and non-formal description of stakeholders.
PERSONA	A more extensive stakeholder description. Guidelines can be used for creating a PERSONA.
STAKEHOLDER	A person influenced by the system and described in the STAKEHOLDER DESCRIPTION.
COMPONENT OVERVIEW	Defines which components make up the system. Can be modeled using either a COMPONENT DESCRIPTION or a FAM.
COMPONENT DESCRIPTION	Describes the system components using a non-formal notation in natural language.
FAM	Is a functional architecture model. Shows a high-level overview of system components and their interaction.
COMPONENT	An element of the SYSTEM. Described in the COMPONENT OVERVIEW.
INTERACTION DESCRIPTION	Describes how the system is used by a STAKEHOLDER or influences a STAKEHOLDER. A STAKEHOLDER interacts with components that have been defined in the COMPONENT OVERVIEW.
REQUIREMENTS DOCUMENT	Document that outlines the goals of the various stakeholders for the system. Can include a USER STORY collection or a GOAL MODEL.
USER STORY	A textual representation of a system requirement, written from the perspective of the stakeholder, using a fixed structure.

Table K.2 – continued from previous page

GOAL MODEL	Graphical representation of STAKEHOLDER goal using a modeling language like iStar.
REQUIREMENT	Shows the goal of a STAKEHOLDER for the system. Is included in a REQUIREMENTS DOCUMENT.
HIGH-LEVEL GOAL	Defines why a SYSTEM is developed based on the PROBLEM DESCRIPTION. Has a scoping status that determines if validation of this element fits the research scope.
HIGH-LEVEL GOAL TRACE LINK	Links a HIGH-LEVEL GOAL to the SYSTEM on the CANVAS.
SYSTEM	The element describing the system on the CANVAS. Has a scoping status that determines if validation of this element fits the research scope.
STAKEHOLDER GOAL	A goal that a STAKEHOLDER has in using the system. Has a scoping status that determines if validation of this element fits the research scope.
STAKEHOLDER ANNOTATION	Annotates the STAKEHOLDER GOAL with a STAKEHOLDER to define for which STAKEHOLDER the goal is defined.
STAKEHOLDER GOAL TRACE LINK	Links a STAKEHOLDER GOAL to a HIGH-LEVEL GOAL on the CANVAS.
IMPLEMENTATION STATUS	Describes whether or not a system COMPONENT has been implemented yet and to what extent.
TOOL-SUPPORTED GOAL	Defines, based on the IMPLEMENTATION STATUS of a COMPONENT, what a goal for using the system is by using the various COMPONENTs. Has a scoping status that determines if validation of this element fits the research scope.
TOOL-SUPPORTED GOAL LINK	Links a TOOL-SUPPORTED GOAL to the STAKEHOLDER GOAL(s) it contributes to.
QUALITY FRAMEWORK ASSESSMENT	Results of assessing the fitness a QUALITY FRAMEWORK.
QUALITY FRAMEWORK	Specifies a set of QUALITY ASPECTs to validate for the system.
QUALITY ASPECT	An aspect that needs to be validated for the system. Has a scoping status that determines if validation of this element fits the research scope.
QUALITY ASPECT TRACE LINK	Links a QUALITY ASPECT to a PORT.

Table K.2 – continued from previous page

PORT	Indicates which COMPONENT is validated using a QUALITY ASPECT for a TOOL-SUPPORTED GOAL.
RESOURCE	Defines what is available for validation, such as time or subjects.
VALIDATION ACTIVITY	Defines what can/will be done to validate a QUALITY ASPECT for a TOOL-SUPPORTED GOAL.
VALIDATION ACTIVITY TRACE LINK	Links a QUALITY ASPECT to a VALIDATION ACTIVITY.
VALIDATION METHOD DESIGN	Defines how a VALIDATION ACTIVITY will be carried out.
GOAL	Can be any of the defined goals and is placed on the CANVAS.
TRACE LINK	Can be any of the defined trace links and is placed on the CANVAS.
CANVAS	Contains all elements of the decomposition tree.

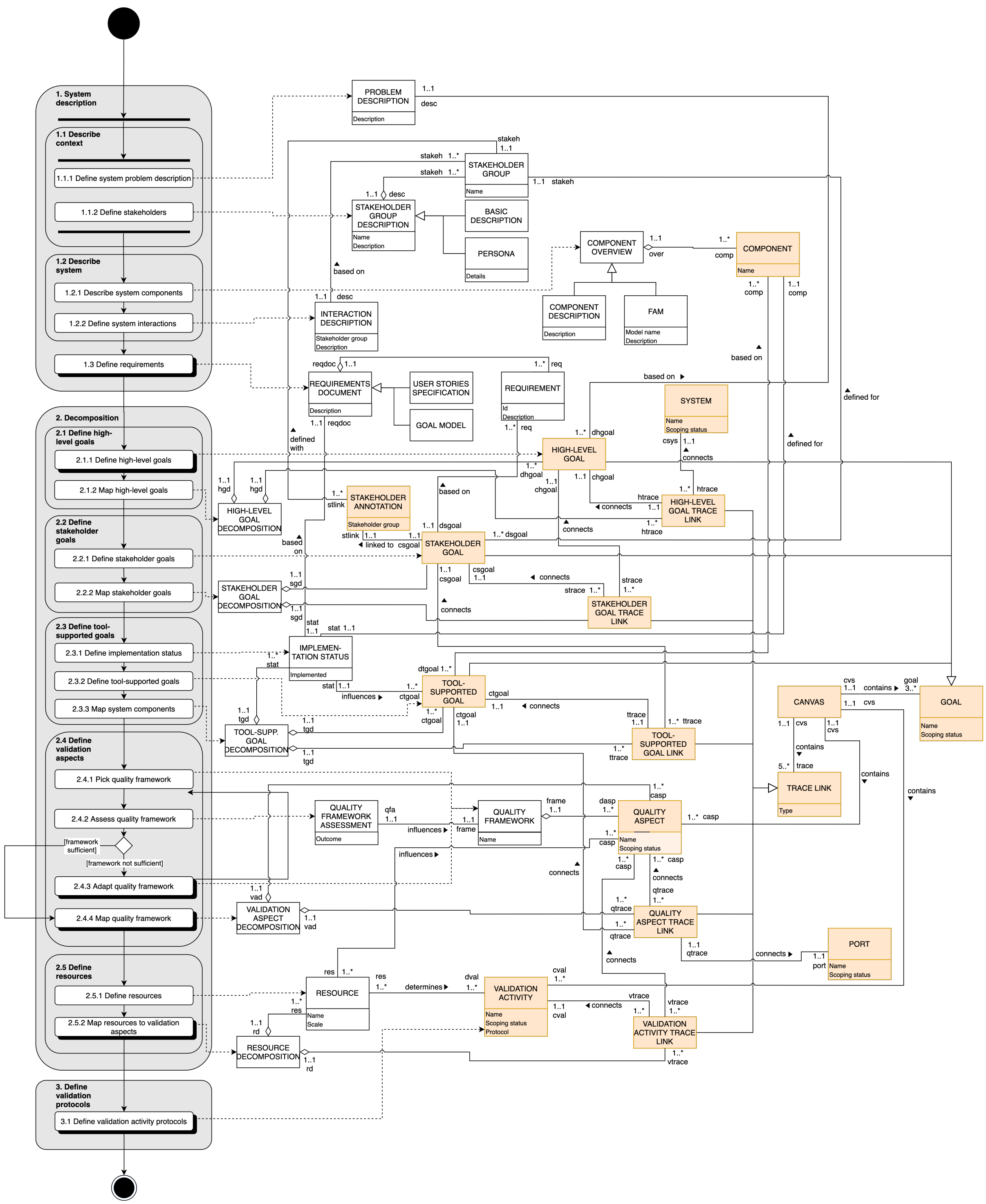

Figure K.1: Validation method PDD

L | Validation Case

In this appendix, the deliverables of applying our validation method to OpenBest are presented.

L.1 Context Description

L.1.1 Problem Description

So-called *responsible enterprises* go beyond legal obligations to act in accordance with ethical values that aim to benefit the environment and society. By taking full advantage of the development of modern ICTs, the field of *responsible software* aims to enable enterprises to transition to becoming responsible enterprises (or organisations) [48].

The shift from a regular organisation to a sustainable organisation often involves a continuous improvement cycle, with the outcome of the cycles resulting in organisational reengineering. Adèr [14] proposes the business ethics continuous improvement cycle (BECIC) as an improvement cycle that is specific to responsible enterprises. Although several interventions exist to guide organisations through this improvement cycle, these interventions currently only answer the question *what* needs to be done.

We aim to design and development of a model-driven repository for sustainability best practices that helps organisations uncovering not just *what* they can do to improve their ethical performance, but also *how* they can improve. This repository, called OpenBest, will support the process of improvement planning for ethical, social and environmental topics by inspiring the choice for improvement actions and defining the concrete action steps. The idea of best practice repositories is not new, but currently available repositories lack flexibility and extensibility: their contents are mostly created by single organisations and mostly accessible to single organisations. This leads to organisations being required to consult several different sources if they want to find best practices created by other organisations. We aim to centralize best practices by all types of organisations in a single repository by improving flexibility of the tool's functionality and to allow organisations adapt the functionality to their needs. When all types of organisations use the repository, knowledge sharing can be promoted among similar organisations, but also extend towards organisations in completely different domains.

L.1.2 Stakeholder Description

The stakeholders of OpenBest are employees of organisations in a domain. A domain is a group of related organisations. For example, a group of European dairy producers or a group of responsible technology start-ups. Within these organisations, two stakeholder groups exist that use OpenBest

User A user can be anyone within a domain that uses OpenBest to store and find best practices. A user is always part of an organisation. They want to find best practices to improve the ethical, social and environmental performance of their own organisation, but they also want to help other organisations by sharing their own organisation's best practices. They also want to create improvement plans that detail what their organisation aims to achieve, how they plan to achieve it and which best practices they will apply. They want to evaluate these improvement plans and the use of best practices.

Domain Administrator A domain administrator is a special type of user that determines for the domain what the structure of best practices is. The domain administrator makes sure that the functionality of the repository adapts well to the domain in question by prescribing this structure in a model. This model is used by the repository to determine which information is required from regular users when they want to store a best practice. A domain administrator is also always a user, but a user is not always a domain administrator.

L.2 System Description

L.2.1 FAM

In the figures below, the functional architecture models (FAM) of OpenBest are depicted. These show the high-level functionality of the system without providing unnecessary detail. The functionality is depicted using modules and sub-modules, their relationships, and the connection to an external system: the database platform.

The high-level FAM in figure L.1 shows the high-level overview of how the modules are related to each other. OpenBest includes an editor module; domain administrators prescribe the structure of best practices in their domain using a model. The interpreter reads these models and determines the feature set of the repository based on this model. The BP storage module and BP retrieval module are responsible for allowing users to store and view best practices. The improvement plan module handles functionality related to the construction of improvement plans.

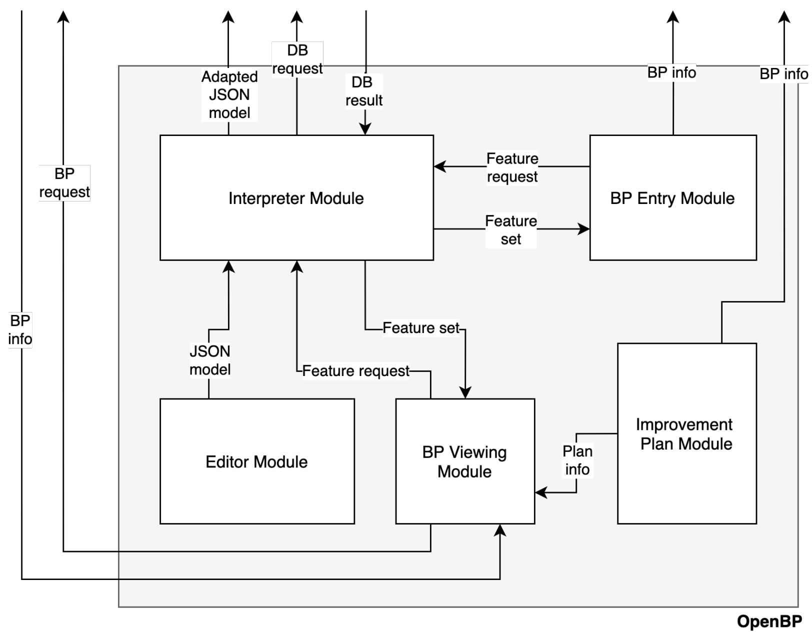

Figure L.1: High-level FAM of OpenBest

The BP entry module FAM in figure L.2 shows how a feature set is requested and received from the interpreter module. The feature set is dependent on what is stored in the database, which is an external service. The database structure is based on an adapted JSON model. The BP entry module also sends best practices to the database to be stored.

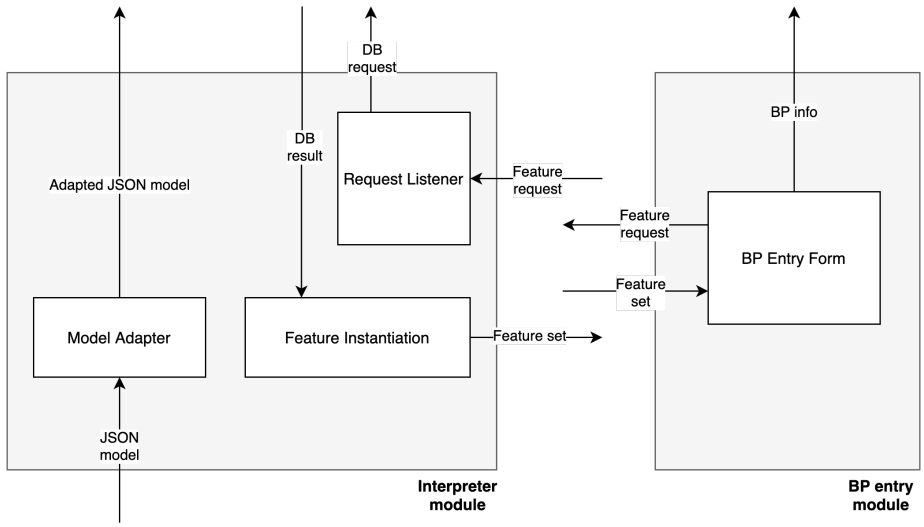

Figure L.2: Interpreter module and BP entry module FAM

The editor module in figure L.3 shows that, based on information entered in a form in the editor interface, a JSON model is compiled and send to the model adapter sub-module. The editor interface includes a core model; a fixed set of mandatory and recommended concepts that are present in every model that is created. Users can extend this model. To construct a JSON model based on the form information, the model compiler uses a textual grammar that specifies rules for construction.

The BP viewing module shows that a feature set is requested here as well, as there are features that enable the rating of best practices that are dependent on data in the database. Other functionality of the viewing interface includes filtering capabilities to find best practices.

The improvement plan module makes sure that users are able to construct improvement plans. Info on the applied best practices is stored in the database.

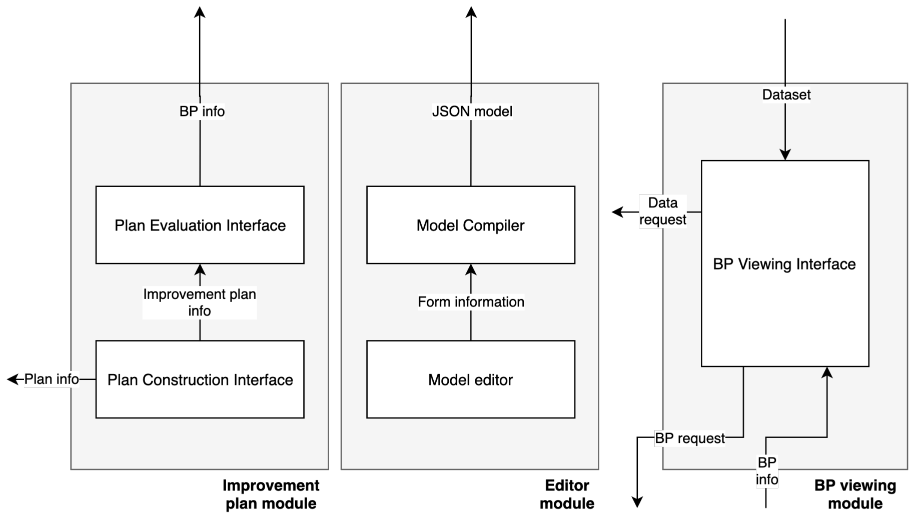

Figure L.3: Editor module, BP viewing module and improvement plan module FAM

L.2.2 Interaction Description

In this section, we define interactions for OpenBest and the two stakeholder groups. This also includes envisioned interactions with system components that are not implemented yet.

- Domain administrator
  - The domain administrator creates a *new domain* by logging in to OpenBest and opening the model editor. The administrator uses the form-based modeler to provide information on the structure of best practices. After the model has been saved, the domain will be instantiated.
  - The domain administrator updates the domain by opening the model editor and making the necessary changes. After the model has been saved, the best practice structure and corresponding features are updated for any *new* best practices (but not the best practices that are already stored).
  - The domain administrator creates a textual model by extending a *core model* of mandatory and recommended concepts.
- User
  - The user stores a best practice using the BP entry form.
  - The user searches for a best practice using filtering capabilities.
  - The user opens a best practice and reads its contents.
  - The user comments on a best practice they have found.
  - The user rates a best practice based on a factor that has been determined by the best practice author.

– The user searches for recommended best practices based on the contextual factors of their organisation and improvement plan.
– The user constructs an improvement plan and indicates which best practices are applied to interventions.
– The user evaluates the success of an intervention after it has concluded and thereby rates the best practice that has been applied.

L.3 Requirements

For brevity we refer to appendix H for an overview of the system requirements.

L.4 Decomposition Tree

The full decomposition tree can be seen in figure L.4.

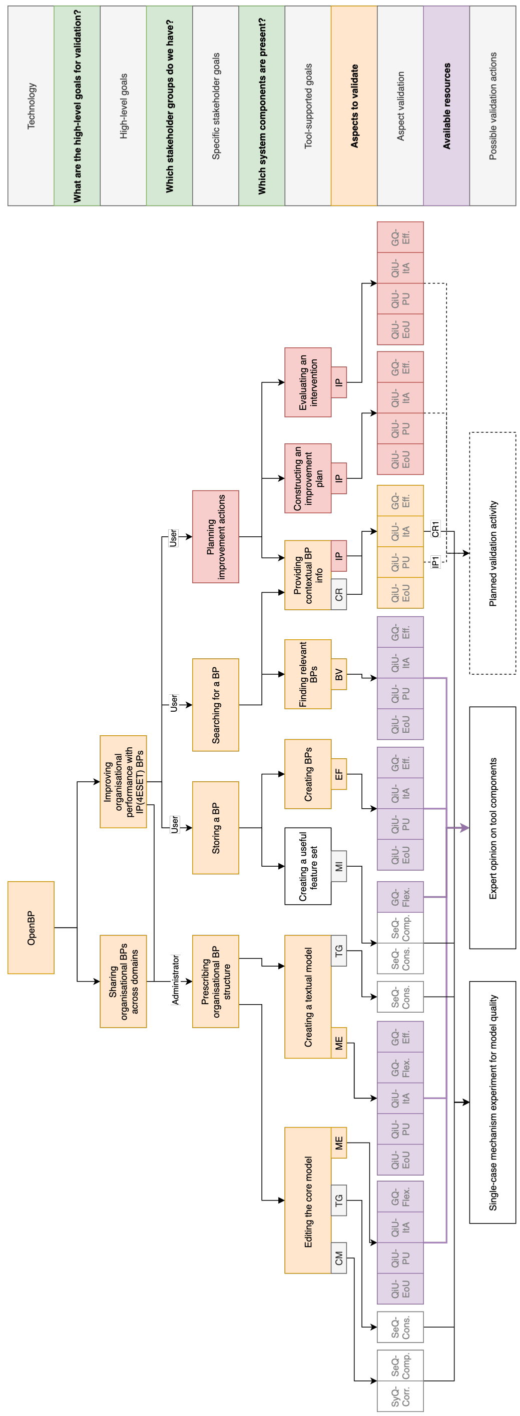

Figure L.4: Decomposition tree for OpenBest

L.5 Expert Opinion - Protocol

L.5.1 Protocol Context

This expert opinion session is used to validate the quality in use and goal quality of OpenBest. Since the tool has not reached enough maturity to be applied in a real-world setting, this validation method will be used to make estimations of the effects. This method is a useful approach to validating these quality aspects for a non-mature version of the tool. We extract expert opinion through interviews.

L.5.2 Variables, Subjects and Instrumentation

Variables With this expert opinion session, we aim to validate the following quality aspects of the use of the model editor, BP entry form and searching and viewing capabilities:

- Perceived ease of use
- Perceived usefulness
- Intention to adopt
- Effectiveness
- Flexibility

Subjects The subjects for this expert opinion session are practitioners working for organisations working on improving their ethical, social and environmental performance. Subjects may also be researchers with an expertise in IP4ESET.

Instrumentation The following instruments are used for this expert opinion session:

- Informed consent forms (section L.5.5)
- OpenBest
- Interview questions

L.5.3 Data Collection Procedure

Prerequisites

Pay attention to the following prerequisites before starting the interview:

- The participant has signed the informed consent form.
- A BP has been stored already to save time.
- Screens with the model editor form and BP entry form filled in have been opened on separate tabs.

Step 1: Explain Intention of Interview

For this research we have created a repository of best practices. A best practice is a solution to an organisational problem. These best practices created by organisations can be stored in this repository. The goal is to let organisations learn from each other's best practices. This interview is meant to validate the added value of this repository as a contribution to improving organisations' ethical, social and environmental performance. We do this by demonstrating the current status of the repository and asking your opinions on several aspects. You can elaborate on your answers, but we may interrupt you when time is running short.

The repository is still under development. We ask you to make estimations about what the effects would be if you were to use this repository in your organisation.

Step 2: Setting Context

Imagine that your organisation could use this repository to store your best practices (approaches to solving problems related to social, environmental and ethical topics) in a centralised location. This repository will also be used by similar organisations, so that you can learn from each other's best practices. These organisations are in the same domain: a group of organisations that use the same version of the repository.

> **Adapt based on interviewee**: This could be, for example: *[provide relevant example for this organisation]*

Since every domain may want to store different pieces of information related to their BPs, the repository can be adapted and tailored to each domain's needs.

Step 3: Model Editor

Prerequisites

- The model editor has been opened on a separate screen with information already filled in to save time.
- The filled in information (such as the name of the domain) is relevant to the organisations that we are interviewing.

Explain

- This form is only accessible to the domain administrator: the person within this domain or organisations responsible for determining the elements that make up a BP.
- When any other user wants to store a BP, they have to fill in the information about the concepts that the administrator defines with this form.
- For example, when a domain administrator says that information should be provided about a concept called Lessons Learned, users have to fill in that information.
- The administrator may choose to update the model at any time, without it affecting BPs that are already stored.

- There are some core elements that should always be present and some elements that we recommend. These are visible in the model editor.

Demonstrate

1. Adding a concept
   - Demonstrate the addition of a **Lessons Learned** concept.
   - Add the group title and description. Explain what these do.
   - Add a Description attribute.
   - Add a relationship.
     - The subcollection is: *users*.
     - The name of the relationship is *Added by*.
2. Creating the model
   - Explain that eventually, the contents of this form are automatically used to determine what users need to fill in.

Questions

Perceived ease of use

1. Do you find the use of the model editor easy to follow? Do you think you could use it?
2. Do you understand how this model editor can be used to determine the inclusion of elements from the core model?

Perceived usefulness

1. Overall, do you find the use of the model editor useful?
2. Do you think that it is useful to have a set of core elements to extend?
3. Do you think that using a form like this is a useful method for prescribing the content structure of BPs?
4. Do you think that prescribing organisational BP structure using a model editor is a useful solution to the problem of variability in the organisational BP domain?

Intention to adopt

This question will not be presented to surrogates, only to possible end-users.

1. I (or someone in my organisation) would use the model editor to prescribe the structure of our BPs.

Flexibility

1. Do you think that this approach provides enough flexibility to prescribe the elements that make up a BP?

Effectiveness

1. Do you think that the use of the model editor is an effective way to prescribe the elements that make up a BP?

Step 4: Model Interpreter and Entry Form

In this step, both the model interpreter and BP entry form are validated, since the former produces visible output to the BP entry form.

Prerequisites

- The BP entry form has been opened on a separate screen with information already filled in to save time.

Explain

- The features in this form depend on what we have just entered in the model editor.
- This form can be used by any user within the domain (i.e. any user within any organisation).

Demonstrate

- The difference in features.
- Adding a second category.
- Pay attention to the **Lessons Learned** concept. This is what we have just added.

Questions Model Interpreter

Flexibility

1. Do you think that the features that are created provide enough flexibility to be useful for any organisation?

Questions Entry Form

Perceived ease of use

1. Do you find the use of the entry form easy to follow? Do you think you could use it?

Perceived usefulness
1. Overall, do you find the use of the entry form useful?
2. Do you think that entering a BP using a form is a useful method for storing BPs?

Intention to adopt
1. I (or someone in my organisation) would use the entry form to store BPs to be shared with other organisations.

Effectiveness
1. Do you think that multiple organisations using this form to store BPs so they can be accessed by other organisations is an effective way to improve organisational performance?

Step 5: Search and View

Prerequisites

- A BP has been stored in the database already.

Explain

- Once we have stored our BP, we can see it in a list of BPs.
- In this list, all other BPs created by other users in this domain are also visible.

Demonstrate

1. The various options to filter BPs.
2. Open a BP. Show all elements and point to what we have just filled in.
3. Show the comments and ratings section and explain that this may foster discussion between users from various organisations.

Questions

Perceived ease of use
1. Do you find the search and viewing functionality easy to follow? Do you think you could use it?

Perceived usefulness
1. Overall, do you find this functionality useful?
2. Do you think that the current searching and viewing functionality is useful to find relevant BPs? Are you missing anything?

Intention to adopt

1. I (or someone in my organisation) would use this functionality to find relevant BPs, possibly created by other organisations.

Effectiveness

1. Do you think that this functionality is an effective way to find relevant BPs and do you think that would lead to improved performance?

Conclude the Interview

Thank the participant and ask for any final remarks. Stop the audio recording.

L.5.4 Validity Evaluation

Conclusion Validity In order to make sure that the conclusions we draw from the expert opinion sessions are reasonable, we separate the session by demonstrating and asking questions about each component individually. Opinions expressed about one system component will therefore unlikely be influenced by opinions on other system components. We also ensure that we do not guide any answers towards a particular direction and only asks subjects to elaborate.

Internal Validity To mitigate threats to internal validity, we arrange different meetings with every subject. We also ensure that we follow the same protocol for every session.

Construct Validity To ensure construct validity, we do not explicitly mention to subjects what we aim to study; we only mention that we would like their opinion on some of the tool components.

External Validity In order to mitigate threats to external validity, we include participants with various backgrounds. All are representative of the target group of users, but apply best practices and IP4ESET practices in general in distinct ways.

L.5.5 Informed Consent

INFORMED CONSENT
Study working title:
Designing a Model-Driven Repository for Sustainability Best Practices

Introduction
You are asked to participate in an interview as part of ongoing research at the Utrecht University on ICT for sustainability. We aim to validate the contribution of a repository of best practices to the field of improvement planning for ethical, social and environmental topics. To this end, an interview of approximately 30 minutes will be conducted where we ask you questions about the repository we have developed.

By sharing your experience and opinion, you are contributing to this body of knowledge. If you want, we will share the results with you. You are able to drop out of this research at any time. This consent form is necessary for us to ensure that you understand the purpose of your involvement and that you agree to the conditions of your participation.

Responsible for this research are Milo Plomp and Sergio España. You can contact us at:

- Main researcher: Milo Plomp (m.plomp2@students.uu.nl)
- Supervisor: Sergio España (s.espana@uu.nl)

Please indicate your choice for the following questions:

☐ **I give permission for the researchers to undertake audio recording during the interview. The audio files are only accessible to the main researchers and will be destroyed after transcribing.**

Please tick the following boxes for agreement:

☐ **I know that participating is completely voluntary. I know that at any moment I can decide not to participate anyway. I do not have to give a reason for that.**

☐ **I understand that the research data, without any personal information that could identify me, may be shared with others.**

☐ **I give permission to keep the collected data for at least 10 years after the end of this investigation.**

PARTICIPANT SIGNATURE

Name ______________________________

Signature ______________________________

Date ______________________________

RESEARCHER SIGNATURE

Name ______________________________

Signature ______________________________

Date ______________________________

L.6 Testing - Protocol

L.6.1 Protocol Context

We run tests on OpenBest as a form of single-case mechanism experiment where we construct test scenarios and observe OpenBests behaviour in these scenarios. This allows us to research the mechanisms that allows OpenBest to contribute to goals.

L.6.2 Variables, Subjects and Instrumentation

We test several quality criteria for the creation of textual models (as extensions of the core model). The following criteria are assessed:

- *Correctness*: does the created model adhere to its grammar specification (the textual grammar)? Are there any statements in the model that are not part of the language?
- *Completeness*: does the created model contain all valid and relevant elements specified in the core model, as well as the information from the model editor?
- *Semantic consistency*: does the model editor, with the textual grammar rules, create models that are semantically consistent?

These quality aspects are tested on models created with the model editor.

L.6.3 Data Collection Procedure

CREATING TEXTUAL MODELS

The following list provides information on the creation of a model set that is suitable for testing the quality aspects:

- Use the model editor to create models
- Ensure that all functionality of the model editor is used and can be assessed
- Ensure enough coverage of at least the following scenarios:
  - Use a varying selection of recommended concepts
  - Create concepts with a varying number of attributes
  - Create concepts that contain a varied amount of sub-concepts on various hierarchy levels
  - Create concepts with references and concepts without references

Overview of the input values An overview of the concept, attributes and relationships entered in the model editor should be created. This overview should also list the placement of sub-concepts in the hierarchy level by placing them under their parent concepts. Table L.1 shows the structure used to document the input values of the model editor.

Table L.1: Structuring model editor data

#	Element	Element info
1	**Concept**	**-**
1	Attribute 1	Text
1	Attribute 2	String, Array
1	*Relationship name*	*Related concept*
1.1	**Sub-concept**	**-**
1.1	Attribute 1	String
...	...	...

Assessing Correctness Output models are correct if they adhere to their grammar. In this case, the textual grammar of OBL. We use the textual grammar as a guide for the construction of textual models. Any inconsistency of output models in relation to the textual grammar is documented with the following information:

- Description of the error
- Error type: e.g. missing element, superfluous element, order error
- Concept related to the inconsistency: mandatory concept, user-defined concept, sub-concept
- Element type of error: concept, attribute, relationship

Assessing Completeness Output models are complete if all elements from the model editor form are included in the output model. Completeness is checked using the information that is documented using the structure in table L.1.

Assessing Consistency Output models are semantically consistent if every form element always translates to the same output. Consistency is validated by checking similarities between the documentation of the model editor input and assessing the consistency of the corresponding output model chunks. We check the consistency of translation for the following model editor information:

1. Name and values of attributes
2. Structure of relationships and inclusion of relationship names
3. Placement of concepts in the hierarchy
4. Ordering of attributes

CREATING FEATURE SETS

The models that have been created using the model editor are uploaded to OpenBest and used to instantiate a feature set.

1. Create an overview of the elements in the textual models that should be interpreted as features

- Structure this overview by grouping the elements under the concept for which they should be displayed

2. Upload the model to OpenBest
3. Assess if features are created for all elements (*completeness*)
4. Describe the instantiated feature and compare results with feature sets instantiated for other models (*semantic consistency)*

L.6.4 Validity Evaluation

There are less contextual factors that influence the validity of this method compared to the expert opinion sessions, as we are not concerned with subjects in this method. We mainly have to mitigate threats to internal validity. We do this by following this protocol for every test. Threats to external validity are hard to mitigate with the currently available resources, as we are limited to validating textual models created by the researcher only. This introduces a bias in the construction of textual models. The protocol for the planned case study in the next section aims to mitigate these threats to validity.

L.7 Case Study - Protocol

L.7.1 Protocol Context

This is a planned protocol; it will not be executed in the current research scope due to a restriction on resources, as well as non-implemented system components. This protocol is therefore created on a higher abstraction level than the previous protocols. It will not include a validity evaluation.

With this case study, we aim to validate a more mature version of the system in a real-world context. The method can be applied to currently non-implemented system components (related to the construction of improvement plans) as well as system components that are (partly) developed but require further validation (and possibly further development).

A case study may be used with various levels of rigour. For example, a case study per organisation may be conducted within one day. The added benefit of this approach over the testing and expert opinion methods is its application in a real-life context. However, a more rigorous approach would be to let organisations work with OpenBP over an extended period of time. Currently, the contribution of OpenBP to improving organisational performance is difficult to validate as this impact is not directly visible. By letting multiple organisations work with OpenBP over an extended period of time, we are more likely to make well-supported claims about OpenBP's contribution to this goal.

L.7.2 Variables, Subjects and Instrumentation

Variables This case study will validate quality aspects related to *semantic quality*, *syntactic quality quality in use* and *goal quality* of all system components:

- Syntactic correctness
- Semantic completeness
- Semantic consistency
- Perceived ease of use
- Perceived usefulness
- Intention to adopt
- Flexibility
- Effectiveness

The first three quality aspects will be determined using a similar approach as in the previous two protocols, but the data used for analysis will now originate from the application of the system in a real-world context. The latter quality aspects can be assessed using surveys and interviews to extract expert opinion. Multiple available subjects would also allow for group discussion.

Subjects The subjects of this case study will be practitioners in organisations working on improving their ethical, social and environmental performance. It is crucial that these subjects are realistic end-users of the system. We wish to study the artifact-context interaction in a setting that is as realistic as possible.

This case study should be conducted in multiple organisations in order to be able to generalise the results. The subjects should have some understanding of the process of improvement planning and/or structure of best practices in their organisation. Different subjects may be used for the various system components.

Instrumentation The instrumentation for this case study is partly based on [14], where a similar case study has been conducted. *This instrumentation is based on conducting a more simple case study, rather than the extended version described in the context description.*

- The OpenBP system
- A schedule for the case study
- Informed consent forms
- Pre-task training for using the system
- Guidelines for the experimental task
- Post-task survey
- Structure for documenting remarks made during the experimental task

L.7.3 Data Collection Procedure

The following procedure provides a high-level overview of steps that should be taken for data collection for the simple version of the case study.

1. The case study should start with introducing the context of the case study: why are we conducting a case study and what is the goal?
2. Subjects should be trained in using OpenBP. This includes all system components that will be used for validation, such as the model editor and the BP entry form.
3. The experimental task should be prepared and outlined. Subjects should be asked to complete a predetermined task.
4. The researcher will document any comments made during the experimental task.
5. After the experimental task, a survey and interview will be used to extract expert opinion on the use of the tool.
6. When multiple subjects are available, a focus group discussion to determine expert opinion may be conducted.