Complex Systems Informatics and Modeling Quarterly (CSIMQ)
eISSN: 2255-9922
Published online by RTU Press, https://csimq-journals.rtu.lv
Article 187, Issue 34, March/April 2023, Pages 1–29
https://doi.org/10.7250/csimq.2023-34.01

The OpenESEA Modeling Language and Tool for Ethical, Social, and Environmental Accounting

Vijanti Ramautar[1⋆] and Sergio España[1,2]

[1] Department of Information and Computing Sciences, Utrecht University, Princetonplein 5, Utrecht, 3584 CC, The Netherlands

[2] Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València, Camí de Vera, 46022 València, Spain

v.d.ramautar@uu.nl, s.espana@uu.nl

Abstract. Assessing business operations' ethical, social, and environmental impacts is a key practice for establishing sustainable development. There is a multitude of methods that describes how to perform such assessments. Often these methods are supported by an ICT tool. In most cases, the tools are developed to support a single method only and do not allow any tailoring. Therefore, they are rigid and inflexible. In this article, we present a novel model-driven approach for alleviating managerial issues that arise as a consequence of the complex landscape of ethical, social, and environmental accounting methods and tools. We have developed an open-source, model-driven tool, called openESEA. OpenESEA parses and interprets textual models, that are specified according to a domain-specific language (DSL). We have performed another iteration of the DSL engineering process, which is in line with the design science paradigm. We have validated the new DSL version by means of a user study. As a result, we present a new version of the openESEA modeling language and interpreter. The results of the user study with regards to performance, perceived usefulness, and perceived ease of use of modeling language are encouraging and provide us with a basis to continue developing new versions with more functionalities. The contributions of this work include a new version of the modeling language, a new version of the interpreter, knowledge surrounding the development of these artifacts, and a protocol for evaluating the quality of textual DSLs. The modeling language and interpreter are relevant for sustainability practitioners and consultants since our tool support has the potential to reduce redundancy in ethical, social, and environmental accounting. Our work is valuable to researchers that aim to assess and reduce the complexity of their modeling languages.

Keywords: Organizational Sustainability, Model-Driven Engineering, Domain-Specific Language, Modeling Language Complexity, Ethical, Social, and Environmental Accounting, Sustainability Reporting.

⋆ Corresponding author

Reference: V. Ramautar and S. España, "The OpenESEA Modeling Language and Tool for Ethical, Social, and Environmental Accounting," Complex Systems Informatics and Modeling Quarterly, CSIMQ, no. 34, pp. 1–29, 2023. Available: https://doi.org/10.7250/csimq.2023-34.01

Additional information. Authors ORCID iD: V. Ramautar – https://orcid.org/0000-0002-3744-0013, S. España – https://orcid.org/0000-0001-7343-4270. PII S225599222300187X. Article received: 15 February 2023. Accepted: 22 March 2023. Available online: 30 April 2023.

1 Introduction

Clients, business partners, citizens, and other stakeholders are more and more interested in the ethical, social, and environmental (ESE) performance of organizations [1]. These stakeholders put pressure on organizations to disclose their sustainability reports publicly. A poor sustainability reputation can even have financial consequences for companies [2]. Though this increase in interest in sustainable business practices has led to a rapid increase of the number of social and not-for-profit enterprises worldwide [3], it has also resulted in a surge in greenwashing; the attempt to capitalize on the growing demand for products and services that are produced by ethically, socially, and environmentally responsible organizations. It becomes increasingly difficult for businesses to be noticed by "green" consumers in the ecosystem of businesses that make false marketing claims about their sustainability performance [4]. The integration of ethical, social, and environmental accounting (ESEA) with traditional financial accounting can help responsible organizations differentiate themselves from greenwashers.

To assist responsible entities with their integrated reporting, this article presents a model-driven approach for ESEA. During the ESEA, organizations assess their performance in material (i.e., relevant) ESE topics [5], [6]. To do this, first, an ESE accountant collects data from organizational stakeholders via surveys or extracts it from information systems. Examples of such data (i.e., direct indicators) are the monthly water consumption or the number of people with an ethnic minority background in managerial positions. This data allows for calculating indirect performance indicators such as the percentage of managers with an ethnic minority background and the annual water consumption. We refer to the set of indicator values collected by conducting an ESEA process as ESE account. Parts of this account are typically published in a sustainability report.

Several factors make ESEA methods complex from the process and ICT perspectives. The ESEA domain abounds with methods, which are supported by ICT tools. Most of these tools are rigid and can solely be used to assess the single ESEA method they were developed for [7]. However, ESEA methods usually overlap in the indicators they require input for; e.g., many ESEA methods ask for the number of full-time workers, percentage of renewable energy, and greenhouse gas emissions. We found that numerous organizations apply multiple ESEA methods, so given the rigidity of the tools, these organizations end up having to use several disconnected tools that ask them for the same data. This phenomenon can be observed in Table 1, where we list a subset of ESEA methods and tools. The table shows that the tools can only assess the methods that they were developed for. The tools cannot be extended to support additional methods or new indicators. Furthermore, we found that organizations often like to extend or tailor the methods to their needs but tools do not allow it. For instance, the B Impact Assessment does not assess the diversity of employees in the organization. If a company wants to assess diversity (e.g., the number of women, men, and non-binary employees) the company has to use a different tool for this because the B Impact Assessment tool does not allow adding indicators.

To reduce the complexity of managing (i.e., defining and applying) ESEA methods, we are engineering the openESEA framework. It consists of a domain-specific language (DSL) that allows modeling ESEA methods and an interpreter tool that allows organizations to execute the methods. At the core of the framework lies the openESEA metamodel, which serves as an ontology that defines the main primitives of ESEA methods. The metamodel also constitutes the abstract syntax of the openESEA DSL. The concrete syntax is specified with textual grammar that is used to model ESEA methods. We have operationalized the framework by means of an open-source, web-based, model-driven tool called openESEA. It can be configured by loading a textual model of an ESEA method; then the tool will automatically support the method application. Our approach allows organizations to not only apply existing ESEA methods, but also extend methods, combine them, or create new ones from scratch, using the DSL, without having to worry about developing or updating the tool support.

Table 1. The tight coupling between ESEA methods and their tools

	Methods						
Tools	B Impact Asessment	Common Good Balance Sheet	STARS	CDP	XES Social Balance	Measurabl	University Sustainability Assessment Framework
B Impact Assessment web-application	✓						
Common Good Calculator		✓					
STARS Reporting Tool			✓				
CDP guidance tool				✓			
Ensenya el cor					✓		
Measurabl platform						✓	
UniSAF spreadsheets							✓

In earlier work, we presented the first version of our framework [7]. Since then, we have made improvements to the DSL and the interpreter. Regarding the DSL, we have extended the initial version with primitives to model surveys, and we have switched from an Extended Backus Naur Form (EBNF) grammar with no editing tool support to an implementation in Eclipse Xtext [8] with its corresponding model editor. Regarding the interpreter[3], we have made several improvements to the technology stack, which are highlighted in Section 6. The changes to the openESEA framework up until the current version (V2) are shown in Figure 1.

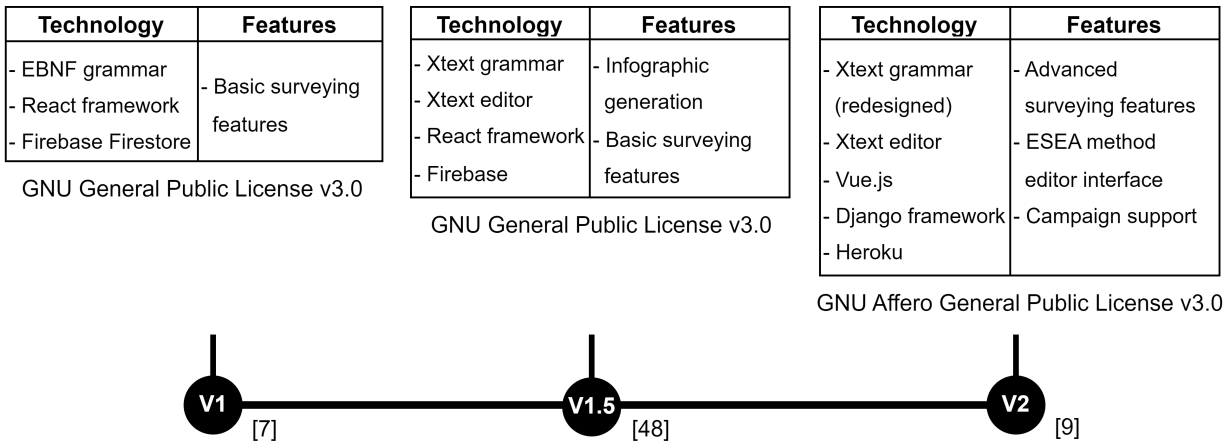


Figure 1. An overview of the changes in the openESEA framework up until V2

While in the following sections, we touch upon all improvements, the article places the focus on the engineering and evaluation of the modeling language for specifying ESEA methods. We explain the DSL development process and the most important primitives of grammar, and we identify potential improvements in the grammar through a user test. Thus, the two main research questions that are answered in this article are:

1. What modeling primitives are needed to specify ESEA methods that are intended to assess organizational sustainability?
2. How can the complexity of modeling ESEA methods be assessed and, if possible, reduced?

[3] Older version: https://github.com/sergioespana/open-sea; newer: https://github.com/sergioespana/openESEA

The first question aims to find a balance between the expressiveness and complexity of the DSL. Whereas the second question is to define a protocol for evaluating the complexity experienced by modelers and to identify potential improvements of the DSL. The contributions of the article are (i) new, more mature versions of the DSL and (ii) the model interpreter, which practitioners and researchers can use to assess organizational sustainability; and, also, (iii) an evaluation of the DSL through user testing. The overall research objective is to reduce the complexity of using the openESEA DSL to model ESEA methods. Herein we extend a paper presented during the 7th Workshop on Managed Complexity (ManComp 2022) [9], by means of providing more extensive theoretical background and in-depth explanations of the modeling language primitives, the usage of the modeling language and the protocol for evaluating the modeling language.

In Section 3 we present the research questions and research method. Section 2 contains the conceptual background on ESEA and related work on managing the complexity of modeling languages. Section 4 explains how we decided on the extensions of the modeling language. In Section 5 we present the DSL (consisting of the metamodel and textual grammar) for specifying ESEA methods. The usage of the modeling language is explained in Section 6. Section 7 explains the user test protocol, its results, and potential improvements of the grammar. The main findings, limitations, and future work can be found in Section 8. At last, this article concludes in Section 9.

2 Background

To increase their ESE performance, organizations usually apply some variation of the continuous improvement cycle depicted in Figure 2. The cycle typically starts with a **materiality assessment** (process P1), during which organizations determine the relevant set of ESE topics, given their needs, the industry sector they operate in, business operations, and the regulatory system of their region [10]. Examples of topics are gender equity, energy management, and greenhouse gas emissions. A commonly used technique to identify and prioritize material topics is producing a materiality matrix [11]. A materiality matrix consists of an X and Y axis where the Y-axis often represents the importance to stakeholders and the X-axis represents the impact on the success of the business. A sustainability manager maps relevant topics on the X and Y axes. Topics in the top right corner of the matrix are considered important.

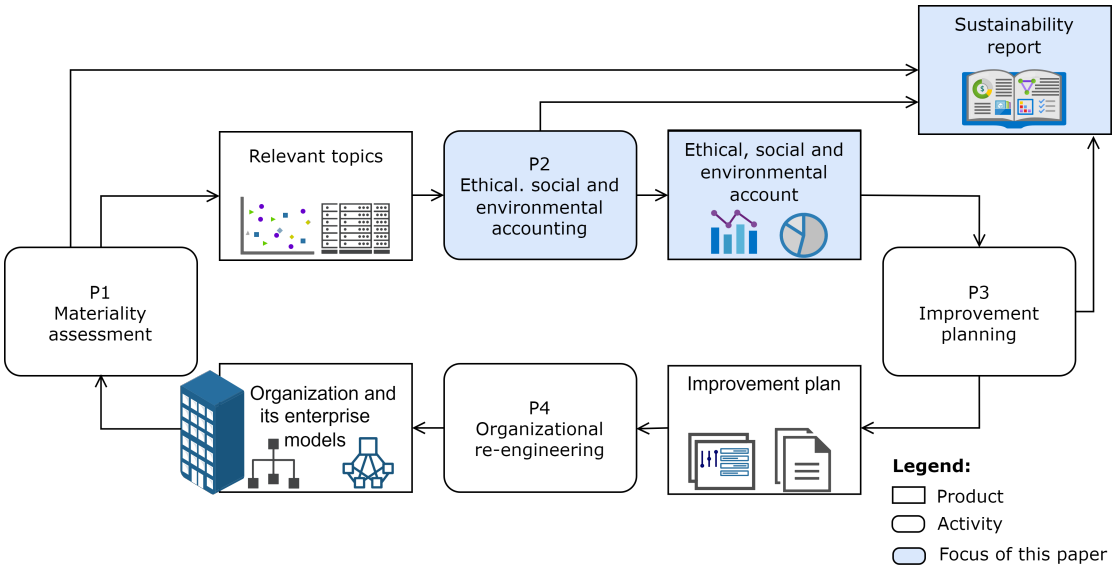


Figure 2. The continuous improvement cycle that organizations can apply to become more sustainable

Once the materiality assessment is completed, organizations assess their performance in the material topics by conducting **ethical, social, and environmental accounting** (P2) [5], [6].

As previously explained, an ESE accountant collects data from organizational stakeholders via surveys or extracts it from information systems. Examples of such data are the total annual energy consumption, the total renewable energy consumption, and the amount of recycled waste. This data allows for calculating indirect performance indicators such as the percentage of renewable energy used, and the percentage of recycled waste. We refer to the set of indicator values collected by conducting an ESEA process as ESE account. Parts of this account are typically published in a sustainability report, intended for specific stakeholder groups or for the general public.

Organizational managers can use ESEA results to formulate a plan intended to increase organizational sustainability. We refer to this as the **improvement planning** process (P3). The improvement plan consists of concrete improvement actions that need to be executed in order to increase the organization's ESE performance. Examples of improvement actions related to gender equity are increasing the number of women executives by making sure that promotion processes are unbiased, improving organizational policies to better work-life balance, and offering career development opportunities. Once the plan is agreed upon, organizations execute the improvement actions by means of an **organizational re-engineering** process (P4). Usually, the changes are not only effectuated in the actual organization but also reflected in the enterprise models (e.g., organization charts, business process models, and policy documents). The effects of these changes are assessed in the next iteration of the cycle.

This article focuses on the ESEA process. The conceptual development of ESEA is attributed to Gray [12]. Over time numerous ESEA methods have been developed. These methods provide guidance and instructions on how to perform ESEA. Often, the methods prescribe a set of ESE topics that should be disclosed and define a procedure to successfully assess and report on these topics. Given that sustainability is a multifaceted concept, it is not directly measurable and therefore requires a set of indicators to measure performance [13]. Hence, ESEA methods usually refine topics further into a set of organizational sustainability performance indicators. Examples of ESEA methods are the Sustainability Tracking, Assessment & Rating System (STARS)[4], the B Impact Assessment (BIA)[5] used by certified B Corporations, and the Common Good Balance Sheet prescribed by the Economy for the Common Good[6]. Some ESEA methods, such as the Global Reporting Initiative Standards[7] and Integrated Reporting framework[8], put emphasis on establishing sustainable reporting guidelines, rather than formulating an approach for measuring and assessing ESE performance [14]. To give an example of an ESEA method, Figure 3 shows two screen captures of the B Impact Assessment. The figure highlights some concepts that are typically found in ESEA methods, such as topics (in red rectangles), indicators (purple rectangles), and answer options (yellow rectangles). Using the menu bar on the left, users can navigate to the automatically generated report and obtain B Corporation certification (green and blue rectangles).

There are several reasons for performing ESEA, such as addressing concerns from the public [15] or obtaining a specific certification [16]. Additionally, ESEA can improve business performance. There has been empirical evidence that shows that ESEA method certifications and ESE disclosures have a positive effect on organizations' financial performance [17], [18]. Küchler and Herzig found that ESEA methods that are applicable to organizations in any industry sector, do not always cover the necessary industry-specific indicators [19]. This limitation is one of the drivers for organizations to apply multiple ESEA methods (industry-specific and non-industry-specific), justifying the need for versatile ICT infrastructure. Given that ESEA methods contain similar concepts this domain lends itself to developing a DSL. We have opted for developing a DSL,

[4] https://stars.aashe.org/about-stars/
[5] https://bimpactassessment.net
[6] https://www.ecogood.org/apply-ecg/common-good-matrix/
[7] https://www.globalreporting.org/standards/
[8] https://www.integratedreporting.org/resource/international-ir-framework/

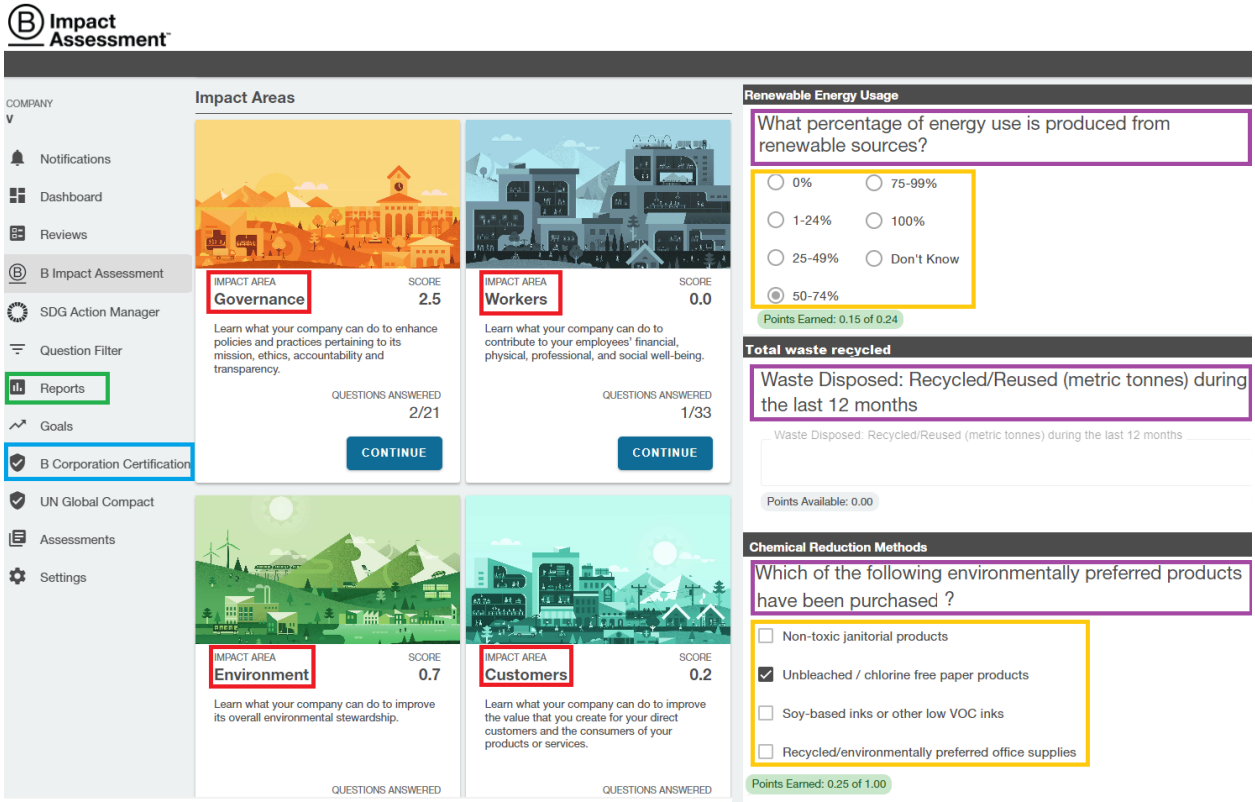


Figure 3. Screen captures of the B Impact Assessment web-based tool, which operationalizes the B Impact Assessment method

instead of using a general-purpose language since we want to execute the models created with the DSL so that we can configure the tool to adapt to the existing methods, and even create new ones.

There are prior works that have focused on managing the complexity of models and modeling languages. For instance, [20] defines a modularisation approach for large models. However, this article focuses on evaluating and improving user performance while understanding, updating, or creating ESEA method models, in the line of earlier work such as [21] and [22]. The importance of grammar for managing model complexity in heterogeneous modeling is further emphasized in [23].

Numerous evaluation protocols for modeling languages have been reported in the literature. Most of these protocols are applied to diagrammatic modeling methods [24], [25]. Our evaluation protocol is highly influenced by previously existing literature, nonetheless, it introduces activities tailored for evaluating an Xtext grammar.

3 Research Method

Since we aim to produce and evaluate the grammar for specifying ESEA method models, produce knowledge around that grammar (e.g. its strengths and limitations), and aim to understand the complexity of creating ESEA method models, we apply Design Science approach [26]. With respect to language development, we follow conventional DSL engineering practices [27]. Figure 4 shows the research method. We use the metamodel, EBNF grammar, and model interpreter from [7] as input for this research. We refer to these artifacts as metamodel V1, openESEA grammar V1, and openESEA interpreter V1, respectively. By executing the research method we aim to create new, more versatile versions (i.e., V2) of the metamodel and the grammar (now specified in Xtext).

3.1 Problem Investigation

Earlier work [7] provides a starting point for improving the modeling language. However, to ensure that we capture all requirements for a new version, we perform another iteration of the research

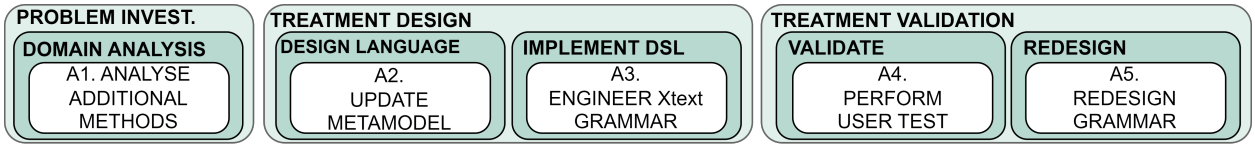


Figure 4. An overview of the research activities

method in [7]. Firstly, we analyze additional ESEA methods (activity A1), by modeling the methods using the Process Deliverable Diagram (PDD) notation [28]. We validate the PDDs with experts in the respective ESEA methods. After the validation interviews, we update and improve the PDDs, if necessary. The validated PDDs are used to create activity and concept comparisons, applying the method comparison approach [29]. The activity and concept comparisons result in a generic method. Using this comparison approach we find commonalities of ESEA methods, which form the backbone of the tool. The common concepts serve as input for the DSL and the generic activities help us identify which core features are not yet supported by the openESEA framework. We write epics and user stories for each of the newly identified features [30]. In this development cycle, we prioritize the DSL features related to the specification of surveys and for the interpreter; we aim to improve the robustness and maintainability. In this article, we report solely on the new version of the DSL, leaving the interpreter evolution out of the scope.

3.2 Treatment Design

Based on the concept comparison (output of activity A1) and the openESEA metamodel V1, we derive new (and update existing) ESEA method metaclasses (activity A2), resulting in metamodel V2. Metamodel V2 is like its predecessor, a UML Class Diagram [31] and it specifies the data structure of an ESEA method and its applications. Metrology standards [32] also inform our design decisions. In activity A3 we engineer a textual Xtext grammar [8] based on the metamodel. While automatic transformation frameworks from (Ecore) metamodels to Xtext grammars exist, we have decided to implement the textual grammar manually to have more control over the result. In this version of the DSL, we have opted for textual grammar. For future versions, we plan to create a diagrammatic DSL and perform user tests to find out which option is preferable.

3.3 Treatment Validation

We run a user test (activity A4) to validate the grammar and to discover potential redesigns that would improve the modeling experience. Section 7 explains the user test design, which is based on the Method Evaluation Model (MEM) [33], and analyses the user test results. After completing the user test, we redesign the grammar (activity A5).

4 ESEA Method Comparison

To produce V1 of our DSL and tool, we analyzed 13 ESEA methods [7]. Now we have analyzed six additional ESEA methods to identify new requirements of the openESEA modeling language. Table 2 shows all 13 methods. For each of the methods, we have created a PDD. As an example, Figure 5 shows the PDD of the B Impact Assessment method. All PDDs can be found in the technical report [34].

To extend the metamodel, we create a super-method and a generic method, based on previously- and newly-modeled PDDs. This article explains the extension of the metamodel with additional metaclasses, therefore, we focus on the results of the concept comparison rather than the activity comparison. During the concept comparison approach [29], we first create a list of super-concepts. This list contains all concepts of one method. Incrementally, the concepts from other methods are added. For each new concept, we check whether the concept is already added to the super-concept

Table 2. The ESEA methods have provided input for extending the modeling language. Newly added methods with respect to [7] are marked with an Asterisk symbol (*).

Method	Organization	URL
B Impact Assessment	B Lab	https://bimpactassessment.net
CDP Company Programs*	CDP	https://www.cdp.net/en/companies
Common Good Balance Sheet	Economy for the Common Good	https://www.ecogood.org/en/common-good-balance-sheet/commongood-matrix
EFQM Model*	European Foundation for Quality Management	https://efqm.org/efqm-model
Fair Trade Software Foundation certification method	Fair Trade Software Foundation	https://dspace.library.uu.nl/handle/1874/368079
GDRC	Global Development Research Centre	http://www.gdrc.org/index.html
GRI Standards	Global Reporting Initiative	https://www.globalreporting.org/standards
ISO 26000 Social responsibility	International Standards Organization	https://www.iso.org/iso-26000-social-responsibility.html
ISO 14000 family	International Standards Organization	https://www.iso.org/iso-14001-environmental-management.html
Measurabl	Measurabl	https://www.measurabl.com
S-CORE*	International Society of Sustainability Professionals	https://doi.org/10.4324/9781849770217
SMETA*	SEDEX	https://www.sedex.com/our-services/smeta-audit/
Social Accounting and Audit	Social Audit Network	https://www.nefconsulting.com/training-capacity-building/resources-and-tools/social-accounting/
Sustainable Development Goals Compass	GRI, United Nations Global Compact & World Business Council for Sustainable Development	https://sdgcompass.org
Sustainability Tracking, Assessment & Rating System (STARS)*	Aashe	https://stars.aashe.org/about-stars/
University Sustainability Assessment Framework (UniSAF)	Green Office Movement	https://www.greenofficemovement.org/sustainability-assessment/
UN Global Compact	United Nations	https://www.unglobalcompact.org/library/231
WFTO Guarantee System*	World Fair Trade Organization	https://wfto.com/what-we-do#our-fair-trade-standard
XES Social Balance	Xarxa d'Economia Solidaria (XES)	https://xes.cat/inici/

list. If it is already present in the list, that concept is not added to the list again. We continue this process until all unique concepts are part of the super-concept list. We refer to concepts that are part of the super-concept list as "super-concepts". After compiling the super-concept list, each super-concept is compared to the concepts of the 13 ESEA methods. We compare the concepts

by creating a table, such as depicted in Table 3. The first column contains all super-concepts and the first row contains all ESEA methods. We compare the super-concepts to the concepts of each method by putting a symbol in the cells at the intersects of super-concepts and ESEA methods. For the symbols we use the following notation, where "c" represents a concept from an ESEA method and "s" depicts a super-concept.

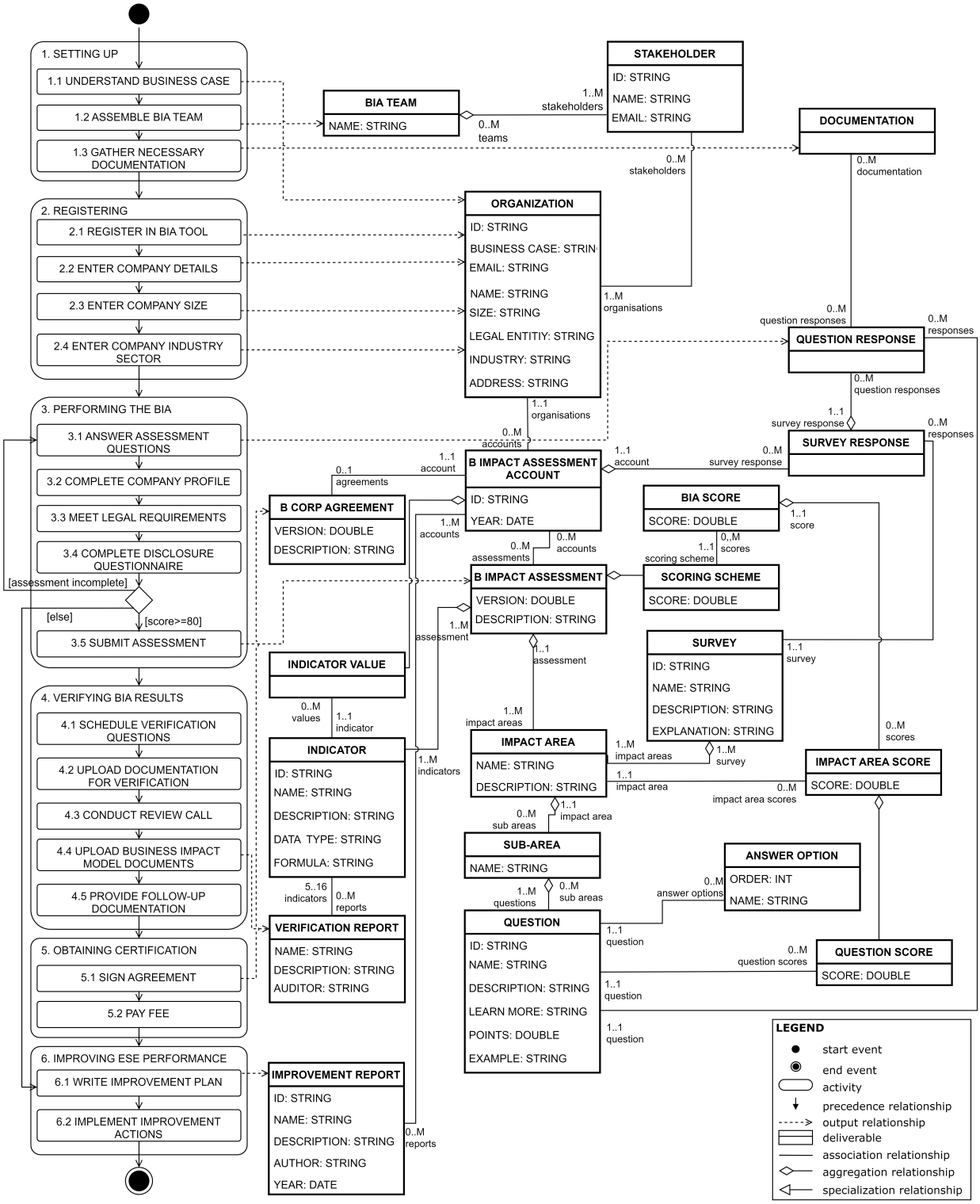


Figure 5. Process deliverable diagram that we have created after reviewing the B Impact Assessment method documentation and tool

- "=": **c** is equivalent to **s** and the name of c is the same as the name of s.
- "[name of concept c]": **c** is equivalent to **s** and the name of c is different from the name of s.
- c "<" s: **c** contains less information than **s**.
- c ">" s: **c** contains more information than **s**.
- c "><" s: A part of **c** overlaps with a part of **s**, but some parts do not overlap.
- c "○" s: **s** is not explicitly mentioned in the method, but it is reasonable to infer its presence.
- c "[empty]" s: **s** is not present in the method.

Table 3 shows a sample of the concept comparison. Based on the method comparison, we create a generic method. The generic method contains concepts that are present in more than one ESEA method. We added nine more concepts to the metamodel because they are part of the generic method. Concepts that were only present in one method were omitted. Moreover, we only focused on the accounting phase (i.e., assessing and reporting) even though many ESEA methods define an auditing protocol as well. We plan to include audit-related classes in the next version of the metamodel. To do this we will extend the method comparison. In Section 5.1 we report on the metaclasses that we added based on the concept comparison.

Table 3. Excerpt of the concept comparison. An asterisk symbol (*) depicts whether a concept is new in V2

	Methods				
Super-concept	B Impact Asessment	Common Good Balance Sheet	SMETA	UniSAF	XES
Survey *	B Impact Assessment	○	SA Questionnaire	○	Module
Question *	=	=	=	○	=
Answer option*	○	><			>
Question response *	=	Evidence	○	○	=
Stakeholder *	=	=	○	=	>
Topic	Impact area	Theme	Pillar	Dimension	Sub-module
Indicator	=	=	=	=	=
Organization	=	Company	=	Institution	=
Certification level	>	=			

5 Modeling Language for Specifying ESEA Methods

The openESEA modeling language consists of two artifacts: the metamodel and the DSL. The metamodel depicts the classes that are necessary to support the application of ESEA methods. A number of these metaclasses are used as the basis for engineering textual grammar. Find a full explanation of all metaclasses, their attributes, and relationships in the technical report [35].

5.1 Metamodel of ESEA Methods

Figure 6 depicts metamodel V2. The metamodel serves as an ontology for managing complexity in ESEA methods. It contains classes with generic names. Classes in ESEA methods that represent the same concepts, but are referred to with different names, can be mapped against the classes in the metamodel. For example, in the B Impact Assessment method, the assessment is structured in "Impact areas". Our metamodel contains a class "Topic". Each impact area of the B Impact Assessment can be expressed by instantiating the topic class. To express the impact area called "Environment", method engineers can instantiate the topic class and provide values to the attribute as follows.

```
Id:impact_area_1
Name: Environment
Description:Evaluates a company’s overall environmental management practices
```

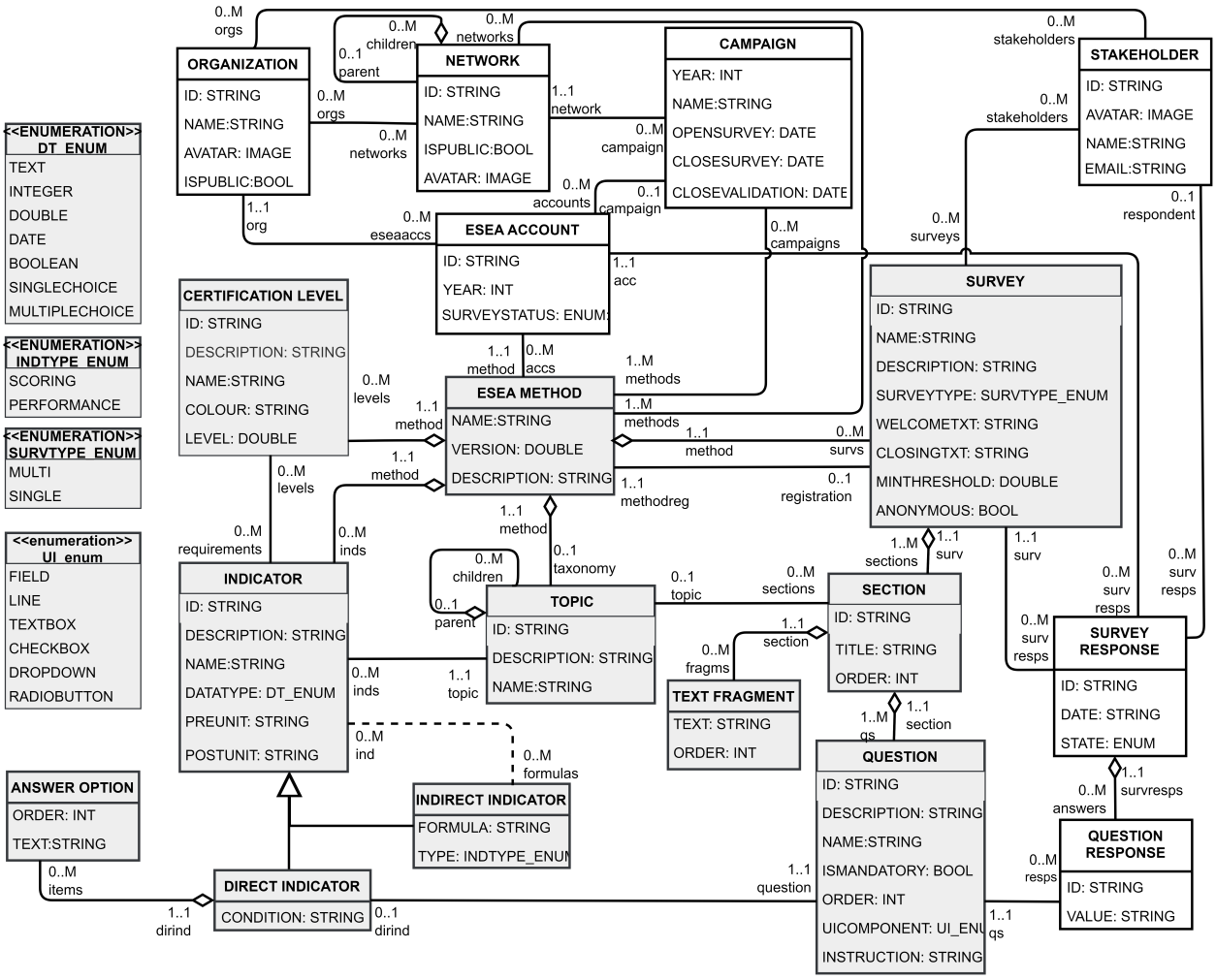


Figure 6. The openESEA metamodel contains the most relevant primitives needed to model ESEA methods

Section 6 provides an elaborate example of how the modeling language is used. V1 of the DSL contained 12 classes which are the following. *ESEA method* captures general information about the method such as the name, version, and description. *Category* stores the names and descriptions of disclosure topics that are assessed as part of the method, for instance, employee diversity, greenhouse gas emissions, and water consumption. The disclosure topics are assessed with *Metric*s such as the number of female, male, and non-binary employees, the CO_2 emission in parts per million, and the annual water consumption in liters. The actual metric value (e.g., the amount of water consumed) is stored in the class `Data`. Based on those metrics data *Indicator*s can be calculated, such as the man-to-woman ratio. All indicator values make up the *ESEA account* and *Report item*s specify what type of chart (e.g., bar chart, pie chart, etc.) should display the indicator values in a sustainability report. A *User* (typically an ESE accountant) applies the ESEA method for the *Organization* they work for. If the organization has applied the ESEA method successfully and fulfilled the *Requirement*s the organization can obtain a *Certification* and become part of a `Network` of sustainable organizations.

In V2 we have opted to change the names of some classes since we found that the terminology in DSL V1 did not coincide with the terminology used in the majority of the ESEA methods or because

we deemed the new names more intuitive (Figure 6). Hence we changed *Category* to *Topic*, *Metric* to *Direct indicator*, *Data* to *Question response*, and *User* to *Stakeholder* in V2 of the DSL.

Based on the concept comparison performed in this research, we have added the class *Survey*, given that the indicator values are collected via surveys. The surveys consist of *Question*s, such as "What was your company's total energy consumption in 2021?" and "What percentage of energy use is produced from low-impact renewable sources?". To structure, the survey questions can be grouped into *Section*s within the survey. For instance, the example questions above can be grouped in the section "Energy Consumption". *Text fragment*s can be placed in-between sections to explain which disclosed topics will be assessed in a section. To facilitate closed questions we added the class *Answer option*s, for instance, to the "Does your company use single-use plastics for packaging of products?" the answer options are "yes", "no", and "not applicable". All question responses are stored in the class *Survey response*. We added a generalization class called *Indicator*, which is specialized in the classes *Direct indicator* and *Indirect indicator* to store information such as the indicator name and description. By analyzing ESEA practices, we found that most ESEA methods are monitored by an entity (typically a not-for-profit organization) that oversees whether organizations deserve to become certified and become part of a network. These monitoring entities usually start a *Campaign* with a fixed start and end date. During this period organizations within the network can apply the ESEA method. If the organizations want to be considered for certification, the accounting should be performed and submitted before the campaign ends. To support this we included the campaign class. Lastly, we removed the *Report item* and *Requirement* classes to reduce the complexity of the DSL. The removal of requirements does not compromise the functionality of the DSL since requirements can be stored as indicators. We removed report items because the corresponding reporting capabilities of the interpreter tool were too limited and we plan to implement more versatile sustainability reporting features in the future.

By adding new concepts to the metamodel we are able to express more aspects of ESEA methods, such as specifying surveys that consist of questions. Table 4 provides an overview of each of the metamodel classes and their definitions.

The metamodel differentiates between metaclasses that are instantiated when the method is engineered and specified (these have a gray background in the metamodel) and metaclasses that are instantiated when the method is applied and executed (these have a white background). According to Brinkkemper [36], method engineering is the engineering discipline to design, construct and adapt methods, techniques, and tools for the development of information systems. While most of the method engineering discipline focuses on system development methods, we adopt the method engineering techniques to engineer methods for ESEA. Therefore in this article, engineering the methods entails designing and constructing ESEA methods. Examples of metaclasses that are instantiated during method engineering are *ESEA method* (which gives the method a name, a description, and the version), *Topic*s, and *Question*. The questions are asked, during execution time, to the people involved in the ESEA data collection (e.g., the ESE accountant or sustainability officer, staff members); however, they are specified during the method engineering. Examples of classes that are instantiated when applying or executing the method are *Organization* (which represent entities that apply the method), and *Question response* (which stores the responses of questions for one specific application of an ESEA method, by a given organization, in a given year). The gray metaclasses provide a proper abstract syntax for the grammar.

5.2 Textual Grammar for Creating ESEA Method Models

We have implemented the DSL as a textual grammar that allows method engineers to create textual models of ESEA methods. We refer to these models as "ESEA method models". The textual models created according to the rules of the grammar then be parsed and interpreted by openESEA, and the tool reacts by offering the proper interfaces and features to support the modeled method. For every gray metaclass in the metamodel, we define a grammar primitive. While there is a multitude of ways

Table 4. The metaclasses and their definitions

Meta class	Definition
ESEA method	A specification of how ethical, social, and environmental accounting should be performed, according to the creator(s) of the method. This includes the specifications of surveys, stakeholder groups, certification levels, topics, indicators, and questions, but also guidelines on, e.g., the reporting format or whether the results of accounting have to be published.
ESE account	The state of the application of an ESEA method at a given moment in time. Organizations typically apply ESEA methods with a given frequency (e.g., every year an ethical, social, and environmental accounting is performed).
Organization	A social entity that is goal-directed is designed as a deliberately structured and coordinated activity system and is linked to the external environment.
Network	A group of responsible enterprises. Often a network prescribes a specific ESEA method and all responsible enterprises within this network have to apply that method to become members (e.g., to be part of the B Corp network, members have to perform the B Impact Assessment). Typically, organizations need to demonstrate a certain performance in order to be granted a membership; for instance, the ESEA method might have a scoring mechanism and the network defines a minimum threshold (e.g., B Impact Assessment produces a score from 0 to 200, and organizations need to score at least 80 to become B Corporations). The network is typically managed (i.e., orchestrated) by an organization, which also maintains (i.e., evolves and provides support to) the ESEA method. For instance, B Corporations is managed by B Labs.
Stakeholder	An individual with an interest or concern in something (e.g., one specific employee or one specific consumer).
Survey	A questionnaire that a certain stakeholder group has to respond to in order to provide data for the direct indicators. Some surveys are meant to be responded to by only one respondent (e.g., a manager), while other surveys are meant to be responded to by several stakeholders of the same stakeholder group (e.g., all employees).
Topic	Topics group indicators concerning the same phenomenon together. For instance, “Gender equity” is a topic that groups all indicators concerning gender equity together (e.g., number of women in the staff, number of women in management positions, etc.). Another example is the topic “Environmental impact” which groups indicators concerning annual CO2 emission, annual electricity consumption, and annual waste together. Trees of topics can exist. This means that topics can be split up into more fine-grained topics (e.g., Workers ->Gender equity or Workers ->Healthcare).
Text fragment	A text that explains or elaborates on the topic.
Section	A section groups text fragments and questions in a survey.
Indicator	An indicator is the definition of a measure that is assessed and reported on during the accounting. For instance, an indicator for the topic “Gender equity” could be the “Gender pay gap”. Indicators can be classified as direct or indirect indicators. “Gender pay gap” is an indirect indicator, while the “Average salary for men” and “Average salary for women” are direct indicators.
Direct indicator	The value of a direct indicator can be provided by a stakeholder via a question in a survey. Therefore a direct indicator does not have a formula. The direct indicator is a specialization of the class “Indicator”, so it inherits all the attributes and relationships.
Indirect indicator	An indirect indicator has a formula. An indirect indicator is calculated by using direct or other indirect indicators. The indirect indicator class can be used to define scoring rules, by creating a scoring indicator and defining a formula. The indirect indicator is a specialization of the class “Indicator”, so it inherits all the attributes and relationships.
Question	Asks for the value of a direct indicator.
Survey response	The response of a survey made by a stakeholder. It contains many questions and responses.
Question response	The response to a question is stored in “Question response”.
Certification level	A certification is an official document attesting to a status or level of achievement. The certification is issued by the network once the applicable requirements are met (note: The requirements can be specified as an indirect indicator).
Campaign	A period during which the ESE accounting can be performed.

of designing grammar rules that operationalize the metaclasses, we have opted for an approach that ensures human readability. For instance, every attribute is written on a new line and we try to choose commonly used, intuitive names for concepts (e.g., for UI components we used common names such as radio button, check box, text field, etc.). Table 5 shows an excerpt of a real-life ESEA method, namely the B Impact Assessment with two topics (Workers and Environmental) and one indicator for each topic. This method part can be expressed by instantiating the ESEA method, topic, and indicator classes. To use the B Impact Assessment in openESEA, a method engineer can specify the method as a textual model, using the grammar rules in the third column. Naturally, grammar consists of more rules, but the example only displays a small subset of rules to exemplify the use of grammar. The grammar starts with an ESEA method rule that captures general information about the method (e.g., name and version). Then a list of topics is created. Here all ESE topics that are part of the method can be specified. In our example, the list of topics contains two topics, Workers and Environment. Then the grammar contains a list of indicators. Here all indicators that are part of the ESEA method are specified. The indicator rules contain attributes to specify any indicator. The method engineer can, for instance, specify the data type of the indicator (e.g., integer, text, Boolean), what topic the indicator belongs to, and whether the indicator is calculated with a formula (i.e., indirect indicator) or if the value is collected directly via a survey (i.e., direct indicator). In the example, there are two indicators; the minimum hourly wage and the annual water consumption. Both indicators are direct and the data types are double and integer, respectively. After all indicators are specified, the grammar specifies a list of surveys and a list of certification levels. For the sake of brevity, we omitted these lists, given that they contain several lines each. For the full Xtext grammar, see the technical report [35] or find it on Github [9].

6 The OpenESEA Technology and Usage of the Language

6.1 Usage of the OpenESEA Modeling Language

Although we use the example of the B Impact Assessment, the openESEA modeling language can be used to create an ESEA method model of any ESEA method. The methods can then be operationalized in the openESEA tool. So far, we have not come across any other tool with these properties and this level of versatility. For future versions of the interpreter, we plan a functionality that makes the tool even more valuable from practitioners' and consultants' perspectives.

We aim to create a repository of ESEA method models (i.e., models created with our DSL of several ESEA methods). Organizations can then select a set of methods they would like to use or create their own method. The methods can be tailored via the widgets in the interpreter. Once the organizations are content with the set of methods, the interpreter performs the model management operations "match" and "merge" on the method models [37]. This way a mapping between the models is created and the models are integrated. As a result, a super-method is created that asks for every indicator and questions only once.

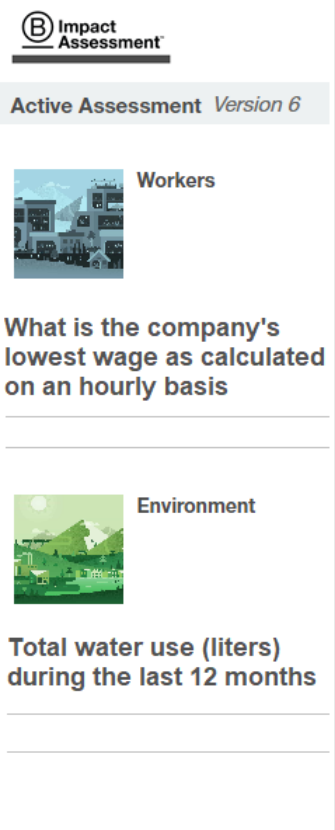
The current version of the DSL contains the basic elements for specifying ESEA methods. The interpreter also includes features that allow uploading and tailoring the methods in the tool, see the screenshot in Figure 7. Integrating model management operations in the openESEA framework would further increase its utility for sustainability professionals.

[9] https://github.com/sergioespana/openESEA

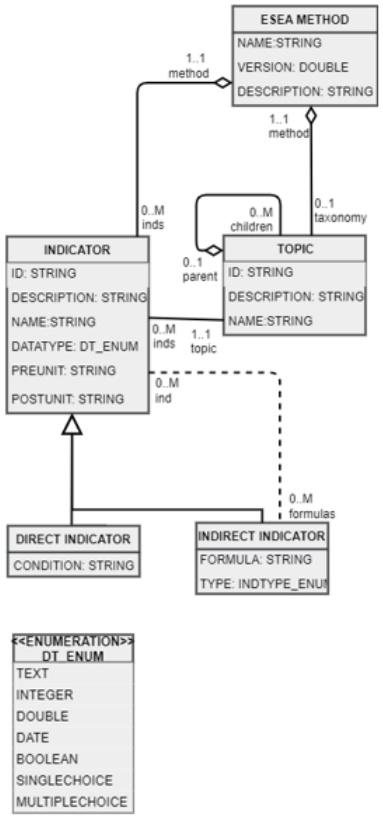
Table 5. The first column contains an excerpt of a real-life ESEA method. The second column displays the meta classes necessary to support the ESEA method excerpt. The third row contains an Xtext grammar excerpt that corresponds to the meta classes. The fourth column shows an ESEA method model fragment that displays information of the real-life ESEA method.

Real life ESEA method	Meta model	Grammar	ESEA method model

Real life ESEA method



Meta model



Grammar

```
ESEA_method:
'Name:' STRING
'Version:' DOUBLE
'isPublic:' BOOLEAN
'Description:' STRING
listTopics+=ListTopics
listIndicators+=ListIndicators
listSurvey+=ListSurvey
listCertificationLevels+=ListCertificationLevels;

ListTopics:
'Topics:'
(topic+=Topic)+ ;

List_of_indicators:
'Indicators:'
(indicator+=Indicator)+;

[…]

Topic:
'topic_id:' name=ID
'Name:' STRING ;

Indicator:
'Indicator_id:' name=ID
'Name:' STRING
('PreUnit:' STRING)?
('PostUnit:' STRING)?
'Topic:' linkTopic=[Topic]
'Indicator_type:' indicator_type=Indicator_type
'DataType:' datatype=Datatype ;


Datatype:
text='text' | integer='integer' | double='double'
| boolean='boolean' | singleChoice=SingleChoice;

Indicator_type:
direct=Direct | indirect=Indirect ;

Direct:
direct='Direct'
('Condition:' expression=Expression)?;
```

ESEA method model

```
Name: 'B Impact
Assessment'
Version: 6.0
isPublic: True
Description: 'A digital
tool that can help
measure, manage, and
improve positive impact
performance'


Topics:

topic_id:workers
Name'Workers'
topic_id:environment
Name:'Environment'

[…]

Indicators:



Indicator_id:
total_water_use
Name: 'Total water use'
PostUnit: 'Litres'
Topic: environment
Indicator_type: Direct

DataType: integer

Indicator_id: Lowest_wage
Name: 'Lowest wage on
hourly basis'
PostUnit: 'Euros'
Topic: workers
Indicator_type: Direct

DataType: double

[…]
```

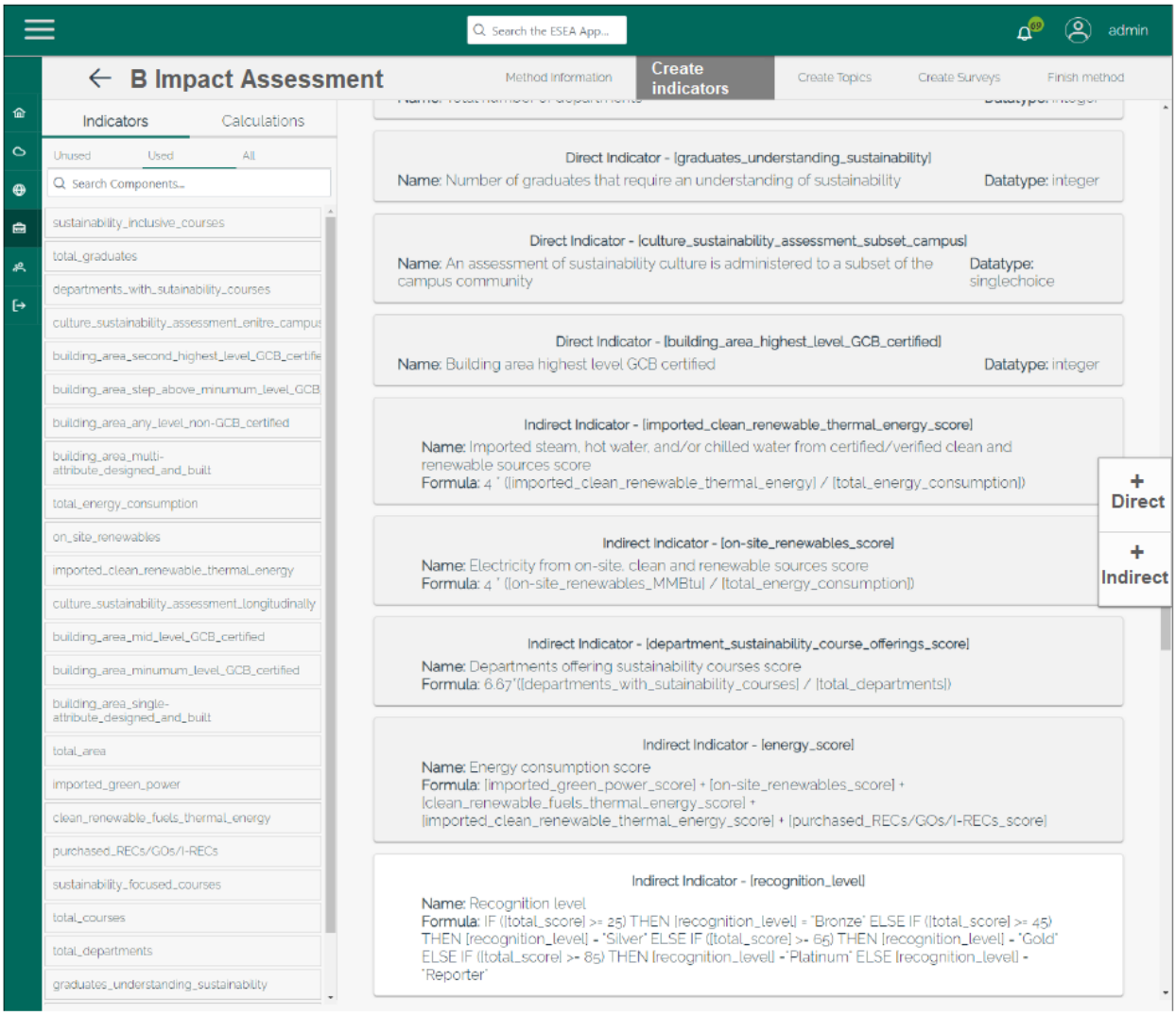


Figure 7. Screenshot of the openESEA screens that allow creating or uploading methods (left) and the tailoring methods (right)

6.2 OpenESEA Technology Stack

For the current version of the modeling language, we have opted for an Xtext grammar, which can be used in combination with the openESEA Xtext editor. ESEA method engineers can create ESEA method models using the openESEA Xtext editor or the method models can be specified through the openESEA interface. If the method models are created with the editor, they can be uploaded to the front end and will be parsed with a JSON schema.

We have re-implemented the interpreter completely to switch from a technology based on the React framework (interface and application tier) and Firebase Firestore, Authentication and Hosting services (back end), to a technology based on the Vue.js framework (interface tier), the Python-based Django framework (application layer), and Heroku Authentication and Hosting services (back end). The current technology provides services that let method engineers create ESEA method models, enable ESE accounts to manage ESE accountings, and allow stakeholders (e.g., employees or customers) to provide input for the accounting through stakeholder surveys. The new architecture of the tool can be observed in Figure 8.

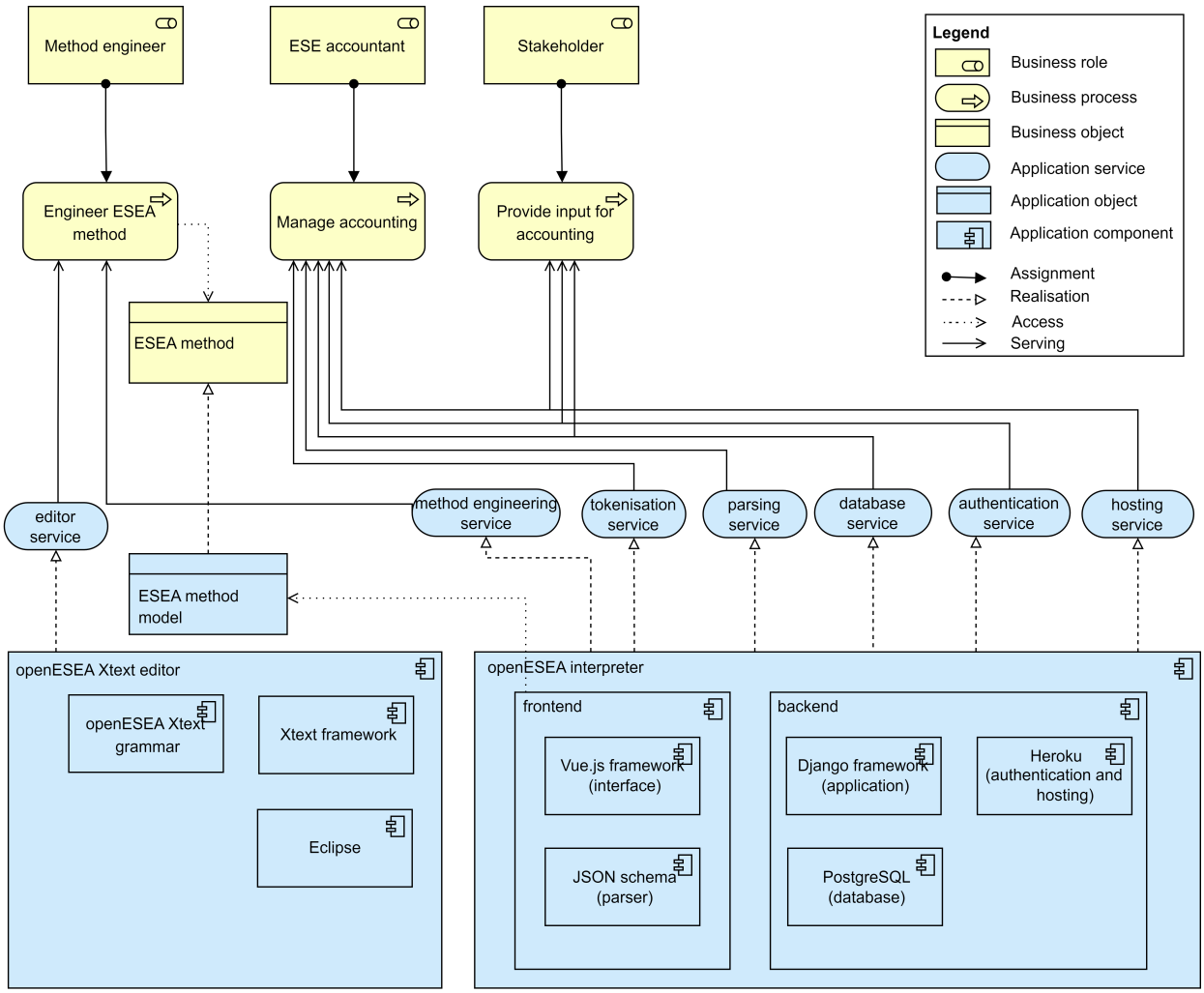


Figure 8. The openESEA architecture represented using ArchiMate [38]

7 Validation of the ESEA Grammar

7.1 User Test Design

We run a user test, to assess the performance of the grammar. The user test is supported by an e-assessment tool. Figure 9 shows the test procedure and variables. We use the reporting guidelines from [39]. The **object of study** is the grammar. We leave the Xtext editor out of the scope since it might interfere with the results. Moreover, in this development iteration, we do not validate the interpreter, given that we are working on a new release of the interpreter that will incorporate major improvements. The **main objective**, assessing the grammar, is refined into two sub-objectives: (i) determine to what extent users are able to successfully create ESEA method models, using the DSL, (ii) discover potential improvements of the grammar by performing qualitative analyses on the user test results. The **test participants** are 75 Information Science bachelor students from Utrecht University, with little to no professional experience, little programming knowledge, and no knowledge of model-driven architectures, textual grammars, and ESEA prior to the user test. The expected future users of the grammar are ESEA method engineers, who have a similar experience with ICT, but greater knowledge of ESEA.

The **test structure** is shown in Figure 10. The test consists of tasks that can be of three types: comprehension, modification, and creation. For each task, we have formulated questions and each question consists of a number of steps. The comprehension questions are the easiest and test whether the participants can understand excerpts of the grammar and excerpts of models created with the grammar. The modification questions ask the participants to make a change in a given model. An

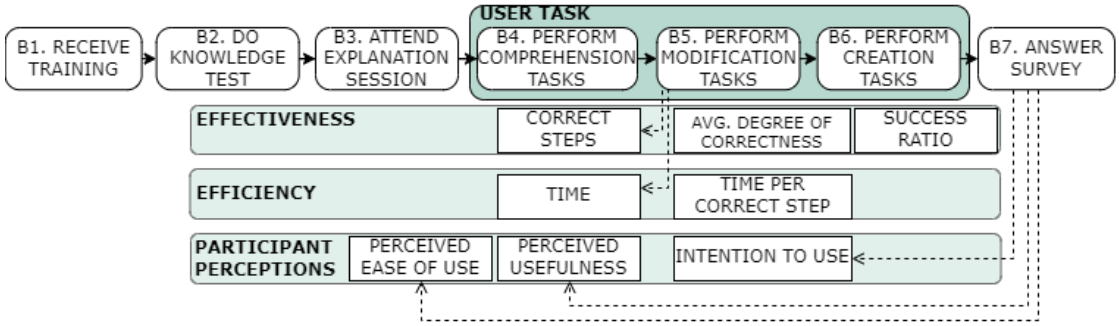
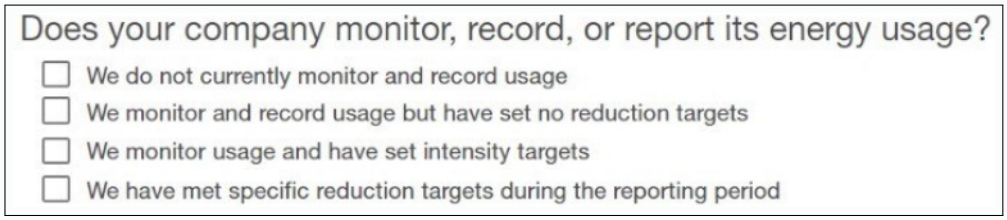


Figure 9. Overview of the user test. Participants first receive training. While carrying out the tasks, users spend some time taking a set of steps. We then elicit their perceptions.

example of a modification question could be filling in the correct data type in an indicator model. The creation questions are the most challenging; they require the participants to create a model from scratch, based on a textual description of an ESEA method or a screenshot of a real ESEA tool. An example of a creation question can be found in Listing 1. In this example, the students are presented with a screenshot of a B Impact Assessment question. The excerpt of the grammar that specifies the *Question* primitive is presented. The students should write the model (which complies with the grammar) that expresses the B Impact Assessment question. Note that the students are allowed to use a manual that explains all grammar components. Thus, they do not have to know the meaning of the attributes (e.g., "isMandatory") by heart.

For comprehension questions, a step typically refers to answering a multiple-choice question. For modification questions, a step refers to making an alteration in a model or filling in a text field. For creation questions, a step refers to writing a line of a model fragment. Thus, the question in Listing 1 contains eight steps.

Listing 1: Example of a creation question




```
Q4. Consider the screenshot (above) and the Question rule (below). Create the Question model for the
question in the screenshot. Assume that answering the question is mandatory.

Question:
'question_id:' name=ID
'Name:' STRING
'Description:' STRING
'isMandatory:' BOOLEAN
'UIComponent:' uicomponent+=UICOMPONENT
'Order:' INT
'Indicator:' linkIndicator=[Indicator]?
'Instruction:' STRING;

enum UICOMPONENT: field="field" | line="line" | textBox="textBox" |
checkBox="checkBox" | dropDown="dropDown" | radioButton="radioButton";
terminal BOOLEAN : ('true'|'false');
```

The **variables** we measure based on the MEM [33] are the effectiveness of using the DSL grammar, the efficiency in the proposed tasks, and the participant perceptions. These variables

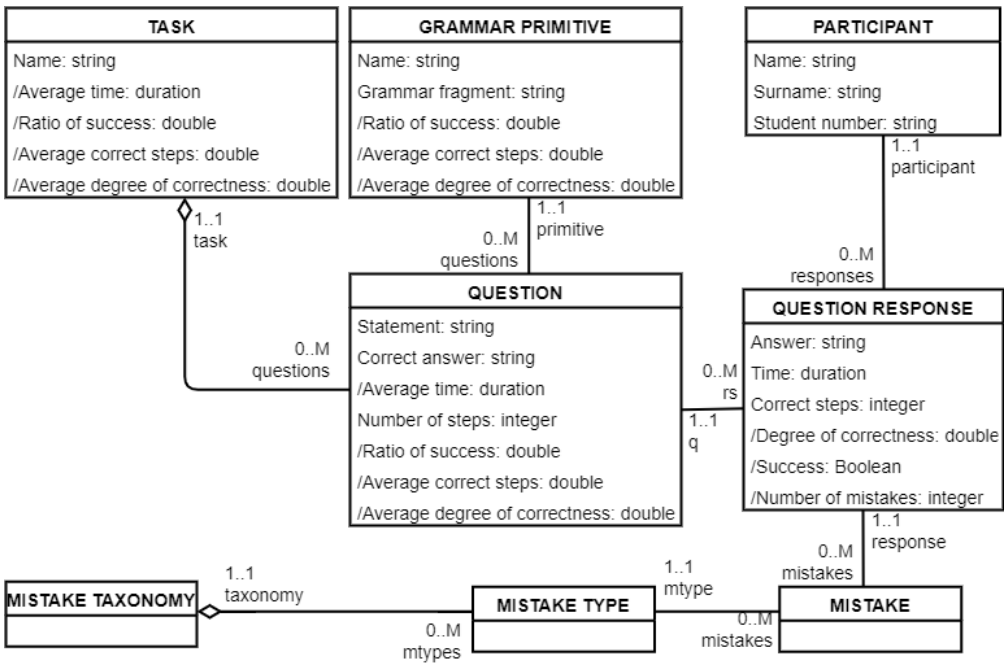


Figure 10. The metamodel of the user test

refer to the environment and structure system dimensions according to the hierarchy of criteria for information system artifact evaluation [40]. We base our approach on earlier work that also used the MEM variables to evaluate languages [41], [42]. For each of the MEM constructs, we define response variables [43], that are represented as attributes in Figure 10. **Effectiveness** refers to how well the DSL achieves its objectives. While assessing the test responses, we produce the values of the *correct steps* variable. With these values, we can calculate the following variables: *average degree of correctness* (see formula 1) measures to what extent the participant correctly conducted the steps of the modeling tasks in the user task, *success* indicates that the participant did not make any mistakes and thus answered the entire question correctly, and *success ratio* (formula 2) reflects the normalized percentage of successful responses.

$$average\ degree\ of\ correctness = \frac{\sum_{q=1}^{|tquestions|} \frac{\sum_{qr=1}^{|responses|} \frac{correct\ steps_{qr}}{steps_{qr}}}{|responses|}}{|tquestions|} \quad (1)$$

$$success\ ratio = \frac{\sum_{q=1}^{|tquestions|} \frac{\sum_{qr=1}^{|responses|} success}{|responses|}}{|tquestions|} \quad (2)$$

Efficiency refers to the effort required to apply the DSL. For each question, the e-assessment tool automatically measures the time that the participant spent on it. As a better variable for efficiency, we define *time per correct step* (formula 3).

$$time\ per\ correct\ step = \frac{\sum_{q=1}^{|tquestions|} \frac{\sum_{qr=1}^{|responses|} \frac{time_{qr}}{correct\ steps_{qr}}}{|responses|}}{|tquestions|} \quad (3)$$

The formulas are aggregating the results per task (comprehension, modification, or creation), where $|tquestions|$ represents the total number of questions per task. When aggregating the average

degree of correctness and success ratio per grammar primitive, $|tquestions|$ should be changed to $|pquestions|$ which represents the total number of questions related to each grammar primitive (per task). Similarly, *responses* represents the set of responses. To assess **participant perceptions**, the MEM offers an adaptable questionnaire that allows measuring the *perceived usefulness*, the *perceived ease of use*, and the *intention to use* the DSL in the future, if confronted with similar tasks during their profession.

In accordance with the **test procedure** shown in Figure 9, the participants receive a ninety-minute training (B1) where we introduce them to ESEA methods. We also train them in DSL grammar. After the training, the participants perform a knowledge test (B2). The knowledge test also consists of three types of questions (comprehension, modification, and creation). The purpose of the knowledge test is to have the participants apply their newly acquired knowledge and receive feedback on their performance. This resembles the training that the future users of the grammar (i.e., ESEA method engineers) will receive. After the participants have finished the knowledge test, they attend an explanation session (B3), where we discuss the answers to the knowledge test. Hereafter, the participants start the user task; i.e., a test where we actually measure their effectiveness and efficiency. They answer comprehension (B4), modification (B5), and creation questions (B6), in that order. The questions are similar to the ones in the knowledge test, but the overall test is longer and more challenging. After the test, the participants are asked to fill in the MEM questionnaire (B7).

7.2 User Test Results

Performance per Primitive. Table 6 shows the average degree of correctness and the success ratio for each grammar primitive per task, as well as the time spent per correct step, the average degree of correctness, and the success ratio aggregated per task. The average degree of correctness per task is quite high, ranging from 86% to 89%. The success ratios range from 72% for the creation task to 83% for the modification task. Overall, a positive sign, indicating that many participants were able to execute questions flawlessly. When evaluating the participant's answers we have chosen to be very strict since every deviation from the grammar rules is a syntactic error. In practice, most syntactic mistakes will be prevented by the usage of the editor, since the editor ensures that the models comply with the syntax.

To put efficiency results in the context of industrial practice, we have estimated the size of a real ESEA method and the total time it would take to author its model using the grammar. We have taken the basic variant of the XES Social Balance as a reference since we have full access to its internal documentation. That method has five topics, 203 direct and indirect indicators, one survey, five survey sections, 86 questions, and five text fragments. The language primitive *Topic* requires three lines, so modeling all five topics of the XES Social Balance implies writing 15 lines. An *Indirect indicator* has eight lines, *Direct indicator* has seven lines, *Answer option* has three lines, *Survey* has eight lines, *Survey section* has four lines, *Question* has eight lines, and *Text fragment* has three lines. As a result, the total size of the model would be 2916 lines. Given that our participants spent 50 seconds per correct line, creating a fully correct method model without the editor would take them $50 * 2916 = 145\,800$ seconds (40 hours and 30 minutes).

Figure 11 depicts two stacked bar charts, plotted on two different y-axes. The stacked bar chart called "Test method" is the collection of model fragments that the user test participants had to model as part of the creation questions. The participants were asked to create one model based on the *ESEA method* primitive, two *Topics*, one *Direct indicator*, one *Survey*, one *Section*, one *Question*, one *Answer option*, and one *Text fragment*. The XES bar chart depicts how often a primitive appears in the XES method. The upper axis in the chart shows how many hours it would take to model our test method and the XES method. The bottom axis shows the total size of both methods in terms of steps.

Table 6. The average degree of correctness and success ratio per primitive per task. Additionally, the aggregated values per task and time per correct step per task. Correctness and success range from 0 to 1, and can be interpreted as percentages.

	Avg degree of correctness	Success ratio	Time per correct step (s)
Comprehension	0.89	0.81	57.18
Section	0.86	0.63	
Text Fragment	0.89	0.89	
Indicator	0.91	0.79	
Certification level	0.97	0.97	
ESEA method	0.80	0.80	
Modification	0.86	0.83	55.63
Indicator	0.75	0.75	
Certification level	1.00	1.00	
Answer option	1.00	1.00	
Question	0.82	0.80	
Survey	1.00	0.99	
Creation	0.87	0.72	50.00
Survey	0.97	0.85	
Section	1.00	0.99	
Text Fragment	0.91	0.87	
ESEA method	0.98	0.93	
Topic	0.99	0.98	
Indicator	0.78	0.15	
Answer option	0.39	0.39	
Question	0.86	0.37	

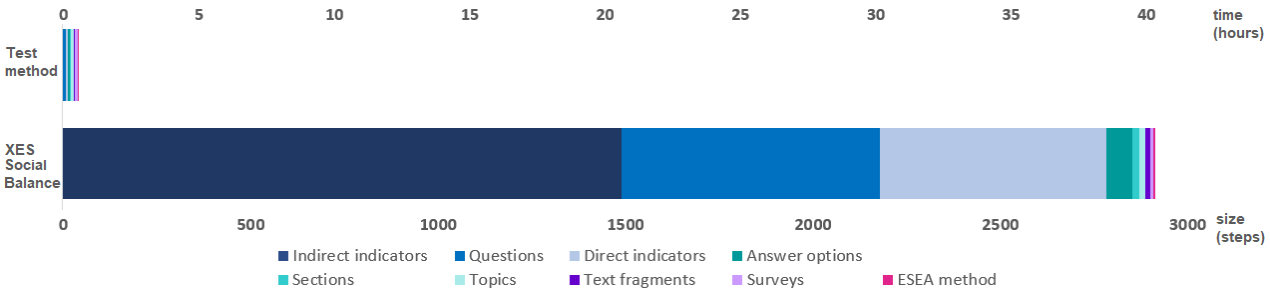


Figure 11. The lower axis and bar charts display the size of our test method compared to the size of a real method, i.e., the XES Social Balance method. The unit is steps, that is, lines of the textual model. The upper axis indicates how long it would take to model each of the methods from scratch; the unit is hours.

Most probably, our estimation of how long it would take to model a real-life ESEA method is an overestimation. We suspect that users of the grammar will become more efficient while producing the model because there are many repetitive actions. For instance, we expect that the ESEA method engineers will spend more time on the first few indicators and become quicker as they progress. They can also copy, paste, and tweak method fragments. Furthermore, in a real-life setting, grammar users will use the editor, which should further improve their efficiency. On the

other hand, the real source of complexity in ESEA method engineering is the participatory design of the method, especially in the case of bottom-up and democratic, design processes, as is the case of the XES Social Balance. But this falls out of the scope of the DSL.

The results per grammar primitive help us pinpoint where we can improve the grammar and training. The grammar primitive *Answer option* in the creation task has the lowest values for the effectiveness variables. The reason for this is that most participants forgot to define the answer options altogether. Perhaps users find it counter-intuitive to define the answer options when modeling a direct indicator (see metamodel, Figure 6). A more intuitive approach could be to define the answer options when modeling its corresponding question. However, a more probable reason for participants forgetting to define the answer options is that users have to select the correct data type for the answer option rule to be triggered. Upon further inspection, we found that in many cases the selected data type was incorrect, making the forgotten answer options an unpreventable follow-up error. The autocompletion feature of the editor will show users exactly which grammar rules are triggered. This will, for instance, prevent the users from forgetting to define answer options. The *Indicator* primitive scores fairly well in the comprehension and modification task. In the creation task, on the other hand, *Indicator* has the lowest success ratio. Most syntactic mistakes in the creation questions related to *Indicator* were non-critical (e.g., capitalization mistakes). Most semantic mistakes were made in the data type. This compromises the utility of the indicator for measuring the intended organizational sustainability performance. In the indicator creation task 65% of the subjects chose the wrong data type. We suspect this is caused by a lack of experience with ESEA and insufficient knowledge about ESEA methods. This can probably be solved by providing a longer, more detailed training session. The success ratio of the *Question* primitive in the creation task is rather low, but most mistakes are non-critical. However, one frequently appearing serious mistake is made in the *order* attribute of *Question*. The order in which questions should be displayed in a survey is indicated with a numeric value in the attribute *order*. The question with the lowest order value is displayed first, followed by the question with the consecutive numeric value, and so on. Overlapping order numbers are not allowed, since multiple questions cannot be in the same place in the survey. Nonetheless, users frequently gave questions with the same order number. To tackle this problem, we should add a constraint to the grammar by extending the validator [44]. The grammar primitives that have proven to cause confusion will be reassessed and possibly updated in the next versions of the modeling language.

Mistake Types. In total, the test subjects made 468 syntactic mistakes and 195 semantic mistakes. Figure 12 shows an overview of the types of mistakes that were made. The semantic mistakes "wrong data type" and "wrong indicator type" relate to the *indicator* primitive. The indicator type specifies whether an indicator is direct or indirect. In other words, whether the value of an indicator is calculated with a formula or not. Data type connotes whether an indicator value is an integer, double, Boolean, text, or multiple choice. We suspect that the grammar users require more training on grammar before they are able to successfully select the correct indicator type and data type. It could also suggest that the grammar should be improved, but, since these are semantic mistakes, we assume that clarifying the meaning of the rules should suffice.

"Missing element" and "extra unnecessary element" mean that the test subject has omitted an element that should have been included in the model or that an element has needlessly been added. These types of mistakes suggest that the subjects in general have not had enough training on how textual grammars work. The concept of how rules can trigger other rules in textual grammar is not yet clear for test subjects that made these types of mistakes. These syntactic mistakes can be avoided by including the editor. This also goes for other syntactic mistakes such as the "capital letter", "data type format", and "rule order" mistakes because the editor will not allow those. Lastly, to avoid the "Overlapping order numbers" mistake we should add a constraint to the *Text Fragment* and *Question* primitives.

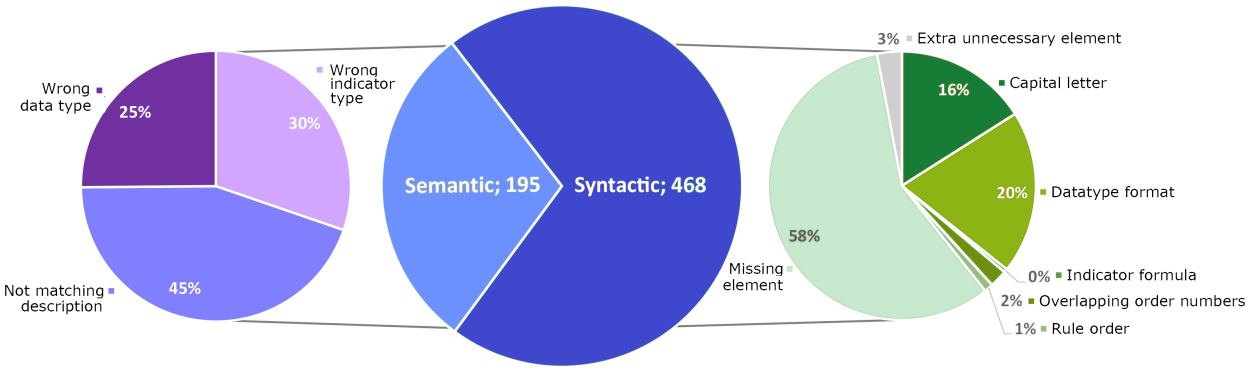


Figure 12. The types of mistakes that participants made in the tasks

Perceptions and Intentions. The results of the perceptions and intentions questionnaire yield Figure 13. It shows whether participants found the grammar easy to use and useful. Additionally, it gives some indication of their intention to use the grammar in combination with a model-driven interpreter, if they become a sustainability officer after their studies. The majority of the participants have indicated that they found it easy to make small changes in the models, understand the models, and create models. Only 24% of the participants found that creating ESEA method models required a lot of mental effort. For perceived usefulness, 67% of the participants felt like using grammar would save them time when engineering ESEA methods, and 48% said that using grammar would increase their productivity. The majority of the participants have indicated that they are willing to use openESEA in combination with the grammar if they ever become an ESE accountant.

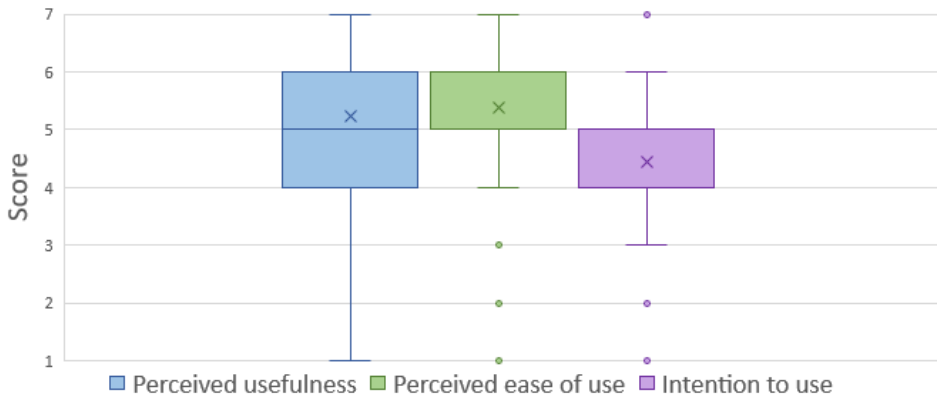


Figure 13. The MEM questionnaire results (n = 62)

Quality of the Grammar. To further assess the quality of the grammar we identified relevant qualities based on [45]. To increase the **physical quality** of the modeling language, the participants are provided with a textual description of the grammar primitives. Moreover, the grammar contains comments that briefly explain the rules.

The **empirical quality** of the grammar is improved with syntax highlighting. In the grammar, data types, ids, and strings are highlighted with different colors. The attributes all start on a new line. Even though the font is the same throughout the grammar, certain concepts are written in bold and italics.

Pragmatic quality is not explicitly included in the scope of the experiment. One way to increase it is by adding comments and guidelines in the grammar specification. The use of the editor would most probably increase the pragmatic quality of the modeling language by including error messages, auto-completion, and additional syntax highlighting.

8 Discussion

8.1 Interpretation of Results

To increase the expressiveness of the DSL we added new concepts to the metamodel. This, of course, increases the complexity of the DSL. However, by means of a user test, we found that users can manage this complexity since they can create models successfully. The user test allowed us to identify four major points of improvement. At least one of these points can be resolved by introducing the editor, two points can be addressed with additional training, and one point requires an extension of the Xtext validator. We consider the user test overall success and deem that, overall, users can effectively create ESEA method models. The success rate and degree of correctness are high and the observed mistakes are mostly non-critical. Based on the user test results we formulate the following key findings about the openESEA modeling language.

Key finding 1: V2 of the openESEA textual grammar does not require any major improvements in terms of ease of use and usefulness. Naturally, the expressiveness of the language remains a point of improvement and will increase with each new release of the openESEA framework, in order to support an increasing number of ESEA method fragments.

Key finding 2: The complexity of using the openESEA textual grammar is manageable, given the promising results of the user study.

Key finding 3: To improve the performance of using the textual grammar, longer training sessions on ESEA and Xtext grammar are needed.

The only model-driven approach for ESEA we found is our own earlier work [7]. The modeling language presented herein builds upon the artifacts from that article, therewith all extensions of these artifacts are new contributions to the model-driven ESEA approach. For testing an Xtext DSL we only found one article that also performed a user test. Jonathan et al. developed a DSL and syntax checker in Xtext for producing mapping configuration files for ASML's Twinscan machine. To test the artifacts, a component test, code review, and user test were performed [46]. Similar to our user test, test subjects were asked to create configuration files. The number of scientific works that describe how to test Xtext DSLs is limited. The test procedure in this article can contribute to an approach for testing Xtext DSLs.

8.2 Validity and Replicability

We have done our best to mitigate the threats to validity, but we acknowledge that the threat to external validity remains. To be more specific, there is a threat to population validity because the set of participants in our user test might not mirror the population of future openESEA users. Though we expect that openESEA users will have similar technical skills as our sample set, future users will be more knowledgeable on sustainability topics. We recognize that this influences the validity of our research, though we do not deem the validity to be compromised because the complexity of using the openESEA modeling language may be less for sustainability experts. Another limitation caused by population validity is that the group of students is not a representative group to measure the intention to use, since they do not (yet) work in the domain of ESEA. Consequently, they might not be able to imagine whether they would use such a tool. Anyhow, in earlier (and ongoing) expert assessment interviews, ESEA experts show appreciation and interest in our approach [7].

We have tried to eliminate the threat to content validity by employing the thoroughly validated Method Evaluation Model [33]. Moreover, we have included a diverse set of grammar primitives in our user test to ensure that every primitive was properly tested. To evaluate whether the questions in our user test were understandable before executing the user test, we performed a pilot test. Based on the pilot test we have reformulated and clarified some questions.

We support the open science movement, hence we publish all our research data in a publicly available data repository [47]. Moreover, we have compiled a technical report containing the PDDs

of the 13 ESEA methods we analyzed [34], and a technical report that presents the entire modeling language [35]. Lastly, the openESEA tool is completely open-source and can be found in the Github repository [10]. By publishing all our artefacts we improve the reproducibility and replicability of our research.

8.3 Ethical Considerations

Since we embedded the user test in a course, we had to strike a balance between our experimental aims and creating an engaging and meaningful learning experience. Time investment and efforts of the students had to be taken into account and we had to find an approach for converting their performance into grades. In terms of the experimental aims, we would have preferred to dedicate more lectures to training the students and to have a user test with more questions per grammar primitive. If the test had contained more questions per primitive, the productivity of the students might have improved, assuming that the increase in experience would have progressed their way forward on the learning curve. This in turn, may have resulted in more accurate time estimations for creating a model from scratch. On the other hand, increasing the size of the user test could also have led to fatigue, resulting in worse performance.

We surveyed the students to discover whether we had compromised the learning goals of our course by embedding a user test. Fortunately, we can conclude that this was not the case. The user test assignment was rated as the best assignment in the course. The question on whether the assignment taught the students something new received a score of 4.5 out of 5. Relevance to the course was rated 4.1 out of 5. The students deemed the assignment to be very interesting (4.6 out of 5) and they also appreciated the workload (4.4 out of 5). All in all, we are content with the user test design and results. After concluding the course we also gathered qualitative feedback from the students and we learned that they found the experience of learning about textual grammar valuable. Moreover, they deemed the exercises that were part of the user test a suitable form of assessment. We even had six students approach us for bachelor and master graduation projects on the openESEA modeling language. A large subset of students also enrolled in a follow-up master course on Responsible ICT, where the openESEA modeling language is discussed in more detail. We consider these actions as evidence of a positive impact on the education of our students.

8.4 Future Work

Developing and evolving the openESEA framework is a complex and continuous project, that is split up into several engineering cycles. Figure 14 displays major milestones we plan for evolving our model-driven approach for ethical, social, and environmental accounting. Every milestone is achieved by performing a problem investigation, treatment design, and treatment validation. The development of the framework started with the conceptualization of the idea. Thereafter, the first versions of the openESEA DSL and interpreter were developed and presented in [7]. This first version was extended with a feature that generates infographics of ESE accounts [48]. OpenESEA V2, which we present in this article, has undergone major architectural improvements, making the back end more robust. Moreover, the DSL has been extended with several new concepts, making V2 more versatile than V1. For V3 we plan to extend the DSL to support the auditing of ESE accounts and we intend to extend the interpreter with features that allow auditors to manage the audit. Next, we plan to develop a model management approach, which allows the matching and merging of ESEA method models (openESEA V4). We also aim to implement Business Intelligence techniques that allow dashboard generation based on ESE accounts. Finally, we have so far opted for engineering textual grammar to support the DSL. We envision a combination of textual and graphical modeling of ESEA method models since a graphical specification of the process dimension of methods may be more intuitive and we already have experience in modeling ESEA

[10] https://github.com/sergioespana/openESEA

methods with PDDs (Figure 14). Lastly, the openESEA technology will be used by researchers and practitioners such as social enterprises and municipalities as part of the *Boosting Social and Community-driven Entrepreneurship for the Transition to an Inclusive and Sustainable Society* (SCENTISS) project.

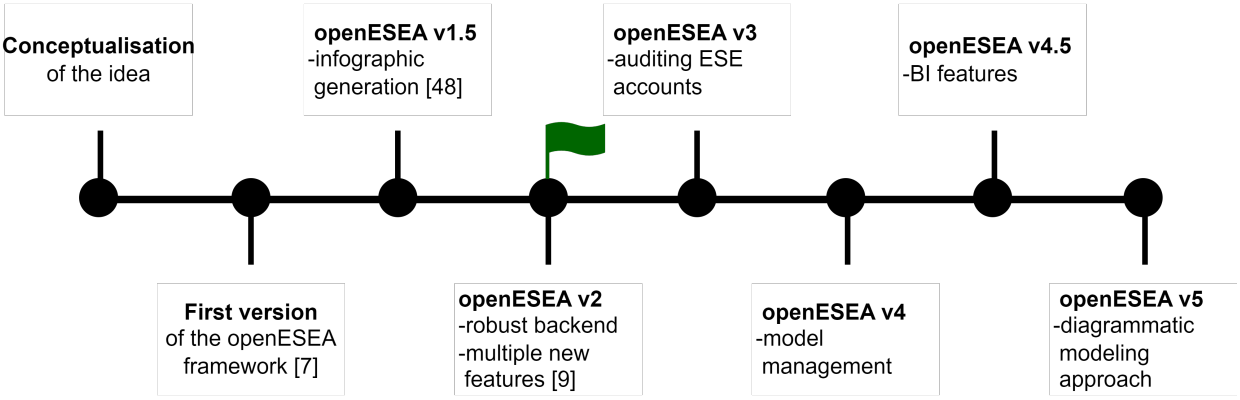


Figure 14. The milestones we envision in the openESEA framework. The flag marks the version we present in this article.

9 Conclusion

With our modeling language organizations can create an ethical, social, and environmental accounting method model that combines all the methods that they wish to apply (e.g., B Impact Assessment, Common Good Balance Sheet, and GRI Standards). By uploading the textual model (which contains three methods) in our open-source, model-driven interpreter, called openESEA, the tool parses and interprets the model. It displays all surveys, questions, topics, indicators, and other necessary elements to execute the three methods specified in the textual model. This novel approach reduces the complexity of managing ESEA methods and eliminates the redundancy caused by the use of rigid ICT tools that are developed to support one method only.

In this article, we contribute an updated modeling language for ethical, social, and environmental accounting methods. The modeling language consists of a metamodel and textual grammar. To design the grammar we have used improvement points from [7], and analyzed additional ESEA methods. We have tested our approach with a user test. The results are promising and form the foundation for further development. We intend that the textual grammar can be used to express any ESEA method, although this remains to be proven. We will continue adding new elements to the modeling language to improve its versatility, while also trying to limit its complexity.

With our work, we expect to simplify and improve the ESEA ICT tool support and hopefully encourage organizations to perform an ESEA. By measuring, reporting, and monitoring ESE performance and impacts, organizations can become more responsible and sustainable entities, therewith improving their business legitimacy.

References

[1] V. Smith, J. Lau, and J. Dumay, "Shareholder use of CSR reports: an accountability perspective," *Meditari Accountancy Research*, 2021. [Online]. Available: https://doi.org/10.1108/MEDAR-02-2020-0769

[2] J. Chung and C. H. Cho, "Current trends within social and environmental accounting research: A literature review," *Accounting Perspectives*, vol. 17, no. 2, pp. 207–239, 2018. [Online]. Available: https://doi.org/10.1111/1911-3838.12171

[3] E. Costa, L. D. Parker, and M. Andreaus, "The rise of social and non-profit organizations and their relevance for social accounting studies," in *Accountability and Social Accounting for Social and Non-Profit Organizations*. Emerald Group Publishing Limited, 2014, vol. 17, pp. 3–21. [Online]. Available: https://doi.org/10.1108/S1041-706020140000017003

[4] E. L. Lane, "Green marketing goes negative: The advent of reverse greenwashing," *European Journal of Risk Regulation*, vol. 3, no. 4, pp. 582–588, 2012. [Online]. Available: https://doi.org/10.1017/S1867299X00002506

[5] R. Gray, C. A. Adams, and D. Owen, *Accountability, social responsibility and sustainability*. Pearson, 2014.

[6] C. A. Adams and C. Larrinaga-González, "Engaging with organisations in pursuit of improved sustainability accounting and performance," *Account. Audit. Account. J.*, 2007.

[7] S. España, N. Bik, and S. Overbeek, "Model-driven engineering support for social and environmental accounting," in *RCIS*. IEEE, 2019, pp. 1–12. [Online]. Available: https://doi.org/10.1109/RCIS.2019.8877042

[8] H. Behrens, M. Clay, S. Efftinge, M. Eysholdt, P. Friese, J. Köhlein, K. Wannheden, S. Zarnekow, and contributors, "Xtext user guide," *Eclipse Fundation*, 2010.

[9] V. Ramautar and S. España, "Managing the complexity in ethical, social and environmental accounting: Engineering and evaluating a modelling language," in *7th Workshop on Managed Complexity, BIR-WS 2022*, 2022, pp. 88–103.

[10] F. Beske, E. Haustein, and P. C. Lorson, "Materiality analysis in sustainability and integrated reports," *Sustain. Account. Manag. Policy J.*, 2020. [Online]. Available: https://doi.org/10.1108/SAMPJ-12-2018-0343

[11] A. Calabrese, R. Costa, N. Levialdi Ghiron, and T. Menichini, "Materiality analysis in sustainability reporting: a tool for directing corporate sustainability towards emerging economic, environmental and social opportunities," *Technological and Economic Development of Economy*, 2019. [Online]. Available: https://doi.org/10.3846/tede.2019.10550

[12] R. Gray and J. Bebbington, "Accounting for the environment," *Sage*, p. 13, 1993.

[13] G. Lamberton, "Sustainability accounting – a brief history and conceptual framework," in *Accounting forum*, vol. 29, 2005, pp. 7–26. [Online]. Available: https://doi.org/10.1016/j.accfor.2004.11.001

[14] S. Sisaye, "The influence of non-governmental organizations (NGOs) on the development of voluntary sustainability accounting reporting rules," *JBSED*, 2021. [Online]. Available: https://doi.org/10.1108/JBSED-02-2021-0017

[15] S. S. Gao and J. J. Zhang, "Stakeholder engagement, social auditing and corporate sustainability," *BPMJ*, 2006. [Online]. Available: https://doi.org/10.1108/14637150610710891

[16] P. Castka and M. A. Balzarova, "A critical look on quality through CSR lenses: Key challenges stemming from the development of ISO 26000," *IJQRM*, 2007. [Online]. Available: https://doi.org/10.1108/02656710710774700

[17] V. Paelman, P. Van Cauwenberge, and H. Vander Bauwhede, "Effect of B Corp certification on short-term growth: European evidence," *Sustainability*, vol. 12, no. 20, p. 8459, 2020. [Online]. Available: https://doi.org/10.3390/su12208459

[18] A. Singh and M. Chakraborty, "Does CSR disclosure influence financial performance of firms? Evidence from an emerging economy," *SAMPJ*, 2021. [Online]. Available: https://doi.org/10.1108/SAMPJ-02-2018-0042

[19] R. Küchler and C. Herzig, "Connectivity is key: holistic sustainability assessment and reporting from the perspective of food manufacturers," *British Food Journal*, 2021. [Online]. Available: https://doi.org/10.1108/BFJ-03-2021-0317

[20] G. Figueiredo, A. Duchardt, M. M. Hedblom, and G. Guizzardi, "Breaking into pieces: An ontological approach to conceptual model complexity management," in *RCIS*, 2018, pp. 1–10. [Online]. Available: https://doi.org/10.1109/RCIS.2018.8406642

[21] D. L. Moody, "Complexity effects on end user understanding of data models: An experimental comparison of large data model representation methods," *ECIS*, p. 10, 2002.

[22] M. E. Manso, M. Genero, and M. Piattini, "No-redundant metrics for UML class diagram structural complexity," in *CaiSE*. Springer, 2003, pp. 127–142. [Online]. Available: https://doi.org/10.1007/3-540-45017-3_11

[23] P. Forbrig, "Managing complexity with heterogeneous modeling." in *BIR Workshops*, 2018, pp. 245–250.

[24] R. Petrusel and J. Mendling, "Eye-tracking the factors of process model comprehension tasks," in *CaiSE*. Springer, 2013, pp. 224–239. [Online]. Available: https://doi.org/10.1007/978-3-642-38709-8_15

[25] R. A. Buchmann and D. Karagiannis, "Modelling mobile app requirements for semantic traceability," *Requir. Eng.*, vol. 22, no. 1, pp. 41–75, 2017. [Online]. Available: https://doi.org/10.1007/s00766-015-0235-1

[26] R. J. Wieringa, *Design science methodology for information systems and software engineering*. Springer, 2014. [Online]. Available: https://doi.org/10.1007/978-3-662-43839-8

[27] M. Mernik, J. Heering, and A. M. Sloane, "When and how to develop domain-specific languages," *ACM Comput Surv*, vol. 37, no. 4, pp. 316–344, 2005. [Online]. Available: https://doi.org/10.1145/1118890.1118892

[28] I. van de Weerd and S. Brinkkemper, "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. [Online]. Available: https://doi.org/10.4018/978-1-59904-887-1.ch003

[29] I. van de Weerd, S. de Weerd, and S. Brinkkemper, "Developing a reference method for game production by method comparison," in *Working Conf. on Method Engineering*. Springer, 2007, pp. 313–327. [Online]. Available: https://doi.org/10.1007/978-0-387-73947-2_24

[30] G. Lucassen, F. Dalpiaz, J. M. E. Van Der Werf, and S. Brinkkemper, "Forging high-quality user stories: towards a discipline for agile requirements," in *2015 IEEE 23rd international requirements engineering conference (RE)*. IEEE, 2015, pp. 126–135. [Online]. Available: https://doi.org/10.1109/RE.2015.7320415

[31] OMG, "Unified Modeling Language (OMG UML), Version 2.5.1," 2017.

[32] ISO/IEC, "International vocabulary of metrology – Basic and general concepts and associated terms (VIM)," ISO/IEC Guide 99:2007, Standard, 2007.

[33] D. L. Moody, "The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods," in *ECIS Proc.*, 2003.

[34] V. Ramautar and S. España, "Domain Analysis of Ethical, Social and Environmental Accounting Methods," Utrecht University, Tech. Rep., 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2208.00721

[35] V. Ramautar and S. España, "The openESEA Modelling Language for Ethical, Social and Environmental," Utrecht University, Tech. Rep., 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2205.15279

[36] S. Brinkkemper, "Method engineering: engineering of information systems development methods and tools," *Inf Softw Technol*, vol. 38, no. 4, pp. 275–280, 1996. [Online]. Available: https://doi.org/10.1016/0950-5849(95)01059-9

[37] G. Brunet, M. Chechik, S. Easterbrook, S. Nejati, N. Niu, and M. Sabetzadeh, "A manifesto for model merging," in *Proceedings of the 2006 international workshop on Global integrated model management*, 2006, pp. 5–12. [Online]. Available: https://doi.org/10.1145/1138304.1138307

[38] M. M. Lankhorst, H. A. Proper, and H. Jonkers, "The anatomy of the archimate language," *International Journal of Information System Modeling and Design (IJISMD)*, vol. 1, no. 1, pp. 1–32, 2010. [Online]. Available: https://doi.org/10.4018/jismd.2010092301

[39] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Science, 2012. [Online]. Available: https://doi.org/10.1007/978-3-642-29044-2

[40] N. Prat, I. Comyn-Wattiau, and J. Akoka, "Artifact evaluation in information systems design-science research–a holistic view," in *PACIS proc. 23*, 2014.

[41] S. España, N. Condori-Fernandez, A. González, and Ó. Pastor, "An empirical comparative evaluation of requirements engineering methods," *Journal of the Brazilian Computer Society*, vol. 16, pp. 3–19, 2010. [Online]. Available: https://doi.org/10.1007/s13173-010-0003-5

[42] F. Gailly and G. Poels, "Experimental evaluation of an ontology-driven enterprise modeling language," in *Advances in Conceptual Modeling. Recent Developments and New Directions: ER 2011 Workshops FP-UML, MoRE-BI, Onto-CoM, SeCoGIS, Variability@ ER, WISM, Brussels, Belgium, October 31-November 3, 2011. Proceedings 30*. Springer, 2011, pp. 163–172. [Online]. Available: https://doi.org/10.1007/978-3-642-24574-9_22

[43] N. Juristo and A. M. Moreno, *Basics of software engineering experimentation*. Springer, 2013. [Online]. Available: https://doi.org/10.1007/978-1-4757-3304-4

[44] L. Bettini, *Implementing domain-specific languages with Xtext and Xtend*. Packt, 2016.

[45] J. Krogstie, *Model-based development and evolution of information systems: A Quality Approach*. Springer, 2012. [Online]. Available: https://doi.org/10.1007/978-1-4471-2936-3

[46] B. Jonathan, R. Avetyan, and S. Abeln, "Create Domain-Specific Language and Syntax Checker Using Xtext," *International Journal of Industrial Research and Applied Engineering*, vol. 4, no. 1, pp. 26–32, 2020. [Online]. Available: https://doi.org/10.9744/jirae.4.1.26-32

[47] V. Ramautar and S. España, "openESEA user test raw data," Utrecht University, Tech. Rep., 2023. [Online]. Available: https://doi.org/10.17632/wvy9fmx3cx.1

[48] S. España, V. Ramautar, S. Overbeek, and T. Derikx, "Model-driven production of data-centric infographics: An application to the impact measurement domain," in *Research Challenges in Information Science: 16th International Conference, RCIS 2022, Barcelona, Spain, May 17–20, 2022, Proceedings*. Springer, 2022, pp. 477–494. [Online]. Available: https://doi.org/10.1007/978-3-031-05760-1_28