

A Process and Tool Support for Human-Centred Ontology Verification

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering und Internet Computing

eingereicht von

Klemens Käsznar, BSc

Matrikelnummer 11776832

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Reka Marta Sabou, MSc PhD

Mitwirkung: Dr.techn. Mag. Fajar Juang Ekaputra

Wien, 3. Juni 2022

Klemens Käsznar

Reka Marta Sabou



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.



A Process and Tool Support for Human-Centred Ontology Verification

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

Klemens Käsznar, BSc

Registration Number 11776832

to the Faculty of Informatics

at the TU Wien

Advisor: Reka Marta Sabou, MSc PhD

Assistance: Dr.techn. Mag. Fajar Juang Ekaputra

Vienna, 3rd June, 2022

Klemens Käsznar

Reka Marta Sabou



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Erklärung zur Verfassung der Arbeit

Klemens Käsznar, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 3. Juni 2022

Klemens Käsznar



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Marta Sabou. She provided continuous guidance, valuable feedback and scientific advice during all stages of this thesis which helped to improve the quality of my work.

Furthermore, I want to thank everybody for their valuable insights during the interview and discussion sessions as part of my research. Moreover, I would like to express my appreciation to Stefani Tsaneva for sharing her experience in the field in several meetings going beyond the interview and discussion sessions.

Last but not least, I am indebted to my parents, my sister, my brother and my friends for always supporting me in every possible way they could.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Abstract

Ontologies explicitly capture domain knowledge in machine-readable formats and act as semantically rich knowledge sources for information systems. Detecting misrepresented knowledge by ontology verification is crucial for avoiding malfunctioning systems, as their decisions rely on correct knowledge. While certain classes of ontology errors can be detected automatically through reasoning, some error classes require human involvement for being solved and are therefore addressed by *human-centred ontology verification*.

Human Computation (HC) is a resource-effective solution to human-centred ontology verification because it avoids employing highly skilled domain experts and engineers. However, a systematic mapping study in this area shows that the process of using HC to solve human-centred ontology verification (i) is not well understood as there is no reference process that is widely used and that (ii) there is no widely-accepted tool support available and most authors rely on the ad-hoc use of a handful of very diverse tools.

To address these gaps, this thesis contributes to better understanding the typical process performed during human-centred ontology verification and the possibilities of supporting it with a tool. To that end, a design science methodology is used to make the following contributions. First, an iterative approach, comprising a systematic literature review, semi-structured interviews and a focus group, defines “VeriCoM 2.0”, a set of three process models describing human-centred ontology verification. Second, a reference architecture featuring four viewpoints is established to enable the implementation of an end-to-end process support platform for “VeriCoM 2.0”. Third, the contributions also include a prototypical implementation of an extensible platform based on the reference architecture. Fourth, a case study evaluates the created artefacts to understand to what extent the preparation of human-centred ontology verification can be supported.

The evaluation of the case study shows that the process models are a helpful tool to plan, conduct and communicate a human-centred ontology verification. Furthermore, the prototypical implementation can support eleven out of nineteen preparation activities of the verification process. Comparing the time effort of implementing the prototypical platform, and thus automating the preparation of the verification, to the time effort of preparing the same verification manually, shows that 29.47% less effort is required for the implementation. An additional comparison reveals that by reusing the prototypical platform and solely customising it to the same verification task, time efforts can be reduced by 85.33% with respect to a manual preparation of the verification task.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Mithilfe von Ontologien kann Wissen in maschinenlesbaren Formaten erfasst und kodiert werden, um als Informationsbasis für Softwaresystem zu dienen. Ein entscheidender Beitrag zur fehlerfreien Funktion solcher Systeme ist die Ontologieverifikation, mit welcher Fehler in Ontologien identifiziert werden sollen. Während ein Teil der Fehler automatisiert durch “Reasoning” identifiziert werden kann, müssen für den verbleibende Teil Menschen in den Prozess involviert werden.

Ein effizienter Ansatz, um letztere Art von Fehler zu identifizieren, stellt “Human Computation” dar, da weder Fachexperten/Fachexpertinnen noch Entwickler/Entwicklerinnen involviert werden müssen. Allerdings ist der Prozess der Ontologieverifikation basierend auf “Human Computation” Prinzipien nicht klar definiert und es gibt nur eine begrenzte Anzahl an Softwaretools, welche Teile des Prozesses automatisieren.

Ziel dieser Arbeit ist es, den Prozess und die Möglichkeiten einer softwarebasierten Unterstützung zu verstehen. Zu Beginn wird der Prozess mithilfe eines iterativen Ansatzes bestehend aus einer systematischen Literaturrecherche, semi-strukturierten Interviews und einer Gruppendiskussion, modelliert. Darauf aufbauend, wird eine Referenzarchitektur entworfen, auf deren Basis Softwareplattformen zur vollständigen Prozessunterstützung implementiert werden können. Abschließend werden alle Artefakte mittels einer Fallstudie, welche eine prototypische Implementierung der Referenzarchitektur einschließt, evaluiert, um festzustellen, inwiefern die Vorbereitungen solcher Ontologieverifikationen unterstützt werden können.

Die Beiträge dieser Arbeit sind wie folgt: (1) “VeriCoM 2.0”, bestehend aus drei Prozessmodellen, (2) eine Referenzarchitektur mit vier Perspektiven, (3) eine Implementierung einer erweiterbaren Plattform zur Prozessunterstützung und (4) eine Fallstudie zur Evaluierung der Artefakte.

Die Evaluierung der Fallstudie zeigt, dass die Prozessmodelle hilfreich für die Planung, Durchführung und Kommunikation von Ontologieverifikationen sind. Darüber hinaus können elf von neunzehn Vorbereitungsaktivitäten durch Softwaretools unterstützt werden. Für die Implementierung eines Software Prototypen, und somit Automatisierung der Vorbereitungsschritte, wird um 25,47% weniger zeitlicher Aufwand benötigt als für die manuelle Durchführung derselben Vorbereitungsschritte. Wird nur der Zeitaufwand notwendiger Anpassungen an der Softwareplattform betrachtet, ergibt sich eine Aufwandsreduktion um 85,33%.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Contents

Abstract	ix
Kurzfassung	xi
Contents	xiii
1 Introduction	1
1.1 Motivation and Problem Definition	1
1.2 Research Questions	2
1.3 Methodology and Contributions	3
1.4 Thesis Structure	6
2 Background and Related Work	7
2.1 Ontology Verification	7
2.2 Human Computation	11
2.3 Human Computation and Conceptual Model Verification	12
2.4 Summary	18
3 Understanding the Process of Human-Centred Ontology Verification	21
3.1 Systematic Literature Review: Collecting an Initial Set of Process Activities	21
3.2 Semi-structured Interviews: Eliciting Activities From Experts	33
3.3 Focus Group: Defining the Final Process Model	45
3.4 Human-centred Ontology Verification Process Model	51
3.5 Summary	60
4 Support-Platform Reference Architecture	61
4.1 Human-centred Ontology Verification Terminological Hierarchy	61
4.2 Reference Architecture	64
4.3 Summary	77
5 Case Study: Supporting Human-Centred Ontology Verification	79
5.1 Case Description	79
5.2 Evaluation Approach	81
5.3 Implementation	82
	xiii

5.4	Evaluation Results	87
5.5	Summary	99
6	Conclusion & Future Work	101
6.1	Conclusions of the Research Questions	102
6.2	Limitations & Future Work	103
	List of Figures	107
	List of Tables	109
	Acronyms	111
	Bibliography	113
	Appendices	123
	Appendix A: SLR Summaries	123
	Appendix B: SSI Invitation EMail	141
	Appendix C: SSI One Pager	142
	Appendix D: SSI Interview Guide	143
	Appendix E: Focus Group Discussion Guide	146
	Appendix F: Individual Process Models	151

Introduction

1.1 Motivation and Problem Definition

Modern information systems, such as recommender systems, search systems or chatbots, rely on rich, high-quality information sources to provide their services. *Ontologies* can act as such information sources by capturing, modelling and formalising domain information explicitly. An *ontology* in computer science is defined as “a formal, explicit specification of a shared conceptualization” [1].

Capturing and formalising information in an ontology is typically either approached manually by employing human experts or automatically by executing computer algorithms. To guide these activities, commonly referred to as *ontology engineering*, a variety of processes and methodologies can be followed [2]. However, mistakes, even when following methodological approaches, cannot be avoided, thus leading to potential mistakes in ontologies.

Identifying and correcting such mistakes in ontologies is crucial to ensure the correct functioning of ontology-based systems. Thus these tasks, typically referred to as *ontology verification*, have become an important part of research. Due to the machine-readable formats used and strong roots in formal logics of ontologies, certain errors, such as logical inconsistencies, can be easily detected by applying an automated process/tool, i.e. a reasoner. At the same time other errors, for example, errors requiring domain knowledge, cannot be detected in an automated matter. Concluding, human input is typically needed to perform thorough ontology verification.

To include humans in the ontology verification process, best summarised as *human-centred ontology verification*, typically domain and/or ontology experts are employed to conduct the verification. These approaches can potentially lead to high resource usage of many kinds, such as monetary or temporal. A promising approach to address these challenges is provided by *Human Computation (HC)* techniques.

Using HC, problems that cannot yet be solved solely by computers are addressed by using an interplay of humans and computers [3]. Using these techniques, a problem is broken down into small tasks, termed *micro-tasks*, that are typically solvable by non-experts. Such micro-tasks commonly only feature a simple question answered by a binary or predefined selection of options. Next, these micro-tasks are published on *crowdsourcing platforms*, such as Amazon Mechanical Turk (AMT)¹, where a large population of *workers* accepts and completes them. Due to the sheer availability of labour force on these platforms, commonly these micro-tasks are executed redundantly, to collect several judgements for one micro-task and to harness the *wisdom of the crowd*. As a final step, aggregation algorithms (e.g. majority voting) are used to aggregate the redundant judgements and a final set of answers is derived for all micro-tasks.

While understanding the generic process for the application of HC techniques is straightforward, applying these techniques to ontology verification is more challenging. Several publications, for example [4] or [5], already address concrete applications of HC to solve human-centred ontology verification tasks. However, the processes and activities, especially those involved before publishing the *micro-tasks*, are often not elaborated in detail and thus not yet understood in a systematic way. In addition, *ontology engineers*, and other stakeholders, such as *information system developers*, lack a unified tool-chain to support them during the execution of a *human-centred ontology verification*.

To address these two gaps, this master thesis aims at (1) systematically understanding the processes and activities of *human-centred ontology verification*, (2) providing a platform that supports *human-centred ontology verifications* and (3) evaluating the extent to which such a well-defined process model and its associated tooling can support ontology engineers. Summarising, the problem to be addressed is at the intersection of the domain of *ontology verification* and the use of *Human Computation (HC)* as a solution thereof.

1.2 Research Questions

To address the problem outlined in the previous section, three research questions have been identified.

RQ1: *What is the typical process of human-centred ontology verification?*

More specifically earlier work [6] introduced the “VeriCoM” approach to verify conceptual domain models (similar to ontologies) using HC (as further detailed in Section 2.3.4). While this is a first step in the direction of understanding the process, extensions and adaptations to the proposed “VeriCoM” process should be identified to make it suitable for human-centred ontology verification as ontologies are a specific type of conceptual structures. A concrete result by answering this research question shall include a systematic and thorough understanding of the process executed, the activities performed and the tools used during human-centred ontology verification. Additionally, a process model to

¹<https://www.mturk.com/>

visualize and communicate the process, which shall be referred to as “VeriCoM 2.0”, is also expected as an outcome of this research question.

RQ2: *What are key requirements for software modules and a reference architecture that automate/support the “VeriCoM 2.0” process?*

To implement an end-to-end process support platform for the “VeriCoM 2.0” process, first, a common set of vocabulary shall be established. Then, the main requirements shall be identified based on the results of **RQ1**. Finally, based on these outcomes, a reference architecture for an end-to-end process support platform for “VeriCoM 2.0” shall be defined.

RQ3: *To what extent does an implementation of the reference architecture support the preparation of human-centred ontology verification?*

Based on the process model defined by **RQ1** and the reference architecture established by **RQ2**, it is important to understand how ontology engineers can benefit from the proposed formulations and automatizations. Therefore the main expected outcome shall include a prototype of an extensible platform, not tied to an existing tool, implementing a subset of the requirements, especially those focusing on preparatory work, as identified earlier. Additionally, an evaluation shall provide insights to what extent the preparation of human-centred ontology verification can benefit from an unified tool support provided by the implemented platform.

1.3 Methodology and Contributions

As an overall approach to the elaborated problem, the *Design Science methodology for information systems research* as proposed by Hevner et al. [7] is followed. On a high-level view, IT artefacts shall be created and evaluated to solve identified organisational problems.

Figure 1.1 depicts the three main cycles of the methodology as elaborated in [8]. The *Design Cycle*, iterating between a develop and evaluation phase, is considered the core element of this research method. That way feedback is collected and the artefacts generated can be refined. In addition to the *Design Cycle* also the *Relevance Cycle* and *Rigor Cycle* need to be addressed. Following enumeration outlines how each of the cycles are addressed.

- *Design Cycle:* The overall *Design Cycle* ensures artefacts are built and evaluated throughout applying this methodology. Due to the scope of this thesis, it is only expected for one such design cycle to be completed.

As elaborated in Section 1.2, investigating each research questions leads to *research contributions* in terms of information artefacts, as follows:

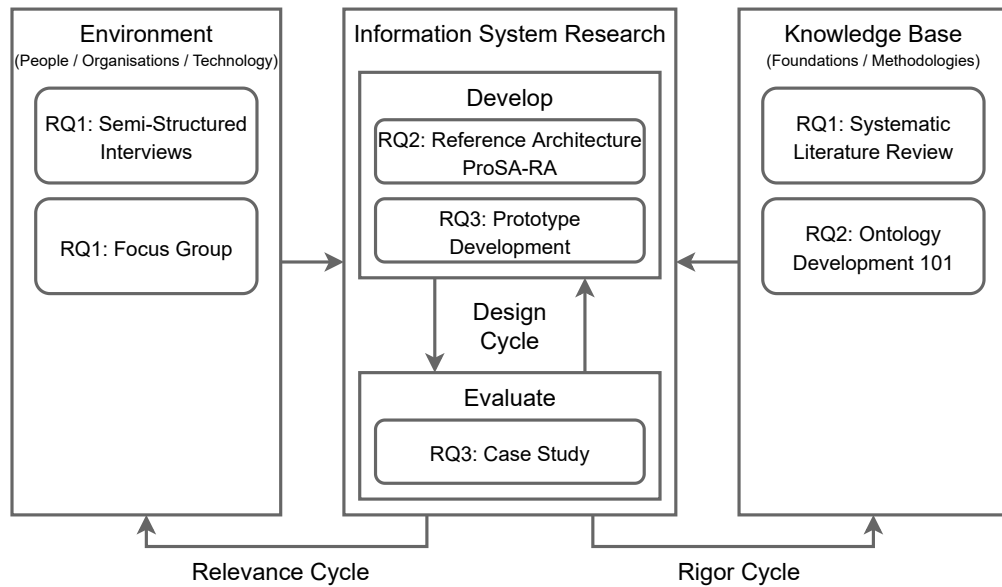


Figure 1.1: Overview of the thesis methodology which relies on Design Science cycles and various methods specific to each research question. Adapted and extended from [7, Figure 2].

- “VeriCoM 2.0” a process model and description of the human-centred ontology verification process (**RQ1**).
- A reference architecture for an end-to-end process support platform for conducting human centered ontology verification (**RQ2**).
- An instantiation of the reference architecture and its evaluation in a case study about a concrete real-life example of human-centred ontology verification (**RQ3**).

- *Relevance Cycle*: The *Relevance Cycle* guarantees that the solution to be developed is relevant to a business problem. In the context of this thesis, the problem can be broken down into two sub-problems. First, there is no systematic understanding of the *human-centred ontology verification* process. Second, no tools supporting the process activities are available. The relevance of the given (sub-)problem(s) is ensured by the following aspects:

- Existing literature is reviewed to gain an initial understanding of the process of *human-centred ontology verification*, to identify requirements for the reference architecture and ensure future work will benefit from the artefacts created
- Stakeholders in need of tool support for *human-centred ontology verification* are identified and interviewed

- *Rigor Cycle*: Within a design science research project, all artefacts created and activities involved should be based on existing knowledge and theories. To incorporate knowledge from existing literature, data collected by a Systematic Mapping Study (SMS) found in [9] will be used. More specifically results will be narrowed down to foster the understanding of the human-centred ontology verification process, “VeriCoM 2.0”.

In addition to the overall *Design Science methodology*, the following methods are used for each research question:

Methods RQ1: To model the process and activities of human-centred ontology verification, a mixed approach is applied to collect data. The mixed approach is composed of conducting *semi-structured interviews* [10], a group discussion following the focus group methodology [11] as well as execution of the data analysis phase of a *systematic literature review* [12].

Following the topology presented by Leech and Onwuegbuzie [13] the approach to be applied can be classified as *partially mixed concurrent dominant status design*. The classification can be explained as follows. Within the scope of this thesis, more emphasis is put on collecting qualitative data (i.e. from the Semi-structured Interviews (SSIs) and the focus group) as opposed to the quantitative data provided by the systematic literature review. Further, both phases are executed concurrently as literature might provide inputs for the interviews and vice versa.

As required by the semi-structured interview, an interview guide as proposed in [10] is created upfront. For the focus group, also a guide and supporting materials are designed. The systematic literature review is meant to give inputs for all materials created.

Conducting an entire systematic literature review would exceed the scope of this thesis, therefore the corpus of literature of an ongoing SMS[9] is narrowed to target RQ1. Further, a data extraction form is created to be used to enrich the existing data of the SMS[9] with information about process steps and tools employed.

Finally, once the data has been collected, a process model is created to visualize and communicate “VeriCoM 2.0”.

Methods RQ2: Once “VeriCoM 2.0” has been defined, the requirements for the end-to-end process support platform are formulated and a reference architecture is established. Particularly, the *ProRA-SA* approach [14] is followed to specify the reference architecture.

Further, following the *Ontology Development 101* methodology [15], a taxonomy is created to define the vocabulary used for the reference architecture.

Methods RQ3: Before actually evaluating to what extent the end-to-end process support platform supports ontology engineers a *prototype* is implemented. The prototype is implemented following the reference architecture defined by RQ2 and focuses on functionality related to preparatory activities of the human-centred ontology verification

process. Important design aspects of the prototype include its standalone characteristics (i.e. not being tied to an existing editor or tool) and its possibility for extension.

The evaluation of the support platform is approached empirically by a *case study* [16, Chapter 5 Case Studies]. In collaboration with ontology engineers, a real-life ontology verification case is defined and it is evaluated to what extent the platform can provide support for such a case. Typically case studies do not allow to control the environment as tightly as experiments do, however, due to their rather flexible design strategy they allow to collect quantitative (i.e. time-savings) as well as qualitative (i.e. rework needed in addition) data to assess the support provided by the platform from multiple perspectives. In general, data collection as proposed in [16] is expected to include interviews and comparison to a baseline, thus providing the intended mix of quantitative and qualitative data needed to answer *RQ3*.

1.4 Thesis Structure

This thesis is structured as follows:

- Chapter 2 Background and Related Work provides background information, reports related work and puts this work into context.
- Chapter 3 Understanding the Process of Human-Centred Ontology Verification addresses **RQ1** and establishes an understanding of the activities and processes during *human-centred ontology verification* (“VeriCoM 2.0”) by conducting a semi-structured literature review, interviews and a focus group.
- Chapter 4 Support-Platform Reference Architecture focuses on **RQ2** and discusses a reference architecture for end-to-end process support platforms for human-centred ontology verifications.
- Chapter 5 Case Study: Supporting Human-Centred Ontology Verification reports on a case study as part of **RQ3**, that involves a prototypical implementation of an end-to-end process support platform for human-centred ontology verification and an evaluation thereof.
- Chapter 6 Conclusion & Future Work summarizes the main findings of this thesis and discusses suggestions for future work.

Background and Related Work

This thesis is situated in the problem space of ontology verification. On the other hand, the main contributions of the thesis focus on applying Human Computation (HC) techniques to support human-centred ontology verification, thus the former can be seen as the solution space of the thesis. Consequently, an understanding of both ontology verification as well as Human Computation (HC) when working on this thesis is required.

To address this aspect, this section discusses the following aspects in more detail. First, in Section 2.1 the problem space of ontology verification is introduced. Second, the solution space Human Computation (HC) and applications thereof are presented in Section 2.2. Finally, Section 2.3 addresses applications of the solution space in the problem space by discussing literature on human computation and conceptual model verification, as ontologies are a special type of conceptual models.

2.1 Ontology Verification

Ontology verification, which focuses on assessing the correctness of an ontology, can be seen as a sub-discipline of ontology evaluation. To present a complete picture of the problem domain, first, an introduction to ontology evaluation is provided by summarizing a referenced survey, then ontology verification, emphasizing human-centred ontology verification, is elaborated in more detail.

2.1.1 Ontology Evaluation

Ontology evaluation is the discipline concerned with assessing selected quality aspects of an ontology [17]. In their survey, Brank et al. [17] identified three different use-cases of ontology evaluation. First, ontology evaluation can help users choose the best ontology for their use case. Second, ontology engineers can be assisted during ontology construction to guide the overall process and the refinement of an ontology. Finally, similarly to

2. BACKGROUND AND RELATED WORK

the first use case, ontology evaluation can help choose the best ontology out of several automatically or semi-automatically constructed ontologies.

Next to the uses cases, the survey [17] identified four different categories of evaluation approaches:

- Comparison to a gold standard [18]
- Using the ontology through an application [19]
- Comparison with a data source to assess domain coverage [20]
- Assessment by humans given predefined criteria, standards, requirements [21]

Finally, Brank et al. [17] outline that ontology evaluation is often addressed on different levels. The survey identified the following six different ontology evaluation levels:

- Lexical, vocabulary or data layer
- Hierarchy or taxonomy
- Other semantic relations
- Context of application level
- Syntactic level
- Structure, architecture, design

By constructing a two-dimensional matrix as shown in Table 2.1, the authors show that human assessment is the only approach feasible to address all evaluation levels.

Level	Approach to evaluation			
	<i>Gold Standard</i>	<i>Application-based</i>	<i>Data-driven</i>	<i>Assessment by humans</i>
<i>Lexical, vocabulary, concept, data</i>	X	X	X	X
<i>Hierarchy, taxonomy</i>	X	X	X	X
<i>Other semantic relations</i>	X	X	X	X
<i>Context, application</i>		X		X
<i>Syntactic</i>	X			X
<i>Structure, architecture, design</i>				X

Table 2.1: Matrix showing which approaches are suitable for certain evaluation levels. Source: [17, Table 1]

Thus, indicating the importance of human involvement in certain ontology evaluation tasks and therefore also in ontology verification tasks.

In addition to the definition of ontology evaluation, the authors present selected approaches from literature which address certain evaluation levels.

In the context of this thesis the survey of [17] is considered sufficient as it introduces the field and is among the most popular surveys in the field, considering its citations. For further reference, pointers to other surveys in the field are provided. Further survey papers addressing ontology evaluation can be found by Raad and Cruz [22], focusing on the classification of ontology evaluation, and by Hlomani and Stacey [23], focusing on the state of the art in the field.

2.1.2 Human-Centred Ontology Verification

According to [24] ontology verification deals with building the ontology correctly, concerning an ontology’s requirements and competency questions and can be seen as part of ontology evaluation. As already briefly touched on earlier and illustrated by the matrix in Table 2.1, an evaluation of ontologies on the level of *structure, architecture and design* can only be addressed by human involvement. Concretely, this evaluation level is concerned with ensuring compliance with pre-defined design principles/criteria, the organisation of the ontology and the possibility for future extension [25].

To illustrate typical mistakes found in ontologies that require human involvement, consider the “Pizza Ontology¹” which is widely used for didactic purposes [26]. For example, one aspect often misunderstood is the open-world reasoning used by semantic web ontologies and tools. More specifically, missing information is not considered to be absent information, as it would be for example with databases. Modelling the only two toppings, *Tomato* and *Mozzarella*, of a *Margherita* pizza only using either (1) existential restriction axioms (`owl:someValuesFrom`) or (2) universal restriction axioms (`owl:allValuesFrom`) will lead to either (1) all pizzas which have *Tomato*, *Mozzarella* and other toppings or (2) pizzas without any topping or only one (i.e. *Mozzarella* or *Tomato*) to be falsely classified as a *Margherita* pizza. To address this modelling mistake and to ensure that in the model a *Margherita* has exactly only *Tomato* and *Mozzarella* topping, a combination of the existential and the universal restriction axioms need to be used.

In addition to the example above, the importance of human involvement in ontology verification tasks can also be deduced from [27]. With their work [27], they provide two important contributions. The first contribution encompasses a catalogue of common defects found in ontologies and the second contribution includes an online tool for automatically detecting a subset of these defects.

The initial version of the catalogue [27] was constructed by assessing literature on ontology- and linked data-evaluation and by manually analyzing ontologies, which then resulted in 40 different pitfalls. As intended by the authors the catalogue shall be extended if needed, which indeed has happened. When reviewing the current version of the catalogue², one pitfall (i.e. *P41. No license declared*) was added since the initial publication.

¹<https://protege.stanford.edu/ontologies/pizza/pizza.owl>

²<http://oops.linkeddata.es/catalogue.jsp>

Based on the catalogue the authors implemented *OOPS! (Ontology Pitfall Scanner!)*³, which is the online tool that allows the identification of certain pitfalls in semantic web ontologies. More specifically, currently, only 33 out of the 41 pitfalls can be automatically detected. For the remaining eight pitfalls human judgement is needed, thus highlighting the importance of human-centred ontology verification as addressed by this thesis.

According to the catalogue, the following defects cannot be detected automatically (the following enumeration is based on [27] and the online catalogue ⁴):

- **P01 - Creating polysemous elements:** refers to ontological elements which represent more than one domain concept
- **P09 - Missing domain information:** are pitfalls representing required information not being included in the ontology
- **P14 - Misusing “owl:allValuesFrom”:** indicates a confusion of the universal quantification restriction and the existential quantification restriction
- **P15 - Using “some not” in place of “not some”:** indicates misused existential quantifier restrictions and negative operators
- **P16 - Using a primitive class in place of a defined one:** indicates that classes are defined using `rdfs:subClassOf` instead of `owl:equivalentClass`
- **P17 - Overspecializing a hierarchy:** suggests that leaf classes are too specific such that no instances can be created for it
- **P18 - Overspecializing the domain or range:** refers to a domain or range not considering all possible characteristics of the conceptualisation
- **P23 - Duplicating a datatype already provided by the implementation language:** are pitfalls related to classes or types which are already provided by the formal language and are re-implemented by the ontology

Considering the elaborated list of pitfalls requiring human judgement, it can be observed that those need to be verified as well to ensure high-quality ontologies and subsequently correct functioning information systems depending on them. For example, if an ontology contains pitfalls, such as *P18*, it might happen that instances cannot be created due to the implemented axioms. Thus, the ontology cannot be used to reflect the domain and systems depending on the ontology might be malfunctioning.

Concluding on [27], the pitfall catalogue and the online tool *OOPS!* foster the importance of human-centred ontology verification and tools to support it.

³<http://oops.linkeddata.es/index.jsp>

⁴<http://oops.linkeddata.es/catalogue.jsp>

An important work with regards to human-centred ontology verification is the master thesis of Tsaneva [28]. Among other things, the author focused on gaining a systematic understanding of human-centred ontology verification tasks by conducting a literature review. The literature review identified several aspects of ontology verification tasks, in particular including aspects of human-centred ontology verification tasks. Similar to this thesis, the literature review is part of the same Systematic Mapping Study (SMS), which focuses on human-centred evaluation of semantic resources. For more information about [9], Section 3.1.1 can be consulted.

2.2 Human Computation

To address the problem space of human-centred ontology verification, Human Computation (HC) is seen as the solution space in the context of this thesis. Law and Von Ahn [29] provide the following definition of the research field “*Human computation is a new and evolving research area that centers around harnessing human intelligence to solve computational problems that are beyond the scope of existing Artificial Intelligence (AI) algorithms.*” In fact, considering the elaboration on human-centred ontology verification tasks in the previous chapter, a similar pattern of solving a problem that requires human intelligence, as is not yet solely solvable by algorithms, can be observed, thus indicating a promising intersection of the human-centred ontology verification problem space and the human computation solution space.

A survey on human computation [30] identifies Von Ahn as one of the early sources of inspiration for HC usage. In one of the first publications [3], Von Ahn motivates HC by outlining the following three successful applications:

- **Labelling Images:** Creating image captions, for example, required for accessibility, is a very laborious and costly task if done by experts. With the application of HC techniques, this task was outsourced to players of an online game [31]. The game itself showed major popularity among players after its release and helped collect useful image captions.
- **Locating Objects in Images:** Similarly to labelling images, a more granular task is labelling objects in images. Typically, such data is needed for machine learning workloads to act as training data. Employing a dedicated workforce to complete this task is rather costly, thus another game [32], based on HC principles was introduced to solve the task.
- **Digitizing Books:** Analogue books or news articles are hard to digitize as scans often show distorted texts and thus Optical Character Recognition (OCR) fails to detect the texts correctly. For humans, on the other hand, this task is considered trivial. By employing CAPTCHA, a mechanism implemented on web pages to make a distinction between humans and computers, this task can be solved in a fashion. Indeed, a follow-up publication of the approach [33], reports an accuracy of 99.1%, indicating the power of such approaches.

Summarizing the publications of Von Ahn, Human Computation (HC) can be seen as a powerful approach to solving various problems not yet solvable by computers. Further, its applications come in various flavours including but not limited to embedding the task in another task or embedding the task in a game (i.e. Games with a Purpose (GWAP)).

In addition to the aforementioned flavours, another popular approach to human computation is crowdsourcing [34]. To that end, typically crowdsourcing platforms, such as Amazon Mechanical Turk (AMT)⁵ or Appen⁶, are facilitated to publish micro-tasks, referred to as Human Intelligence Tasks (HITs), which are perceived simple by humans but unsolvable by computers, to be completed by a crowd of online workers.

Diving further into these platforms, within [34] the authors identified a set of categories of answers to be collected on crowdsourcing platforms. One category of answers is referred to as *commonsense*, where humans are asked to solve a task that requires information about the world that a computer cannot possess. This type of answer or task is especially interesting for this thesis, as human-centred ontology verification tasks often require the evaluators as well to rely on their knowledge about the world to assess the correctness of an ontology. This further indicates, that HC is a promising approach for completing certain ontology verification tasks. The next section presents relevant literature on applying HC techniques to ontology verification and conceptual model verification.

2.3 Human Computation and Conceptual Model Verification

Ontology verification can be seen as an instantiation of the verification of a conceptual model and thus the latter can be seen as the core discipline addressed by this thesis. First, Sections 2.3.1, 2.3.2 and 2.3.3 present selected literature at the intersection of the problem space (i.e. ontology verification) and the solution space (i.e. Human Computation (HC)), resembling cases of human-centred ontology verification. Second, literature focusing on the application of HC on conceptual model verification, thus providing a broader scope on the topic, is outlined in Section 2.3.4.

2.3.1 Verifying Ontology Hierarchies

An important series of publications [4, 35, 36, 37] in the context of using human computation techniques to perform ontology verification was conducted by Mortensen et al. More specifically, their work is concerned with verifying the correctness of parent-child relationships by publishing crowdsourcing tasks on AMT.

As outlined in [4], a publication summarizing their previous experiments and findings, the following questions were investigated:

1. “Is crowdsourcing ontology verification feasible?” detailed in [35]

⁵<https://www.mturk.com/>

⁶<https://appen.com/solutions/crowd-management/>; formerly also known as CrowdFlower

2. “What is the optimal formulation of the verification task?” detailed in [36]
3. “How does this crowdsourcing method perform in an application?” detailed in [37]

To address each of the questions in detail, the authors used the same experiment/task design and typically only changed one variable to investigate the effects on the overall outcome. The task published on AMT for each experiment asked questions like *Every Dog is a Mammal*. and the workers were expected to provide a binary answer (i.e. either *TRUE* or *FALSE*) [35].

In order to understand the feasibility of publishing such tasks/questions on AMT to verify ontological relationships, the authors experimented with different ontologies in [35]. Specifically, they tested two upper-level ontologies, namely *BWW (Bunge-Wand-Weber)*[38] from the business process modelling domain and *SUMO (Suggested Upper Merged Ontology)*[39] representing general-purpose terms, one ontology representing common-sense terms and relationships based on *WordNet*⁷ and finally a domain- and application-specific ontology, *CARO (Common Anatomy Reference Ontology)*[40] reflecting information from the anatomy context. The experiments showed, that crowd workers can perform similar to undergraduate students or domain experts if some context/additional information for the relationship to be verified is provided.

Next, in [36] the optimal formulation of the relationship verification task is investigated. To that end, the authors experimented with different question formulations, whether to provide context or not and whether to use qualification tests. Again the common-sense ontology extracted from *WordNet*⁸ and the specialised anatomy ontology *CARO*[40] were used. The results showed that the polarity of a question (i.e. positive or negative formulation) depends on the concrete instance to be verified while using indicative answers (i.e. *TRUE* or *FALSE*) are assumed to provide better results. As for providing context and including qualification tests, beneficial effects on quality are observed if those are included.

Finally, in [37] the crowdsourcing-based verification approach was applied to the popular biomedical ontology *SNOMED CT*. Within the previously outline publications and experiments, wrong relationships were artificially generated to conduct the experiments and assess the worker performance. In [37] in contrast, no errors were introduced artificially, rather existing errors identified by other authors in *SNOMED CT* [41] were asked to be identified by the crowd workers. By employing a Bayesian interference-based aggregation algorithm, the crowd was able to nearly reproduce the same set of errors as found by other authors [41] in *SNOMED CT*.

Concluding on this array of work, it can be seen that ontology engineering tasks can be completed with HC techniques while achieving similar quality when compared to experts undergraduate students and other authors in the field, in fact, the difference is shown

⁷<https://wordnet.princeton.edu/>

⁸<https://wordnet.princeton.edu/>

to be statistically not relevant. However, these publications are limited to verifying relationship hierarchies only, thus not targeting a generic approach of human-centred ontology verification. As the feasibility of applying HC to ontology verification tasks is shown by this array of work, this thesis is meant to broaden the scope of the verification tasks by investigating and proposing a more generic process.

2.3.2 Verifying Ontology Restrictions

Apart from the Systematic Literature Review (SLR) provided in [28] and also briefly touched on in Section 2.1.2, the author also implemented an approach to verify ontology restrictions. In particular, the usage of the *universal quantification* and the *existential quantification* restrictions were investigated.

Once the common errors related to these restrictions are discussed, the HC approach is introduced. The HITs focus on verifying these restrictions in the *Pizza Ontology*⁹ by asking the crowd workers to select one out of five different pre-defined judgements (c.f. Figure 5.1). These judgements indicate correctness of the axiom, absence of one restriction or the need for a replacement of one restriction. In addition, various forms of contextual information, as well as representations of the relations to be verified, can be included for each task.

With the evaluation, the authors focused on understanding the impacts of different formalisms as well as visualisations and the impacts of previous knowledge in the field. The evaluation was conducted as a controlled experiment using AMT and by employing a crowd of university students. The results indicated that the majority (i.e. 92.58%) of the collected responses were correct. Further, the results indicated that the VOWL formalisms[42], as well as that prior modelling knowledge, impact the outcome positively.

Concluding on the HC approach for verifying ontology restrictions in [28], again the feasibility of employing HC techniques to ontology verification, as also outlined for a different verification aspect in Section 2.3.1, is demonstrated. Further and also similar to the approach outlined in Section 2.3.1, only one aspect of human-centred ontology verification is considered, thus this thesis is meant to complement this by providing a generic understanding of the human-centred ontology verification process.

2.3.3 Protégé Plugin for Human-centred Ontology Verification

Wohlgenannt et al. [5] approached human-centred ontology verification by two means. First, they tried to understand what ontology engineering tasks are already solved by crowdsourcing approaches. Second, a plugin, namely “*uComp*”, for the popular ontology engineering tool *Protégé*¹⁰ was implemented, which allows a set of ontology verification tasks to be published on crowdsourcing platforms directly from the engineering tool.

⁹<https://protege.stanford.edu/ontologies/pizza/pizza.owl>

¹⁰<https://protege.stanford.edu/>

Based on a literature study the authors identified the following set of crowdsourced ontology engineering crowdsourcing (enumeration based on [5]):

- **T1. Specification of Term Relatedness:** For this type of task, the crowd workers are asked to specify if two terms are related to each other.
Example: Are “cat” and “dog” related to each other?
- **T2. Verification of Relation Correctness:** Based on an already provided relation between two concepts, the workers are asked to verify its correctness.
Example: Is a “cat” “a type of” “human”?
- **T3. Specification of Relation Type:** With these tasks, as opposed to T2 the workers are asked to specify the relationship between two concepts.
Example: Which of the following relations: “is a type of”, “is the same as”, describes the relation between “animal” and “dog” best?
- **T4. Verification of Domain Relevance:** A concept and a domain are shown and the workers judge whether the concept is relevant for the domain or not.
Example: Is “Mammal” a relevant term in the domain of “Dogs”?

Based on the identified crowdsourced ontology engineering tasks the *Protégé* plugin, “*uComp*”, was implemented. The tasks supported by the plugin include assessing the domain relevance, verifying the correctness of subsumption relations or instances and specification of relation types. The tool also allows publishing the crowdsourcing tasks and collecting the answers directly to the editor.

As part of their work [5], they conducted two groups of evaluations, one focusing on the feasibility and the other one on the scalability of using the plugin for crowdsourcing ontology engineering tasks.

With the feasibility evaluations, the focus was on identifying time and cost savings, as well as the quality of the obtained work by employing a group of domain experts. It is shown that time and thus also cost-saving across various tasks can be achieved by using the plugin when compared to manually conducting the tasks.

As the feasibility evaluation does not target the full potential of crowdsourcing platforms, that a big workforce is available 24 hours a day, the second set of scalability evaluations addressed this aspect. The evaluation showed that significant cost savings can be achieved and that the tasks can potentially be faster completed using crowdsourcing, while also yielding considerable high quality results.

While the plugin shows promising results and parallels to this thesis, it lacks two aspects that shall be addressed by this thesis. First, only a very high-level view of the process of conducting crowdsourcing engineering tasks is provided. Within this thesis, one of the contributions includes a detailed process model of human-centred ontology verification thus addressing this aspect. Further, the implemented tool is tightly integrated into an

existing tool, thus hampering the portability to other environments and also limiting the application scenarios. To address this aspect, the end-to-end process support platform implemented by this thesis acts as a standalone component that can be accessed via a well-defined API.

2.3.4 VeriCoM: Verifying Conceptual Models Approach

Moving to a broader scope than ontologies, conceptual model verification is considered next. Important work in this context was conducted by [6] and [43]. Within [6] conceptual model verification was generalized, formalized and eventually applied to detect errors in Extended Entity Relationship (EER) diagrams, while [43] extends this work and experimented with different task designs for EER diagram verification.

More detailed, the main contributions of [6] include (1) a formalization of conceptual model verification to enable conducting such verification across several areas and communities, (2) an approach describing the execution of conceptual model verification using human computation techniques, referred to as Verifying Conceptual Models (VeriCoM), and (3) an experiment-based evaluation of VeriCoM for verifying EER diagrams using a textual specification document.

The proposed formalization includes the following core elements:

- *Conceptual Domain Model M* : a conceptual domain model which is subject of verification
- *Set of Model Elements ME* : each model M is composed of a set of model elements ME , which in turn can be further divided into subsets depending on the concrete application area
- *Frame of Reference FR* : some form of reference, such as a specification document or general human knowledge, providing input to verify the conceptual model
- *Expected Model Elements EME* : elements to be expected in the model based on the information provided by FR
- *Evidence of an EME EV_{eme}* : evidence justifying the existence of an EME based on a FR or a specification
- *Set of Defects D* : a set of errors which are identified during the model verification

Following this formalization, conceptual model verification of a model M using a frame of reference FR to detect a set of defects D is a function γ as follows:

$$\gamma(M, FR) \rightarrow D \quad (2.1)$$

Using this formalization the authors present a generic human computation approach, namely *Verifying Conceptual Models (VeriCoM)*, to verify conceptual models using a textual specification involving four stages as depicted in Figure 2.1.

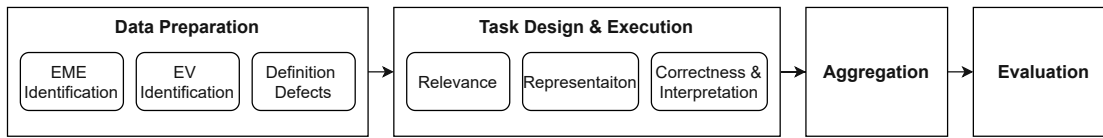


Figure 2.1: Main stages of VeriCoM.

In an initial *Data Preparation* stage, the *EMEs* which are found in the context of the textual specification, as well as their evidence are identified. Further, depending on the context of the verification, the defect types need to be defined to guide the creation of task interfaces.

Then, after the preparation is done, the human computation task needs to be defined and executed. The *Task Design & Execution* should guide the workers through three steps during execution: first judge whether an *EME* is relevant, next locate the element in the model and finally assess the correctness and interpretations of the model elements.

Lastly, as typically redundant results are collected during human computation tasks, the results need to be *aggregated* and finally, an *evaluation* yields insights about the quality of the obtained work (i.e. the set of defects *D*).

Also within [6], the authors evaluated the generic VeriCoM approach in a controlled experiment targeting identifying defects in a EER diagram. For preparation, the authors seeded in known defects to the EER diagram and thus had a gold standard on hand. Over the course of four different workshops, students executed the human computation tasks as shown in Figure 2.2.

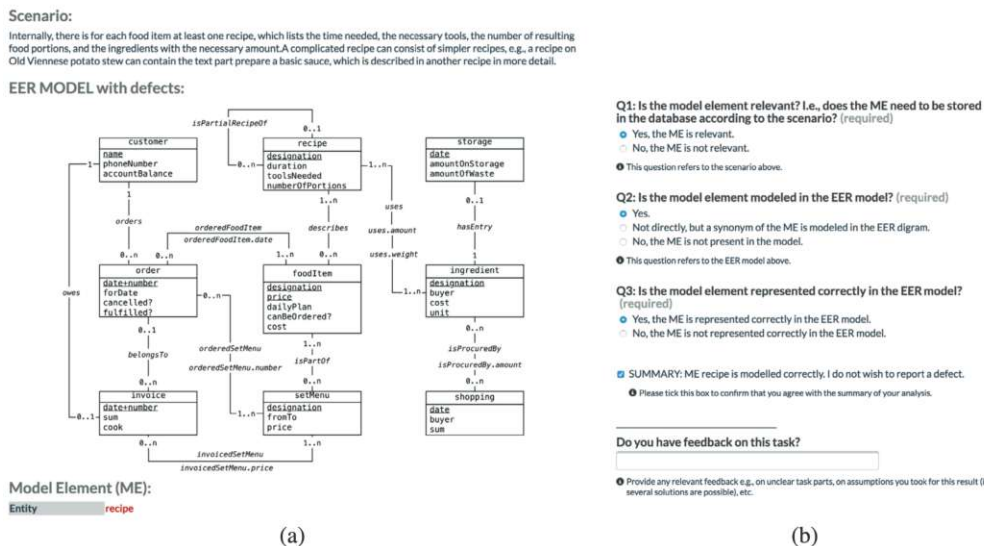


Figure 2.2: HIT interface used by [6] a) showing data related to the model and b) showing the verification questions. Source: [6, Figure 2]

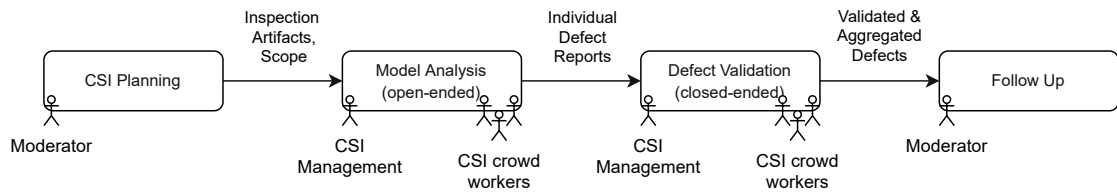


Figure 2.3: Main stages of CSI.

Then the results were compared against a gold standard. The experiment showed promising results for EER diagram verification and the overall VeriCoM approach.

In their follow up work [43], the authors extended their previous work with an additional *closed task* step to ease aggregation, as within [6] *open tasks* (i.e. workers were asked to submit their answers following a controlled language in free text) were used which are typically harder to aggregated than *closed tasks* (i.e. workers were asked to provide answers from a given selection). This extended task design, referred to as Crowdsourced Software Inspection (CSI) and based on [44], is composed of four stages as depicted in Figure 2.3. As the approach is based on VeriCoM[6] the elements *Data Preparation*, *Aggregation* and *Evaluation* thereof can be found in the CSI process under *CSI Planning* and *Follow Up*. For the *Task Design & Execution* stage of VeriCoM, two explicit steps, namely (1) *Model Analysis* and (2) *Defect Validation* can be observed in CSI. Within (1) *Model Analysis* the workers are asked to identify errors using an open question design, whereas in (2) *Defect Validation* the workers validate the open answers from the (1) and assess whether a defect can be considered a real defect or not.

Similarly to [6], this extended approach presented in [43] was evaluated using a controlled experiment. It is shown that combing these two different task designs (i.e. open and closed) aggregation can be eased and the workers are more creative when submitting their answers.

To conclude, [6, 43] form an important foundation of this thesis, as conceptual model verification is addressed on a generic level and the problem itself is formalized, such that approaches for more specific applications of conceptual model verification, such as ontology verification, can be developed. Further, it can be observed that the experiments outline the preparation process for EER diagrams in detail (especially those conducted in [6]). However, as *ontologies* are structurally different (e.g. they include logical based axioms or hierarchies) from EER diagrams the preparation activities are different and need to be determined by this work.

2.4 Summary

To conclude this chapter the main aspects found during the literature study are outlined. First, *Ontology Verification* as part of *Ontology Evaluation* and as the problem space of this thesis is already being studied in the literature. Work on ontology evaluation and also

more specific work on ontology verification, such as *OOPS!*[27], reveal the importance of human involvement in the evaluation and verification process as selected aspects can only be addressed by humans. A corpus of studies requiring human involvement is also used in Chapter 3 as part of the SLR for which summaries of the reviewed studies can be found in Appendix A: SLR Summaries.

Second, Human Computation (HC) itself as part of the solution space is also well-studied in literature for various kinds of problems such as labelling images. However, specific applications of HC techniques to solve ontology verification tasks are rare. Specific approaches include verifying ontology hierarchies or verifying ontology restrictions.

Finally, the process of human-centred ontology verification is not yet well-studied in literature and a gap can be identified. Further, also the tools supporting human-centred ontology verification are very rare and those that exist (e.g. “*uComp*”[5]), are tightly tied to existing platforms hampering their portability and usability. Thus, the literature study strengthens the need for a process model and an end to end tool support.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Understanding the Process of Human-Centred Ontology Verification

A vital aspect for the implementation of an end-to-end process support platform for human-centred ontology verification lies in the understanding of the process itself, which is reflected by **RQ1**.

To establish this process understanding, an iterative approach is applied. First, Section 3.1 presents a SLR and defines an initial set of activities based on the corpus of literature from a related SMS. Then, using the techniques of SSIs as presented in Section 3.2, a group of experts is interviewed to elicit their views on the process. Finally, the data from the SLR and SSIs are combined during a focus group as discussed in Section 3.3 and the final process model “VeriCom 2.0” is defined in Section 3.4.

3.1 Systematic Literature Review: Collecting an Initial Set of Process Activities

A foundation for understanding the process for human-centred ontology verification and for crafting a well founded semi-structured interview guide is established by conducting parts of a Systematic Literature Review (SLR) focused on collecting process related information. As defined in [12] a SLR uses “*a well-defined methodology to identify, analyse and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable*”.

As part of another research project in the domain of this thesis, an ongoing Systematic Mapping Study (SMS)[9] is conducted investigating human-centred evaluation of semantic

resources. In the context of [9] semantic resources encompass ontologies, linked data datasets and knowledge graphs. Since the SMS overlaps with the problems addressed by this thesis, it forms a starting point for the research investigations of this thesis connected to the *Rigor Cycle* of the Design Science method (c.f. Figure 1.1).

Typically a SLR consists of three phases: (1) planning the review (c.f. Section 3.1.1), (2) executing the review and (c.f. Section 3.1.1) (3) reporting the review (c.f. Section 3.1.4). Since this thesis targets solely human-centred ontology verification, a subset of the publications identified by the SMS[9] is extracted to form the corpus of relevant literature for the herein conducted SLR. Thus parts of the planning, as well as execution of the SMS[9] will be reused, and solely the reporting (including data analysis) phase, on a reduced set of literature specifically targeting studies focusing on ontologies or ontological structures, is conducted extensively within the scope of this thesis.

3.1.1 Systematic Mapping Study Planning and Execution

Within the planning phase of the aforementioned SMS[9] on human-centred evaluation of semantic resources, the research questions and the search queries were defined. For the step of executing the study, several digital libraries were queried and the results were further refined. The following paragraphs provide a more detailed overview of these phases of the SMS[9] used as a basis for the SLR of this thesis.

Planning: First, in the context of the SMS[9] five research questions were defined targeting:

1. the characteristics of the evaluated semantic resources,
2. the goal of the evaluation,
3. the context and setting of the evaluation,
4. the characteristics of the human evaluators, and
5. the methodological and tooling aspects of the evaluation.

Based on (1) the research questions of the SMS[9] itself, (2) a set of seed papers and (3) related studies, three clusters of search keywords as presented in Table 3.1 were identified. In the context of [9], using the definition of the search keywords, literature is considered relevant if it matches at least one keyword from each of the three clusters.

Table 3.1: Clusters of keywords used by the SMS (adapted from [9])

Cluster	Keywords
C1 (Semantic resource)	knowledge graph, linked data, ontolog*, OWL, RDF, semantic web, SPARQL, vocabular*

Table 3.1 continued from previous page

Cluster	Keywords
C2 (Human involvement)	crowd*, expert evaluation, expert review, expert sourcing, game*, gamification, GWAP, human computation, human in the loop, layman, laymen, microtask, user evaluation, user study, user testing
C3 (Evaluation task)	anomaly, assessment, bias, defect*, error*, evaluat*, pit-fall*, quality, refinement, validat*, verif*

Execution: Figure 3.1 provides an overview of the execution phase as conducted by the SMS.

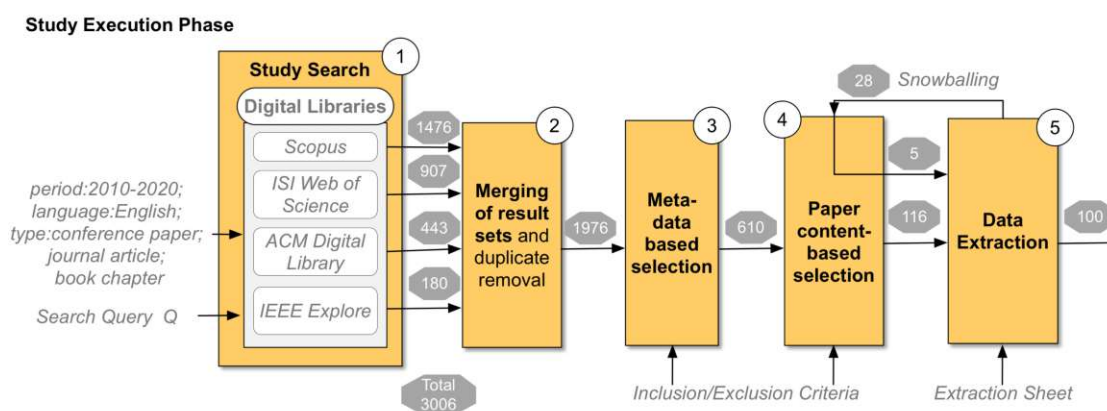


Figure 3.1: Overview of the SMS execution phase. Source: [9]

As a starting point of [9], the query was executed on the following digital libraries: WebOfScience¹, ACM Digital Library², IEEE Xplore³ and Scopus⁴ to obtain an initial corpus of relevant literature. Based on the keywords and further search restrictions (i.e. search period, language, search scope and publication type) a corpus of 3006 matching publications was found. Through duplication elimination, the corpus was reduced to 1976 distinct studies.

In a subsequent two-fold study selection step, the corpus was further reduced to 116 studies. The first selection decision was based on the meta-data (including title, abstract and keywords) of each publication based on predefined inclusion and exclusion criteria. In the second selection step, the remaining publications were selected based on a review of their entire content.

¹<https://www.webofknowledge.com/>

²<https://dl.acm.org/>

³<https://ieeexplore.ieee.org/>

⁴<https://www.scopus.com/>

As a final step, data extraction was performed using a predefined form. Throughout this step, some additional papers were identified by snowballing while other papers were excluded, resulting in a total of 100 papers for analysis. The concrete outcome of the SMS in [9], which forms the basis for further steps of the SLR of this thesis, includes but is not limited to a) a corpus of relevant literature and b) already populated data extraction forms.

3.1.2 Customised Data Extraction

The SMS in [9] has a much broader focus than this thesis and especially **RQ1**, thus some customisation, as outlined next, needs to be applied to tailor the data extraction steps towards process activities and tooling of *human-centred ontology verification*.

First, a set of SLR-specific research questions is defined, which can be seen as a more granular view on **RQ1**:

- **SLRQ1:** What are the steps involved in human-centred ontology verification?
- **SLRQ2:** What tools typically support the process?

The aim of both SLR-specific research questions combined is to serve as an initial foundation for answering **RQ1**. More specifically, a set of activities and their supporting tools and/or libraries used alongside the process of human centre ontology verification shall be identified.

Second, the extraction form initially used for the referenced SMS is extended to reflect **SLRQ1** and **SLRQ2** of the SLR herein. Table 3.2 presents the name, description, expected type of answer and rationale for the inclusion of the eleven fields added to the existing form. The majority of the additional fields (i.e. fields 3 to 10) is focused on extracting relevant activities and tools alongside the preparation, execution and follow-up phase of the human-centred ontology verification found in the studies. While the remaining fields (i.e. fields 1,2 and 11) target collecting general information which can be used for later reference (e.g. when assembling the requirements specification) or for further filtering. Generally, the data form is populated by reading the full publications, thus being in line with the approach of the initial SMS.

Table 3.2: Fields added to the data extraction form of the SMS.

Name	Description	Answer	Rationale
1) Targets ontology evaluation	Whether the ontology is the target of the evaluation or other aspects such as the tool.	Binary (yes/no)	Enable further filtering
2) Notes on evaluation	Capturing interesting and/or additional aspects of the evaluation.	Free text	Highlighting interesting facts
3) Outlines preparation	Ordinal subjective measure how much detail about the human-centred evaluation is outlined.	Ordinal from 0 to 3	Enable further filtering; Relevance assessment
4) Preparation activities	List of activities related to the preparation of human-centred ontology verification.	List of activities	Identify set of activities
5) Preparation tools	List of tools used to realise the preparation activities.	List of tools	Identify set of tools

Table 3.2 continued from previous page

Name	Description	Answer	Rationale
6) Execution activities	List of activities related to the execution of human-centred ontology verification.	List of activities	Identify set of activities
7) Execution tools	List of tools used to realise the execution activities.	List of tools	Identify set of tools
8) Includes screenshot	Whether screenshots of the evaluation interface are included or not.	Binary (yes/no)	Act as further reference
9) Follow-up activities	List of activities related to follow-up steps of human-centred ontology verification.	List of activities	Identify set of activities
10) Follow-up tools	List of tools used to realise the follow-up activities.	List of tools	Identify set of tools
11) Notes	Any arbitrary information found in the publication.	Free text	Capturing interesting aspects

Third, as the corpus of literature extracted by the referenced SMS in [9] includes various kinds of semantic resources and not solely ontologies as targeted by this thesis, further filtering is applied. More specifically, the corpus of literature is filtered using the existing extraction forms on the field *CODED D8d Resource Type (reviewer)*, capturing the evaluated type of resources, for the following contents:

- The studies evaluation targets the *TBox of an ontology* (i.e. field value *Ontology (TBox)*) or
- The studies evaluation targets semantic triples (i.e. field value *Ontology triples*)

The rationale for extending the filter criteria also to include ontology triples is twofold. On one hand, larger ontologies can be broken down into triples, thus semantic triples forming sub-parts of ontologies, and on the other hand, those studies are more likely to include detailed information about the evaluation as often HIT focus on smaller structured as opposed to whole ontologies.

To conclude, the customised data extraction processes includes SLR-specific research questions, an extended extraction form and additional filtering by the content of the SMS extraction forms.

3.1.3 Data Analysis

Once the literature is filtered, read and the data is extracted, the data needs to be analysed to answer **SLRQ1** and **SLRQ2**. The analysis approach consists of three main phases, (1) *collecting quantitative statistical data*, (2) *coding the data* and (3) *deriving answers for SLRQ1 and SLRQ2*, as detailed in the following paragraphs.

To gain an overview of the collected data, the following *quantitative statistical data* is calculated from the extraction forms:

- Size of initial corpus of literature: The initial count of literature from [9]
- Size of filtered corpus of literature: The count of literature after filtering for resources types *Ontology* or *Ontology triples*
- Number of distinctive human-centred evaluations: Count of different human-centred evaluation approaches found, as several work presents more than one evaluations
- Number of human-centred evaluations for resource type *ontology*: Count of papers evaluating full ontologies
- Number of human-centred evaluations for resource type *ontology triples*: Count of papers evaluating only ontology triples

It remains to note that the size of the corpus of literature is not expected to equal the number of evaluations, as a study might conduct more than one evaluation.

Next, a *coding approach* is applied to the extraction form to unify the information, enable filtering and further quantitative and qualitative analysis. More detailed, the coding process will be applied to the fields extracting relevant information for the process phases (c.f. Table 3.2 fields 3 to 7 and 9 to 10) as these are lists of free text. The remainder of the fields does not need any coding as either the answer is already coded because a

binary or ordinal scale is used or the fields reflect arbitrary notes where no common pattern is to be expected.

To code the selected fields a two-step process is applied. First, for each field a set of distinctive and non-ambiguous codes is extracted. For example, for the field 4) *Preparation activities* this might include a set of codes representing process steps while for the field 5) *Preparation tools* this might be a set of codes representing the types of tooling. Second, each evaluation gets assigned the codes for the fields found in the studies.

Finally, as quantitative data is collected and the data is coded, the information is used to *derive answers for SLRQ1 and SLRQ2*, as described next.

3.1.4 Results of the Systematic Literature Review

The main results of the SLR addressing **SLRQ1** and **SLRQ2** include a set of activities during the process of human-centred ontology verification, a classification of the used tools and further insights strengthening the lack of an end-to-end platform and providing insights for following steps. Before presenting the results of **SLRQ1**, **SLRQ2** and a final summary, a quantitative overview is provided. In addition to the results of the SLR summaries of the literature reviewed can be found in Appendix A: SLR Summaries.

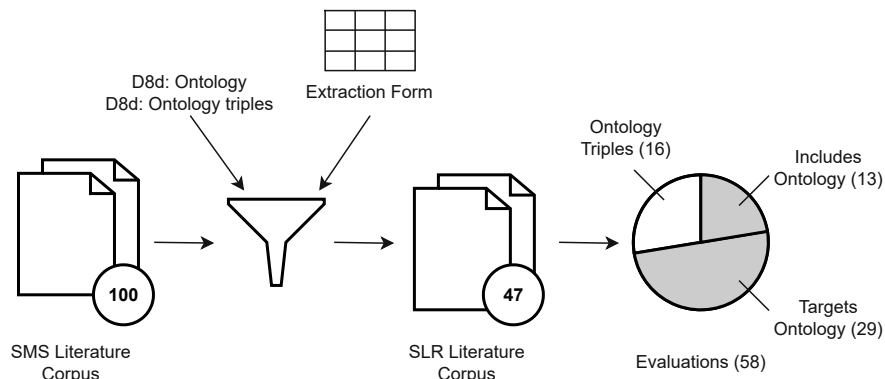


Figure 3.2: Literature selection process and quantitative overview.

Considering Figure 3.2 a quantitative overview of the studies and the human-centred evaluations therein is presented. After filtering the initial corpus of 100 papers used for the SMS, which this SLR is based on, more than half of the papers are excluded (i.e. 53 of 100) as they do not evaluate *ontologies* or *ontology triples*. Thus resulting in a corpus of literature for the SLR of 47 papers (c.f. *SLR Literature Corpus* in Figure 3.2). Further, considering the pie-chart of Figure 3.2 it can also be observed that more human-centred evaluation steps than studies can be found (i.e. 58 evaluations in 47 studies), indicating that human involvement is needed for various kinds of evaluation tasks. When comparing the different resources evaluated, about three-quarters (i.e. 29 + 13 out of 58) of the

evaluations focus on *ontologies* whereas only the remaining quarter focuses on *ontology triples*. However, it needs to be emphasised, that just half of the evaluations directly target an ontology (i.e. 29 of 58) while the remaining ontologies are evaluated through different means. One common scheme found for this different means of evaluation, is that an application or system which uses an ontology is evaluated.

SLRQ1: As briefly outlined earlier, analysing the data and relating it to **SLRQ1** revealed a set of activities conducted during human-centred ontology verification. To provide a more granular view, the results presented next are grouped according to the process phases (1) preparation, (2) execution and (3) follow-up.

- *Preparation Activities:* Through coding, a set of 15 different preparation activities or process steps is identified and presented in Table 3.3. Note that only steps which are explicitly mentioned in the publications were considered during extraction from population.

Code	Description	Number Identified
EE	Extract relevant ontological elements	21
GS	Seed in control questions	11
HCQ	HIT: Create questions	8
HUI	HIT: Create UI	7
PT	Submit to crowdsourcing platform	6
EC	Extract/Provide context	5
HCA	HIT: Create potential answers	5
HPT	HIT: Populate template	5
CS	Create survey	5
PR	Prepare presentation	4
CQ	Collect competency questions	4
NL	Translate to natural language	4
GV	Generate visualizations	3
SA	Specify aspect of verification	2
TQ	Prepare training questions	2
MISC	Any further steps not codeable	10

Table 3.3: Coded set of preparation activities order by occurrences.

Most often, studies mentioned the *extraction of relevant ontological elements* (i.e. *EE*) during preparation. Extracting ontological elements for evaluation can serve several purposes, two of the most common include extracting sub-parts of ontologies that should be evaluated, as for example in [45], or extracting elements from ontologies to provide different modelling possibilities for certain conceptualisations, as for example in [46]. Relating the studies mentioning *EE* to the resource types, it shows that regardless of the evaluation resource (i.e. 21 of 11 studies evaluate ontologies and the remaining 10 evaluate ontology triples), this preparation activity can be considered important.

Another activity observed with a high frequency compared with all other identified preparation activities is the *creation and inclusion of control questions or qualification tests* (i.e. *QC*). These are specific questions, where the correct answers are known upfront, which are included in HITs or surveys to filter out low-performing or non-trustworthy workers. When relating the studies including *GS* to the frame of reference that is required to answer the human-centred evaluation task, it shows that all of these studies use human knowledge as their frame of reference. This especially hints, that if the frame of reference of an evaluation includes human knowledge, workers' quality should be ensured by control questions such that for example spamming can be prevented.

Next, a group of different preparation activities, of which each is explicitly mentioned in at least five papers, is related to HITs and crowdsourcing platforms. The group involves *specifying HIT questions* (i.e. *HCQ*), *creating HIT UIs and templates* (i.e. *HUI*), *extracting context* which can be presented alongside questions to help workers understand the task (i.e. *EC*), *creating a potential set of candidate answers* if a multiple choice HIT design is used (i.e. *HCA*), *populating previously created UI templates* with ontological constructs to be evaluated (i.e. *HPT*) and finally also *publishing the HITs* on a crowdsourcing platform such as AMT (i.e. *PT*).

In contrast, if the target of deployment is not a crowdsourcing platform, authors often apply some *form of survey to collect feedback* about ontological constructs or overall ontology quality (i.e. *CS*). Further, a common scheme can be observed when relating publications that employ a survey to their resource type and the overall goal of the studies. These studies typically evaluate entire ontologies and the study's goal for the majority of these cases (i.e. 4 out of 5) includes the creation of an ontology. This might indicate that different collection mechanisms (e.g. either a survey or a crowdsourcing platform) should be used depending on the task to be fulfilled.

The remaining activities outlined in Table 3.3 are identified with a frequency of less than five, thus especially for those further investigation is needed.

To conclude **SLRQ1** with regards to *preparation activities* the most distinctive preparation steps include extracting ontological elements, seeding in quality questions and tasks related to either HITs or surveys.

- *Execution Activities*: The coding approach for the *execution activities* does not reveal any interesting insights, as the main tasks to be observed are either laymen completing HITs on crowdsourcing platforms or experts filling out provided surveys. However, these two predominant activities can also be deduced from the preparation steps as then either some form of crowdsourcing relevant step is included or a survey is being created during the preparation activities. Apart from the aforementioned insights, no outstanding activities are to be expected.

- *Follow-up Activities*: For *follow-up activities*, a similar pattern as with the *execution activities* can be observed, as coding does not reveal any interesting insights. The main activities include *collecting results*, in the case of crowdsourcing also aggregating results, and finally *calculating performance metrics*. One reason, for the shallow insights to

follow-up activities, might be that the authors of the studies are more likely to emphasize presenting the results rather than elaborating detailed processes on how follow-up steps are conducted.

- *Further Remarks on SLRQ1*: For completeness of **SLRQ1**, the following two aspects need to be mentioned.

First, it is important to emphasize that the obtained set of process steps is not complete and further investigation by conducting semi-structured interviews is needed. This also manifests when considering the frequency of tasks occurrences presented in Table 3.3, as the most frequent task (i.e. extracting relevant ontological elements) is observed 21 times whereas a total of 58 evaluations (c.f. Figure 3.2) are found across all publications, thus not even half of the assessed publications outline their preparation steps explicitly. Further, this aspect also strengthens the importance of this thesis' topic.

Second, for creating a process model of human-centred ontology verification as intended by **RQ1** an ordering of activities is required. It was expected, that as a side product of **SLRQ1** it is possible to obtain an ordering of the extracted activities, however, due to a very diverse corpus of literature and most studies not explicitly mentioned activity orderings, deducing general sequences of process steps is not possible.

SLRQ2: With **SLRQ2** relevant tools and libraries supporting the human-centred ontology verification process should be identified and classified. Following the same scheme as for **SLRQ2**, the following results are presented according to the process phases (1) preparation, (2) execution and (3) follow-up.

- *Tools Supporting Preparation*: Generally, information about tools supporting the preparation for human-centred ontology verification is rather rare within the reviewed literature. Indeed, only eight out of the 58 identified evaluations mention using any tool or library for preparing a human-centred ontology verification. The eight tools identified can be split into two distinctive groups: (i) existing tools or (ii) custom implemented tools. Five evaluations used one of the following existing tools *Hadoop* to preprocess data [47], *OWL API* for filtering relationships [48], *Snorocket Reasoner* to classify axioms [45], *Qualtrics* for creating a survey [49] or *Hootation API* for creating natural language statements [47].

The remaining three identified tools supporting preparation are custom implementations done by the authors of the studies. Within [50] a framework for enabling GWAP is presented which is capable of handling the whole process for several aspects of data linking. Another GWAP, to create and extend ontologies, was developed in [51]. Herein the implemented tooling covered all aspects of preparation including extraction of relevant constructs, natural language processing, breaking down constructs and generating the structures needed for the gameplay. A tool or rather a Protégé plugin, targeting directly ontology verification was implemented in [5]. For further details of this plugin consult Section 2.3.3.

Connecting the set of tools with the preparation activities identified within **SLRQ1**, *little tool support can be observed*, thus also underlining the importance of this thesis. Indeed, apart from the results of this SLR, the editorial paper [52] concerned with assessing the state of research in the intersection of the semantic web and human computation community emphasizes the need for reusable tools in this context.

- *Tools Supporting Execution*: In contrast to the tools involved for preparation, more information with regards to the tools used for the execution evaluations can be identified.

More specifically, a rather big portion of evaluation (i.e. 20 evaluations) is executed and supported by *crowdsourcing platforms* like *AMT* or *CrowdFlower*⁵. This indicates that the APIs of these platforms should be integrated into the end-to-end process support platform.

Another portion (i.e. 9 evaluations) was conducted and supported by *custom created tools*. This includes, but is not limited to, game platforms, as well as custom deployed web-pages allowing HIT execution. The reason for implementing custom tools might include that existing platforms do not support the desired execution.

A remaining of seven evaluations used some form of survey tools and a further two evaluations were supported by *Protégé*.

- *Tool Supporting Follow-up*: Hardly any information of supporting tools used for follow-up activities such as metrics calculation or inter-rater agreement calculation. Only three evaluations revealed the tools used. These include two usages of *SPSS*, an statistics and analytical tool, and *one online tool* used to calculate an inter-rater agreement.

Potential reasons are similar to those assumed for the *follow-up activities* identified with **SLRQ1**. Further, as often common metrics such as for example *Precision* or *Recall* are calculated, a wide variety of tools to be used can be found online, thus mentioning them in the studies is not considered of high significance.

- *Further Remarks on SLRQ2*: A common pattern that the analysis revealed is that very few publications report on tools used to support the process of human-centred ontology evaluation. Even though no evidence is provided by this SLR it can be assumed that many steps are performed manually due to the absence of reusable tools. Thus generally the importance of providing such tools to the relevant research communities is given and need to be addressed by this thesis.

Summary and Conclusions of the SLR: Concluding on the SLR the following aspects need to be considered.

With regards to **SLRQ1** focusing on identifying *process activities*, mixed results are observed. As for preparation activities, a set of 15 distinctive activities is identified. Generally, only a small set of preparation activities, including extracting relevant ontological elements, seeding in control questions, tasks related to crowdsourcing and surveys can be

⁵Note that CrowdFlower is rebranded as Appen)

identified in more than five different evaluations. Considering activities during *execution* or *follow-up phases*, literature does not reveal interesting insights, as most steps can be deduced from the preparation or can be commonly assumed.

For **SLRQ2** reporting on the tools used to support the process phases, even less information is identified. Commonly a categorization in tools being either a custom implementation or an off-the-shelf solution is found. The evaluation execution is typically supported by either crowdsourcing platforms, survey tools or custom deployed tools. However, reusable tools supporting the *preparation* and *follow-up phases*, are hardly mentioned in the publications and considering the number of evaluations identified, a need for such tools can be deduced.

Finally, combining all results found indicates that further work in the direction of thoroughly understanding the process activities and implementing reusable supporting tools is needed. The data presented in this section is considered to be an enabler for further research steps conducted by this thesis. Thus, a follow up semi-structured interview uses the data and aims at eliciting process steps, their ordering and required tools. Further, the semi-structured interviews are meant to complement the SLR as they should (1) uncover missed aspects and (2) strengthen and support the SLR results.

3.2 Semi-structured Interviews: Eliciting Activities From Experts

As outlined in the previous section, additional information for establishing a generic process model for human-centred ontology verification is needed, as the results of the SLR are not extensive enough. To address this aspect a group of experts (i.e. ontology engineers) is asked to join SSIs and to elaborate their experience with human-centred ontology evaluations. This way the quantitative data collected by the SLR can be enriched by qualitative data from a set of experts.

The overall aims of the SSIs include the following aspects:

1. Collecting a set of steps involved in human-centred ontology verification seen from the perspective of experts
2. Clarifying and extending the activities found by the SLR
3. Providing information for a follow-up focus group to finalise the process model

3.2.1 Interview Preparation and Approach

A vital step of a SSI is the preparation to ensure the needed information can be elicited from the experts. To that end, the insights from three different publications are used to guide the preparation as well as the execution of the interviews. More specifically, [10] and [53] are concerned with introducing SSIs in software engineering research and [54]

complements the aforementioned introductory papers with experience from 280 conducted interviews in software engineering studies.

Following the narrative of these publications, the preparation includes the three following steps, which are outlined in more detail in the upcoming paragraphs. First, the *participants need to be selected*, dates need to be scheduled and background information/material shall be provided. Next, an *interview guide* and an agenda for the interviewer need to be prepared. Finally, certain *guidelines for conducting the interview* need to be established.

Selecting Participants, Scheduling and Introducing the Interview Generally, participants need to be knowledgeable in the field of human-centred ontology verification to enable the extension of the results of the SLR towards a generic process model. To ensure this criterion, the selected group of participants are all sourced from a group of ontology engineers employed or working with the *Semantic Systems Research Lab*⁶ at *TU Wien*.

As suggested by [10], [53] and [54] the participants should be informed about the research and the interview beforehand, such that they can put themselves into context and prepare for potential questions. In the context of this thesis, this is addressed by two means:

- **Invitation E-Mail Template**⁷: The invitation mail will be sent to the participants in time well before the interview in order to briefly introduce the research, request them to join the interviews and schedule the interview appointments. In accordance with [10], a duration estimation for the interview (i.e. about one hour) is included so that the participants can schedule their time. Further, a link to an appointment scheduling tool⁸ is included to ensure the participants are aware of the date and time well in advance and no scheduling issues are to be expected.
- **One-Pager**⁹: A one-page document is attached to the invitation mail to complement it with additional background information relevant for the participants as such preparation is considered beneficial by all reviewed literature [10, 53, 54]. In particular, the problem of the thesis and the overall contributions are presented. Further the current state of the SLR, the need and content of the SSI, and information for a follow-up focus group are outlined in detail. To ensure the participants are not biased when preparing for the interviews or when answering questions, the main activities identified as a result of the SLR are not shared upfront.

In addition to the appointment scheduling tool shared with the invitation mail, a confirmation of the time slot is sent to the participants. Further, a couple of days before the interviews, the participants are reminded about their upcoming meeting and the *one-pager* is attached again for preparation purposes.

⁶<https://semantic-systems.org/>

⁷Refer to Appendix B: SSI Invitation EMail

⁸Doodle <https://doodle.com/>

⁹Refer to Appendix C: SSI One Pager

Preparing the Interview Guide and Agenda Semi-structured Interviews (SSIs) are situated between highly structured interviews, which are collecting a large set of answers from many participants by using closed-ended questions, and focus groups, which are fluidly collecting information from a group of interviewees [10]. To succeed with this type of interview, typically an interview guide is used by the interviewers. Among the reviewed papers [10, 53, 54], it is agreed on, that an interview guide proposes a set of open-ended questions, which collect the needed information while allowing certain aspects (e.g. including clarification questions; ordering of questions) to be fluidly adapted to the participant’s information flow during the interview.

The interview guide (note: the interview guide can be found in the following document¹⁰) for eliciting process information about human-centred ontology verification used by this thesis consists of a mixture of instructions to the interviewer, open-ended questions and outlines of information to be shared with the interview participant.

To make the interview process as seamless as possible, the following recommendations from the literature for designing the interview guide have been taken into account:

- The first set of questions should be considered easy to set the stage and to establish a connection with the participant. A typical example of such questions targets the participant’s background on the topic. Next, questions collecting non-sensitive information should be used to work towards the needed information. Then, the questions should target the success-critical and sensitive information. Finally, the interview should be ended with a brief review of the questions and gratitude towards the participant. To that end, also questions about the future of the topic can be discussed with the participant. (Recommendation based on [10, 54])
- The agenda is typically subject to change, as participants might outline unexpected information due to the open-ended nature of questions [10, 53, 54]. To react to such situations, the questions of the interview guide are prioritized as suggested by [10], indicating which questions *must* be asked, which *can* be asked and which are only asked *if time and context allows*. Additionally, the interview guide is structured using *if-then* statements, as proposed by [53], to account for the need for more detailed information requests and foreseeable changes in the agenda.
- Considerable effort is also put towards the difficulty of the question, as it needs to be ensured that the questions are not too hard or too detailed, as this prevents a broad and meaningful insight [54].
- As indicated by [54] reflexive questions (e.g. “*What could have been done differently?*”) are included in the interview guide as these typically stimulate the thinking of the participants and enable rich insights.

¹⁰Refer to Appendix D: SSI Interview Guide

Conducting the Interviews Also, aspects, as presented next, beyond following the proposed interview guide should be considered during the interview.

Conducting the interview is just one step of collecting the information, however, the information should also be accessible for future analysis steps. Thus, it is agreed on in all three publications [10, 53, 54], that the interviews should be audio-recorded and transcribed. To that end, if the interview are conducted in English, the “Otter.ai¹¹” application on a mobile phone is used as it supports recording and transcribing the interviews in one place. For German interviews “oTranscribe¹²” is used. As specifically outlined in [10], the consent for recording needs to be given by the participant and thus this aspect is included in the interview guide. Further, to avoid technical difficulties the setup is tested beforehand [10, 54].

In addition to recording the interviews, some information (i.e. the elicited process steps) is collected using pen and paper, visible to both the participant and interviewer, to enable a vivid discussion as well a review with the participants towards the end of the interview. This is also in line with [54], as their experience shows that using more interactive interview formats and visual tools (e.g. in the concrete case of the SSIs of this thesis, that is drawing a process model with the participant) helps the interview process. Following [10], all of this additional information not captured by the recordings is immediately transferred to a digital document to ensure no information is lost and everything can be reconstructed correctly.

Last, also the interviewer is expected to adhere to a set of guidelines to elicit the most information. Summarizing from the literature the following aspects need to be considered:

- The interviewer should always act neutral and in a non-judgemental way to avoid offending the participant and stopping the flow of information [10, 54].
- The interviewer should apply techniques of active listening by nodding and occasionally repeating the answers given by the participants [10, 54].
- The interviewer should know the interview guide by heart such that changes to the agenda can be done fluidly [10].
- The interviewer should not hesitate to ask clarification questions [10].
- The interviewer should either include very easy questions or probing questions if a participant does not share a lot of information [53, 54].
- The interviewer should gently but rather passively (e.g. by stopping practices of active listening) guide the participant back to the topic if he/she drifts off too much [53, 54].

¹¹<https://otter.ai/>

¹²<https://otranscribe.com/>

3.2.2 Results of the Semi-Structured Interviews

Within this section, the results of the interviews are presented. In total four SSIs spread across two days were conducted. Due to the ongoing pandemic situation, half of the interviews were conducted in a remote setting, while the remaining half of the interviews was conducted on-site. The interview population consisted of one senior researcher and professor, one PhD student, one MSc graduate and one MSc student, all working or having worked with human-centred ontology verification.

Importance of Ontology Evaluation: After the participants introduced themselves, the participants were asked to rate the importance of ontology evaluation as part of the whole process of ontology engineering on a scale from one (1) to five (5), where one (1) represents “not important” and five (5) represents “very important”. Except for one expert, all agreed on rating the activity of ontology evaluation as being “very important” (5). In fact, one participant even stated that if it would be possible, she would like to rate six, as this step should be one of the core activities of the engineering process. Another participant specifically included that ontologies often form the knowledge base of several systems and thus require special quality assessment measures for correct functioning systems. The remaining participant, who did not rate the activity as “very important” (5), rated the activity as “important” (4), with mentioning a tendency towards rating it higher. Thus, overall experts consider ontology evaluation as one of the most vital steps in the ontology engineering process.

Process Activities: The core part of the interview was focused on eliciting the activities of the process of human-centred ontology verification. To that end, the following three questions and additional probing questions (probing questions not listed below; these can be found in the interview guide) were used:

- **Q14:** *Starting with the preparation phase of the evaluation, can you list specific activities / steps that you would perform before you show the ontology and the task to your ontology evaluators?*
- **Q15:** *Now suppose that all of the activities associated with the preparation phase are completed, what are the steps that can be expected during the execution of the evaluation task with humans?*
- **Q16:** *Finally all the data is collected from the evaluators. What are the steps you are performing to conclude the ontology evaluation?*

During the interview, all the steps and activities mentioned by an expert were noted on a piece of paper/digital whiteboard visible to them. Once the expert mentioned all steps, he/she was asked to come up with an ordering of the activities to obtain an individual process model. Thus for each of the expert interviews, an individual process model and a transcript / meeting notes are obtained. Note that for analysis and reporting

purposes the handwritten information was transferred to the digital whiteboard after the interviews. The individual process models from the experts can be found in Appendix F: Individual Process Models.

Following the structure proposed by the extraction form used in the SLR as shown in Table 3.2, the analysis of the interview data will also be grouped according to the three phases (1) *preparation*, (2) *execution* and (3) *follow-up*. Further, within this chapter, the activities are not described in detail as this is presented with the final process model in Section 3.4

Preparation Activities: As the first step of the analysis, an aligned set of *preparation activities* was extracted by analyzing both the individual process models and the transcripts / interview notes. This step is similar to the coding process performed in the SLR and is required to create an aligned and distinctive set of activities as experts often refer to the same task with different names. For example, one expert mentioned preparing “Feedback questions” while another expert mentioned preparing a “Post Study”. Revisiting the transcript / interview notes revealed that both experts aim at creating surveys to collect feedback about the tasks and overall workflow of the verification.

Table 3.4 lists the aligned set of activities of all experts and identifies the overlap between the *preparation activities* identified in the SLR and elicited by the experts. Overall nine out of fifteen *preparation activities* found in the SLR are also found by the experts. The activities “*Extract relevant ontological elements (EE)*”, “*HIT: Create questions (HCQ)*” and “*Generate visualizations (GV)*” were mentioned by all experts indicating their importance. Further, “*Seed in control questions (GS)*” and “*Specify aspect of verification (SA)*” were mentioned by three experts. The remaining part of activities elicited is only mentioned by two (one activity), one (three activities) or none (six activities) of the experts. Nevertheless, also these activities, mentioned with lower frequencies, are considered in the follow-up focus group as the process model shall support a wide range of different human-centred ontology verifications.

In addition, the experts mentioned ten steps not identified by the SLR. Only one of these additional activities, namely “*Specify evaluation environment*”, has been mentioned by all experts. One of the reasons why this task is not explicitly found in the SLR, might be that the evaluation environment is often implied by the study design in use and thus not explicitly mentioned in literature. Further, “*Collect / load ontology*” and “*Find a crowd / evaluators*” has been mentioned by three experts. Similar to the overlap analysis, the remaining activities were only identified by either two or one expert(s).

Table 3.4: Overlap of preparation activities from the SLR and SSI. Activities which are non-coded are solely identified by the experts.

Code	Description	SLR	E1	E2	E3	E4
EE	Extract relevant ontological elements	X	X	X	X	X
GS	Seed in control questions	X	X	X	X	

Table 3.4 continued from previous page

Code	Description	SLR	E1	E2	E3	E4
HCQ	HIT: Create questions	X	X	X	X	X
HUI	HIT: Create UI	X		X		
PT	Submit to crowdsourcing platform	X				
EC	Extract/Provide context	X		X		
HCA	HIT: Create potential answers	X		X	X	
HPT	HIT: Populate template	X				
CS	Create survey	X				
PR	Prepare presentation	X				
CQ	Collect competency questions	X				
NL	Translate to natural language	X				
GV	Generate visualizations	X	X	X	X	X
SA	Specify aspect of verification	X		X	X	X
TQ	Prepare training questions	X	X			
	Prepare self assessment test		X		X	
	Implement follow up scripts		X			
	Prepare feedback form		X		X	
	Specify evaluation environment		X	X	X	X
	Find a crowd / evaluators		X	X		X
	Collect / load ontology			X	X	X
	Batch design			X		
	Specify evaluation scope			X		
	Prepare instructions				X	
	Inspect overall quality measures					X

As the experts were also asked to define an ordering of the activities they mentioned, a preliminary preview of the final process model can be created by applying the following set of rules to merge the individual process models:

1. Find all common activities mentioned by all experts and add those to the preliminary process model.
2. Starting with one expert's process model, insert all the activities, mentioned by this expert happening before a common activity. Note that if several activities are inserted before a common activity, the ordering of activities of the expert's process model shall be retained.
3. For all the remaining expert's process models add the steps, which have not yet been added, with respect to the position of the activity in the expert's process model.
4. Further, mark all the activities where a disagreement of ordering can be observed (i.e. one expert considered an activity to be done before a common activity while

3. UNDERSTANDING THE PROCESS OF HUMAN-CENTRED ONTOLOGY VERIFICATION

another expert considers the same activity to be done after a common activity) for further discussion.

Note that the *preparation activities* identified solely by the SLR cannot be added to the preliminary process model, as no information about ordering with respect to another activity was extracted. Also note, revisiting the transcripts or interview notes helped structure the preliminary process model through qualitative insights.

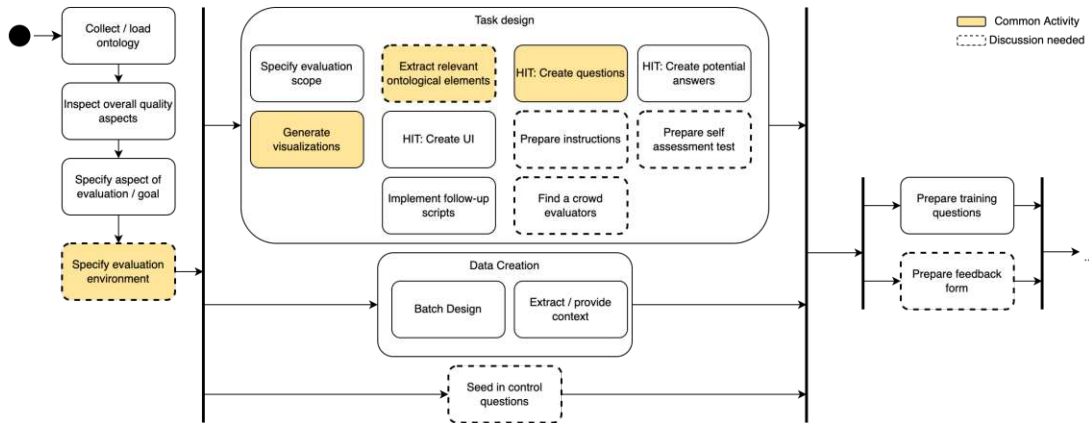


Figure 3.3: Preliminary process model of preparation activities.

Figure 3.3 depicts a preliminary process model of the *preparation activities* created by following the proposed rules. Activities highlighted in yellow represent common activities mentioned by all experts which were used as a starting point to define the preliminary ordering. Further, activities with dashed borders require further discussion as different views with regards to their ordering or their relevance can be obtained from the transcripts or individual process models.

The experts also commonly mentioned that certain activities can be executed in parallel. In the preliminary process model this is indicated by activities situated between two vertical lines. For example, the activity “*Prepare training questions*” was mentioned to be able to be performed parallel to “*Prepare feedback from*”.

Apart from the ordering of the *preparation activities*, three of the experts also explicitly proposed to group certain activities under “*Task Design*”, which helps to structure the process model. Such an activity group encompasses several activities where the experts did not identify any specific sequential ordering. In fact, a similar pattern can also be found with the SLR as certain activities were also grouped under the prefix HIT. One expert also added that the activities found in “*Task Design*” should be iterated upon in cycles. In addition to the activity group “*Task Design*”, one expert also proposed a group of activities referred to as “*Data Creation*”.

Comparing Table 3.4 and Figure 3.3 it can be observed that six *preparation activities* (i.e. “*Submit to crowdsourcing platform*”, “*HIT: Populate template*”, “*Create survey*”,

“Prepare presentation”, “Collect competency question”, “Translate to natural language”) are not included in the preliminary process model. The reasons for this are two-fold. For “Submit to crowdsourcing platform” and “HIT: Populate template”, at least one experts considers them as part of the execution phase and none of the experts considers any of these two activities to be part of the preparation phase, thus the classification needs to be revisited in the focus group. The remaining four activities (i.e. “Create survey”, “Prepare presentation”, “Collect competency questions”, “Translate to natural language”) are not mentioned by the experts and thus, no ordering with respect to other activities can be defined and the activities cannot yet be added to a preliminary process model.

Concluding on the *preparation activities* identified within the interview, Figure 3.3 depicts a preliminary process model which requires further discussion to reflect the preparation process of human-centred ontology verification. In particular following aspects/questions need to be clarified:

- Q_{P1} : Where should the activity “Specify evaluation environment” be placed? The interviews suggest the following placements: before, after or inside “Task Design”.
- Q_{P2} : Where should the activity “Extract relevant ontological elements” be placed? The interviews suggest a placement either inside “Task Design” or before “Task Design”.
- Q_{P3} : How does “Prepare instructions” differ from “Prepare training questions”. Can these two activities be merged?
- Q_{P4} : Is the self assessment test needed? If yes, where should it be placed? The interviews suggest a placement either inside “Task Design” or parallel to “Task Design”.
- Q_{P5} : Where should the activity “Find a crowd” be placed? The interviews suggest the following placements: before, after or inside “Task Design”.
- Q_{P6} : Where should the activity “Seed in control questions” be placed? The interviews suggest a placement either after “Task Design” or parallel to it.
- Q_{P7} : Should “Prepare feedback form” be part of the “Task Design” or should it be placed after “Task Design”?
- Q_{P8} : Are the activities “HIT: Populate Template” and “Submit to crowdsourcing platform” part of the preparation or part of the execution?
- Q_{P9} : Should the following activities “Create Survey¹³”, “Prepare presentation”, “Collect competency questions” and “Translate to natural language” be included to the process model? If yes for any of them, where to place them.
- Q_{P10} : Can any ordering for the activities inside “Task Design” be defined?

¹³Note that this activity refers to survey being sent out about the ontology.

Execution Activities: Similar to the previous section, first a set of distinctive and aligned *execution activities* elicited from the experts is presented. However, for the *execution activities*, it is not possible to determine the overlap with the results found with the SLR, as the results of the latter are high level within this area.

Description	E1	E2	E3	E4
HIT: Populate template		X		
Publish tasks		X	X	X
Monitoring & Revision		X	X	
Advertise tasks	X	X	X	
Host support meeting	X			
Conduct self assessment test	X			
Conduct qualification test	X		X	
Show tutorial	X			
Answer tasks	X			
Collect feedback	X		X	

Table 3.5: Consolidated execution activities elicited by the experts.

Table 3.5 outlines the *execution activities* elicited by the experts. After consolidating the individual process models and transcripts / meeting notes, a set of ten distinctive activities during the execution of a human-centred ontology verification can be found. The majority of the activities have only been mentioned by one or two of the experts and common activities are not mentioned very frequently. Only the two activities “*Publish tasks*” and “*Advertise tasks*” are mentioned by three experts, while no activity is being mentioned by all experts.

Considering the combined view of all experts presented in Table 3.5, three experts seem to agree on the distinction of the three process phases (i.e. phases (1) *preparation*, (2) *execution* and (3) *follow-up*), while one expert stood out as a different distinction between the three phases can be observed. This aspect surfaced during the analysis of the *execution phase* since this expert mentioned “*Extract relevant ontology elements*”, “*Collect results*” and “*Aggregate results*” to be part of the execution. The activity “*Extraction of relevant ontology elements*” is specified by all other experts to be part of the *preparation phase*. While the remaining activities “*Collect results*” and “*Aggregate results*” are seen as part of the *follow-up phase* by two other experts. Considering the other experts’ views, a tendency towards these three activities not being part of the *execution phase* can be observed, however further clarification is needed.

Following the analysis methodology of the *preparation activities*, it would be beneficial to create a preliminary process model outlining a possible ordering of the activities. However, as many of the activities are only mentioned by one or two experts and none of the activities are mentioned by all experts, a merging following the same methodology is not possible. As the set of activities is rather small when compared to the *preparation*

activities, the ordering of the activities will be defined during the focus group.

To conclude on the *execution activities* outlined in Table 3.5 following aspects need further clarification:

- Q_{E1} : Is the task “Extract relevant ontology elements” part of the *preparation phase* or part of the *execution phase*?
- Q_{E2} : Is the task “Collect results” part of the *execution phase* or part of the *follow-up phase*?
- Q_{E3} : Is the task “Aggregate results” part of the *execution phase* or part of the *follow-up phase*?
- Q_{E4} : How can the activities of the *execution phase* (i.e. consult Table 3.5) be ordered?

Follow-up Activities: In line with the previous analysis steps, as a first step, the activities elicited by the experts are aligned to form a distinctive set of activities. Table 3.6 outlines a set of eight activities expected to be performed during the *follow-up phase* of a human-centred ontology verification. Similar, to the *execution phase* most of the activities are mentioned only by one or two experts. Further, “*Create statistics*” is the only activity that is mentioned by three experts, while none of the activities are mentioned by all the experts. Also, note that an overlap analysis between the task identified during the SLR and elicited by the experts is not possible, as the results of the SLR are not founded enough.

Description	E1	E2	E3	E4
Collect data	X			
Analyse data	X	X		
Create statistics	X		X	X
Improve ontology	X		X	
Aggregate results		X	X	
Report results		X		X
Pre-process data	X		X	
Calculate trustworthiness			X	

Table 3.6: Consolidated follow-up activities elicited by the experts.

Analysing the transcripts / interview notes, again revealed a pattern of experts referring to the same activities with different names. For example, one expert mentioned one task to be referred to as “*Create statistics*” and another expert mentioned “*Calculate measures*”, while both experts referred to generating statistics about the collected data.

Another pattern that can be observed is that the granularity of the tasks mentioned to be expected during the *follow-up phase* is different between the experts. While two

experts only elicit three activities, which could potentially be also activity groups (e.g. as also observed with “Task Design” in the *preparation activities*) the remaining experts tend to have a more detailed view of the activities. For example, in one expert’s process model the activity “*Analysis / Analyze data quality*” can be found, while another expert mentioned activities such as “*Pre-process data*”, “*Calculate trustworthiness*” and “*Calculate measures*”. Revisiting the transcripts for these parts revealed that the expert who just mentioned “*Analysis / Analyze data quality*”, specifically included one goal of this activity to be “[...] *try to do to reduce spammers, [...]*”. Thus showing a similarity to “*Calculate trustworthiness*”. This brings rise to the question if certain *follow-up activities* elicited from the experts, can be grouped under “*Analysis*”.

Similar to the *execution activities*, none of the *follow-up activities* is mentioned by all the experts and thus the rules proposed earlier for merging the individual process models together to one preliminary process model for the *follow-up phase* cannot be done. Again, as the set of activities is rather small, when compared to the *preparation activities*, the ordering is defined during the focus group.

Concluding on the *follow-up activities*, the following aspects / questions need to be clarified:

- Q_{F1} : Is it beneficial for the final process to create an activity group “Data analysis”? If yes, which of the *follow-up activities* should be added to it?
- Q_{F2} : How can the activities of the *follow-up phase* (i.e. consult Table 3.6) be ordered?

Reflexive Questions: If time allowed (i.e. total interview duration was under one hour and questions regarding process activities have already been completed) the participants were also asked up to two reflexive questions to collect their opinion about the benefits of having a process model and what they would do differently for further verification compared to some of the previous verifications they did. The concrete questions are as follows:

- **Q23:** *Now, this brings me to the last question of my interview: As an outcome of my thesis, it is expected to implement a process model and tool that supports ontology engineers like you when preparing for such evaluations. How do you think your future work would benefit from it?*
- **Q19:** *Considering your previous experience with human-centred ontology evaluations, is there anything you would do differently now?*

All experts have been asked question Q23, while question Q19, due to time reasons, has only been asked to two experts.

For question Q23 targeting at benefits of having a process model and tool support for human-centred ontology verification, all experts mentioned that a process model would

help to structure their work. One expert specifically mentioned that from his experience, literature that describes the steps of human-centred ontology verification in depth is not available and having a baseline process model would have helped him a lot.

Another aspect mentioned by two experts is that a process model or tool support would save them time, as the preliminary step of understanding the process is not needed.

A further benefit elaborated by one of the experts is that a process model can help with the replication of an verification, as the steps are well defined and deviations during the verification on a process level can be reduced. Within this regard, another expert mentioned that the process model would prevent people to reinvent the wheel as they just would need to rely on an established process.

Finally, one expert mentioned an aspect especially important for scientific projects. If an ontology is part of a research project, this typically includes evaluating the ontology. Thus this requires the scientists to define an verification approach. Having a process model for human-centred ontology verification in place, the scientists could refer to this approach and focus more on their actual research work.

Only one out of the two experts who have been asked question *Q19* stated an aspect they would do differently for a future human-centred ontology verification. The expert mentioned that through a feedback questionnaire, people acting as evaluators said that the questions were very repetitive. Thus, for a future verification, she would take care of including more variety in the task design.

3.3 Focus Group: Defining the Final Process Model

As a result of the SLR, discussed in Section 3.1.4, a set of activities conducted during human-centred ontology verifications was extracted . Further, the results of the SSI, reported in Section 3.2.2, collected activities as seen from the perspective of experts in the field of ontology engineering. Both approaches have been designed to be independent of each other to ensure that the experts in the SSI are not biased and their personal views on the process can be elicited. To combine both results, clarify open aspects and collect feedback for a final agreed process model, a discussion with some of the interviewed experts following the focus group methodology is outlined next.

3.3.1 Focus Group Preparation and Approach

Similar to the SSI approach used, a guide is created to structure the interview and collect the desired information from the participants. Further, additional material including a short informational e-mail, presentation slides and whiteboard canvases are created to prepare and support the discussion process.

Following [11], a focus group typically encompasses the following steps:

1. Defining the research problem

2. **Planning the focus group event**
3. **Selecting the participants**
4. **Conducting the focus group session**
5. **Data analysis and reporting**

Defining the research problem: As outlined by [11], the focus group methodology is suitable, among other things, to evaluate and collect feedback on potential solutions. In the case of this thesis, the focus group will act as a tool to collect feedback on a merged view of activities collected from the SLR and SSIs. Further, the experts are asked to provide insights with regards to how the activities should be ordered in a process model. So the problem to be addressed by the focus group is to *collect inputs with regards to establishing a final process model for human-centred ontology verification*.

Planning the focus group event: To avoid potential bias and ensure good results from a focus group, the discussions should be properly planned before the meeting [11].

As a first step, the availability of the participants was collected using an online appointment scheduling tool¹⁴ well in advance of the actual meeting to avoid potential conflicts. In line with literature [11, 55], the appointment is scheduled to take two hours.

Another step taken to give the participants the chance to prepare for the focus group and also to remind them about the date and time, an informational email is sent to them four days before the focus group. As bias towards process activities is not expected since the participants are the same as for the SSI and each of them already elicited their individual view on the process activities, a document listing the activities¹⁵ found during the analysis of the SSIs is attached to provide further context.

The document outlining the identified activities from the SSI is structured according to the three process phases (1) *preparation*, (2) *execution* and (3) *follow-up*, thus the focus group meeting is also structured according to these phases. Further, to be able to collect insightful data, a semi-structured approach, as with the SSIs is followed, and a discussion guide is created. The discussion guide¹⁶ structures the focus group in five parts: (1) introduction, (2) - (4) the three process phases and (5) a round-up of the meeting. The questions are based on the aspects requiring clarification as outlined in Section 3.2.2. Additional for questions where an ordering needs to be defined, the discussion is structured in several rounds. For the first round, the participants are asked to write the first activity on a piece of paper and then the participants will reveal them. In the following rounds, the subsequent activities are ordered in the same fashion. Letting the experts write down the activities rather than having a verbal discussion to start with, should help to avoid bias towards one expert.

¹⁴Calendly: <https://calendly.com/>

¹⁵Full list of activities is obtained by combining Table 3.3, 3.4, 3.5, 3.6

¹⁶Refer to Appendix E: Focus Group Discussion Guide

To support the moderator and provide also visual information for the participants a presentation and three digital whiteboards are prepared. The presentation slides aim at introducing the main goal of the focus group session and also introduce the overall procedure to the participants. For each of the three process phases, a whiteboard showing the activities identified during the SSIs and SLR is created to allow discussing and ordering of the activities. The whiteboard for preparation activities uses the preliminary process model as depicted in Figure 3.3, while for (2) *execution* and (3) *follow-up* the whiteboards are created from scratch.

Selecting the participants: Next to preparing material used during the discussion, also the participants need to be selected. According to [11], depending on the research question to be addressed, participants should either be people with little to no experience in the topic or experts that can rely on experience in the field. As the process model to be created shall support ontology engineers and researchers, well-founded information is needed and thus for the focus group, the same group of experts as with the SSIs is invited to join the discussion session.

Conducting the focus group session: As the nature of the designed focus group is semi-structured, the moderator should follow the proposed discussion guide. In addition, the recommendations outlined in Section 3.2.1 for the moderator of a SSI should be followed as well.

Further, as there is only one moderator during the interview, a mobile phone is used to record the session. To that end *Otter.ai*¹⁷ is used as it is already used with the SSIs and allows recording and transcribing the recording in one application.

In addition to following the interview guide and recording the discussion, the whiteboards are extended during the discussion to reflect the ideas of the participants. This way the ideas collected from the participants are always present throughout the interview and the ideas can be referred to them at later stages. Also, the final whiteboards provide an important data source for subsequent analysis steps.

Data analysis and reporting First, after the focus group is conducted the audio recording is transcribed. Next, a coding process, as also performed with the SLR and SSIs, is applied to extract important information about the process and identify potential schemes.

Finally, once the transcript is processed, the digital whiteboards are enriched with relevant information from the transcript and this information is then used to create the final process model for human-centred ontology verification.

¹⁷<https://otter.ai/home>

3.3.2 Results of the Focus Group

One focus group session with three of the four experts, which also participated in the interviews, was held. Within this section, the main discussion points of the focus group, apart from the detailed ordering of activities that form the final process model, are presented. The ordering of the activities, which was part of the expert discussion, is presented in detail with the final process model in Section 3.4.

Following the same approach similar to the previous chapters, the main results are reported according to the three processes phases and in chronological order, as proposed in the discussion guide. Also, if the discussion point was inspired by one of the aspects requiring further clarification as outlined in Section 3.2.2, this is indicated by the aspect's numbering (e.g. Q_{P4}) as well.

Preparation Activities: Starting with aspect Q_{P3} as outlined in Section 3.2.2, whether the activities “*Prepare training questions*” and “*Prepare instructions*” can be merged, following insights apart from the ordering of the activities are outlined. Analysis of the discussion revealed, that the experts consider both activities vital for the preparation phase, however, they suggest two adoptions. “Prepare instructions” should be renamed to “*HIT: Prepare instructions*” to better capture that these are the instructions shown to the evaluators during the verification. Further, they also proposed the introduction of an activity group “*Quality control*”, which is composed of “*Prepare training questions*” and “*Seed in control questions*” (Note that this activity was addressed for requiring more clarification under aspect Q_{P6}) as depicted in Figure 3.5.

Considering aspect Q_{P2}/Q_{E1} , the experts stated that the activity “*Extract relevant ontology elements*”, should be part of the activity group “*Data creation*”. The main scheme mentioned for this aspect is that the activity targets splitting the ontology elements, which includes working with the data (i.e. in that case the ontology) and thus a natural fit to “*Data creation*” can be deduced.

For the activity “*Prepare self assessment*”, also reflected by aspect Q_{P4} , the experts suggest removing it as it might be only relevant for verifications conducted in experimental settings like research projects. Further, one expert added that there is data-driven evidence that a self assessment test might not be useful and the results thereof differ from the actual performance of workers can be observed.

Aspect Q_{P8} focuses on understanding whether “*HIT: Populate Template*” and “*Submit to crowdsourcing platform*” should be considered as part of the *preparation* or as part of the *execution* phase. First, a common scheme of experts suggesting that these activities can be part of either phase can be observed. However, the ongoing discussion revealed that the experts consider these two activities to be part of the *execution* phase, as interaction with a crowdsourcing engine/platform is required. Apart from the assignment to a phase, the experts also discussed whether these activities could be merged, as depending on the platform used, these two steps might be done in one. To be able to reflect several different platforms, the majority of the experts suggest keeping these two activities separated.

Next, the experts were asked to discuss whether the *preparation* activities solely identified during the SLR (i.e. “*Create survey*”, “*Prepare presentation*”, “*Collect competency question*” and “*Translate to natural language*”) as considered by aspect Q_{P9} should be included in the process model.

For “*Create survey*” the experts agree that this activity should not be part of the process model, as creating a survey to collect general aspects of an ontology is rather done during experimental evaluations. However, one expert added that if the evaluation environment is not micro-task oriented, such surveys might be placed instead of the activity group “*Task design*” and the process can still act as a useful baseline for the verification. Also the discussion of the activity “*Collect competency question*”, revealed a similar pattern that this is typically part of verifications during scientific studies but it does not directly have to be involved for human-centred ontology verification.

On the two remaining activities, the experts suggested including them in the process model. According to the discussion, the activity “*Prepare presentation*” should be included as an own activity, as it helps to explain to evaluators what the verification is about. Ideally, the preparation should be prepared after the task is designed as this might help to include the right content. The remaining activity “*Translate to natural language*” sparked a discussion about whether it should be included as an own activity or it should be part of “*Generate visualizations*”, as both are some form of representations of an ontologies or parts of an ontologies. The conclusion on this aspect is that a new activity “*Specify presentation modality*” should be introduced which implicitly includes both activities “*Generate visualizations*” and “*Translate to natural language*”.

The final point of discussion with regards to preparation activities was whether the activities in the group “*Task design*” can be ordered (i.e. aspect Q_{P10}). Here the experts agree, that all activities to be iterative and to be executed in parallel, as one activity might not be completed before some result of another activity is present and vice versa.

Based on the elaborated feedback and also ordering of activities the experts mentioned during the focus group the process model for the preparation phase of human-centred ontology verification is discussed in Section 3.4.1 and depicted in Figure 3.5. Also, all remaining aspects listed in Section 3.2.2 requiring further clarification not addressed herein are addressed by Section 3.4.1.

Execution Activities: As a starting point, the experts were asked to discuss aspects Q_{E2} and Q_{E3} focusing on eliciting information on whether the activities “*Collect results*” and “*Aggregate results*” are part of the execution or part of the follow-up phase. A scheme of experts agreeing on that it depends on whether the verification is seen from a dynamic view or seen from a batch view, the assignment of these two activities can be different. However, as the preparation phase also includes an activity “*Batch Design*” a batch view was agreed upon. Based on this consideration, the experts suggest moving these two activities to the *follow-up phase*.

All further points of discussion with regards to *execution activities* came up during ordering the activities and were not captured in Section 3.2.2. First, the activity “*Conduct self assessment test*” is no longer required, as the *preparation phase* should not include a self assessment test according to the experts. Next, as the experts decided on adding the activity “*Prepare presentation*” to the *preparation phase*, an activity “*Show presentation*” needs to be added to the *execution phase*.

Another activity, that started a discussion while ordering the activities was “*Host support meeting*”. On the one hand, the experts agree, that this activity can be vital when doing a synchronous verification, where all experts work during the same time, while for asynchronous approaches, where the evaluators have the change to complete the verification tasks in a larger given time frame, it might not be possible to host a support meeting. Thus the experts suggest merging the activities “*Monitoring & Revision*” and “*Host support meeting*” as the support meeting can be seen as some form of monitoring.

The last point of consideration brought up by the experts for the *execution activities* is to rename “*Answers tasks*”, which focuses on collecting answers for the verification questions of the ontology (i.e. core questions of the verification), to “*Conduct verification*”. The rationale behind this suggestion can be found in the concept of different actors being involved in the process of human-centred ontology verification. One expert mentioned that “*Answer tasks*” is the only activity listed that is supposed to be completed by an evaluator or crowd worker, while the remaining activities are all to be completed by an ontology engineer (or the person who wants an ontology to be verified). Thus to ensure a single perspective of the final process model the activity name “*Conduct verification*” is considered more appropriate and concise.

As elaborated in the previous section, the final process model for the *execution phase* can be found in Section 3.4.2 and is depicted in Figure 3.6.

Follow-up Activities: The first and also the main discussion of the focus group with regards to follow-up activities was whether to create an activity group “*Data analysis*” (reflected by Q_{F1}). In conclusion no activity group “*Data analysis*” is suggested by the experts, however, the experts proposed to create an activity group “*Create data quality statistics*” as part of this discussion. This activity group encompasses “*Calculate trustworthiness*” and “*Calculate inter rater agreement (IRA)*” and further it should be included instead of the activity “*Create statistics*”, as “*Create statistics*” is too generic and can cause confusion. Further, one expert specifically mentioned that there could be more activities inside this activity group and trustworthiness and IRA are just examples thereof.

Apart from structuring and grouping the activities one expert also suggested renaming the activity “*Aggregate results*” to “*Aggregate data*”. The rationale behind this suggestion is that for the experts, results are always the outcome of some analysis and with the aggregation, data from the crowdsourcing engine will be used and not analysis findings.

Again, the final process model based on the expert suggestions and inputs on ordering can be found in Section 3.4.3 and is depicted in Figure 3.7.

3.4 Human-centred Ontology Verification Process Model

To conclude the SLR, SSIs and focus group outlined in this chapter and to address **RQ1**, this section combines all the information and discusses the agreed process model for human-centred ontology verification, referred to as “VeriCoM 2.0”, in detail.

Before going into detail about the agreed process model “VeriCoM 2.0” and its activities, it is important to mention the following three considerations. First, “VeriCoM 2.0” is targeted toward micro-tasking environments such as crowdsourcing platforms. Second, the envisioned process focuses on batch style verification rather than on dynamic style verification. For a batch style verification, verification tasks are bundled into batches and the requester of the verification waits until all tasks of the batch are completed, whereas, for the dynamic style verification, verification tasks might be removed or added before completion. Third, certain activities identified during the earlier stages of this chapter (i.e. SLR, SSIs or focus group) are renamed to adhere to a *verb-object/verb-noun* naming pattern (i.e. “Batch design” as listed in Table 3.4 is renamed to “Design batches”) to adhere to best practices and to be precise [56, Chapter 3 Business Processes - What Are They Anyway?].

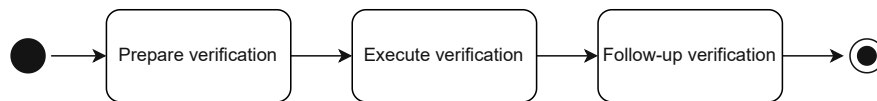


Figure 3.4: High level process view of human-centred ontology verification.

Figure 3.4 provides a high-level view of “VeriCoM 2.0” and its activities reflect the three process phases (1) *preparation*, (2) *execution* and (3) *follow-up* that guided the previous analysis. For each of the three high-level activities (i.e. “Prepare verification”, “Execute verification” and “Follow-up verification”) a more detailed view and discussion is provided are the following sections.

3.4.1 Prepare Verification

The first step when conducting a human-centred ontology verification involves several preparatory activities. Figure 3.5 outlines a detailed sequence of activities ordered, and also verified, by the experts during the focus group. In total nineteen activities, depicted by rounded squares, can be observed. A full black circle indicates the start of the process while a black circle surrounded by a white circle represents the end of the process. Activities that are situated between two vertical lines are expected to happen in parallel. Further, big rounded squares that contain nested activities (e.g. “Task design”),

are referred to as *activity groups* and are used for grouping activities as they represent familiar or connected tasks.

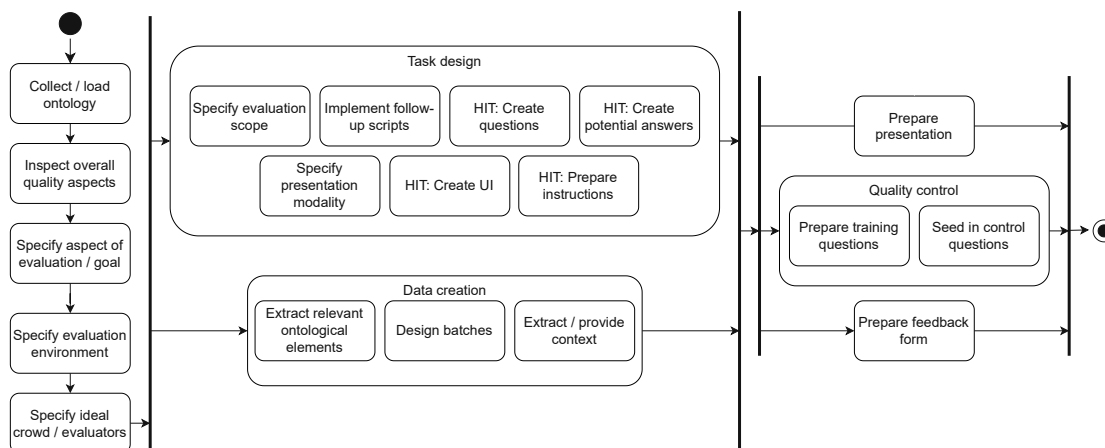


Figure 3.5: Final process model for the preparation phase of human-centred ontology verification.

In the following, each activity is defined and discussed in detail with respect to the ordering defined by experts and depicted in Figure 3.5.

Collect / load ontology: As a starting point, the ontology to be verified needs to be collected or loaded such that the following tasks can be completed. Ontologies are typically shared in an *OWL* or a *XML* representation.

Inspect overall quality aspects: To get an overview of the ontology, standard metrics and quality aspects should be inspected. For example in the Protégé editor metrics include but are not limited to *number of axioms*, *number of classes* or *number of data properties*.

Specify aspect of evaluation / goal: After gaining an overview of the ontology it is crucial for further steps to precisely specify the aspect that should be evaluated and to specify the overall goal of the verification. Examples from literature for example include verifying ontology hierarchies [4]. Depending on the methodology used, also collecting competency questions from experts prior to the evaluation can help to determine the verification goal. For more examples, Section 2.1 or Section 2.3 can be consulted.

During several interviews as well as the focus group discussion session, it was repeatedly emphasized by the experts that there exists a strong link to further activities (e.g.

“Analyse data” or “Report results” of the *follow-up* phase) and thus the importance of specifying the goal to reach good results of the evaluation cannot be overemphasized.

Specify evaluation environment: Specifying the evaluation environment refers to deciding on a crowdsourcing platform or medium that enables conducting the verification. Apart from crowdsourcing platforms, GWAP or conducting the verification using pen and paper are also possible.

The discussions revealed that the evaluation environment should be specified before designing the verification task or other material such as a presentation, as this choice might influence the following activities.

Specify ideal crowd / evaluators: In addition to the evaluation environment, also the crowd or evaluators’ characteristics should be specified at an early stage of the process as these could have an impact on how the task is should be designed. These characteristics include but are not limited to aspects like age, gender, locality or experience in a field of the workers. One expert mentioned that for example, a crowd of laymen needs a different task formulation than a crowd of experts. To ensure the crowd has some required skills, *qualification tests*, assessing the knowledge of the workers, can be specified and used as part of the verification.

Special consideration should be taken with regards to avoiding creating a potential bias through the crowd selection. One expert mentioned this becomes especially valid when ontologies are used for AI applications as regulations need to be followed and audits are to be expected. Another expert added to this point that platforms such as AMT allow specifying certain demographic aspects before publishing the actual tasks, which could help avoid problems in this regard.

Task design: Based on the basic aspects of the evaluation the verification task itself needs to be designed. As the design of the verification task might not be trivial and several activities are to be expected, this is represented in the process model as an activity group referred to as “Task design”. As a combined result of the SLR, SSIs and focus group a total set of seven nested activities (elaborated in detail next) are identified.

As already elaborated with the results of the focus group (c.f. Section 3.3.2), for these seven activities it is important to emphasize that no order is defined as the activities are expected to happen in an iterative and redundant fashion. For example, creating the questions might go hand in hand with creating the UI, as the question must be displayed in the UI and the UI must care for the ideal representation of the question.

Task design - Specify evaluation scope: For each of the tasks, its scope needs to be defined to be in line with the evaluation goal. More specifically, with the scope, one unit of verification is specified.

An example mentioned by an expert in the SSIs is that there might be a different scope between verifying subclass relationships and verifying the relevance of concepts. For the relationships, it might be sufficient to show the evaluators triples, while for the relevance, more ontological elements might be needed to ensure a correct judgement.

Task design - Implement follow-up scripts: As the task design might impose a structure on the final data and implications for analysis arise, scripts that can process the data might be implemented during this step.

Task design - Specify presentation modality: Verifying an aspect of an ontology requires the ontology elements to be presented to the evaluators. In literature, several different representations such as *Rector*[26], *Warren*[57] or *VOWL*[42] exist and depending on the task, the right modality should be specified.

Task design - HIT: Create questions: The core part of the verification is represented by the questions asked to the evaluators. Special emphasis should be put on designing the questions such that the required information can be collected.

Within that regard, one possibility for finding the right question for the task is to employ different types of questions. During the SSIs, the experts commonly mentioned a distinction between *open-ended* and *closed-ended* questions. Typically for *open-ended* questions, the evaluators can submit their answers using a free text form with little to no restrictions. On the other hand, with *closed-ended* questions typically a single choice or multiple-choice design with predefined answers is employed.

Task design - HIT: Create potential answers: For certain types of questions (e.g. multiple-choice questions), a pre-defined set of answers is required. Thus these need to be created during the task design and in general, this step can be closely related to creating the questions, as the answers need to be synchronised with the questions.

For example in [46] the authors employed, among other designs, a single-choice design where each set of answer possibilities is based on extracted triples from DBpedia¹⁸.

Task design - HIT: Create UI: Another important task of “Task design” is to design the User Interface (UI) for answering the questions. As the name suggests, the UI builds the bridge between the evaluator and the evaluation platform (i.e. crowdsourcing platform) and thus the UI can influence the obtained results [58]. The main elements of a UI include, but are not limited to, the question, answer possibilities, ontology representations and instructions.

Task design - HIT: Prepare instructions: In order to guide the evaluators through the UI and support them in case of unclarities, for each type of question/UI appropriate

¹⁸<https://www.dbpedia.org/>

instructions should be provided. Note that during the focus group it was agreed that the instructions are different from training questions, as the instructions are rather passive, while the training questions require the evaluators to actively perform training tasks.

Data creation: Parallel to the activities in “Task design” the activities grouped as “Data creation” are executed. The main scheme of activities to be found in this group is related to working with the ontology (i.e. data) and preparing it for verification. The activities inside this group are not expected to happen sequentially but rather in an iterative manner in small cycles to find the ideal composition of data.

Data creation - Extract relevant ontological elements: Ontologies can have complex structures and thus are often not suited for micro-tasking environments, as only small tasks (i.e. HITs) should be used. To that end, the ontology should be split or scoped according to the goal of the verification. For example, this could mean extracting a well-defined group of ontology axioms consisting of classes related to each other by `rdfs:subClassOf` properties.

Data creation - Design batches: As already introduced earlier, “VeriCoM 2.0” is targeted toward batch verifications, which means that the elements (e.g. axioms, classes etc.) to be verified should be grouped in batches which in turn can then be published for verification. However, the discussions revealed that special consideration with regards to batch design should be taken as too small or too big batches might not be completed by the workers.

Data creation - Extract / provide context: For certain verification tasks, such as verifying specific domain ontologies, it might be beneficial to add context to the ontological elements under verification. Such context can be retrieved from *WordNet*¹⁹ or similar databases. One example of context inclusion showing a beneficial effect for the workers can be found in [35] (refer to Section 2.3.1 for detailed information).

Prepare presentation: Once the tasks are designed and the data is created, a presentation should be created to inform the evaluators what the verification is about and what they are expected to do. Further, the analysis of the transcripts revealed that the experts suggest preparing the preparation after the task is designed as the design might yield important inputs for the presentation.

Quality control: Even if the crowd / evaluators are well-specified at the beginning of the preparation phase, spammers or unqualified workers might be among them. Thus a vital aspect of the preparation of human-centred ontology verification is to prepare quality control mechanisms, which is reflected by the process model’s activity group “Quality control”. To that end, the results suggest including the two activities of “Prepare

¹⁹<https://wordnet.princeton.edu/>

training questions” and “Seed in control questions”. However, the activities used for quality control are not limited to those activities, for example, as also mentioned with *Specify ideal crowd / evaluators*, qualification tests can be included to ensure high-quality results can be obtained.

Quality control - Prepare training questions: At the beginning of a task / HIT a set of training questions should be shown, to train the evaluators / workers with the given question format and final UI. To that end, the experts suggest using the same design as with the verification task, however, in their experience, they used a different ontology than the one to be evaluated.

Quality control - Seed in control question: A common scheme to ensure high-quality results and to detect potential scammers, already identified during the SLR and also mentioned throughout most of the expert interviews and discussions, is to employ control questions. Typically this is a set of questions, using the same design and the same ontology as the verification questions, with the difference that the correct answers are already known before the verification. These control questions can then be shown before the actual tasks or can be randomly included during the verification such that evaluators that answer these questions (or a certain percentage of these questions) wrong can be filtered out after completion.

Prepare feedback form: In parallel to “Prepare presentation” and the activities of “Quality control” also a feedback form should be prepared. Typical questions include whether the task was clear or how hard the task was perceived by the workers / evaluators. This feedback can especially be helpful if the human-centred ontology verification should be repeated and the process should be improved.

3.4.2 Execute Verification

Once the preparatory activities are completed, the human-centred ontology verification needs to be executed. Typically the verification is conducted on a crowdsourcing platform by a crowd of specified evaluators or even laymen. However, execution typically involves more steps than just conducting the verification tasks itself. To that end, Figure 3.6 depicts a total of nine ordered activities expected to happen. For the process model the same modelling principles as outlined in Section 3.4.1 apply. The following paragraphs describe these nine activities in detail.

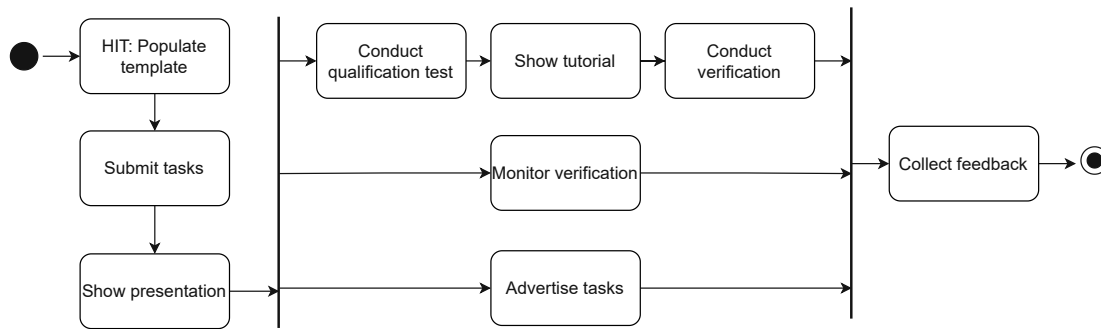


Figure 3.6: Final process model for the execution phase of human-centred ontology verification.

HIT: Populate template: As the first step of execution, the prepared UI elements and templates need to be populated with the data. At this stage, the tasks are not yet publicly accessible and according to the experts, this step can be used to verify and preview the task design. This way, if certain aspects need a revision, it can be done before the evaluators actually see the tasks.

Submit tasks: After the templates are populated and are considered to be suitable, the tasks can be submitted / published such that the evaluators can see the tasks. As already discussed in Section 3.3.2, populating the templates and submitting the tasks are activities which are closely related to each other and not every platform provide both options, however to ensure compatibility to various platforms both activities are included in the process model.

Show presentation: To ensure the workers know what the goal of the verification is and what they are expected to do, the prepared presentation should be shown to them. Note that the presentation might be included within the verification platform or if the verification is conducted synchronously (i.e. all evaluators are conducting the tasks at the same time) more traditional settings such as sharing a set of presentation slides through an additional meeting can be considered.

Conduct qualification test: To ensure that high-quality results can be obtained and spammers can be detected, the qualification tests should be conducted. Note that in “VeriCoM 2.0”, due to abstraction reasons the qualification test is placed before “Conduct verification”, however, as already outlined during “Seed in control questions”, these types of questions can also be randomly included during the verification questions itself.

Show tutorial: Before conducting the verification, it needs to be ensured that the workers are familiar with the task and thus the tutorial or training questions should be shown to them beforehand.

Conduct verification: Once the qualification test and the tutorial are successfully completed, the actual verification needs to be conducted. To that end the evaluators are required to answer the predefined questions about the shown ontology elements. Further, the evaluation environment (e.g. crowdsourcing platform) is typically responsible for showing open batches of questions (i.e. batches of questions that have not received enough answers from workers) and for collecting the answers from the evaluators.

Monitor verification: Parallel to the qualification test, tutorial and actual verification, the process should be closely monitored. Monitoring the tasks helps to identify and correct potential problems early on. To that end, crowdsourcing platforms typically provide management interfaces that can be used.

A scheme found through the discussions, is that the monitoring step could require the requester of the verification to stop the process and to go back to any previous step to do some revision of the process. For example, one expert mentioned that she had misconfigured the crowdsourcing platform to only collect one judgement for each task, even though she wanted to collect redundant judgements from several evaluators / workers. Thus, the tasks require re-configuration and probably need to be submitted again.

Apart from monitoring the tasks themselves, in a synchronous evaluation setting, a support meeting could be hosted to clarify potential questions from the evaluators during the execution.

Advertise task: Another part happening in parallel to the qualification test, tutorial and actual verification is to advertise the tasks. The tasks can be advertised through different means such as newsletters, web pages or any other communication means.

One expert mentioned, that advertising the tasks can be of particular importance if no influx of new evaluators / workers can be observed. She mentioned that towards the end of a verification, there might be a set of 100 tasks left and through advertising, it needs to ensure that also these tasks get completed.

Collect feedback: Finally, after the verification itself is completed, feedback should be collected using the prepared feedback form. This way potential problems with the workflow can be identified and the following verifications can be improved.

3.4.3 Follow-Up Verification

As the final step in the verification, concluding and follow-up activities need to be performed. The main scheme of activities found can be summarized by processing the obtained data. Figure 3.7 depicts the reviewed and detailed process model for the follow-up activities, composed of eight activities and one activity group. As with the previous process models, the modelling principles as outlined in Section 3.4.1 are used. In the following paragraphs, the activities are discussed in detail.

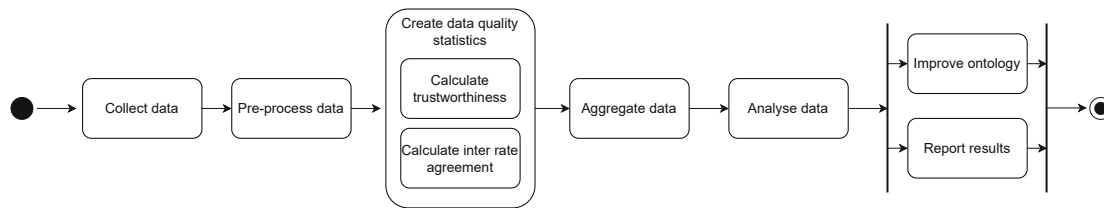


Figure 3.7: Final process model for the follow-up phase of human-centred ontology verification.

Collect data: Initially, the data reflecting the answers or judgements from the evaluators need to be collected from the crowdsourcing platforms to enable further steps. To that end, the experts mentioned that typically `csv` files are used.

Pre-process data During pre-processing, the collected data needs to be brought in the final format, which is compatible with the prepared analysis scripts (c.f. “Implement follow-up scripts” as part of the *preparation* phase). It is important to emphasise that pre-processing activities focus solely on a *syntactic level* and no interpretation of data is done yet.

Create data quality statistics: To gain an overview of the collected data, data quality statistics should be calculated. In the process model, this is reflected by the activity group “Create data quality statistics”, which has two nested activities “Calculate trustworthiness” and “Calculate inter rater agreement”. As already discussed earlier, these two are just examples of data quality statistics and different metrics can be used as well.

Create data quality statistics - Calculate trustworthiness: Based on the control questions and other measures provided by the evaluation environment, a score reflecting how much a worker can be trusted can be calculated. That way potential scammers can be identified and his/her data can be excluded from the following steps and higher quality results can be obtained.

Create data quality statistics - Calculate inter rater agreement: Based on the different answers collected on one question by several workers, an inter rater agreement can be calculated. To that end, various statistical methods such as *Cohen’s Kappa*[59] can be facilitated.

Aggregate data: In micro-tasking environments typically redundant judgements are collected for each task. To obtain a conclusion on a task these redundant answers need to be aggregated. Several approaches such as *majority voting* or variations of the *dawid-skene* method [60] can be employed.

Further, based on the calculated data quality statistics, weights might be assigned to workers and certain responses might be excluded from the aggregation to yield a high-quality set of aggregated answers.

Analyse data: Once the data is processed, the data needs to be analysed in order to obtain the final results of the verification. Depending on the verification goal specified during *preparation*, different analysis might be performed. For example, following the VeriCoM approach[6, 43] (outlined in Section 2.3.4) a final set of defects or errors found during verification can be obtained.

Another important aspect mentioned during the discussions is that the analysis can be done on several levels. Common levels include assessing each worker on an individual level or analysing the set of aggregated data.

Improve ontology: Once a final set of results is obtained through analysis, this can be used to improve certain aspects of the ontology. Of course, this is tightly linked to the goal specified during the *preparation* and depending on it, the results can be used to improve certain aspects of the ontology.

Report results: Parallel to improving the ontology, the results should be summarized and reported to the requester of the verification. Again, a close link to the evaluation goal can be observed and reporting should be in accordance with the goal.

3.5 Summary

To conclude this chapter and also **RQ1**, the main findings are elaborated. Generally, an iterative approach of four phases was used to understand the typical process of human-centred ontology verification by modelling “VeriCoM 2.0”.

First, in Section 3.1 a SLR on a corpus of literature conducting human-centred ontology verification was used to identify an initial set of activities expected during the process. Further, the number of papers studied also indicates the importance of the topic.

Second, experts, who already conducted human-centred ontology verifications, were interviewed in order to elicit their views on the process as presented in Section 3.2.

Next, as the results of the SLR were not shown to the experts prior to the SSIs, in Section 3.3 a focus group was conducted to collect feedback on the information of the SLR and to combine it with the information from the SSIs.

Finally, all the collected data was used to answer **RQ1** by creating a set of three process models referred to as “VeriCoM 2.0”, depicted in Figures 3.5, 3.6 and 3.7, that address the process of human-centred ontology verification structured in the three phases (1) *preparation*, (2) *execution* and (3) *follow-up*.

Support-Platform Reference Architecture

A reference architecture can be used as a base for creating concrete systems of a given domain[61]. As one of the main goals of this thesis is to implement a prototype of such a platform, this chapter introduces a reference architecture for systems that support human-centred ontology verification.

Before establishing the reference architecture a taxonomy of the domain is introduced in Section 4.1. The taxonomy shall support the reference architecture by capturing important and commonly used vocabulary for human-centred ontology verification. Based on the taxonomy, the *ProSA-RA*[14] approach is followed, to create a set of artefacts that resemble the reference architecture as discussed in Section 4.2.

In summary, the artefacts discussed in this chapter(i.e. taxonomy and reference architecture specification) address **RQ2**.

4.1 Human-centred Ontology Verification Terminological Hierarchy

This section introduces a lightweight ontology / taxonomy based on the process model (c.f. Chapter 3), that shall support the creation of the reference architecture as it establishes a common understanding and vocabulary of the domain of human-centred ontology verification.

4.1.1 Approach to Create the Taxonomy

As a methodology *Ontology Development 101*[15] is used, to ensure a well-founded taxonomy is created. The process is structured in the following seven steps (Note:

enumeration based on [15]):

1. Step: Determine the domain and scope of the ontology
2. Step: Consider reusing existing ontologies
3. Step: Enumerate important terms in the ontology
4. Step: Define the classes and the class hierarchy
5. Step: Define the properties of classes
6. Step: Define the facets of the slots
7. Step: Create instances

Since the main purpose of the created ontology is to support the creation and usage of the reference architecture with a common vocabulary, no instances as proposed by *Step 7* are created.

Next, during *Step 5* only uni-directional object properties and no additional data properties are created, due to the aforementioned purpose of the ontology. Note that the authors of [15] support this approach if the ontology should act as a communication tool.

Further, the result of *Step 3* to *Step 6* are not reported individually, as the creation process follows an iterative nature.

4.1.2 Taxonomy for Human-Centred Ontology Verification

Starting with *Step 1* the domain and the scope of the ontology/taxonomy are defined as follows. As given by the nature of the topic, the domain of the taxonomy is *human-centred ontology verification*. The main usage of the taxonomy is to act as a communication tool to establish a common understanding and terminological hierarchy of human-centred ontology verification. Further, the taxonomy should accompany the process model and not replace the process model. Reasons, why the ontology is helpful alongside the process model, include: (1) enabling easier exploration and navigation of terms by introducing a hierarchy and (2) making a clear distinction of concepts as elaborated by the experts[15].

Next, during *Step 2* related ontologies are searched, which could then be reused. Considering a range of sources (i.e. Ontohub¹, Ontology Design Patterns², Github³ and a literature search) no appropriate taxonomy can be identified. Nevertheless, it is worth mentioning that in [62] the authors propose an ontology for *Verification & Validation*, however, it has a different scope as it does not focus on the domain of human-centred ontology verification and thus is not suitable for the task on hand.

¹<https://ontohub.org/>

²<http://ontologydesignpatterns.org/wiki/Ontology:Main>

³<https://github.com/>



Figure 4.1: Terminological hierarchy of terms used for human-centred ontology verification.

Once the general aspects of the taxonomy are specified, the taxonomy was created during iterative cycles of *Step 3* to *Step 6*. Figure 4.1 shows the final taxonomy. As a formalism *OWL*⁴ was selected and the exported file can be obtained at [hcovtax.owl](https://drive.google.com/file/d/1LMCnTdxFOiSwRKjMXomfvuUulHYIOBk6/view)⁵.

⁴<https://www.w3.org/TR/owl2-syntax/>

⁵<https://drive.google.com/file/d/1LMCnTdxFOiSwRKjMXomfvuUulHYIOBk6/view>

The taxonomy encompasses a total of 36 concepts, of which 11 form the base of the hierarchy and the remaining concepts refine the base by introducing up to two additional hierarchical levels. The initial enumeration of terms used to build the hierarchy was inspired by the process models discussed in Section 3.4. It is important to mention that not all activities of the process models are reflected by the taxonomy, as they are not of main relevance when establishing a domain understanding. For example, the activity “Collect data” of the *follow-up phase* (c.f. Figure 3.7) is not included in the taxonomy as it reflects information only relevant to the process model and not the vocabulary of the domain.

Following [15], a combined approach, starting mostly from mid-level concepts, was used to organise the terms into a hierarchy and to further refine it. Finally, uni-directional object properties linking all concepts are introduced. For example, the concept `HumanCentredVerification` is linked to `EvaluatorPopulation` by an object property `employs`.

Concluding on the taxonomy, it outlines the most important domain vocabulary used for conducting human-centred ontology verification and in turn can be used to support the creation of a reference architecture.

4.2 Reference Architecture

A reference architecture is “an architecture that encompasses the knowledge about how to design concrete architectures of systems of a given domain”[14]. In the context of this thesis, a reference architecture for human-centred ontology verification shall be established to enable the design and implementation of an end-to-end support platform. The following sections establish the reference architecture according to the *ProSA-RA*[14] process. As part of the process also a set of exemplary system requirements as well as architectural requirements are identified. Thus the results of this section address **RQ2**.

4.2.1 ProSA-RA Approach

Using *ProSA-RA*[14], the establishment of a reference architecture is systematized as a process of four steps. Figure 4.2 provides an overview of the process and relates the steps to the research questions of this thesis. The results of **RQ1** are important information sources for *Step RA-1*. *Step RA-1* to *Step RA-3* address **RQ2** while *Step RA-4* is addressed indirectly by **RQ3**.

Step RA-1 - Information Source Investigation: First, a set of different information sources is identified which should help understand the processes and activities of human-centred ontology verification. To that end, the authors of [14] identify five different information sources: (1) people, (2) software systems, (3) publications, (4) reference models and reference architectures and (5) domain ontologies. Most of the relevant

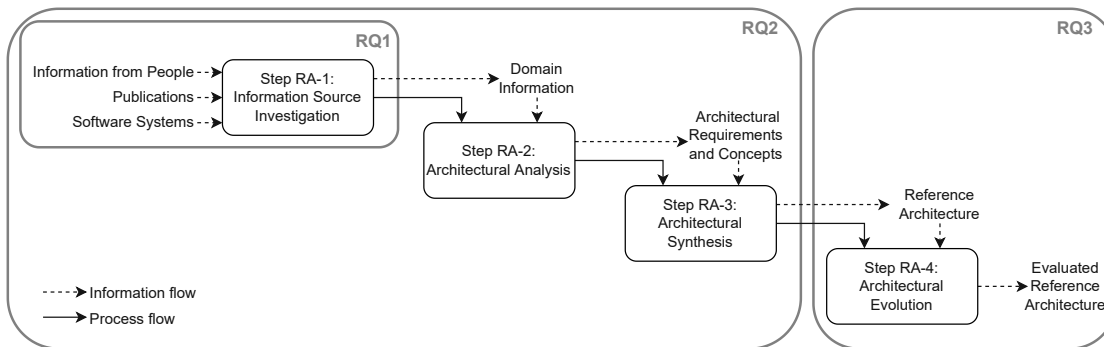


Figure 4.2: *ProSA-RA* approach overview and relevant resource questions. Adapted and extended from [14, Fig. 1]

information sources are already investigated during the work for Chapter 3 and are reused in this regard.

Step RA-2 - Architectural Analysis: The identified information sources are then used to define a set of requirements for a system in the source domain (i.e. human-centred ontology verification). Then these system requirements are aggregated to form architectural requirements. To ensure readable and uniform requirements, [63] is followed for writing the requirements. Further, each requirement is provided with a unique identifier and its origins to enable effective referencing and tracing.

Finally, the architectural requirements are mapped into concepts of an ontology or taxonomy to identify important system concepts. To that end, the taxonomy proposed earlier in this chapter (refer to Section 4.1.1) is used.

Step RA-3 - Architectural Synthesis: Based on the architectural requirements and concepts the reference architecture is then established. This encompasses the selection of relevant architectural patterns and applying all collected information about requirements into a coherent reference architecture. Further, the following four viewpoints of the architecture should be provided (the following enumeration is based on [14]):

- *Crosscutting viewpoint:* specifies general information of the reference architecture
- *Runtime viewpoint:* specified the dynamic behaviour of the systems enabled by the reference architecture
- *Deployment viewpoint:* specifies the hardware structure of typical systems
- *Sourcecode viewpoint:* specifies the software structure and modules

Whenever applicable, the viewpoints are built according to the *UML 2.5* standard⁶.

⁶<https://www.omg.org/spec/UML/2.5>

Step RA-4 - Architectural Evaluation: Once the reference architecture is established, the created artefacts need to be evaluated. For that reason, the authors propose to use the checklist-based approach *FERA* (*Framework for Evaluation of Reference Architectures*)[64]. However, this approach is highly focused on embedded systems and the full checklist is not available any longer, thus this approach cannot be used in the context of this thesis.

To overcome this issue, an indirect evaluation approach is used. During **RQ3** an instantiation, of the reference architecture is provided and evaluated in a case study. Thus, the evaluation of the reference architecture is provided in Chapter 5 as part of the conducted case study.

4.2.2 Establishment of a Platform Reference Architecture

Before applying the *ProSA-RA* approach to establish a reference architecture, the scope and goal of the reference architecture are defined as follows. The scope of the reference architecture encompasses all systems that provide end-to-end process support platforms for human-centred ontology verification. Further, the goal of the reference architecture is to establish a structure that enables and guides the implementation of the aforementioned systems.

Step RA-1 - Information Source Investigation: A total of ten different groups of information sources to determine a set of requirements is identified. Table 4.1 lists all identified groups of information sources, including a classification according to [14] and the main purpose.

Table 4.1: Identified information sources for building the reference architecture

ID	Information Source	Classification	Purpose
IS1	Experts (c.f. Chapter 3)	People	Understanding the process and the currently used tools
IS2	Semantic web libraries	Software systems	Understanding which ontology engineering tasks can be supported

Table 4.1 continued from previous page

ID	Information Source	Classification	Purpose
IS3	Automated evaluation tools	Software systems	Understanding which evaluation tasks are already automated and how
IS4	Ontology editors	Software systems	Understanding how ontology engineers typically work with ontologies
IS5	Crowdsourcing platforms	Software systems	Understanding how these platforms can be integrated
IS6	Related Work (c.f. Chapter 2)	Publications	Provide a general understanding of the problem domain
IS7	Process models (c.f. Section 3.4)	Reference models	Understanding which tasks should be supported
IS8	Taxonomy (c.f. Section 4.1.1)	Domain ontologies	Establishing an understanding of the common vocabulary and the reference architecture

In addition to the information outlined in Table 4.1, it is important to mention the following four aspects.

The information for *semantic web libraries* is restricted to Java-based libraries that are recommended by *W3C*⁷, as the prototype of the platform is going to be implemented in Java and libraries targeting other platforms usually do not differ much. Thus these

⁷https://www.w3.org/2001/sw/wiki/SemanticWebTools#General_Development_Environments.2C_Editors.2C_Content_Management_Systems.2C_E2.80.A6

include *Apache Jena*⁸, *OWL Api*⁹ and *rdf4j*¹⁰.

For *automated evaluation tools*, the selection is restricted to *OOPS!* (for more information, refer to Section 2.1.2) and to *OntoMetrics*¹¹.

Only *Protégé*¹² is selected as an ontology editor as it is considered a mature platform that is backed by a big community [65].

The selection of crowdsourcing platforms is inspired by the results of the SLR (Section 3.1) and thus *AMT* and *CrowdFlower* are selected. However, as *CrowdFlower*¹³ does not exist anymore only *AMT* can be used.

Step RA-2 - Architectural Analysis: Following *ProSA-RA*[14], the three elements (i.e. *System requirements*, *Architectural requirements* and *Mapping to domain concepts*) based on the information sources from the previous steps are outlined next.

First, the *system requirements* are defined based on a series of meetings with a researcher and a PhD student in the field, who represent both the customer as well as a user, when seen from a stakeholder analysis perspective. During these meetings, they elicited the following initial requirements (Note that the initial requirement ids are prefixed with *I-*) as outlined in Table 4.2.

Table 4.2: High-level requirements obtained from stakeholder meetings. Trace M reflects requirements to be obtained from a meeting note.

ID	Description	Trace
I-RE1	The system shall support the process of human-centred ontology verification as reflected by the process models (c.f. Section 3.4).	M
I-RE2	The system shall be tool independent (i.e. The system shall not be tied to an existing editor or tool).	M
I-RE3	The system shall allow extension for further tasks by using an appropriate modular architecture.	M
I-RE4	The system shall have a documented structure and interfaces.	M

In line with [14], based on an envisioned system and the information sources, a set of refined system requirements is defined based on the initial requirements of the stakeholders. Table 4.3 lists the refined requirements of a system that shall support verification of ontology restrictions as conducted in [28]. Note that certain aspects, like context extraction from WordNet (c.f. S-RE12), are assumptions of the systems and can be easily replaced.

⁸<https://jena.apache.org/>

⁹<http://owlcs.github.io/owlapi/>

¹⁰<https://rdf4j.org/>, formerly known as *Sesame*

¹¹<https://ontometrics.informatik.uni-rostock.de/ontologymetrics/>

¹²<https://protege.stanford.edu/>

¹³<https://www.mturk.com/>

Table 4.3: High-level requirements obtained from stakeholder meetings. Source IS from an identified information source, Source I-RE from an initial requirement.

ID	Description	Sources
S-RE1	The system shall support loading the ontology in OWL format.	I-RE1, IS1, IS7, [28]
S-RE2	The system shall calculate overall quality aspects of the ontology.	I-RE1, IS1, IS3, IS4, IS7, [28]
S-RE3	The system shall be capable of publishing the tasks on AMT.	I-RE1, IS1, IS5, IS7, [28]
S-RE4	The system shall create a set of evaluation questions.	I-RE1, IS1, IS6, [28]
S-RE5	The system shall create answers to the corresponding questions.	I-RE1, IS1, IS6, [28]
S-RE6	The system shall create axiom representation in Rector formalism.	I-RE1, IS1, IS6, IS7, [28], [26]
S-RE7	The system shall create axiom representation in Warren formalism.	I-RE1, IS1, IS6, IS7, [28], [57]
S-RE8	The system shall create axiom representation in VOWL ¹⁴ formalism.	I-RE1, IS1, IS6, IS7, [28]
S-RE9	The system shall populate task templates with the question, answers, representation and instructions.	I-RE1, IS1, IS7
S-RE10	The system shall extract all concepts and relations with ontological restrictions.	I-RE1, IS1, IS2, IS7, [28]
S-RE11	The system shall group tasks in batches of a given size.	I-RE1, IS1, IS7
S-RE12	The system shall provide additional context for the concepts from WordNet ¹⁵ .	I-RE1, IS1, IS7
S-RE13	The system shall create training questions from a different ontology.	I-RE1, IS1, IS7
S-RE14	The system shall seed in control questions.	I-RE1, IS1, IS7
S-RE15	The system shall provide information about the current status of the crowdsourcing tasks.	I-RE1, IS1, IS5, IS7
S-RE16	The system shall collect the data from AMT.	I-RE1, IS1, IS5, IS7
S-RE17	The system shall calculate the inter-rater agreement based on <i>Cohen's Kappa</i> .	I-RE1, IS1, IS7, [59]

Table 4.3 continued from previous page

ID	Description	Sources
S-RE18	The system shall aggregate the results using majority voting.	I-RE1, IS1, IS7, [59]
S-RE19	The system shall provide an interface to obtain the results.	I-RE1, I-RE4, IS1, IS7, [59]
S-RE20	The system shall be tool independent. (i.e. The system shall not be tied to an existing editor or tool)	I-RE2
S-RE21	The system shall allow extension for further tasks by using an appropriate modular architecture.	I-RE3

Second, the system requirements are aggregated to form the architectural requirements of the system. Table 4.4 lists the architectural requirements and which the system requirements are used as sources. It can be observed that $A-RE1$ to $A-RE9$ reflect functional requirements while $A-RE10$ and $A-RE11$ reflect non-functional requirements. More specifically, $A-RE10$ specifies the architecture to be integrable and $A-RE11$ specifies the architecture to be extensible.

Table 4.4: Architectural requirements obtained from aggregating the system requirements.

ID	Description	Sources
A-RE1	The system shall provide the capability of loading an ontology.	S-RE1
A-RE2	The system shall provide the capability of calculating overall quality aspects.	S-RE2
A-RE3	The system shall provide the capability of interacting with crowd-sourcing platforms.	S-RE3, S-RE15, S-RE16
A-RE4	The system shall provide the capability of creating verification tasks.	S-RE4, SE-RE5, SE-RE6, SE-RE7, SE-RE8, SE-RE9
A-RE5	The system shall provide the capability of extracting the required data.	S-RE10, S-RE12
A-RE6	The system shall provide the capability of grouping tasks.	S-RE11

¹⁴<http://vowl.visualdataweb.org/v2/>

¹⁵<https://wordnet.princeton.edu/>

Table 4.4 continued from previous page

ID	Description	Sources
A-RE7	The system shall provide the capability of including quality control mechanisms.	S-RE13, S-RE14
A-RE8	The system shall provide the capability of processing the data.	S-RE17, S-RE18
A-RE9	The system shall provide the capability of obtaining the results.	S-RE19
A-RE10	The system shall provide the capability of being integrable from other tools and editors.	S-RE20
A-RE11	The system shall provide the capability of providing extensions.	S-RE21

Finally, to provide input for further architectural decisions, the ten architectural requirements are mapped to the domain concepts of the taxonomy presented in Section 4.1.1. The obtained mapping is reported in Table 4.5.

Table 4.5: Mapping of the architectural requirements to the taxonomy.

ID	Requirements Summary	Concept
A-RE1	Ontology loading	Ontology
A-RE2	Quality aspect creation	VerificationAspect
A-RE3	Crowdsourcing platforms	CrowdsourcingPlatform
A-RE4	Verification task design	TaskDesign
A-RE5	Data extraction	SplittingApproach
A-RE6	Grouping tasks	BatchDesign
A-RE7	Quality control	QualityControl
A-RE8	Data processing	DataProcessing
A-RE7	Result provision	VerificationAspect
A-RE10	Integrability	-
A-RE11	Extendability	-

It is important to mention that for the mapping the most specific concepts from the taxonomy are chosen. Further, *A-RE10* and *A-RE11* are not mapped to the taxonomy as these requirements do not represent domain-specific aspects and thus are not part of the taxonomy.

Step RA-3 - Architectural Synthesis: Based on the architectural requirements outlined in Table 4.4 the reference architecture is built. As suggested by [14], the four viewpoints, (1) *crosscutting*, (2) *runtime*, (3) *deployment* and (4) *sourcecode*, are specified to describe the reference architecture. Figure 4.3 provides an overview of the artefacts used to specify the different viewpoints of the reference architecture.

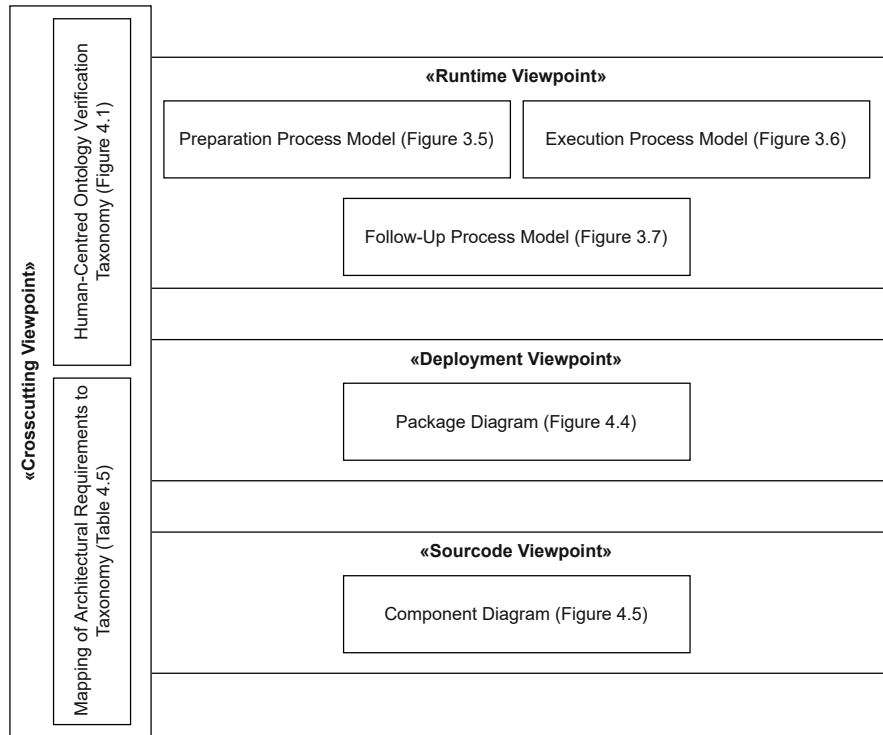


Figure 4.3: Viewpoints and diagrams used to build the reference architecture.

Starting with the *crosscutting viewpoint*, the conceptual domain of the process support platform for human-centred ontology verification is addressed. The core concepts that describe the domain are specified in the column *Concept* in Table 4.5. A more detailed specification of the domain vocabulary, providing additional hierarchical information, is described in Section 4.1.1 and depicted in Figure 4.1.

The *runtime viewpoint* is responsible for specifying the dynamic behaviour of the intended systems. As the reference architecture shall enable systems that support human-centred ontology verification, the dynamic behaviour is described by the process models defined in Section 3.4. More specifically, Figure 3.5 specifies the activities during preparation, Figure 3.6 specifies the activities during execution and Figure 3.7 specifies the activities during the follow-up of human-centred ontology verification.

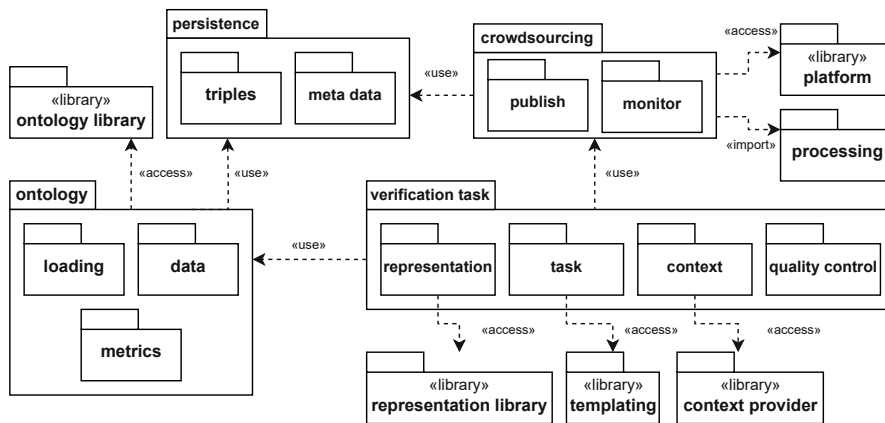


Figure 4.4: Package diagram (UML 2.5) depicting the deployment viewpoint of the reference architecture.

Figure 4.4 depicts a package diagram according to UML 2.5 standard, specifying the *deployment viewpoint* of the reference architecture. The main purpose of this diagram is to provide an overview of the packages and show the dependencies between the modules.

The following high-level packages are specified:

- **ontology**: Encompasses all software modules responsible for working with the ontology. To support ontology related tasks an external library `ontology library` can be used. For further refinement of the package, operation-specific sub-packages, such as `loading`, are introduced.
- **persistence**: To ensure data is stored in one central place to be used across several sessions, the `persistence` package is responsible for storing ontologies, extracted triples, as well as, metadata related to a verification.
- **processing**: Encompasses all software modules responsible for processing the data obtained from the crowdsourcing platform.
- **crowdsourcing**: Includes all modules related to working with the crowdsourcing platform. More specifically a library, `platform`, can be used to publish and monitor tasks. A `meta data` package is introduced to be able to store information about ongoing or completed tasks. Further, it is important to mention that the `processing` package is *imported* and thus according to the UML standard, all elements of the latter are publicly available through the `crowdsourcing` package for further use by other packages.
- **verification task**: With this package, all functionality of the platform shall be connected, as it uses the `ontology` package as well as the `crowdsourcing` package. The sub-packages are organised according to the specific operations of the

verification task package. To support the individual operations, support libraries can be used. A representation library can be used for presenting ontology elements. Tempting is a library that shall allow populating question templates with data and the context library can be used to provide additional context for the elements to be verified.

The last viewpoint of the reference architecture, the *sourcecode viewpoint*, is specified as a UML 2.5 component diagram as depicted in Figure 4.5. It is used to show the overall architectural style as well as the interfaces required and provided by the systems. In addition, this viewpoint also explicitly addresses the non-functional architectural requirements *A – RE10* and *A – RE11* from Table 4.4.

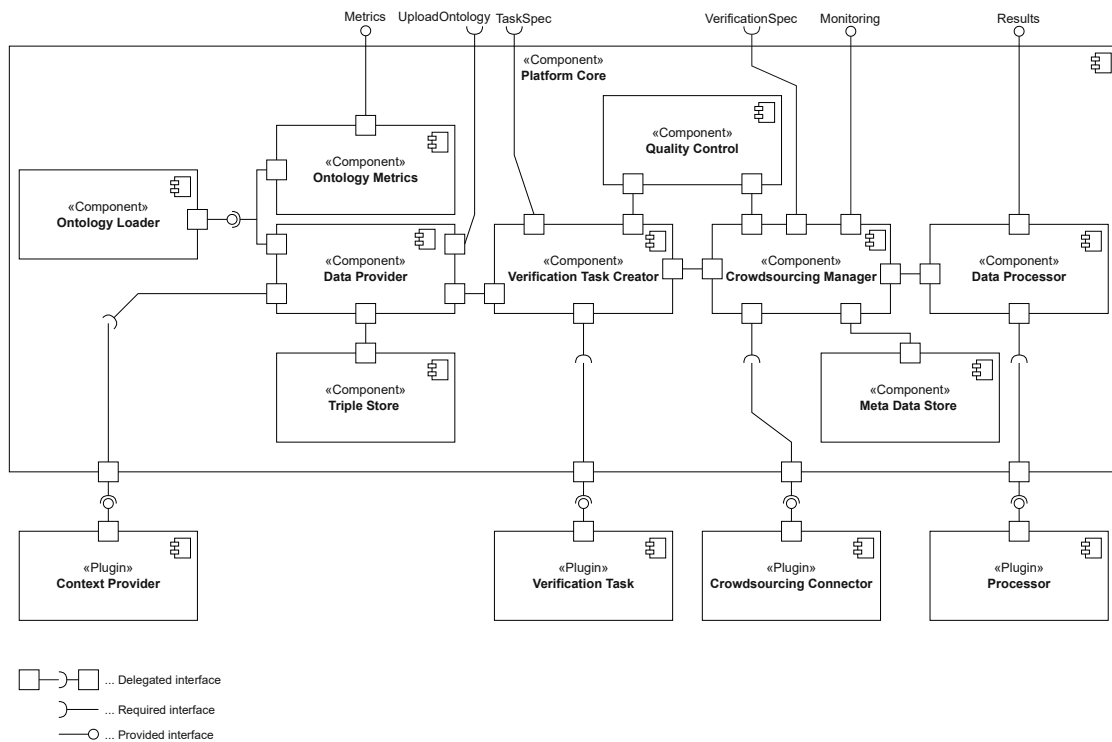


Figure 4.5: Component diagram (UML 2.5) depicting the sourcecode viewpoint of the reference architecture.

As a starting point for the creation of the component diagram, an architectural style that fits the architectural requirements is selected. The selection is guided by [66], which provides an introduction to common software architecture patterns and compares their characteristics.

To realise the process support platform reference architecture, a *Microkernel Architec-*

*ture*¹⁶ is selected, which features two main elements, the core and the plugins. The general business logic is encapsulated in a *core component* and concrete functionalities or extensions can be provided by the *plugins*[66]. Plugins have well-defined interfaces and new plugins can easily be added through mechanisms provided by most run-time environments. These mechanisms often follow the *Strategy pattern*¹⁷, such that the correct algorithm is selected during runtime and not during implementation. To indicate the differentiation among core components and plugin components, (core-)components are annotated with the default «*Component*» stereotype while plugins are annotated with a new stereotype «*Plugin*» in Figure 4.5.

One of the main reasons for selecting this architectural pattern is the high agility as proposed by the comparison in [66]. In the context of [66], agility is a concept that refers to the ability of the architecture to respond to change, thus enabling easy extension of functionality as required by *A – RE11*. Other reasons why a microkernel architecture might be selected include the ease of deployment, high testability and high performance[66].

Limitations of this architectural pattern are low scalability and hard development. However, for the envisioned platform, scalability is not of concern as the user group of the systems is relatively small, compared to other systems (e.g. banking systems) that require thousands of transactions to be realized in a matter of seconds where scalability is then required. Further, using up-to-date technology, for example, *Java Service Provider*¹⁸, and the artefacts of the reference architecture, development can be supported and the main concerns of [66] with regard to development can be alleviated.

What follows next, is the description of the components and interfaces specified in the sourcecode viewpoint. On a high-level view, a core component *Platform Core* encompassing nine components, which provide the main business capabilities of the platform, is specified in Figure 4.5. Apart from the capabilities provided by nine components inside the core, an implicit capability of the core is the orchestration of the different process steps, as well as, the loading of the required plugins.

For interaction purposes, three provided interfaces (i.e. *Metrics*, *Monitoring*, *Results*) should be realised by the systems implementing the reference architecture. These interfaces allow (1) collecting general quality metrics of the ontology (*Metrics*), (2) retrieving the current status of the crowdsourcing tasks (*Monitoring*) and (3) obtaining a set of processed results once all tasks are completed (*Results*). Further, three required interfaces, *UploadOntology*, *TaskSpec* and *VerificationSpec*, should be implemented that allow uploading the ontology and specifying the aspect of the verification, including the verification task and the crowdsourcing platform parameters. The aforementioned interfaces can also be seen as the entrance points to start a human-centred ontology verification. In the reference architecture, no technology for realising these interfaces is

¹⁶Comparable to a plugin-oriented architecture: <https://spring.io/blog/2010/06/01/what-s-a-plugin-oriented-architecture>

¹⁷https://en.wikipedia.org/wiki/Strategy_pattern

¹⁸<https://www.baeldung.com/java-spi>

specified to ensure that systems are not tied to any existing platforms and integrability as reflected by $A - RE10$ can be achieved.

In addition to the required and provided interfaces, four delegated interfaces are provided by the core. These are responsible for forwarding specific tasks to plugins. Further, these interfaces can also be seen as extension points of the architecture and enable extension through (1) different context providers, (2) different task designs, (3) different crowdsourcing platforms and (4) different data processing approaches. Again, it is important to emphasize, that the plugins provide concrete functionality and the components in the core are responsible for selecting an appropriate plugin for the task and orchestrating the process.

The following enumeration describes each of the components in the *Platform Core* in detail:

- *Ontology Loader*: Ontologies typically are specified in machine-readable formats, thus this component is responsible for loading and parsing ontologies in these formats for further use by the system.
- *Ontology Metrics*: The *Ontology Metrics* component is responsible for calculating overall quality metrics of the ontology to support ontology engineers to decide what verification task shall be performed. To support this aspect the *Ontology Loader* is facilitated and the calculated results can be obtained by a provided interface *Metrics* of the core.
- *Data Provider*: With the *Data Provider* component, relevant ontological elements can be extracted. To support this functionality, the component needs to be connected to the *Ontology Loader*. Further, the extracted elements, depending on the verification task, might be enriched with additional context and stored through the interface to *Triple Store*. To that end, several different *Context Provider* plugins can be included by facilitating a delegated interface.
- *Triple Store*: This component provides the capability of storing extracted ontological elements as triples. Through this mechanism, relevant ontological elements can be stored upon loading and extracting, such that once a verification is completed, these can be (re-)used for further purposes.
- *Verification Task Creator*: The verification tasks, including question design, answer design, templating and representation of ontological elements, are created within this component. As already mentioned earlier, a required interface *Task* should be implemented that allows specifying the verification tasks. To support different verification tasks, a delegated interface is responsible for loading *Verification Task* plugins that create the concrete tasks. Further, interfaces to *Data Provider* are used to collect the required data and an interface to *Crowdsourcing Manager* is used to publish the tasks on a crowdsourcing platform.

- *Quality Control*: The quality control component is responsible for providing the possibility to seed-in control questions or to create training questions for the HITs. To ensure the same task design as for the verification itself is used, an interface to use the *Verification Task Creator* is required.
- *Crowdsourcing Manager*: To complete the verification tasks, this component is responsible for creating, grouping, publishing and monitoring the crowdsourcing tasks as well as for collecting the results from the platform. To that end, plugins (i.e. *Crowdsourcing Connector*) are used to be able to interact with different crowdsourcing platforms. Further, an interface to the *Quality Control* component is needed to include training and control questions. To keep track of the published tasks the interface to the *Meta Data Store* is used to store meta data about the HITs. Tasks are created and published by interactions with the required interface *VerificationSpec*. Further, this information can be obtained by the provided interface *Monitoring* of the core.
- *Meta Data Store*: The *Meta Data Store* provide the capability to store information about ongoing verification tasks. This might include used ontological elements, used task design or the current status of crowdsourcing tasks.
- *Data Processor*: Once the HITs are completed, the results need to be obtained from the *Crowdsourcing Manager* and processed. Thus this component provides the functionality of processing the data obtained from the crowdsourcing platform. To ensure several different processing steps can be included (e.g. aggregation or trustworthiness), a delegated interface to load *Processors* needs to be implemented. The processed results can then be collected through the provided interface *Results* of the core.

4.3 Summary

This chapter addresses **RQ2** and thus focuses on defining requirements and establishing a reference architecture to enable the implementation of platforms that support human-centred ontology verification.

To build a common vocabulary for establishing, describing, as well as, using the reference architecture, a taxonomy capturing important concepts is presented in Section 4.1.

Then, the reference architecture is defined in Section 4.2 following the *ProSA-RA* approach [14]. This includes, collecting important information sources, electing system and architectural requirements, and specifying different viewpoints of the final reference architecture. More specifically, the viewpoints include the taxonomy (c.f. Figure 4.1), the process models (c.f. Section 3.4), a package diagram (c.f. Figure 4.4) and a component diagram (c.f. Figure 4.5). These artefacts provide the basis for the implementation of a prototype used for the case study discussed in the next chapter.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Case Study: Supporting Human-Centred Ontology Verification

This chapter investigates **RQ3**, more specifically, it is evaluated how the process models, presented in Chapter 3, and the reference architecture, presented in Chapter 4, can support ontology engineers when conducting human-centred ontology verification. To that end, a case study following the methodology in [16, Chapter 5 Case Studies] is conducted.

First, the use case under study and the evaluation approach are defined in Section 5.1 and Section 5.2, respectively. Next, as the use case is supported by an implementation of the reference architecture (c.f. Chapter 4), the prototypical implementation is outlined within Section 5.3. Finally, the evaluation results are reported and conclusions are derived in Section 5.4.

5.1 Case Description

The use case investigated by this thesis is the verification of ontological restrictions. In more detail, the selected use case is based on the verification during the experiment in [28], as also outlined in Section 2.3.2. Thus the use case encompasses the identification of modelling mistakes related to universal (\forall) and existential (\exists) quantifiers in ontologies using Human Computation (HC) techniques. According to [28], these modelling mistakes are often related to an incorrect assumption that either (1) the universal restriction implies also the existential restriction or (2) that missing information is incorrect.

The verification conducted in [28] uses the Pizza ontology¹. Typical mistakes, which

¹<https://protege.stanford.edu/ontologies/pizza/pizza.owl>

5. CASE STUDY: SUPPORTING HUMAN-CENTRED ONTOLOGY VERIFICATION

should be identified by such a verification, can be illustrated as an example of the Pizza ontology. A pizza *Margherita* has two toppings *Tomato* and *Mozzarella*. Modelling these two toppings only using either (1) an existential quantification restriction (`owl:someValuesFrom`) or (2) universal quantification restriction (`owl:allValuesFrom`), would either lead to (1) all pizzas with a tomato and mozzarella topping amongst other toppings or (2) pizzas with no toppings to be classified as a pizza *Margherita*.

In [28], Human Intelligence Tasks (HITs) are published on Amazon Mechanical Turk (AMT) to realise the verification of ontological pizza definitions involving such restrictions. Figure 5.1 depicts a sample user interface for a HIT published on AMT annotated with descriptions.

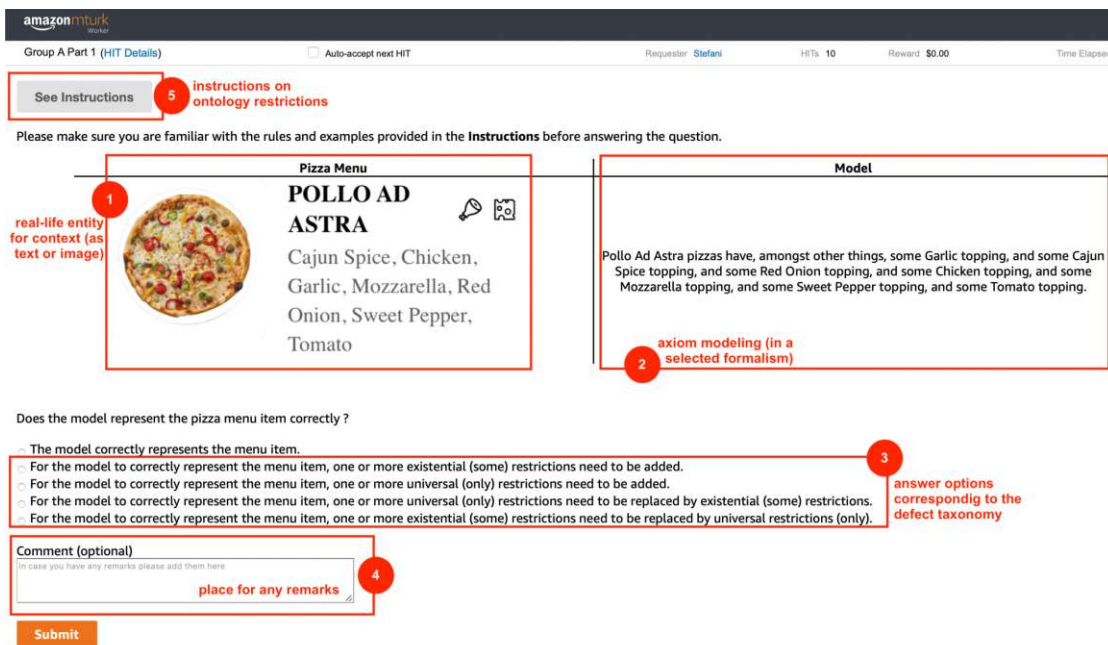


Figure 5.1: Example interface of a HIT used by [28]. Source: [28, Figure 4.1]

Based on the information in the publication and during meetings with the author, the preparation and the creation of the HITs were performed manually with minimal automation and tool support. Figure 5.2 shows the process model for the preparation of human-centred ontology verification annotated with information relevant for the case study. Activities depicted in yellow, were conducted as part of the verification in [28]. Further, each of the yellow activities is annotated with *(M)*, *(SA)* or *(A)* to indicate if the activity was conducted *manually*, *semi-automatically* or *automatically*.

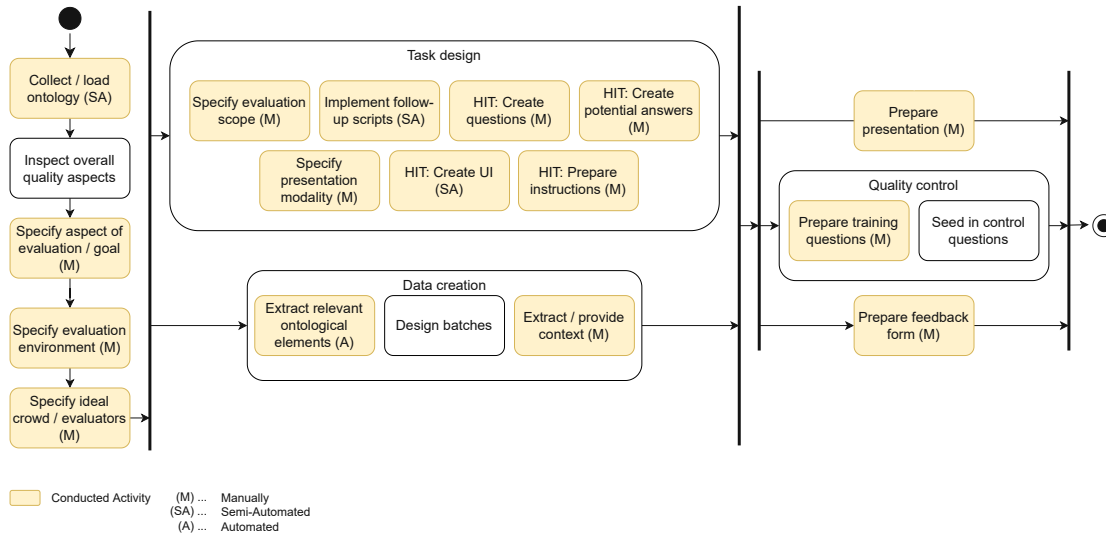


Figure 5.2: Activities conducted for preparing the verification in [28].

As already outlined earlier, the case studied in this thesis replicates the verification as part of the experiment in [28], with the difference that the activities shall be supported/automated by the process models and a prototypical implementation of the reference architecture for human-centred ontology verification, presented in Section 3.4 and Section 4.2, respectively. For further understanding, it is important to emphasize that the prototypical implementation realised as part of the case study includes (1) a core platform and (2) a set of plugins to realise the concrete verification (for more information about the software architecture consult Section 4.2 or Section 5.3). The main aim of the case study is to find out how well these artefacts can support ontology engineers while preparing human-centred ontology verifications (c.f. **RQ3**).

5.2 Evaluation Approach

Based on the case outlined in the previous section, data is collected to address **RQ3** and to evaluate the reference architecture presented in Section 4.2. As the case study is based on automating a manual process, the evaluation is realised by a comparison between the automated approach against the already existing manual approach (i.e. a baseline).

In more detail, the aspects that shall be evaluated can be summarised as follows:

- *EA-1*: Which of the preparation activities (c.f. Figure 3.5) can be supported by an implementation of the reference architecture? Which activities cannot (yet) be

supported?

- *EA-2*: How much time is needed to implement the platform and the required plugins to conduct the verification of ontological restrictions? How does this compare to performing the steps manually / without an end-to-end process support platform?
- *EA-3*: Which aspects are improved when the platform is used in comparison to the manual approach?
- *EA-4*: Which aspects are missing / different when using the platform?

As the evaluation aspects are mostly related to comparison against the manual approach, this baseline needs to be established first. This is realised by collecting information from the author of [28], who has performed the verification of ontological restrictions manually as part of an experiment. More specially, the author is asked to specify which steps of the generic process model depicted in Figure 3.5 are required for this verification (i.e. Figure 5.2 outlines the conducted steps) and to provide time estimates for the manual approach that is followed in [28] for each of the required steps. Additionally, the author is asked to share any artifacts that supported her during preparation of the verification to collect additional insights.

The data for the case study, is collected by indicating which activities of the process model presented in Figure 3.5 are automated by the prototypical information, by tracking time efforts and by collecting qualitative information / aspects during implementation.

The evaluation aspects *EA-1*, *EA-2* and partially also *EA-4* can be addressed by comparing the data of the case study against the baseline. Further, *EA-3* and *EA-4* are addressed by analysing qualitative information collected during implementation and from the author of [28].

5.3 Implementation

As proposed in Section 5.1, the case study includes the implementation of a prototype of the reference architecture presented in Chapter 4. The main aim of the prototype is to address **RQ3**, however as the implementation strictly follows the reference architecture, the prototype also evaluates the reference architecture and so addresses **RQ2** as well.

Following the reference architecture, the prototype is composed of two parts:

- *Platform Core implementation*: Following Figure 4.5 of the reference architecture, all sub-component except the component `Ontology metrics` and its related interfaces of the *Platform Core* are implemented. The main responsibility of the core is to orchestrate the communication between the sub-components and to provide the ability to load customized plugins depending on the verification task on hand. Additional to the core, a Software Development Kit (SDK) enables extension of the system by describing the interfaces and data structures required by the plugins.

- *Plugin implementation*: To be able to extract the needed axioms from the Pizza ontology and to create the required HITS, a *Context Provider* plugin and a *Verification Task* plugin are implemented. Further, a *Crowdsourcing Connector* plugin to connect to AMT and a default *Processor Plugin* to collect raw results from a crowdsourcing platform are implemented.

As a methodology for all parts of the implementation Test Driven Development (TDD)[67] is followed and if applicable, clean code practices as presented in [68] are applied. This ensures, that high-quality code, that is well-tested, well-structured and easy to extend, is produced.

The technologies used for the prototype are as follows:

- *Java 18²*: As a programming language, the latest version of Java is used due to its high popularity³ and familiarity.
- *Spring 2.6.7⁴*: To ensure the code is extensible and implementation overhead can be reduced, the popular Java framework *Spring* is used.
- *Swagger⁵*: All external interfaces (e.g. *VerificationSpec*) are realised using HTTP *REST Services*⁶ and for documentation purposes, the interfaces are documented using *Swagger*. This also enables the generation of a web interface that allows executing calls against the interfaces. Further, an *OpenApi 3.0* specification can be generated, that allows external parties to automatically create clients, which can interact with the platform core.
- *Apache Jena 4.4.0⁷*: Using the *Apache Jena* library, it is possible to programmatically interact with OWL ontologies. The main applications of the library in the context of the prototype include ontology loading and extracting relevant ontological elements.
- *Thymeleaf 3.0.15⁸*: To allow specifying templates for a human-centred ontology verification and populating the templates with resolved variables the popular Java library *Thymeleaf* is used.
- *AWS MTurk 2.17.174⁹*: For the implementation of the *Crowdsourcing Connector Plugin* to connect to AMT the Amazon Web Services (AWS) library is used.

²<https://docs.oracle.com/en/java/javase/18/>

³According *PYPL PopularitY of Programming Language* <https://pypl.github.io/PYPL.html>

⁴<https://spring.io/>

⁵<https://swagger.io/>

⁶<https://www.kennethlange.com/books/The-Little-Book-on-REST-Services.pdf>

⁷<https://jena.apache.org/>

⁸<https://www.thymeleaf.org/>

⁹<https://aws.amazon.com/sdk-for-java/>

- *Apache Maven 3.6.3*¹⁰: To organise the project structure, manage dependencies and build the prototype, *Apache Maven* is used.

The two following sections outline the (1) platform core and SDK implementation, as well as (2) the adaptations of plugins needed to realise the human-centred ontology verification outlined in Section 5.1.

5.3.1 Platform Core and SDK

The platform core implementation strictly follows the proposed component structure presented in Figure 4.5 and the package organisation presented in Figure 4.4, both as specified by the reference architecture (c.f. Chapter 4). The final implementation can be found at the platform core git repository¹¹.

For each of the component, a Java interface is created, which is documented using *Javadoc*, to enable extension and the concrete implementation by creating a *Spring Component*. The majority of the components of the core, are responsible for coordinating the flow of data. As this is already described by the reference architecture and the process models, these are not discussed in detail.

The components *Tripe Store* and *Meta Data Store* are different, as they are responsible for persisting data and thus will be briefly discussed. As for the triple store, a file-based solution is implemented. Each ontology uploaded to the platform is persisted in an own folder. Additionally, sub-ontologies created by extraction of ontological elements during verification task creation, are also persisted in the respective ontology folder. The meta data store uses a object relation mapping to persist the information of published verifications in a relational database. The prototype, uses a file-based *H2 database*¹².

What is described next, is the dynamic loading of plugins to realise new kinds of human-centred ontology verifications. The mechanism for loading the plugins is based on two components: (1) a set of available plugins implementing pre-defined plugin interfaces from the SDK and (2) a plugin registry.

The SDK defines four Java interfaces, *IVerificationTaskPlugin*, *IContextProviderPlugin*, *ICrowdsourcingConnectorPlugin* and *IProcessorPlugin*, and their respective methods that need to be implemented for concrete instantiations of the plugins. Following functionality is provided by the interfaces:

- *IVerificationTaskPlugin*: Allows extracting ontological elements, specifying a template and a mechanism to resolve templating variables to define the Graphical User Interface (GUI) of a HIT.

¹⁰<https://maven.apache.org/>

¹¹<https://github.com/k-klemens/hc-ov-core>

¹²H2 Database Engine: <https://www.h2database.com/html/main.html>

- *IContextProviderPlugin*: Allows to extract provide contextual information for a given set of ontological elements.
- *ICrowdsourcingConnectorPlugin*: Allows connecting to a crowdsourcing platform to publish verification tasks, monitor them and to obtain the results.
- *IProcessorPlugin*: Allows processing the results obtained for a published verification.

As an elaborate discussion of all methods and data-structures would be out of scope for this thesis, please consult the *Javadoc* of the git project of the SDK¹³ for more detailed information. Further, a set of sample plugins can be found at the *sample plugin repository*¹⁴. The implementation of the already mentioned *Processor Plugin* for obtaining the raw results is found at *RawDataProcessorPlugin.java*¹⁵.

Figure 5.3 depicts the sequence of setting up the plugin registry and loading required plugins. Note that the exact method calls might vary depending on the selected run-time environment, however, the approach of initializing and loading is still the same.

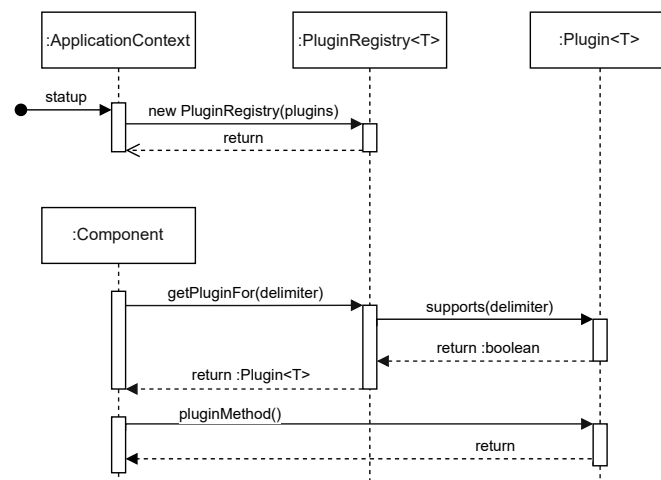


Figure 5.3: Sequence diagram of plugin registry initialisation and plugin loading.

During startup the *ApplicationContext* creates a *PluginRegistry* for each type *T* of plugin and initializes the registry with a list of available plugins. Next, once any *Component* requires some specific plugin, this components calls a registry with a given delimiter describing the plugin. Subsequently the plugin registry calls all available *Plugins'* *support* methods with the delimiter, to determine whether a suitable plugin is available or not. Once a suitable plugin is found by the registry, the plugin is returned to the component and the component can make use of the provided functionality.

¹³<https://github.com/k-klemens/hc-ov-sdk>

¹⁴<https://github.com/k-klemens/hc-ov-sample-plugins>

¹⁵<https://github.com/k-klemens/hc-ov-core/tree/master/src/main/java/at/kk/mcsc/hcov/core/service/processing/plugin>

As already introduced when describing the technology stack, the interfaces are implemented using REST over HTTP and documented using *Swagger* and *OpenApi 3.0*. In total, six different interfaces are provided by the platform. As a detailed description of each of the interfaces, would exceed the scope of this section, more detailed information can be found in the respective *OpenApi 3.0* specification¹⁶ and in Section 4.2.

5.3.2 Plugin Implementation

The core platform implementation as such, does not allow to create and publish HITs for a concrete ontology verification. To create and publish the verification tasks for the case study three plugins are implemented by using the SDK as a maven project dependency:

- *RestrictionVerificationPlugin*¹⁷: This plugin is responsible for defining how the universal quantification and existential quantification axioms are extracted from a given ontology. Further, a HTML template and a method on how to extract values from the ontology for each variable in a template are specified to define the GUI of the HITs. By using a configuration property the axioms can be rendered as “*Warren*” or as “*Rector*” formalism.
- *PizzaMenuContextProviderPlugin*¹⁸: As shown in literature [35], including context to ontology verification task helps achieve better quality results, thus a *IContextProviderPlugin* is implemented for the Pizza ontology to summarise all the toppings of a Pizza in a restaurant menu style. The contextual information of a given set of ontological elements include the name of pizza, a list of specified toppings and a URL to a random image of a pizza.
- *AMTCrowdsourcingConnector*¹⁹: Following the use case description, the created tasks shall be published on AMT. The functionality of the crowdsourcing connector includes publishing the tasks on AMT, retrieving the current status of the published verification and also obtaining the raw results from the platform. For each requested verification, first a *HITType* is created which is then used to create the concrete *HITs*. Additionally, by providing a *QuestionForm*²⁰ and a *AnswerKey*²¹

¹⁶<https://github.com/k-klemens/hc-ov-core/blob/master/src/main/resources/openapi.yaml>

¹⁷<https://github.com/k-klemens/hc-ov-pizza-verification-plugins/blob/master/src/main/java/at/kk/msc/hcov/plugin/pizza/RestrictionVerificationPlugin.java>

¹⁸<https://github.com/k-klemens/hc-ov-pizza-verification-plugins/blob/master/src/main/java/at/kk/msc/hcov/plugin/pizza/PizzaMenuContextProviderPlugin.java>

¹⁹<https://github.com/k-klemens/hc-ov-amt-connector>

²⁰https://docs.aws.amazon.com/AWSMechTurk/latest/AWSMturkAPI/ApiReference_QuestionFormDataStructureArticle.html

²¹https://docs.aws.amazon.com/AWSMechTurk/latest/AWSMturkAPI/ApiReference_AnswerKeyDataStructureArticle.html

as a configuration parameter, a new qualification type (i.e. qualification test for quality control purposes) is created and used with the *HITType*.

Note that no concrete *IProcessorPlugin* is implemented, as this is not scope of the case study. However, the processor plugin to obtain unprocessed raw results included with the core can be used for demonstration purposes.

5.4 Evaluation Results

Following the evaluation approach defined in Section 5.2, this section addresses the evaluation aspects *EA-1* to *EA-4* individually for the evaluation of the described case study in Section 5.1.

The screenshot shows the Amazon Mechanical Turk interface. At the top, there are navigation tabs for 'HITS', 'Dashboard', and 'Qualifications'. A search bar contains 'Pizza-Toppings-Verification-Rector' and a 'Filter' button. Below the search bar, it says 'All HITS' and 'Your HITS Queue'. A message indicates '1-1 of 1 results containing 'Pizza-Toppings-Verification-Rector''. The main section is titled 'HIT Groups' and includes options to 'Show Details' and 'Hide Details', and a dropdown for 'Items Per Page' set to '20'. A table lists the HIT details:

Requester	Title	HITS	Reward	Created	Actions
k-klemens	Pizza-Toppings-Verification-Rector Type	22	\$0.00	36s ago	Preview, Qualify

Below the table, there is a detailed view of the HIT:

Description	Time Allotted	Qualifications Required	Your Values
This is a test ontology verification of the restrictions in the pizza ontology	60 Min Expires in 9m	✘ Sample qualification test for: Pizza-Toppings-Verification-Rector is not less than 80	None Take Test

At the bottom of the screenshot, there are navigation links for 'Help', 'Contact', 'Legal', 'Service Health', and 'Feedback', along with the Amazon logo and copyright information: '© 2005-2022 Amazon Mechanical Turk Inc, or its affiliates. All rights reserved.' and 'An amazon company'.

Figure 5.4: Screenshots of the published verification showing an overview of the verification.

However, before the individual evaluation aspects are addressed in detail, a summary of the interaction steps with the platform, necessary to conduct the steps required for the case study is outlined. Note, that the implementation of the required plugins is seen as a prerequisite to the following sequence.


5. CASE STUDY: SUPPORTING HUMAN-CENTRED ONTOLOGY VERIFICATION

amazonmturk Worker Return

Pizza-Toppi... (HIT Details) Auto-accept next HIT k-klemens HITs 22 Reward \$0.00 Time 0:15 of 60 Min

[See Instructions](#)

Please make sure you are familiar with the rules and examples provided in the **Instructions** before answering the question.

Pizza Menu	Model
 <p>La Reine Ham, Mozzarella, Mushroom, Olive, Tomato</p>	<p>La Reine pizzas have, amongst other things, some Ham topping, and some Mozzarella topping, and some Mushroom topping, and some Olive topping, and some Tomato topping, and also only Ham, Mozzarella, Mushroom, Olive, and/or Tomato toppings.</p>

Does the model represent the pizza menu item correctly ?

- The model correctly represents the menu item.
- For the model to correctly represent the menu item, one or more existential (some) restrictions need to be added.
- For the model to correctly represent the menu item, one or more universal (only) restrictions need to be added.
- For the model to correctly represent the menu item, one or more universal (only) restrictions need to be replaced by existential (some) restrictions.
- For the model to correctly represent the menu item, one or more existential (some) restrictions need to be replaced by universal restrictions (only).

Comment (optional)
In case you have any remarks please add them here

[Report this HIT](#) | [Why Report](#) Return

Figure 5.5: Screenshots of the published verification showing an example HIT.

1. The `Pizza.owl` is uploaded with a given name to the platform by using the *UploadOntology* interface.
2. A verification specification following the interface definition of *VerificationSpec* is defined to create the verification of ontological restrictions of the pizza ontology. This specification includes the name of the verification, the name of the ontology to be loaded, which plugins to execute and the configuration of each plugin. As for the plugins, the specification defines the implemented plugins outlined in Section 5.3.2 and the processor plugin, allowing to obtain the raw data from AMT, to be executed. The configuration of the plugins includes the representing formalism to be used, general information required for AMT and the location of a qualification test.
3. The defined verification specification is then used to invoke the *VerificationSpec* interface to automatically extract the needed ontological elements, provide contextual information, create the UIs and publish the tasks, including the qualification test, on AMT. Once the tasks are published by the platform, all meta-data is stored

and a summary is returned. Figure 5.4 and 5.5 shows screenshots of the created verification on AMT.

5.4.1 EA-1: Automated Activities

To provide an overview of how the platform core can support ontology engineers, it is outlined which activities can be supported and how this is realised for the concrete case study. Figure 5.6 provides an overview of the supported activities by colour-coding the preparation process model (c.f. Figure 3.5) in green. In total seven out of the nineteen preparation activities are fully supported and an additional four out of the nineteen preparation activities are partially supported by the platform. Comparing this with the manual approach of the case study, (i.e. baseline; c.f. Figure 5.2) nine out of the sixteen conducted activities can be supported by the platform.

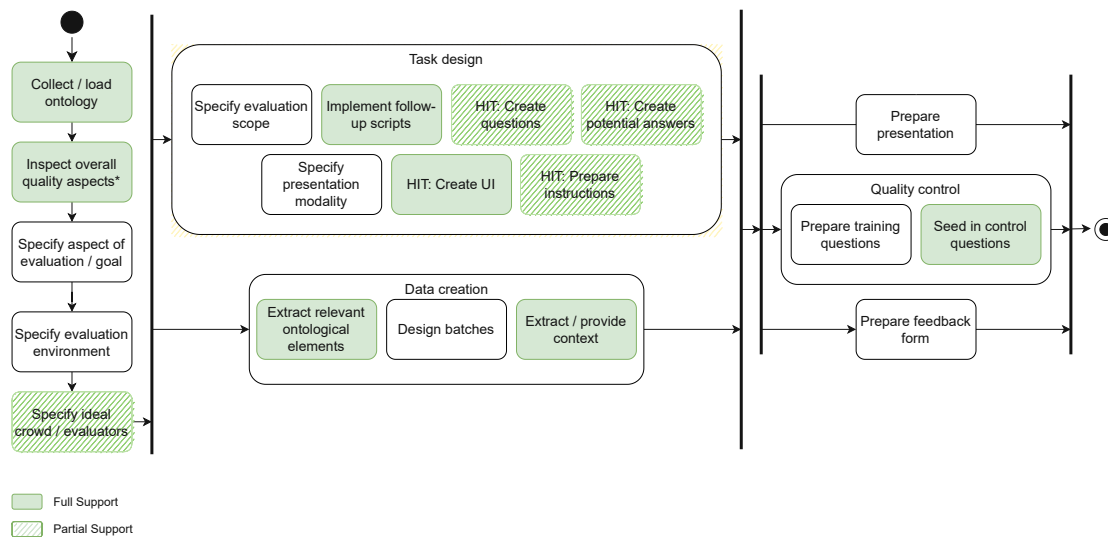


Figure 5.6: Preparation activities of a human-centred ontology verification coded to indicate whether they can be supported by the platform core.

The following enumeration discusses how the platform can support activities / groups of activities, how these are realised for the concrete case study of verifying ontological restrictions and how the support compares to the baseline (i.e. manual approach for preparing the verification of the case study and its activities depicted in Figure 5.2):

- **Collect / load ontology:** To prepare for a verification the given ontology needs to be uploaded to the system and this is supported, and also realized for the given use-case, by providing / using a REST interface and a file-based triple store. In

comparison with the baseline, this activity can be fully automated as only the .owl file and an ontology name need to be provided and all further operations are handled by the platform.

- **Inspect overall quality aspects:** The platform provides a component and data-structures to calculate metrics. However, as this was not part of the studied case, this aspect has not been implemented with concrete functionality and is just stubbed to demonstrate the possibility of calculating quality aspects.
- **Specify aspect evaluation / goal, Specify evaluation environment:** As these two activities require human-decisions which cannot be made, these activities cannot be supported by the platform. Thus these activities remain with manual execution.
- **Specify ideal crowd / evaluators:** Specifying demographic aspects of the crowd are activities which require human decision and thus cannot be supported. However, implementing certain configuration properties with a *Crowdsourcing Connector Plugin* allows supporting certain aspects of specifying the ideal crowd. For the given case study, the *Crowdsourcing Connector Plugin* was implemented to support publishing qualification tests, which then can ensure that the crowd workers have certain qualifications.
- **Specify evaluation scope, Specify presentation modality:** Similar to the activities *Specify aspect evaluation / goal* and *Specify evaluation environment*, specifying the evaluation scope and the presentation modality require human decisions and thus these cannot be automated and remain with manual execution for the case study.
- **Implement follow-up scripts:** By implementing a *Processor Plugin*, follow-up scripts or more precisely follow-up processing methodologies can be specified. As the focus of the case study is on preparation activities, no special processing plugin is implemented. For demonstration purposes of the case study, the processor plugin, allowing to extract the raw data from the crowdsourcing platform, provided with the core is used.
- **HIT: Create questions, Create potential answers, Preparing instructions:** The support of this group of tasks is highly dependent on the concrete verification and thus can only be partially supported. For the given use case, the questions, the instructions as well as the potential answers were implemented with the *Verification Task Plugin* and thus in comparison to the baseline can be automated once a methodology for it is implemented.
- **HIT: Create UI:** Using the templating mechanism of *Thymeleaf* provided with the platform, HTML documents, which can be used as the UI for a micro-task, can be automatically created by implementing a *Verification Task Plugin*. In the case study, this was realised by providing different templates that allow to automatically

create UIs for different representation mechanisms of ontological representations through the *RestrictionVerificationPlugin*. In comparison with the baseline, where this activity was conducted semi-automatically, creating the UIs for the verification tasks can be fully automated.

- **Extract relevant ontological elements:** By implementing a *Verification Task Plugin*, methodologies for extracting relevant ontological elements can be specified and thus automated. For the given case study, the *RestrictionVerificationPlugin* automatically extracts all restrictions of *Pizza* classes to sub-ontologies.
- **Design batches:** Designing or grouping tasks in batches is not implemented with the platform, as it was not required by the case studied. However, adapting the *Crowdsourcing Connector Plugin* allows realising such functionality, if it is supported by the crowdsourcing platform.
- **Extract / provide context:** By implementing a *Context Provider Plugin* contextual information can be automatically extracted for a given set of ontological elements. Comparing this with the baseline of the studied case, where contextual information was extracted manually, the platform allows automating this step by using the *PizzaMenuContextProviderPlugin*.
- **Prepare presentation, Prepare feedback form:** Both activities are highly related to the verification task on hand and are very likely to differ for each verification, thus these activities are not supported by the platform.
- **Prepare training questions:** The creation of training questions is not supported by the platform, as these do typically not include any extracted information and are hard-coded.
- **Seed in control questions:** Using a *Verification Task Plugin* and predefined templating variables, the platform supports the creation of randomly seeded control questions. As the studied case does not include control questions, this aspect is not evaluated in the scope of this thesis.

In addition to the discussion of the supported activities, it is important to emphasize, that each of the automated activities, requires the corresponding plugins to be implemented first. Further, going beyond preparation activities, the platform and the implemented plugins allow publishing, monitoring and retrieving the results of a created verification on AMT.

Summarising *EA-1*, the majority of the preparation activities required for the case study (c.f. yellow-coloured activities in Figure 5.2) can be supported once the plugins are implemented. Activities, which are not supported by the platform (e.g. “Specify evaluation scope”) require human decisions or are not expected to be reusable once automated, thus these activities cannot or intentionally were not implemented with the platform and plugins.

5.4.2 EA-2: Implementation Effort

This section of the evaluation discusses the implementation effort of the platform and of the implemented plugins to support the tasks required for the case study and draws a comparison with the manual approach as performed in [28].

Table 5.1 provides an overview of the effort required to implement the platform core, the plugins and to perform any additional activities, such as meetings and creating documentation. Note that all efforts are provided in *person-hours*. The majority of the effort was spent during the implementation of the core, while only a smaller part of the effort was required to implement the customization plugins. It is important to emphasize, that all implementation steps benefited of the reference architecture (c.f. Chapter 4). Thus the provided efforts for the core and plugin only reflect the implementation and not any specification or design efforts. As the main foundation of the platform core is provided by the *reference architecture*, an effort estimation, based on a personal week planner (six full working days were spent), for the creation and documentation of the *reference architecture* is included.

	Effort (h)
Platform core and SDK implementation	55
<i>RestrictionVerificationPlugin</i> implementation	11.5
<i>PizzaMenuContextProviderPlugin</i> implementation	3
<i>AMTCrowdsourcingConnector</i>	14
Misc. effort (e.g. meetings; documentation)	5.5
Reference architecture design and documentation	48*
Total effort	137
- <i>Core implementation</i>	55
- <i>Plugin implementation</i>	28.5
- <i>Misc. effort</i>	5.5

Table 5.1: Efforts for implemented the platform and required plugins to replicate the verification of [28]. (Efforts annotated with * are based on an estimation)

To provide a comparison between preparation efforts involved with the manual approach and the approach of implementing the required functionality to enable platform support for the verification of the case study, the author of [28] provided the time effort required for the preparation activities of the process model (c.f. Figure 5.2) as reported in Table 5.2. Note that for certain activities only estimates based on commit logs and weekly working hours were available. Further, the activity *Specify ideal crowd / evaluators* is split, as half of the effort for the activity was spend on implementing the *qualification test* and the remaining effort was spent for defining a self-assessment test and defining the required qualifications.

²²Listed for completeness; not scope of the case study

Activity	Effort (h)	Supported By	Reused
Collect / load ontology	0.25*	CORE	
Specify aspect of evaluation / goal	50		X
Specify evaluation environment	8		X
Specify ideal crowd / evaluators	10		X
Specify ideal crowd - QT implementation	10	CCP	
Specify evaluation scope	2*		X
Specify presentation modality	8		X
Implement follow-up scripts ²²	100	PP	-
HIT: Create questions	0.5	VTP	X
HIT: Create potential answers	20	VTP	X
HIT: Create UI	80	VTP	
HIT: Prepare instructions	20	VTP	X
Extract relevant ontological elements	100*	VTP	
Extract / provide context	4	CPP	
Prepare presentation	5		X
Prepare training questions	8		X
Prepare feedback form	1*		X
Total effort	426.75		
- <i>Supported and not reused</i>	<i>194,25</i>		
- <i>Reused</i>	<i>132.50</i>		
- <i>Not scope of case study</i>	<i>100.00</i>		

Table 5.2: Efforts for each activity *without platform support* as conducted in [28] representing the baseline for the effort based evaluation. “Supported by” column encoding: *CORE*: Platform core, *VTP*: Verification task plugin, *CCP*: Crowdsourcing connector plugin, *CPP*: Context provider plugin, *PP*: processor plugin; Efforts annotated with * are based on an estimation.

In addition to the required time efforts, Table 5.2 also includes information on which component is implemented to support the respective preparation activities. Further, as the author of [28] shared her resources for the preparation of the verification (e.g. a question template), the information in the table also indicates which resources are reused. Using this classification, efforts related to activities in which resources are reused, are omitted for comparison with Table 5.1, as these efforts would distort the comparison. A visual comparison reflecting these efforts is presented in Figure 5.7.

On a high level, the effort for implementing the platform-support (i.e. core and plugin implementations) is 137.00 hours (c.f. Table 5.1; Lower bar in Figure 5.7), while the comparable effort spent as part of preparation activities (i.e. activities supported by the platform and not reused) without platform support (i.e. baseline) is 29.47% higher with 194.25 hours. A potential reason why the platform-support shows an advantage over the manual approach might be related to the process models defined as part of this

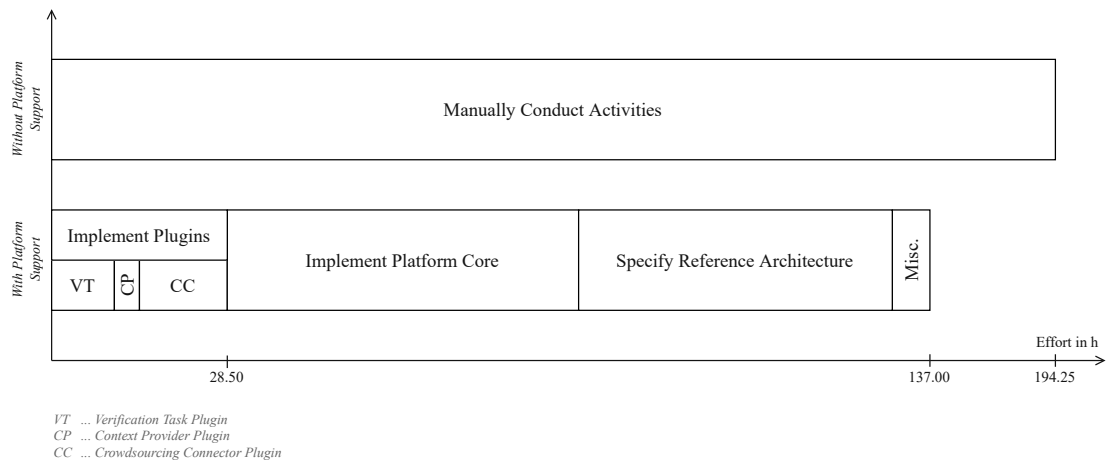


Figure 5.7: Comparison of efforts between the approach *without platform support* and *with platform support*.

thesis (c.f. Chapter 3). Having a well-defined process model on hand does not require an engineer to spend effort for defining which steps are potentially required, as these are already specified with the generic process model. Thus the implementation does not start from ground zero and certain aspects are already defined upfront. However, this aspect is not evaluated and thus remains an unverified hypothesis.

Another aspect important to emphasize, is that for future verifications, efforts for the implementation of the platform core are not required and thus a second comparison is provided under the assumption that the platform core is already available. In numbers, this means that 28.5 (c.f. row “Plugin implementation” in Table 5.1) hours were spent on implementing the plugins while the comparable efforts for preparing the verification is 194.25 hours (c.f. row “Supported and not reused” in Table 5.2; Upper bar in Figure 5.7). Phrased differently, the effort can be reduced by 88.33%. Hence, the approach of extending an existing implementation of the platform with plugins is even more effective, when compared to manually creating the verification tasks.

The following enumeration provides a more detailed comparison between the implementation efforts of the plugins only and the related effort for the manual activities (i.e. sum of related efforts, theoretically supported by a plugin as mentioned in the *Supported By* column of Table 5.2).

1. *Verification Task Plugin*: Implementing the *RestrictionVerificationPlugin* took 11.5 hours while significantly more effort, a total of 180 hours, was spent on preparation activities if no tool supported is used. Reasons for that might include, that certain functionality, such as coordinating and converting data between different activities, are all handled and defined already with the platform core and thus not need to be taken care of, while with the manual approach these steps are still to be expected.

2. *Context Provider Plugin*: Manually extracting and providing the context for ontological elements required 4 hours, while automating the approach with the *PizzaMenuContextProviderPlugin* took 3 hours. For the given case study, the efforts are not very different when comparing an automated to a manual approach. However, the larger the ontology gets in size, the more time-consuming manual context extraction gets, and thus the implementation of a context provider plugins shows an advantage if an ontology of a certain size is used.
3. *Crowdsourcing Connector Plugin*: For the implementation of the *AMTCrowdsourcingConnector* plugin, a total of 14 hours was spent, while the related manual activities only required 10 hours effort. However, as the plugin functionality goes beyond just publishing a qualification test, by also supporting publishing tasks or monitoring HITs, it is expected that about the same time is required independent of if a manual or a platform-supported approach is employed.

Concluding this evaluation aspect, it can be seen that implementation efforts of the platform for the given case study are smaller than with a manual approach and thus the benefits of the platform can be observed. However, it needs to be considered that the implementation of the platform is a replication of [28] and that for the given case study, certain aspects, such as the *process models* had a positive influence on the required efforts for implementation.

5.4.3 EA-3: Improved Aspects

This section outlines four aspects that can be improved when using the prototypical platform implementation to replicate the human-centred ontology verification of [28]. The information is based on qualitative data collect during a series of informal meetings and are potential ideas and need further studies to be evaluated and fully claimed. However for reasons of completeness, the ideas are outlined and discussed next.

Centralized Orchestration and Storage: As the platform implements end-to-end process support for human-centred ontology verification, the process activities, as well as, data can be orchestrated centrally.

Having predefined interfaces (e.g. *VerificationSpec*) to interact with the platform enables *centralized orchestration of all process activities*. This way the users interacting with the platform do not need to take care of any execution dependencies between the process activities, as this is taken care of by the platform core based on the specification. In addition to the execution dependencies, data structures to be used by the components of the platform are defined, hence data conversion happens automatically and in a predefined manner.

Apart from the centralized orchestration, the platform also provides *centralized data storage*. More specifically the prototype includes a file-based triple-store, to store the ontologies and extracted sub-ontologies used to create the HITs, and a relational database,

to store information about published verifications. Having these storage components, the possibility of missing data and incompatible data can be reduced. Further, as also the information about the verification itself is stored, exact replications of a verification can be enabled. Users could benefit from that information if a verification needs to be redone as, for example, an ontology grows and newly added data needs to be verified.

Extensibility and Reusability: One of the requirements proposed by the stakeholders in Section 4.2.2, specifies the need for an extensible platform. In the context of the case study, the platform demonstrates the capability of extension by using the SDK and implementing plugins. In this way, a wide spectrum of verification tasks can be supported by implementing new plugins while benefiting from the already existing infrastructure provided by the platform core.

Another aspect, which is closely related to extensibility, is reusability. Once a plugin is implemented, future human-centred ontology verifications might not require all plugins to be implemented, as for example a *Crowdsourcing Connector Plugin* is already implemented for the desired crowdsourcing platform. Thus overall implementation efforts are expected to shrink as the availability of plugins grows.

Considering the column *Reused* in Table 5.2, the case study also demonstrates that verifications already done without tool-support can be automatized by partially reusing artifacts from previous verifications. These artifacts can help when implementing extension plugins and hence, overall implementation effort can be reduced.

Data Scalability: Considering Figure 5.2, for the manual approach of the case study, certain activities, such as *Extract / provide context* were performed completely manually. These manual activities might become infeasible when ontologies, that require verification, grow. For example, with the given *Pizza ontology*, only 22 sub-ontologies /axioms to be verified were extracted, however with larger ontologies, such as *SNOMED CT*²³, more sub-ontologies are expected to be extracted. Hence, automation is required to efficiently conduct the verifications. As the platform allows to automate these manual activities by implementing respective plugins, data scalability in terms of growing ontologies is provided.

Platform Independence: Current tools supporting human-centred ontology verification (e.g. [5]) are often tightly integrated to existing editors or platform, thus reducing their usability if different editors or platforms are used. The prototypical platform implementation overcomes this limitation by providing well defined interfaces (e.g. REST over HTTP documents with OpenApi 3.0) and hence the core logic is abstracted from the user interfaces. In that way, already existing editors or platforms could provide extensions that communicate with the interfaces of the end-to-end process support platform and so integrate its functionalities.

²³<https://bioportal.bioontology.org/ontologies/SNOMEDCT>

5.4.4 EA-4: Limitations and Differences

Complementary to the aspects improved when using the prototypical platform, also the limitations and differences between the manual and the tool-supported approach of preparing the human-centred verification of the case study need to be discussed. To that end, this section discusses three aspects. Similar to the results presented with *EA-3*, the limitations are based on qualitative data collected during a series of informal meetings and further evaluations is needed to proof them.

Automation of Specification Activities: Considering Figure 5.6, certain preparation activities of human-centred ontology verification are concerned with the specification of verification aspects (e.g. *Specify evaluation environment*). These activities typically require human decisions to be made and thus cannot be automated by the prototypical platform, as they are highly individual for each human-centred ontology verification. Considering Table 5.2, this means that 78 out of 426.75 hours of effort are related to activities concerned with specification and hence about a fifth of the effort of the case studies effort cannot be automated and remain manual.

Supported Representation Mechanisms: For the human-centred ontology verification in [28] three different representation mechanisms of ontological restrictions were used: Rector[26], Warren[57] and VOWL[42]. However, the *Verification Task Plugin*, responsible for rendering the ontological restrictions implemented as part of the case study, does not support the VOWL formalism, which uses a graph to represent ontologies. During plugin implementation several possibilities were investigated to include VOWL support, however, none is considered feasible.

First, the methodology of [28] was analysed. The author manually created the representations by uploading the extracted sub-ontologies to *WebVOWL*²⁴, a web-based VOWL rendering platform, then took a screenshot of the graph and included it in the respective HIT. However, as the goal was to automate the preparation activity of creating the verification tasks, this approach is considered unsuitable.

Second, a set of libraries was identified that could potentially be included with the plugin to render the sub-ontologies. To that end *WebVOWL*²⁵, *OWL2VOWL*²⁶ and *ProtegeVOWL*²⁷ were identified:

- *WebVOWL*: As already outlined previously, one unsuitable option for using *WebVOWL* is manually rendering the representations and using screenshots. Another option of using *WebVOWL* is directly including it in the *Verification Task Plugin*, however as the plugin implementations are based on *Java* and *WebVOWL* is based on *JavaScript* this is also not possible.

²⁴<http://vowl.visualdataweb.org/webvowl.html>

²⁵<http://vowl.visualdataweb.org/webvowl.html>

²⁶<https://github.com/VisualDataWeb/OWL2VOWL>

²⁷<https://github.com/VisualDataWeb/ProtegeVOWL>

- *OWL2VOWL*: The library supports converting a `.owl` ontology to a *JSON* object, that contains the needed information for rendering a VOWL graph. However, the *JSON* representation is not intended to be used by humans and it would require implementing the rendering logic specified in the *VOWL 2.0 Specification*²⁸, which would exceed the scope of this thesis.
- *ProtegeVOWL*: *ProtegeVOWL* is a *Protégé* plugin to visualise ontologies in the VOWL formalism. An analysis of the plugin revealed that it is tightly integrated into the editor and would introduce a major dependency to the *Protégé* editor, which by definitions of the requirements of the reference architecture (c.f. Section 4.2.2) should be avoided and hence also this option is considered unsuitable.

It is important to note that the comparison of the implementation efforts outlined with evaluation aspect *EA-2* in Table 5.1 and Table 5.2, still remain valid, as a major part of the efforts was related to researching possibilities on how to include *VOWL*.

Published Task Structures: Apart from the supported rendering mechanisms, a difference between the automated approach using the prototypical platform implementation and the manual approach in [28] can be observed in the structure of the published tasks.

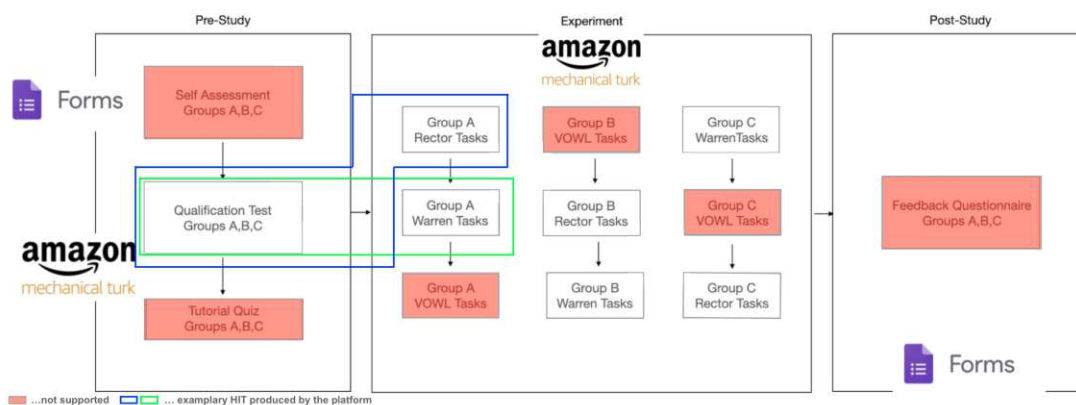


Figure 5.8: Experiment and verification design of [28] annotated with task design as produced by the platform. Source: Adapted from [28, Figure 4.2]

Figure 5.8 provides an overview of how the published tasks created by the platform are different from the published tasks created during the experiment/verification in [28]. During the verification two steps (i.e. *Self Assessment* and *Feedback Questionnaire*) were published using *Google Forms*, as these are highly specific to the experiment and not specific to human-centred ontology verification, this aspect is not supported by the platform. The remaining steps (i.e. *Qualification Test*, *Tutorial Quiz*, *Rector Tasks*, *Warren Tasks* and *VOWL Tasks*) were all published on AMT, however, the structure of

²⁸<http://vowl.visualdataweb.org/v2/>

how the tasks are published with tool-support differ from the manual approach and are discussed in the following paragraphs.

First, the *Tutorial Quiz* is equivalent to the outcome of the preparation activity *Prepare training questions* (c.f. Figure 3.5). As already outlined with *EA-1*, these questions are not supported as they are typically hard-coded and thus do not benefit from automation.

Second, as discussed with the first limitation in this section, the *VOWL* representation mechanism is not supported and thus also not replicated with the case study.

Third, it remains to discuss the structure of the published tasks for *Qualification Test*, *Rector Tasks* and *Warren Tasks*. In [28], three different types of HITs were published on AMT, one for each task. However, as the platform is focused on human-centred ontology verification using crowdsourcing platforms, the level of control is not as high as with the experiment in [28], which the verification of the studied case is part of. Thus it is not possible to separate the qualification tasks from the actual verification tasks (i.e. *Rector Tasks* or *Warren Tasks*), as this could result in workers not passing the qualification test accepting and working on the verification tasks. To avoid such cases, the *Qualification Tests* was divided to only either include ontological restrictions in *Warren* or *Rector* formalism. Then, these adapted qualification tests were published together either with the *Warren* or *Rector* tasks as coherent units and so ensuring that the crowdsourcing platform requires the correct qualification before allowing workers to accept and work on verification tasks. A comparison of this task structures can be observed by the blue and green outlines in Figure 5.8.

A final difference related to the verification tasks is concerted with the element extraction methodology. With the manual approach, ontological restrictions on the properties *hasTopping* and *hasBase* of the *Pizza ontology* are used. The element extraction mechanism implemented for the tool-supported approach, only extracts restrictions on *hasTopping* properties, as for the *hasBase* properties only two sets of relevant ontological elements would be extracted and the general extraction approach remains equivalent.

5.5 Summary

This chapter addresses **RQ3** and investigates how the process models (c.f. Chapter 3) and the reference architecture (c.f. Chapter 4) can support the preparation of human-centred ontology verifications.

To assess the level of support, in Section 5.1, a case study replicating the human-centred ontology verification as found in [28] and the experiment therein is defined. The main evaluation approach is outlined in Section 5.2 and can be summarized as providing tool-support for the verification and comparing it to the approach without tool-support.

First, based on the process models and the reference architecture, a prototypical implementation of an end-to-end process support platform and a SDK are implemented (c.f.

Section 5.3.1). Next, the SDK is used to provide three extension plugins for the platform to automate the required steps for the case study (c.f. Section 5.3.2).

The evaluation results, as reported in Section 5.4, show that a majority of the preparation activities can at least be partially supported by the end-to-end process support platform. In addition, the process models are a useful tool to communicate the process, as it is used to specify the case study. Further, comparing the implementation efforts of the platform and required extensions to the preparation effort without tool support, promising results can be observed. The implementation efforts are 29.47% smaller than the efforts required for performing the preparation activities manually. If the comparison excludes the implantation efforts of the platform core and solely focuses on the plugin implementation, the efforts are 85.33% smaller when compared to the manual approach. Also, certain positive aspects, including centralized orchestration, centralized data storage, extensibility, scalability and platform independence, can be observed, while only minor limitations, mainly related to the concrete ontology verification task studied, can be identified.

Conclusion & Future Work

Ontology verification, focusing on identifying modelling errors, is a vital activity in the ontology engineering process to ensure the correctness of the represented knowledge. Although certain classes of ontology errors can be identified automatically by algorithms, still, a set of error classes remains, that requires human knowledge to be identified. The latter type of verification also termed *human-centred ontology verification*, can for example be approached by *Human Computation (HC)* principles. This approach involves breaking down the verification problem into smaller *micro-tasks* and publishing them on crowdsourcing platforms to be solved by layman workers. In contrast to a more traditional approach of employing domain experts and ontology engineers, such a crowdsourcing approach, as described above, promises to be more time-, as well as, cost-effective.

Based on the literature for *human-centred ontology verification*, two gaps to be addressed by this thesis can be identified. First, there is no agreed-upon methodology/reference process for conducting *human-centred ontology verifications* using HC techniques. Second, there is no widely-accepted single tool that supports the whole process, thus most authors rely on different, or no tools, during the preparation, execution and follow-up work of *human-centred ontology verifications*.

As a result, the main research goals of this thesis can be summarized as follows. To begin with, a process model capturing the activities involved along the whole process of *human-centred ontology verification* shall be defined to enable effective planning, implementation and communication thereof. Next, based on the process model a reference architecture shall be established to allow the implementation of an extensible platform that provides tool support for the whole process. Finally, it shall be studied to what extent the process can be supported by an implementation of the reference architecture and which efforts are related to the implementation.

Section 6.1 presents the conclusions on the research questions and Section 6.2 discusses the limitations of the work as well as concrete starting points for future work.

6.1 Conclusions of the Research Questions

RQ1: *What is the typical process of human-centred ontology verification?*

An iterative approach, comprising a SLR, four SSIs and a focus group, was used to address this research question. On a high level, three distinct phases of the human-centred ontology verification process supported by Human Computation (HC) techniques, namely (1) *preparation*, (2) *execution*, and (3) *follow-up*, are revealed.

As a starting point for defining the phases, a small subset of the reviewed literature elicits activities related to *human-centred ontology verification* as summarized in Table 3.3. Moreover, the results of the SLR confirm the lack of tool-support, as only a small set of heterogeneous and platform-dependent tools that support individual activities of the process are identified.

Independent from the results of the SLR, also the experts described divergent views on the process phases during the SSIs, indicating the importance of a well-defined process. These divergent views are best illustrated by considering the overlap of activities in the Tables 3.4, 3.5 and 3.6.

The results of the SLR and SSIs were then used during the focus group to define “VeriCoM 2.0” (Verifying Conceptual Models (VeriCoM)), a set of three process models, agreed upon by the interviewed experts, defining the sequence of activities during each of the three process phases. The process models are depicted in Figures 3.5, 3.6 and 3.7, and specify nineteen preparation, nine execution and eight follow-up activities, respectively.

In conclusion, “VeriCoM 2.0”, providing a comprehensive view of the process, and the lack of tool support, present the opportunity to implement a platform to support and partially automate the process of *human-centred ontology verification*.

RQ2: *What are key requirements for software modules and a reference architecture that automate/support the “VeriCoM 2.0” process?*

The contribution of this research question establishes a reference architecture that enables the implementation of a platform providing end-to-end tool support for human-centred ontology verifications conducted on crowdsourcing platforms. The high-level requirements to define the reference architecture are defined in Section 4.2.2 and include (1) the ability to provide tool-support for “VeriCoM 2.0”, (2) tool independence, (3) extendability for future verification tasks and (4) a well-documented structure.

To that end, Figure 4.3 provides an entry point to the comprehensive documentation of the reference architecture organised in four viewpoints. First, the *runtime viewpoint*, based on “VeriCoM 2.0”, specifies the dynamic behaviour. Second, the *deployment viewpoint* depicted in Figure 4.4 organises the applications in packages to illustrate dependencies and to document the overall structure. Third, the *sourcecode viewpoint* represented in Figure 4.5, defines a set of components and corresponding interfaces to be implemented. In addition, the structure of the components follows a plugin-based approach to enable the extension of the platform. Fourth, a hierarchical taxonomy, as

discussed in Section 4.1, specifies the domain vocabulary of human-centred ontology verification representing the *crosscutting viewpoint*.

RQ3: *To what extent does an implementation of the reference architecture support the preparation of human-centred ontology verification?*

To gain insights into the extent of support provided by an implementation of the reference architecture, a case study, involving a human-centred ontology verification of ontological restrictions, was conducted.

As depicted in Figure 5.6 the results show that a prototypical implementation can fully support and automate seven out of nineteen preparation activities and further, a set of four out of the nineteen preparation activities can be partially supported. Activities not supported by the implementation either require human decisions or are hard-coded for each verification, so for these activities any reusability through automation is not likely.

Section 5.4.2 provides a comparison of the efforts required to implement a prototypical platform to the efforts required to manually prepare the verification exercise. The customization and extension efforts, required to tailor a platform to the verification of ontological restrictions, are 28.50 hours, while the efforts of the comparable manual activities are 194.25 hours. Even if all implementation efforts (i.e. efforts related to the implementation of the platform itself and the implementation of the extensions) of a platform are considered, only 137.00 hours are spent. Thus it shows, that implementing and using a platform is a more efficient approach than manually preparing the verification exercise.

In summary, the extensive set of supported activities and comparable low implementation and customisation efforts of the platform show the effectiveness of the support provided by an implementation of the reference architecture.

6.2 Limitations & Future Work

Based on the research questions and the conclusions presented in the previous section, this section identifies limitations and proposes ideas for future work.

Process Model Formalism: The process models summarised as “VeriCoM 2.0” and defined in Section 3.4 are based on an *informal flow-chart representation*, which is sufficient in the context of this thesis as solely a sequence of activities is described. However, adhering to a more elaborate standard, such as the *Business Process Model and Notation (BPMN)*¹, would allow the creation of a richer model. For example, the process models could be enriched with information about roles and responsibilities or information about activity results.

¹<https://www.omg.org/spec/BPMN/>

Process Model Classification: Another aspect related to the first research question is that “VeriCoM 2.0” is *not classified and linked to models of related domains*. Identifying similar processes that focus on human-centred verification of information resources, for example from the natural language processing domain, could help identify possible synergies based on parallels in the models and thus improve the overall process.

Reference Architecture Evaluation: The reference architecture, proposed as a direct result of RQ2 in Chapter 4, is only *indirectly evaluated through the case study’s evaluation* in Chapter 5. The reason for this is that the checklist-based approach proposed by the followed methodology, *ProSA-RA*[14], is no longer publicly accessible. A more sophisticated evaluation approach could involve interviews with domain experts and software architects that evaluate the architecture against its requirements. Based on this data, the reference architecture could then be extended and improved.

Prototypical Implementation: Albeit the implemented platform, as part of the case study evaluated in Chapter 5, is well tested and able to create the required tasks for the verification of ontological restrictions, it still needs to be considered *a prototypical implementation of the reference architecture*. This is mainly due to the simplified implementation of the *Triple Store* and the *Ontology Metrics* components, however, for the context of the case study, these are deemed sufficient.

The implementation of the *Triple Store* is file-based and could be replaced by a standalone solution, such as *GraphDB*², to obtain a more performant platform.

For the *Ontology Metrics* component, the implementation defines the required interfaces, however, no methodology to calculate actual metrics is provided. Based on existing libraries or publications, a set of useful metrics could be identified and implemented with the component as part of future work.

Case Study Scope: The scope of the case study is limited by two factors: (1) the *studied task* and (2) the *collected data*.

Because the case study focuses on the verification of ontological restrictions found in a small ontology, the Pizza ontology, it is expected that not the full scope of efficiency that can be enabled by automation is studied. The verification of a larger ontology would have resulted in more tasks and thus the implementation efforts are expected to be even smaller when compared to manually preparing all aspects of the verification. However, this remains an untested hypothesis and could be investigated in future work.

The data collected as part of the case study is mainly of quantitative nature. Hence, claims about potential benefits, apart from effort reduction, such as centralized orchestration or storage remain to be verified through qualitative methods.

²<https://graphdb.ontotext.com/>

Plugin Availability: Closely related to the previous limitation, only *four plugins to extend the platform core* are implemented as part of the case study. As one of the expected advantages of the automation of a human-centred ontology verification is to be able to reuse certain parts, future work could implement extension plugins for other types of ontology verification problems.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Figures

1.1	Overview of the thesis methodology which relies on Design Science cycles and various methods specific to each research question. Adapted and extended from [7, Figure 2].	4
2.1	Main stages of VeriCoM.	17
2.2	HIT interface used by [6] a) showing data related to the model and b) showing the verification questions. Source: [6, Figure 2]	17
2.3	Main stages of CSI.	18
3.1	Overview of the SMS execution phase. Source: [9]	23
3.2	Literature selection process and quantitative overview.	28
3.3	Preliminary process model of preparation activities.	40
3.4	High level process view of human-centred ontology verification.	51
3.5	Final process model for the preparation phase of human-centred ontology verification.	52
3.6	Final process model for the execution phase of human-centred ontology verification.	57
3.7	Final process model for the follow-up phase of human-centred ontology verification.	59
4.1	Terminological hierarchy of terms used for human-centred ontology verification.	63
4.2	<i>ProSA-RA</i> approach overview and relevant resource questions. Adapted and extended from [14, Fig. 1]	65
4.3	Viewpoints and diagrams used to build the reference architecture.	72
4.4	Package diagram (UML 2.5) depicting the deployment viewpoint of the reference architecture.	73
4.5	Component diagram (UML 2.5) depicting the sourcecode viewpoint of the reference architecture.	74
5.1	Example interface of a HIT used by [28]. Source: [28, Figure 4.1]	80
5.2	Activities conducted for preparing the verification in [28].	81
5.3	Sequence diagram of plugin registry initialisation and plugin loading.	85
5.4	Screenshots of the published verification showing an overview of the verification.	87
5.5	Screenshots of the published verification showing an example HIT.	88
		107

5.6	Preparation activities of a human-centred ontology verification coded to indicate whether they can be supported by the platform core.	89
5.7	Comparison of efforts between the approach <i>without platform support</i> and <i>with platform support</i>	94
5.8	Experiment and verification design of [28] annotated with task design as produced by the platform. Source: Adapted from [28, Figure 4.2]	98
1	Individual Process Model of Expert 1	152
2	Individual Process Model of Expert 2	153
3	Individual Process Model of Expert 3	154
4	Individual Process Model of Expert 4	155

List of Tables

2.1	Matrix showing which approaches are suitable for certain evaluation levels. Source: [17, Table 1]	8
3.1	Clusters of keywords used by the SMS (adapted from [9])	22
3.2	Fields added to the data extraction form of the SMS.	25
3.3	Coded set of preparation activities order by occurrences.	29
3.4	Overlap of preparation activities from the SLR and SSI. Activities which are non-coded are solely identified by the experts.	38
3.5	Consolidated execution activities elicited by the experts.	42
3.6	Consolidated follow-up activities elicited by the experts.	43
4.1	Identified information sources for building the reference architecture	66
4.2	High-level requirements obtained from stakeholder meetings. Trace M reflects requirements to be obtained from a meeting note.	68
4.3	High-level requirements obtained from stakeholder meetings. Source IS from an identified information source, Source I-RE from an initial requirement. . . .	69
4.4	Architectural requirements obtained from aggregating the system require- ments.	70
4.5	Mapping of the architectural requirements to the taxonomy.	71
5.1	Efforts for implemented the platform and required plugins to replicate the verification of [28]. (Efforts annotated with * are based on an estimation)	92
5.2	Efforts for each activity <i>without platform support</i> as conducted in [28] repre- senting the baseline for the effort based evaluation. “Supported by” column encoding: <i>CORE</i> : Platform core, <i>VTP</i> : Verification task plugin, <i>CCP</i> : Crowd- sourcing connector plugin, <i>CPP</i> : Context provider plugin, <i>PP</i> : processor plugin; Efforts annotated with * are based on an estimation.	93



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Acronyms

- AMT** Amazon Mechanical Turk. 2, 12–14, 30, 32, 53, 68, 69, 80, 83, 86, 88, 89, 91, 98, 99, 127, 130, 132, 133
- AWS** Amazon Web Services. 83
- CSI** Crowdsourced Software Inspection. 18, 107
- EER** Extended Entity Relationship. 16–18
- GUI** Graphical User Interface. 84, 86
- GWAP** Games with a Purpose. 12, 31, 53, 134, 135, 138, 139
- HC** Human Computation. ix, 1, 2, 7, 11–14, 19, 79, 101, 102
- HIT** Human Intelligence Task. 12, 14, 17, 27, 30, 32, 40, 55, 56, 77, 80, 83, 84, 86, 88, 95, 97, 99, 107, 126–128, 130, 132, 134, 135, 137, 139
- NLG** Natural Language Generation. 124
- OCR** Optical Character Recognition. 11
- SDK** Software Development Kit. 82, 84–86, 96, 99, 100
- SLR** Systematic Literature Review. 14, 19, 21, 22, 24, 27, 28, 32–34, 38, 40, 42, 43, 45–47, 49, 51, 53, 56, 60, 68, 102, 109
- SMS** Systematic Mapping Study. 5, 11, 21–28, 107, 109
- SSI** Semi-structured Interview. 5, 21, 33–38, 45–47, 51, 53, 54, 60, 102, 109
- TDD** Test Driven Development. 83
- UI** User Interface. 54, 56, 57, 88, 90, 91
- VeriCoM** Verifying Conceptual Models. 16–18, 60, 102, 107



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Bibliography

- [1] R. Studer, V. R. Benjamins, and D. Fensel, “Knowledge engineering: Principles and methods,” *Data & knowledge engineering*, vol. 25, no. 1-2, pp. 161–197, 1998.
- [2] R. Iqbal, M. A. A. Murad, A. Mustapha, and N. M. Sharef, “An analysis of ontology engineering methodologies: A literature review,” *Research journal of applied sciences, engineering and technology*, vol. 6, no. 16, pp. 2993–3000, 2013.
- [3] L. von Ahn, “Human computation,” in *2008 IEEE 24th International Conference on Data Engineering*, 2008, pp. 1–2.
- [4] J. Mortensen, M. A. Musen, and N. Noy, “Developing crowdsourced ontology engineering tasks: An iterative process,” in *CrowdSem*, 2013.
- [5] G. Wohlgenannt, M. Sabou, and F. Hanika, “Crowd-based ontology engineering with the ucomp protégé plugin,” *Semantic Web*, vol. 7, pp. 379–398, 05 2016.
- [6] M. Sabou, D. Winkler, P. Penzerstadler, and S. Biffl, “Verifying conceptual domain models with human computation: A case study in software engineering,” *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 6, no. 1, pp. 164–173, Jun. 2018. [Online]. Available: <https://ojs.aaai.org/index.php/HCOMP/article/view/13325>
- [7] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS quarterly*, pp. 75–105, 2004.
- [8] A. R. Hevner, “A three cycle view of design science research,” *Scandinavian journal of information systems*, vol. 19, no. 2, p. 4, 2007.
- [9] M. Sabou, M. Fernandez, M. Poveda-Villalón, M. C. Suárez-Figueroa, and S. Tsaneva, “Human-centric evaluation of semantic resources: A systematic mapping study,” in preparation for ACM Comp. Surveys.
- [10] W. C. Adams *et al.*, “Conducting semi-structured interviews,” *Handbook of practical program evaluation*, vol. 4, pp. 492–505, 2015.
- [11] J. Kontio, J. Bragge, and L. Lehtola, *The Focus Group Method as an Empirical Tool in Software Engineering*, 01 2008, pp. 93–116.

- [12] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” 2007.
- [13] N. L. Leech and A. J. Onwuegbuzie, “A typology of mixed methods research designs,” *Quality & quantity*, vol. 43, no. 2, pp. 265–275, 2009.
- [14] E. Y. Nakagawa, M. Guessi, J. C. Maldonado, D. Feitosa, and F. Oquendo, “Consolidating a process for the design, representation, and evaluation of reference architectures,” in *2014 IEEE/IFIP Conference on Software Architecture*, 2014, pp. 143–152.
- [15] N. F. Noy, D. L. McGuinness *et al.*, “Ontology development 101: A guide to creating your first ontology,” 2001.
- [16] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.
- [17] J. Brank, M. Grobelnik, and D. Mladenic, “A survey of ontology evaluation techniques,” in *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*. Citeseer Ljubljana, Slovenia, 2005, pp. 166–170.
- [18] A. Maedche and S. Staab, “Measuring similarity between ontologies,” in *EKAW*, 2002.
- [19] R. Porzel and R. Malaka, “A task-based approach for ontology evaluation,” in *Proc. of ECAI 2004 Workshop on Ontology Learning and Population*, Valencia, Spain, August 2004.
- [20] C. Brewster, H. Alani, S. Dasmahapatra, and Y. Wilks, “Data driven ontology evaluation,” in *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04)*. Lisbon, Portugal: European Language Resources Association (ELRA), May 2004. [Online]. Available: <http://www.lrec-conf.org/proceedings/lrec2004/pdf/737.pdf>
- [21] A. Lozano-Tello and A. Gomez-Perez, “Ontometric: a method to choose the appropriate ontology,” *J. Database Manag.*, vol. 15, pp. 1–18, 04 2004.
- [22] J. Raad and C. Cruz, “A survey on ontology evaluation methods,” in *Proceedings of the International Conference on Knowledge Engineering and Ontology Development, part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2015.
- [23] H. Hlomani and D. Stacey, “Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey,” *Semantic Web Journal*, vol. 1, no. 5, pp. 1–11, 2014.

- [24] A. Gómez-Pérez, *Ontology Evaluation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 251–273. [Online]. Available: https://doi.org/10.1007/978-3-540-24750-0_13
- [25] A. Gomez-Perez, “Some ideas and examples to evaluate ontologies,” 03 1995, pp. 299–305.
- [26] A. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, and C. Wroe, “Owl pizzas: Practical experience of teaching owl-dl: Common errors common patterns,” vol. 3257, 10 2004, pp. 63–81.
- [27] M. Poveda-Villalón, A. Gómez-Pérez, and M. C. Suárez-Figueroa, “OOPS! (Ontology Pitfall Scanner!): An On-line Tool for Ontology Evaluation,” *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 10, no. 2, pp. 7–34, 2014.
- [28] S. S. Tsaneva, “Human-centric ontology evaluation,” Master’s thesis, Wien, 2021.
- [29] E. Law and L. von Ahn, “Human computation,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 5, no. 3, pp. 1–121, Jun. 2011. [Online]. Available: <https://doi.org/10.2200/s00371ed1v01y201107aim013>
- [30] A. J. Quinn and B. B. Bederson, “Human computation: a survey and taxonomy of a growing field,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’11. New York, NY, USA: Association for Computing Machinery, 2011, p. 1403–1412. [Online]. Available: <https://doi.org/10.1145/1978942.1979148>
- [31] L. von Ahn and L. Dabbish, “Labeling images with a computer game,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’04. New York, NY, USA: Association for Computing Machinery, 2004, p. 319–326. [Online]. Available: <https://doi.org/10.1145/985692.985733>
- [32] L. von Ahn, R. Liu, and M. Blum, “Peekaboom: A game for locating objects in images,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’06. New York, NY, USA: Association for Computing Machinery, 2006, p. 55–64. [Online]. Available: <https://doi.org/10.1145/1124772.1124782>
- [33] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, “recaptcha: Human-based character recognition via web security measures,” *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.
- [34] M.-C. Yuen, I. King, and K.-S. Leung, “A survey of crowdsourcing systems,” in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, 2011, pp. 766–773.

- [35] N. F. Noy, J. Mortensen, M. A. Musen, and P. R. Alexander, “Mechanical turk as an ontology engineer? using microtasks as a component of an ontology-engineering workflow,” in *Proceedings of the 5th Annual ACM Web Science Conference*, ser. WebSci '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 262–271. [Online]. Available: <https://doi.org/10.1145/2464464.2464482>
- [36] J. Mortensen, P. R. Alexander, M. A. Musen, and N. Noy, “Crowdsourcing ontology verification,” in *ICBO*, 2013.
- [37] J. Mortensen, M. Musen, and N. Noy, “Crowdsourcing the verification of relationships in biomedical ontologies,” *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, vol. 2013, pp. 1020–9, 11 2013.
- [38] P. F. Green and M. Rosemann, “Integrated process modeling: An ontological evaluation,” *Inf. Syst.*, vol. 25, pp. 73–87, 2000.
- [39] I. Niles and A. Pease, “Towards a standard upper ontology,” in *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, ser. FOIS '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 2–9. [Online]. Available: <https://doi.org/10.1145/505168.505170>
- [40] M. A. Haendel, F. Neuhaus, D. Osumi-Sutherland, P. M. Mabee, J. L. Mejino, C. J. Mungall, and B. Smith, *CARO – The Common Anatomy Reference Ontology*. London: Springer London, 2008, pp. 327–349. [Online]. Available: https://doi.org/10.1007/978-1-84628-885-2_16
- [41] A. L. Rector, S. Brandt, and T. Schneider, “Getting the foot out of the pelvis: modeling problems affecting use of SNOMED CT hierarchies in practical applications,” *J. Am. Medical Informatics Assoc.*, vol. 18, no. 4, pp. 432–440, 2011. [Online]. Available: <https://doi.org/10.1136/amiajnl-2010-000045>
- [42] S. Lohmann, S. Negru, F. Haag, and T. Ertl, “Visualizing ontologies with VOWL,” *Semantic Web*, vol. 7, no. 4, pp. 399–419, 2016. [Online]. Available: <http://dx.doi.org/10.3233/SW-150200>
- [43] M. Sabou, K. Käsžnar, M. Zlabinger, S. Biffl, and D. Winkler, “Verifying extended entity relationship diagrams with open tasks,” *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 8, no. 1, pp. 132–140, Oct. 2020. [Online]. Available: <https://ojs.aaai.org/index.php/HCOMP/article/view/7471>
- [44] D. Winkler, M. Sabou, S. Petrovic, G. Carneiro, M. Kalinowski, and S. Biffl, “Improving model inspection processes with crowdsourcing: Findings from a controlled experiment,” in *Systems, Software and Services Process Improvement*, J. Stolfa, S. Stolfa, R. V. O’Connor, and R. Messnarz, Eds. Cham: Springer International Publishing, 2017, pp. 125–137.

- [45] J. M. Mortensen, E. P. Minty, M. Januszyk, T. E. Sweeney, A. L. Rector, N. F. Noy, and M. A. Musen, “Using the wisdom of the crowds to find critical errors in biomedical ontologies: a study of SNOMED CT,” *J Am Med Inform Assoc*, vol. 22, no. 3, pp. 640–648, May 2015.
- [46] J. Waitelonis, N. Steinmetz, M. Knuth, and H. Sack, “Whoknows? evaluating linked data heuristics with a quiz that cleans up dbpedia,” *Interact. Techn. Smart Edu.*, vol. 8, pp. 236–248, 11 2011.
- [47] M. Amith, F. J. Manion, M. R. Harris, Y. Zhang, H. Xu, and C. Tao, “Expressing Biomedical Ontologies in Natural Language for Expert Evaluation,” *Stud Health Technol Inform*, vol. 245, pp. 838–842, 2017.
- [48] J. M. Mortensen, N. Telis, J. J. Hughey, H. Fan-Minogue, K. Van Auken, M. Dumontier, and M. A. Musen, “Is the crowd better as an assistant or a replacement in ontology engineering? an exploration through the lens of the gene ontology,” *Journal of Biomedical Informatics*, vol. 60, pp. 199–209, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1532046416000277>
- [49] R. Lin, M. T. Amith, C. Liang, R. Duan, Y. Chen, and C. Tao, “Visualized emotion ontology: A model for representing visual cues of emotions,” *BMC Medical Informatics and Decision Making*, vol. 18, 07 2018.
- [50] G. R. Calegari, A. Fiano, and I. Celino, “A framework to build games with a purpose for linked data refinement,” *CoRR*, vol. abs/1811.02848, 2018. [Online]. Available: <http://arxiv.org/abs/1811.02848>
- [51] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer, and J. Lehmann, “Crowdsourcing linked data quality assessment,” in *The Semantic Web – ISWC 2013*, H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, and K. Janowicz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 260–276.
- [52] M. Sabou, L. Aroyo, K. Bontcheva, A. Bozzon, and R. Qarout, “Semantic web and human computation: The status of an emerging field,” *Semantic Web*, vol. 9, pp. 291–302, April 2018, © 2018 IOS Press. [Online]. Available: <https://eprints.whiterose.ac.uk/133568/>
- [53] C. B. Seaman, “Qualitative methods in empirical studies of software engineering,” *IEEE Trans. Software Eng.*, vol. 25, pp. 557–572, 1999.
- [54] S. Hove and B. Anda, “Experiences from conducting semi-structured interviews in empirical software engineering research,” vol. 2005, 10 2005, pp. 10 pp.–.
- [55] R. A. Krueger and M. A. Casey, “Focus group interviewing,” *Handbook of practical program evaluation*, vol. 3, pp. 378–403, 2010.

- [56] A. Sharp and P. McDermott, *Workflow modeling: tools for process improvement and applications development*. Artech House, 2009.
- [57] P. Warren, P. Mulholland, T. Collins, and E. Motta, “Improving comprehension of knowledge representation languages: A case study with description logics,” *International Journal of Human-Computer Studies*, vol. 122, pp. 145–167, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1071581918305068>
- [58] A. Finnerty, P. Kucherbaev, S. Tranquillini, and G. Convertino, “Keep it simple: Reward and task design in crowdsourcing,” ser. CHIItaly ’13. New York, NY, USA: Association for Computing Machinery, 2013. [Online]. Available: <https://doi.org/10.1145/2499149.2499168>
- [59] J. Cohen, “Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit.” *Psychological bulletin*, vol. 70 4, pp. 213–20, 1968.
- [60] V. B. Sinha, S. Rao, and V. N. Balasubramanian, “Fast dawid-skene: A fast vote aggregation scheme for sentiment classification,” *arXiv preprint arXiv:1803.02781*, 2018.
- [61] B. Gallagher, “Using the architecture tradeoff analysis method to evaluate a reference architecture: A case study,” Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU/SEI-2000-TN-007, 2000. [Online]. Available: <http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=5109>
- [62] M. Kezadri-Hamiaz and M. Pantel, “First steps toward a verification and validation ontology.” 01 2010, pp. 440–444.
- [63] A. A. Efremov and K. I. Gaydamaka, “IncoSE guide for writing requirements. translation experience, adaptation perspectives,” in *Proceedings of the CEUR Workshop Proceedings, Como, Italy*, 2019, pp. 9–11.
- [64] J. F. M. Santos, M. Guessi, M. Galster, D. Feitosa, and E. Y. Nakagawa, “A checklist for evaluation of reference architectures of embedded systems (S),” in *The 25th International Conference on Software Engineering and Knowledge Engineering, Boston, MA, USA, June 27-29, 2013*. Knowledge Systems Institute Graduate School, 2013, pp. 451–454.
- [65] M. Winter, C. Brooks, and J. Greer, “Towards best practices for semantic web student modelling.” NLD: IOS Press, 2005, p. 694–701.
- [66] M. Richards, *Software architecture patterns*. O’Reilly Media, Incorporated 1005 Gravenstein Highway North, Sebastopol, CA . . . , 2015, vol. 4.
- [67] K. Beck, *Test Driven Development. By Example (Addison-Wesley Signature)*. Addison-Wesley Longman, Amsterdam, 2002.

- [68] R. C. Martin and J. O. Coplien, *Clean code: a handbook of agile software craftsmanship*. Upper Saddle River, NJ [etc.]: Prentice Hall, 2009. [Online]. Available: https://www.amazon.de/gp/product/0132350882/ref=oh_details_o00_s00_i00
- [69] N. Alrajebah and H. Al-Khalifa, “Extracting ontologies from arabic wikipedia: A linguistic approach,” *Arabian Journal for Science and Engineering*, vol. 39, pp. 2749–2771, 04 2013.
- [70] N. Anand, J. R. Van Duin, and L. Tavasszy, “Framework for modelling multi-stakeholder city logistics domain using the agent based modelling approach,” *Transportation Research Procedia*, vol. 16, pp. 4–15, 12 2016.
- [71] D. Aréchiga, F. Crestani, and J. Vegas, “Ontology-driven word recommendation for mobile web search,” *The Knowledge Engineering Review*, vol. 29, no. 2, p. 186–200, 2014.
- [72] R. Barros, P. Kislansky, L. do Nascimento Salvador, R. Almeida, M. Breyer, L. G. Pedraza, and V. Vieira, “EDXL-RESCUER ontology: an update based on faceted taxonomy approach,” in *Proceedings of the Brazilian Seminar on Ontologies (ONTOBRAS 2015), São Paulo, Brazil, September 8-11, 2015*, ser. CEUR Workshop Proceedings, F. Freitas and F. Baião, Eds., vol. 1442. CEUR-WS.org, 2015. [Online]. Available: http://ceur-ws.org/Vol-1442/paper_19.pdf
- [73] R. Bouzidi, A. De Nicola, F. Nader, and R. Chalal, “Ontogamif: A modular ontology for integrated gamification,” *Applied Ontology*, vol. 14, pp. 1–35, 06 2019.
- [74] T. J. Bright, E. Yoko Furuya, G. J. Kuperman, J. J. Cimino, and S. Bakken, “Development and evaluation of an ontology for guiding appropriate antibiotic prescribing,” *J. of Biomedical Informatics*, vol. 45, no. 1, p. 120–128, feb 2012. [Online]. Available: <https://doi.org/10.1016/j.jbi.2011.10.001>
- [75] P. Charlton, G. Magoulas, and D. Laurillard, “Enabling creative learning design through semantic technologies,” *Technology, Pedagogy and Education*, vol. 21, no. 2, pp. 231–253, 2012. [Online]. Available: <https://doi.org/10.1080/1475939X.2012.698165>
- [76] Y. Chen, X. Peng, B. Zhong, and H. Luo, “Application of ontology in vulnerability analysis of metro operation systems,” *Structure and Infrastructure Engineering*, vol. 12, no. 10, pp. 1256–1266, 2016. [Online]. Available: <https://doi.org/10.1080/15732479.2015.1110602>
- [77] C.-C. Chou, A.-P. Jeng, C.-P. Chu, C.-H. Chang, and R.-G. Wang, “Generation and visualization of earthquake drill scripts for first responders using ontology and serious game platforms,” *Advanced Engineering Informatics*, vol. 38, pp. 538–554, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474034617304317>

- [78] P. Ciancarini, A. D. Iorio, A. G. Nuzzolese, S. Peroni, and F. Vitali, “Evaluating citation functions in cito: Cognitive issues,” in *ESWC*, 2014.
- [79] F. Corno, L. De Russis, and A. Monge Roffarello, “A high-level semantic approach to end-user development in the internet of things,” *International Journal of Human-Computer Studies*, vol. 125, pp. 41–54, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1071581918301228>
- [80] G. Demartini, D. Difallah, and P. Cudré-Mauroux, “Large-scale linked data integration using probabilistic reasoning and crowdsourcing,” *VLDB Journal*, vol. 22, no. 5, pp. 665–687, Oct. 2013, copyright: Copyright 2013 Elsevier B.V., All rights reserved.
- [81] K. Eckert, M. Niepert, C. Niemann, C. Buckner, C. Allen, and H. Stuckenschmidt, “Crowdsourcing the assembly of concept hierarchies,” in *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, ser. JCDL ’10. New York, NY, USA: Association for Computing Machinery, 2010, p. 139–148. [Online]. Available: <https://doi.org/10.1145/1816123.1816143>
- [82] N. M. El-Gohary and T. E. El-Diraby, “Domain ontology for processes in infrastructure and construction,” *Journal of Construction Engineering and Management*, vol. 136, no. 7, pp. 730–744, 2010.
- [83] A. Dumitrache, L. Aroyo, and C. Welty, “Achieving expert-level annotation quality with crowdtruth: The case of medical relation extraction,” 2015.
- [84] A. Getman and V. Karasiuk, “A crowdsourcing approach to building a legal ontology from text,” *Artificial Intelligence and Law*, vol. 22, pp. 313–335, 09 2014.
- [85] H. Guo, S. Gao, J. Krogstie, H. Trætteberg, and A. I. Wang, “An Evaluation of Ontology Based Domain Analysis for Model Driven Development,” *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 11, no. 4, pp. 41–63, October 2015. [Online]. Available: <https://ideas.repec.org/a/igg/jswis0/v11y2015i4p41-63.html>
- [86] Y. Gutiérrez, D. Tomás, and I. Moreno, “Developing an ontology schema for enriching and linking digital media assets,” *Future Generation Computer Systems*, vol. 101, pp. 381–397, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167739X18323859>
- [87] D. Kontokostas, A. Zaveri, S. Auer, and J. Lehmann, “Triplecheckmate: A tool for crowdsourcing the quality assessment of linked data,” in *Proceedings of the 4th Conference on Knowledge Engineering and Semantic Web*, 2013. [Online]. Available: http://jens-lehmann.org/files/2013/kesw_triplecheckmate.pdf
- [88] Y.-L. Kuo, J. Hsu, and F. Shih, “Contextual commonsense knowledge acquisition from social content by crowd-sourcing explanations,” *AAAI Workshop - Technical Report*, pp. 18–24, 01 2012.

- [89] M. Lassaad, H. Raja, and H. H. B. Ghezala, ““onto-computer-project”, a computer project domain ontology : Construction and validation,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 3, 2020. [Online]. Available: <http://dx.doi.org/10.14569/IJACSA.2020.0110345>
- [90] A. Malhotra, M. Gündel, A. M. Rajput, H.-T. Mevissen, A. Saiz, X. Pastor, R. Lozano-Rubi, E. H. Martinez-Lapsicina, I. Zubizarreta, B. Mueller, E. Kotelnikova, L. Toldo, M. Hofmann-Apitius, and P. Villoslada, “Knowledge retrieval from pubmed abstracts and electronic medical records with the multiple sclerosis ontology,” *PLOS ONE*, vol. 10, no. 2, pp. 1–12, 02 2015. [Online]. Available: <https://doi.org/10.1371/journal.pone.0116718>
- [91] C. Mazo, L. Salazar, O. Corcho, M. Trujillo, and E. Alegre, “A histological ontology of the human cardiovascular system,” *Journal of Biomedical Semantics*, vol. 8, p. 47, 10 2017.
- [92] T. Mioch, L. Kroon, and M. A. Neerincx, “Driver readiness model for regulating the transfer from automation to human control,” in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, ser. IUI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 205–213. [Online]. Available: <https://doi.org/10.1145/3025171.3025199>
- [93] S. Nazir, Y. H. Motla, T. Abbas, A. Khatoun, J. Jabeen, M. Iqra, and K. Bakhat, “A process improvement in requirement verification and validation using ontology,” in *Asia-Pacific World Congress on Computer Science and Engineering*, 2014, pp. 1–8.
- [94] J. Evermann and J. Fang, “Evaluating ontologies: Towards a cognitive measure of quality,” *Information Systems*, vol. 35, no. 4, pp. 391–403, 2010, vocabularies, Ontologies and Rules for Enterprise and Business Process Modeling and Management. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306437908000689>
- [95] A. Pomp, J. Lipp, and T. Meisen, “You are missing a concept! enhancing ontology-based data access with evolving ontologies,” in *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, 2019, pp. 98–105.
- [96] V. Presutti, A. Nuzzolese, S. Consoli, A. Gangemi, and D. Reforgiato Recupero, “From hyperlinks to semantic web properties using open knowledge extraction,” *Semantic Web*, vol. 7, pp. 351–378, 05 2016.
- [97] H. Rijgersberg, M. Wigham, and J. Top, “How semantics can improve engineering processes: A case of units of measure and quantities,” *Advanced Engineering Informatics*, vol. 25, no. 2, pp. 276–287, 2011, information mining and retrieval in design. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474034610000753>

- [98] T. J. Bright, E. Yoko Furuya, G. J. Kuperman, J. J. Cimino, and S. Bakken, “Development and evaluation of an ontology for guiding appropriate antibiotic prescribing,” *Journal of Biomedical Informatics*, vol. 45, no. 1, pp. 120–128, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1532046411001675>
- [99] M. Sabou, A. Scharl, and M. Föls, “Crowdsourced knowledge acquisition: Towards hybrid-genre workflows,” *International Journal on Semantic Web Information Systems*, vol. 9, pp. 14–41, 07 2013.
- [100] D. Thakker, S. Karanasios, E. Blanchard, L. Lau, and V. Dimitrova, “Ontology for cultural variations in interpersonal communication: Building on theoretical models and crowdsourced knowledge,” *Journal of the Association for Information Science and Technology*, vol. 68, 10 2016.
- [101] A. Vizcaíno, F. García, M. Piattini, and S. Beecham, “A validated ontology for global software development,” *Computer Standards Interfaces*, vol. 46, pp. 66–78, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0920548916300046>
- [102] E. Younesi, A. Malhotra, M. Gündel, P. Scordis, A. T. Kodamullil, M. Page, B. Müller, S. Springstubbe, U. Wüllner, D. Scheller, and M. Hofmann-Apitius, “PDON: Parkinson’s disease ontology for representation and modeling of the Parkinson’s disease knowledge domain,” *Theor Biol Med Model*, vol. 12, p. 20, Sep 2015.
- [103] M. Zhitomirsky-Geffet, E. S. Erez, and B.-I. Judit, “Toward multiviewpoint ontology construction by collaboration of non-experts and crowdsourcing: The case of the effect of diet on health,” vol. 68, no. 3, p. 681–694, mar 2017.
- [104] Y. Zhou, S. Schockaert, and J. Shah, “Predicting ConceptNet path quality using crowdsourced assessments of naturalness,” in *The World Wide Web Conference on - WWW '19*. ACM Press, 2019. [Online]. Available: <https://doi.org/10.1145%2F3308558.3313486>
- [105] S. K. Kondreddi, P. Triantafillou, and G. Weikum, “Combining information extraction and human computing for crowdsourced knowledge acquisition,” in *2014 IEEE 30th International Conference on Data Engineering*, 2014, pp. 988–999.
- [106] A. Malhotra, E. Younesi, M. Gündel, B. Müller, M. T. Heneka, and M. Hofmann-Apitius, “ADO: a disease ontology representing the domain knowledge specific to Alzheimer’s disease,” *Alzheimers Dement*, vol. 10, no. 2, pp. 238–246, Mar 2014.

Appendices

Appendix A: SLR Summaries

Following summaries only extract relevant parts of the publication with regards to the human centred ontology evaluation approaches found.

S2 - Extracting Ontologies from Arabic Wikipedia: A Linguistic Approach [69]

The authors extracted an ontology from Arabic Wikipedia using a linguistic approach by facilitating the semi-structured nature of the articles. For evaluation, three experiments were conducted, where two evaluation approaches (i.e. evaluating correctness and consistency using an online survey; evaluating the structure of a subset of sentences by domain experts) have human involvement. With regard to the human centred evaluation approaches no information about the process of preparing the evaluation is provided.

S3 - Framework for modelling multi-stakeholder city logistics domain using the agent based modelling approach [70]

In this paper agent based modelling is used to foster the understanding of the interactions between heterogeneous stakeholders in the city logistics domain. As a foundation for the agent based modelling approach a conceptual model is needed, for which in this case an ontology (i.e. “GenOLOn” Generic City Logistics Ontology) was used. To ensure the correctness of the agent based model, the authors stress the need for human centred qualitative ontology evaluation. By collecting data from stakeholder interviews and other city logistics models the used ontology was evaluated. Apart from the types of evaluation, no further information yielding insights to the process human centered ontology validation are provided.

S4 - Ontology-driven word recommendation for mobile Web search [71]

Differences in web search can be observed whether it is accessed from a desktop or a mobile device. As mobile devices have become more important the authors want to improve web search by employing an ontology capturing contextual information of the users (e.g. location, network connection, device size). Through a survey, the authors

evaluated whether the ontology is able to represent the context of the mobile search environment. Little information is provided for the questions of the survey, however further information about preparation is not included.

S5 - EDXL-RESCUER ontology: An update based on faceted taxonomy approach [72]

Within this work, crowdsourcing shall provide information to command centres for emergency and crisis management. An ontology (i.e. “EDXL-RESCUER”) was created, to support the coordination and exchange of information between legacy systems. Evaluation of the created ontology was performed by applying a brainstorming technique with stakeholders of the systems. During the brainstorming sessions, stakeholders were asked to find synonyms or correlate terms used in their daily context.

S7 - OntoGamif: A modular ontology for integrated gamification [73]

The main goal of the authors of this publication is to create an ontology covering the gamification domain. The evaluation of the ontology was addressed on multiple levels. Only on the functional level, a human centred evaluation was performed. It encompassed collecting feedback about ontological elements (i.e. missing concepts and relationships, the relevance of concepts and relationships) from domain experts using questionnaires. The form used to collect the feedback is available but the process of creating the form is not outlined.

S8 - Development and evaluation of an ontology for guiding appropriate antibiotic prescribing [74]

Contextually this work is situated in the domain of clinical decision support systems for guiding prescription antibiotics. As these systems are often developed locally, hampering reuse and sharing knowledge, this study aimed to develop an ontology for this domain. As for human centred evaluation, the goal was to assess the correctness as well as the usefulness of the created ontology. A workshop like structure with two domain experts was employed to reconstruct the hierarchy of the ontology concepts, hence assessing correctness. During the usefulness evaluation, the domain experts were asked to complete tasks in protege. A basic outline of the evaluation tasks is provided but no information about the preparation of the tasks can be identified within this reference.

S11 - Expressing biomedical ontologies in natural language for expert evaluation [47]

Ensuring the correctness of ontologies often required involving domain experts in the evaluation process, however, those often have only little to no engineering knowledge. Thus the authors applied a Natural Language Generation (NLG) approach aimed at translating ontological axioms to natural language statements to enable the evaluation.

S13 - Enabling creative learning design through semantic technologies [75]

The authors aim at supporting the creation of learning designs by employing a tool called “Learning Designer” which is backed by an ontology. Using the tool, learning designs are semantically annotated by concepts of the ontology. One of the key aspects emphasized is, that the ontologies allow common concepts and vocabularies to be established. By approaching creating learning design by digital means, several aspects are changed, as for example, the common concepts of the ontology allow analytic tasks or the relations of the ontology enable sharing, re-use and similarity discovery. The evaluation was not targeted at the ontology, rather it was targeted at the whole system itself. A group of learning design practitioners was asked to work with the tool in order to assess whether the backed ontology aligns with the user’s views.

S14 - Application of ontology in vulnerability analysis of metro operation systems [76]

The authors focused on creating an ontology to enable multiple stakeholders, involved in the process of making metro systems less susceptible to errors, to coordinate their efforts and to use a common knowledge base. In the scope of a case study the created ontology is evaluated by expert users in the domain with regards to its comprehensiveness and navigational ease. The first of two evaluation steps encompassed a detailed introduction to the research project and an exploration of the ontology in the “Protégé” editor. In the second evaluation step, the experts were asked to fill out a questionnaire about certain quality aspects of the ontology. Apart from the questionnaire design, no further detailed information about the evaluation, for example, it would have been of interest how the first step was conducted in detail, is provided.

S16 - Generation and visualization of earthquake drill scripts for first responders using ontology and serious game platforms [77]

In this study, the authors are concerned with creating drill scripts (i.e. scenarios of disaster events) for training emergency units and first responders. As the creation of these drill scripts can be rather timely and costly, more efficient approaches to generation are needed. To address this issue a system based on an ontology and a serious game was created. As a major part of the evaluation, the ontology in use was only evaluated indirectly by using the system with a group of senior domain experts. Apart from the evaluation of the whole system, also the usefulness of the ontology itself was evaluated by one expert and the research team. The evaluation was conducted by examining the final drill scripts created by the ontology with the target of identifying inconsistent or unrealistic scenarios. Even though the results are clearly outlined, the process of conducting the evaluation is not described.

S17 - Evaluating citation functions in CiTO: Cognitive issues [78]

Assessing citations of scholarly articles is often a key activity within scientific communities. Characterising citations is a difficult task for both humans and computers. To enable the classification of citations, the authors use the “CiTO” (Citation Typing Ontology) as a model for providing semantic annotations for them. The main goal of the evaluation is to study the human mental models while annotating the citations. To that end, the annotations made by humans using the ontology were compared and questionnaires were collected. During the experiment, the participants were asked to read a citation and assign a property from the ontology. Using the results from the experiment, suggestions to improve the ontology were derived. The process for preparing the evaluation is briefly outlined, while the results are reported in detail.

S18 - A high-level semantic approach to End-User Development in the Internet of Things [79]

Using end-user development for IoT devices allows non-professional software developers (e.g. smarthome owners) to develop their own IoT applications through high-level application interfaces. The platforms available often do not provide a high level of abstraction thus disallowing non-professionals to develop their applications or lack the ability to abstract common functionalities from devices such that the context cannot be changed easily. To address these issues the author developed an ontology that shall help to include semantic information of the devices. By providing a platform backed by the ontology, novice users shall be able to create their own applications. The platform was then evaluated in an end-user study, where the goal was to study the effectiveness and efficiency as well as advantages and drawbacks enabled by a higher-level abstraction through the ontology. The study design and results are outlined in a detailed manner, however, insights solely targeting the evaluation process of ontology are not provided.

S20 - Large-scale linked data integration using probabilistic reasoning and crowdsourcing [80]

By employing a novel approach of combining algorithms and HITs the complex tasks of entity linking and instance matching in the context of linked open data are addressed by the authors. For both entity linking and instance matching first the system (i.e. “ZenCrowd”) performs a set of initial steps to propose a potential link or match respectively. Each of these proposed results is assigned a confidence value. Results having high confidence values will be directly used as a result of a task.

If a result is considered promising for an instance matching task but further judgement is needed whether is correct, a HIT is generated. The concrete steps for creating the HITs are based on using web page templates of tasks to be published on a crowdsourcing platform. The templates are filled with (1) the name of the source instance, (2) contextual information by querying a graph database and (3) a selection of top-k relevant matches from the preceding algorithmic steps. The authors experimented with two different task

templates and found the “molecule” interface, directly showing data relevant to the proposed pairs, to be the more effective task design.

A similar approach can be found for the entity linking tasks. A crowdsourcing task was created showing the workers an entity and a set of URIs, from which several could be selected.

The human centred components of the instance matching task are outlined in great detail yielding insights about the process of preparing the HITs. However, for entity linking the human centred components are outlined only on a high level. Even though this publication focuses on ontology triples rather than the evaluation of full-sized ontologies, insights with regards to the human centred process can be obtained.

S22 - Crowdsourcing the Assembly of Concept Hierarchies [81]

Creating concept hierarchies can be a demanding task as oftentimes expert knowledge is required. The authors propose an approach for creating concept hierarchies through *is a* relationships by employing crowdsourcing techniques. To provide a frame of reference for comparison, the crowdsourced results are compared to an expert based created hierarchy (i.e. “InPhO”(Indiana Philosophy Ontology)).

An experiment on AMT was conducted to compare expert results with crowd worker results. The preparation steps are weakly outlined in the paper as only a screenshot of the task interface and selected aspects (e.g. seeding in a hidden gold standard) are reported.

S23 - Domain Ontology for Processes in Infrastructure and Construction [82]

The authors developed an ontology to cover the infrastructure and construction process domain knowledge in an explicit way to act as a core domain model to foster a shared understanding. The evaluation was addressed by three different views, two of them having human involvement.

One human centred evaluation targeted at assessing the coverage of the proposed competency questions by the writers. No process information with regards to this evaluation is provided by the authors.

The second evaluation approach including humans was based on conducting interviews with domain experts. First, an introduction to the research project was presented to the experts. Second, the ontology itself was presented by navigating through the hierarchy and additionally allowing experts to interactively browse the ontology in the Protégé editor. The final part of the interview was based on questionnaires to collect information about the participants themselves, abstraction and categorization effectiveness, navigational ease, and overall evaluation of the ontology. As for the abstraction and categorization effectiveness, the experts were asked to track the ontological hierarchy and rate the effectiveness of the hierarchy on a predefined scale. To address navigational ease, the

participants were given a concept, then asked to locate it in the ontology and finally to assess the difficulty of the navigation task. For the overall evaluation, the participants were given five questions, which are not specified further in the publication. On a high-level view process information about how the preparation of the ontology evaluation was done can be deduced from the outlined steps. However, no further information about the interfaces, tools or extraction processes of certain information for evaluation (e.g. how are the concept hierarchies extracted and shown to the experts) are provided by the study.

S24 - Achieving expert-level annotation quality with CrowdTruth: The case of medical relation extraction [83]

The authors apply a semantic annotation approach aiming at generating ground truth data in the domain of clinical natural language processing. Ground truth generation was addressed by extracting certain terms out of medical texts and asking participants to select a relations type (referred to as “RelExt”) as well as the relation direction (referred to as “RelDir”).

Overall the evaluation targets a comparison of crowdsourced relation annotation and expert-based relation annotations. For the target of this thesis, this is not of interest, however, the tasks found to generate the ground truth could provide useful information for constructing evaluation tasks related to ontological relations. The main process steps include extracting possible related terms by using distant supervision, creating tasks to select the concrete relation type and creating tasks to select the relation directions. The authors provide screenshots about the HITs, brief information about the pre-processing methodology and information about the crowdsourcing platform used. Further information about the process is not provided.

S27 - A crowdsourcing approach to building a legal ontology from text [84]

Within this study, the authors created an ontology in the domain of law. The ontology was populated by undergraduate law students during their learning activities. Experts (i.e. professors of the law department) were asked to assess the quality of the created ontology. The evaluation targeted verifying completeness in terms of concepts and their relations. Apart from the verified aspects, no further information about the evaluation process was provided.

S28 - An evaluation of ontology based domain analysis for model driven development [85]

Model driven development requires thorough domain analysis to achieve high software quality and high requirements coverage. Usually, these domain analysis approaches are heavyweight and do not fit well into iterative development processes such as in the context of pervasive game development. The authors addressed this issue by developing

an ontology (“PerGO” Pervasive Game Ontology) to make the results of domain analysis reusable and the whole process of domain analysis more lightweight. The evaluation of the ontology was addressed in three rounds, where the last round was conducted in a human centred fashion by collecting questionnaires from students. Before conducting the survey, a video and a presentation about the research and the ontology were shown to the study participants to introduce the context. No further information about the process of human centred ontology evaluation is provided by the authors.

S29 - Developing an ontology schema for enriching and linking digital media assets [86]

In this study, the authors developed an ontology (“DMA ontology” Digital Media Asset ontology) to include semantic information to various kinds of digital media. The motivation behind the development was to cover the requirements of all stakeholders involved in the life cycle of digital media assets. From a high level perspective, the evaluation of the ontology was conducted in three steps, namely, qualitative evaluation, quantitative evaluation and an evaluation with regards to the competency questions.

Only the latter evaluation required human involvement, as important stakeholders were asked to define a set of competency questions in order to assess the correctness and completeness of the ontology. The assessment was realized by executing SPARQL queries on the ontology. As these stakeholders typically are not familiar with querying ontologies using SPARQL, ontology experts transformed the competency questions to SPARQL queries and executed them. Eventually, the stakeholders assessed the results. The process is only outlined on a high-level as described within this paragraph and no information with regard to the used tools is provided by the authors.

S32 - Triplecheckmate: A tool for crowdsourcing the quality assessment of linked data [87]

With their work, the authors addressed quality assessment of linked open data. More specifically a tool (i.e. “Triplecheckmate”), following a proposed generic quality assessment methodology, was implemented to assess the quality of DBpedia triples (other data sources accessible by a SPARQL endpoint are available as well). The proposed generic quality assessment methodology consists of four steps (1) Resource selection, (2) Evaluation mode selection, (3) Resource evaluation and (4) Data quality improvement, which are described in the paper in more detail.

The process for assessing the quality of a triple starts with a user logging into the system, then selecting the types of triples to be checked and finally judging randomly shown triples. For each random triple, the user is asked to provide information whether it is considered correct or incorrect. If a triple is marked incorrect further information about the quality problem according to a given but extensible taxonomy needs to be provided. Further resources by the authors demonstrate the user interface and workflow of evaluating triples by videos.

S34 - Contextual Commonsense Knowledge Acquisition from Social Content by Crowd-Sourcing Explanations [88]

Harnessing commonsense, as well as contextual knowledge can yield powerful structures to deduce new knowledge. However, the authors claim that contextual knowledge is often not collected and thus they address this issue by employing a human computation approach. In essence, triples annotating social media content with contextual knowledge are created by using HITs and crowdsourcing.

To evaluate how the proposed tool can support the generation of contextual knowledge, an experiment on AMT using Twitter post was conducted. The main preparation steps included pre-processing of the tweets (i.e. duplicate elimination, removing links) and ordering of the tweets. For the task itself, ground truth questions were seeded in for higher trustworthiness of the crowdworkers.

S36 - “Onto-computer-project”, a computer project domain ontology: Construction and validation [89]

Within the scope of this work, the authors proposed an ontology to capture project information in the domain of computers. An ontology creation, as well as an evaluation approach, is presented.

The evaluation approach presented consists of six steps, where starting from a tabular format of the ontology, several expert feedback is collected and the consistency is checked using a reasoner. Even though the evaluation method is presented it is unclear how such an evaluation process is conducted in detail as the descriptions are rather high level and the created ontology is not evaluated within the scope of this work.

S40 - Visualized Emotion Ontology: a model for representing visual cues of emotions [49]

Understanding the emotions of patients can help improve the quality of medical services. An array of models exists that describe human emotions, however, most of these models are not yet formally represented in a machine-readable format, thus the authors address this gap by modelling an ontology (“VEO” Visualized Emotion Ontology) for describing emotions. Further, the authors also generated visual representations of the emotions, based on basic shapes and a set of colours, based on the ontological data.

The ontology itself was evaluated by using automated tools, the visual representations, on the other hand, were evaluated through a survey conducted on AMT. The participants were asked to rate the validity of a statement matching an emotion expressed by the generated visualisations on a scale from 1 to 5. Even though the ontology is indirectly evaluated by means of human computation techniques, no interesting insights for the preparation process of human centred ontology evaluation can be deduced from the publication.

S45 - Knowledge Retrieval from PubMed Abstracts and Electronic Medical Records with the Multiple Sclerosis Ontology [90]

The authors developed an ontology to collect information on multiple sclerosis. Apart from providing an integrated knowledge base to medical professionals another goal of the authors is to incorporate new knowledge from the growing scientific publications in this error.

The human centred evaluation approach of this paper was to evaluate competency questions proposed by experts. No further aspects are provided.

S47 - A histological ontology of the human cardiovascular system [91]

Within this publication, the authors proposed the implementation of a histological ontology of the human cardiovascular system. The three main goals include establishing a common knowledge base, enabling reuse and tracking of changes.

To validate the ontology a three-fold approach was selected. First, the ontology was scanned for pitfalls using “OOPS!” [27]. The second evaluation approach was human centred as two surveys were conducted. The first survey included students, which suggested improvements for the ontology, whereas the second survey was used to collect feedback from experts and to revalidate the results of the first survey. The surveys were designed to ask binary questions (yes/no) by employing a three-step process: first identifying elements to be validated, grouping elements and identifying the characteristics of the elements to be validated. The third evaluation approach included assessing the coverage of the competency questions using SPARQL queries.

S48 - Driver Readiness Model for Regulating the Transfer from Automation to Human Control [92]

Truck platooning is a scenario of autonomous driving where the first truck is driven by a chauffeur and the next trucks are following via a virtual tow bar. The drivers of the virtually towed trucks must be able to take over the truck in certain scenarios, thus the driver needs to be in a state of readiness. To support the assessment and reasoning about a drivers state and for guiding systems in the take-over process from automated to manual driving, an ontology was developed capturing the necessary concepts and relations.

Evaluation of the ontology was approached by expert interviews collecting information about clarity, coherence, completeness and consistency. No further information with regard to the expert interviews is provided.

S49 - Developing crowdsourced ontology engineering tasks: An iterative process [4]

This publication provides an overview of previous experiments of the authors addressing ontology verification using crowdsourcing. In particular, all experiments targeted at several aspects of verifying conceptual hierarchies using HITs.

A common task design and preparation approach was used for all experiments conducted. First, the ontological axioms to be verified were selected. Second, a task as a HTML form was created to verify the hierarchies and published on AMT. Finally, the results were collected, spam was removed and the final set of responses was compared to a gold standard.

S50 - Using the wisdom of the crowds to find critical errors in biomedical ontologies: a study of SNOMED CT [45]

Withing their work, the authors addressed the crowd-based verification of hierarchies in the medical ontology “SNOMED CT”. HITs were developed to collect information on whether a relationship between concepts is considered correct or incorrect.

The preparation involved filtering and extracting the `SubClassOf` axioms of the ontology using the “Snorocket” classifier. In a further preparation step, context information for each axiom was included in the task design to ensure high-quality results can be achieved. The tasks were then executed on CrowdFlower and aggregated using a Bayesian method.

S51 - Is the Crowd Better as an Assistant or a Replacement in Ontology Engineering? An Exploration through the Lens of the Gene Ontology [48]

With this work, the authors provide a replication study of their previous work [?] of verifying relationships in biomedical ontologies. In contrast to their earlier work, the “Gene Ontology” instead of the “SNOMED CT” ontology was the target of evaluation. As for the process, the same task design as in [?] was used. However, apart from the ontology, another set of differences from their previous work can be seen for the selection of the crowdsourcing platform, the exploration of the results and the general conclusion on when to apply crowdsourcing.

S52 - A process improvement in requirement verification and validation using ontology [93]

The success of software development processes is heavily dependent on the quality of the requirements specification. Providing a high-quality software requirements specification can be challenging for various reasons. One particular challenge outlined by the authors is that stakeholders participating in the requirements engineering process often do not share a common vocabulary and lack a formal language of the process. To address this

issue, a common understanding of the process was provided by developing an ontology to support the requirements engineering process.

The created ontology was evaluated in the scope of a case study in which experts were asked to rate certain quality aspects of the ontology. In addition, the authors defined competency questions to be translated to *DL queries* to assess whether the ontology is capable of providing the expected answers.

S56 - Mechanical Turk as an Ontology Engineer? Using Microtasks as a Component of an Ontology-Engineering Workflow [35]

Within this publication, the authors addressed several different aspects for verifying hierarchies in ontologies using crowdsourcing. First, to assess whether AMT workers perform different from students or not, the authors replicated a study from [94]. Evermann and Fang conducted an experiment where students were asked to assess the hierarchies found in the “BWW” ontology and the “SUMO” ontology. The authors of this publication replicated the study with a worker population from AMT.

The two follow up studies addressed what effect the domain of an ontology has on the quality of the work by repeating the process with “WordNet” and the “CARO” ontology. Generally, hierarchies extracted from “WordNet” reflect common sense whereas the axioms extracted from the “CARO” ontology reflect biomedical knowledge.

All experiments followed the same approach of generating binary questions (true/false) asking whether a natural language sentence, which was generated from the ontologies hierarchies, was correct or incorrect. Thus the preparation at least involved extraction of relevant ontological axioms and translation of these axioms to natural language. In addition, a set of control questions was included in the final task design and the overall question ordering was randomized. Further, certain task designs also required contextual information to be extracted and included.

S63 - You are Missing a Concept! Enhancing Ontology-Based Data Access with Evolving Ontologies [95]

Data lakes can be used as one paradigm to store large amounts of data, however, as the number of data sources increases, the data lakes become less transparent, discoverable and understandable. To address these issues the data can be accessed using “OBDA” (Ontology-Base Data Access), where an ontology establishes a common understanding of the data. However, the authors claim that also “OBDA” faces a challenge whenever new data not previously captured by any concepts in the ontology is introduced to the data lake, as the ontology needs to be extended. Thus the authors propose a dynamic approach including an extensible ontology on the “ESKAPE” OBDA platform.

During the evaluation, participants were asked to build and extend the ontology of the approach. As the evaluation was targeted at the assistant to create the ontological

concepts and not on ontology verification, no further information with regards to human centred ontology verification processes can be found.

S64 - From hyperlinks to Semantic Web properties using Open Knowledge Extraction [96]

The authors attended the generation of semantic web triples by employing a novel approach harnessing the contextual information provided by hyperlinks from natural text web pages. This open knowledge extraction approach is described extensively in the publication.

The evaluation of the extracted triples was approached by HITs on CrowdFlower. Overall, five different types of tasks were used to address several aspects of the developed approach. Two of the tasks expected binary answers, two expected answers from a predefined scale and one task expected open answers by the crowd workers. To be able to ensure trust of the workers, qualification questions were seeded into the questionnaire.

S67 - A Framework to Build Games with a Purpose for Linked Data Refinement [50]

GWAP are a flavour of human computation techniques, where problems are solved by letting users play fun computer games. The authors of this publication present a GWAP approach and technical framework to solve several aspects of data linking.

To create a new GWAP using the framework, first the data linking case to be solved needs to be defined. Second, the database needs to be populated with data resources to be linked and operational data (e.g. aggregation specific data). Finally, the application can either be run or be more specialised towards the final use-case.

Three instantiations are shown to outline the generality of the framework in the context of creating ground truth data for machine learning applications. Showing an asset (e.g. an image) to the user and letting them classify it, thus linking the asset to a label, is the common scheme found across the games presented.

S68 - How semantics can improve engineering processes: A case of units of measure and quantities [97]

Attaching information of measurements and quantities to scientific as well as engineering work, can help support various aspects, including but not limited to the comparison of work across different unit systems or rigorous analysis of a study. To that end, the authors evaluated existing ontologies in the field and developed a new improved ontology based on the evaluation.

The evaluation of the existing ontology is conducted by the authors with regards to the following criteria: completeness, quality of formal definitions, understandability,

extensibility and completeness of documentation. For this part of the evaluation, no further information apart from the assessed criteria and results are provided.

To evaluate the improved ontology, three demo applications were created, whereof one was evaluated in a structured walkthrough with domain experts. Similar to the evaluation of the existing ontologies, no information about the preparation or execution of the evaluation is provided.

S72 - Ontology development and evaluation for urinal tract infection [98]

The authors developed a biomedical ontology for the context of urinal tract infections using “UMLS” (Unified Medical Language System). Very little detail with regards to the expert evaluation conducted over the course of this study is provided within the publication.

S73 - Crowdsourced Knowledge Acquisition: Towards Hybrid-genre Workflows [99]

Within their work, the authors aimed to develop a hybrid crowdsourcing workflow combining a GWAP approach and a mechanised labour approach (i.e. paid-for crowdsourcing) to be able to collect data from the game players and subsequently validate them using HITs. To outline the feasibility of the approach, a literature study was conducted to reveal the advantages and disadvantages of both approaches.

The crowdsourcing workflow consists of a total of three steps. First, crowd workers were asked to judge if automatically extracted concepts of the ontology are potentially related. Then, “ClimateQuiz”, which is a Facebook game and thus is a GWAP, was employed to let players assign one out of ten relations to the concepts extracted in the first step. Finally, the results from the GWAP were validated in another crowdsourcing task, by showing a binary question (true/false) to the workers.

For both of the mechanised labour approaches (c.f. first and last step), the preparation steps are outlined on a high level. To prepare the data, either concepts are automatically extracted from an ontology or the results of the game are filtered. Further, some qualification questions and training questions are included in the task design to achieve higher quality results. No further information with regards to preparation activities is provided.

S82 - Ontology for cultural variations in interpersonal communication: Building on theoretical models and crowdsourced knowledge [100]

User generated content from social media platforms can yield insights into cultural variation in interpersonal communication and thus classifying the content could foster understanding and enable computer-based applications of the domain. One challenge in classifying the content also referred to as semantic tagging, is the lack of an ontology

to relate the content to concepts thereof. Therefore the authors addressed this gap by creating an ontology (“AMOn+”) and populating it with relevant instances from literature and DBPedia.

The created ontology was evaluated with regards to its fit-for-purpose. As a first step of the evaluation, a corpus of relevant texts was collected from various sources. Then parts of the corpus were semantically tagged by an expert to act as a gold standard, which in turn was then also validated by another expert. Using an automated tool and the created ontology, the same corpora of text was annotated and eventually, the results of the automated tool were compared to the experts gold standard.

S87 - A validated ontology for global software development [101]

Software development is a discipline often involving teams, which are distributed geographically across several locations, thus exhibiting different characteristics. To enable efficient study and application of this global software development approach, a common model capturing concepts and their relations is needed according to the authors. Thus, within their work, the authors developed an ontology capturing the advantages, challenges and concepts of global software development projects.

The construction of the ontology was attempted using input from a systematic literature mapping study and relating it to previous work in this field. Based on the results of a first evaluation through an expert survey, the initially created ontology was further improved. To assess the clarity, coherence, extensibility, focus on knowledge rather than the implementation of the ontology and the ability to be applied in various scenarios (i.e. generic), a second survey with experts was conducted. The survey used in the first evaluation was extended and a standardized scale for providing answers was employed.

S90 - PDON: Parkinson’s disease ontology for representation and modeling of the Parkinson’s disease knowledge domain [102]

The authors developed an ontology in the domain of Parkinson disease to be able to structure and semantically enrich this domain of research. The publication presents the construction process, the three steps evaluation process as well as two selected application scenarios of the ontology.

The evaluation steps encompass structural, functional and expert evaluation. As of interest by this thesis, the expert evaluation was targeted at collecting competency questions from domain experts and using them to query literature based on the ontology. The query results using the ontology were then compared to the results of a similar query on PubMed. Apart from the step of collecting competency questions from domain experts, no further information was provided with regards to the human centred evaluation of the ontology.

S93 - Toward multiviewpoint ontology construction by collaboration of non-experts and crowdsourcing: The case of the effect of diet on health [103]

It is claimed by the authors that expert-created ontologies often reflect only one viewpoint of a domain. For capturing a whole domain, however, it is required to integrate several different viewpoints of experts into one multi-viewpoint ontology. To address this issue, a methodology for creating such ontologies by non-experts using crowdsourcing techniques is proposed, which is in turn evaluated in the domain of diet and health. On a high-level view, the methodology involves a two-stage process, where the first step was to create a multi-viewpoint ontology by guiding domain non-expert through relevant scientific literature and the second step is to let crowd workers classify all the concepts identified earlier either as a true concept, a viewpoint concept or as an erroneous statement.

To evaluate the approach a series of three experiments was conducted. Each experiment focused on the second stage of the methodology, thus focusing on human centered evaluation of ontological constructs. The HITs all contained some form of qualification test and slightly different question design to classify the ontological constructs according to the approach elaborated earlier. Even though screenshots of the HITs and extensive description of the rationale behind each task design, little to no information with regards to the preparation of the evaluation can be found.

S94 - Predicting conceptnet path quality using crowdsourced assessments of naturalness [104]

Knowledge graphs are typically composed of a set of concepts, which are interrelated with each other, allowing for example recommender systems to reason about a given input. However, some path that connects two concepts might not be considered as natural or meaningful as another path connecting the same two concepts. Thus, the authors propose an approach where crowdsourcing is used to rank paths from ConceptNet and use these results to train a neural network for ranking paths according to the “naturalness”.

The crowdsourcing task was designed to show the workers two different paths connecting the same two concepts and letting them select the more natural path. In addition to the paths extracted from ConceptNet also control questions were introduced to ensure workers quality and trust. As the crowdsourcing task was used to collect training data for a neural network, the further evaluation experiments presented targeted the neural network and not the evaluation of the linked data.

S95 - Crowd-based ontology engineering with the uComp Protege plugin [5]

Wohlgenannt et al. approach human centred ontology verification by providing tool support for the ontology engineers. A plugin for the popular *Protégé*³ ontology editor

³<https://protege.stanford.edu/>

is created allowing five different ontology verification tasks to be crowdsourced. These tasks include assessing the domain relevance, verifying the correctness of subsumption relations or instances and specification of relation types. The tool also allows publishing the crowdsourcing tasks and collecting the answers directly to the editor. It is found that when employing crowd workers to solve these tasks similar results in terms of performance can be reached while reducing the time spent on the tasks when compared to a group of experts.

Seen from a perspective of the preparation steps involved for creating the tasks following information can be collected: (1) the task needs to be specified, (2) the parts of the ontology to be evaluated need to be selected, (3) additional information (i.e. task-specific information, generic information, additional information) should be filled in, and (4) finally the task is published on a crowdsourcing platform.

SB2 - Guess What?! Human Intelligence for Mining Linked Data [51]

The authors present a GWAP that allows to create and extend ontologies. The game is designed to be played by at least two players and involves two stages. First, the players are shown a conjunction of ontological classes for which they should come up with instances that fit this definition. Second, the collected instances are shown to all other players of a turn to evaluate whether the found instances are correct, incorrect or undecidable.

To prepare for the game first some seed concepts are extracted from a knowledge base (e.g. DBPedia), then further information (e.g. superclass, object properties) is retrieved from related RDF repositories. Using natural language processing complex constructs are broken down into smaller fragments and logical operators are extracted. Finally, the extracted data is used to generate the conjunctions shown in the first stage of the game. Generally, the game preparation workflow and its tools are outlined in detail in the paper, however, as the main focus of the game is on knowledge acquisition and not ontology evaluation little information with regards to preparing human centred ontology evaluation can be deduced.

SB3 - WhoKnows? - Evaluating Linked Data Heuristics with a Quiz that Cleans Up DBpedia [46]

Within their work, the authors develop a GWAP originally intended to evaluate property relevance ranking heuristics which turned out to had a side effect for detection of inconsistencies and doubtful facts in linked open data triples. The users play one out of three variants of a question-based game (i.e. one-to-one, one-to-n, hangman) and immediately get feedback about their answers. If users are in doubt about the feedback of their answers they were able to express their concerns via a dislike button.

To provide an initial set of data for the game, first, data was extracted from DBPedia and pre-processed using Hadoop. Further, the data was filtered, assigned to categories

and weights according to the difficulty of the possible resulting questions were calculated. Finally, the questions were generated employing a three-step process of (1) weighing the triples and properties, (2) assigning difficulties to the triples and (3) generating false answers.

The developed game has been deployed and evaluated as a Facebook game. One part of the evaluation was focused on the property ranking heuristics while the other part focused on deriving information from the dislike clicks of the players. Manual assessment of the dislike clicks revealed a considerable amount of triples to contain errors or inconsistencies. Thus this human centred approach can be used to detect these defects in triples or ontologies.

SB4 - Combining information extraction and human computing for crowdsourced knowledge acquisition [105]

The authors developed a workflow that first automatically extracts information from web pages and then validates and extends these statements by employing a GWAP approach. More specifically the HITs focus on collecting, confirming or assessing instances of binary relations between the extracted concepts.

Once the information extraction component has provided concepts and their relations the human computation engine needs to prepare the HITs. In that regard, a two-stage approach is used where first, the question to be asked to the player is constructed and subsequently, candidate answers are generated. The question generation process is focused on questions of interests of a user by relating questions to interests and using salient entities of movies and books which are derived from other knowledge sources (e.g. Wikipedia). To generate possible candidate answers, first, a ranking of candidates is generated and then a diversification process is applied. Diversification, in this context, prunes candidates which are rather similar to each other to ensure the answers represent a diverse spectrum and the final answer is distinctive enough. The process for ranking and diversifying the candidates uses multiple statistical language models from the domain of information retrieval. Finally, the questions (i.e. entities), candidates (i.e. relations) and contextual information are filled into predefined question templates.

SB5 - ADO: A disease ontology representing the domain knowledge specific to Alzheimer's disease [106]

This study presents a biomedical ontology for Alzheimer's disease. The ontology was constructed by extracting information from various sources (e.g. scientific publications) and also connections to other biomedical ontologies.

The evaluation of the ontology was approached on three levels: structural, functional and based on competency questions. The evaluation level with regards to the competency questions can be considered human centered as competency questions specified by experts were used to assess the semantic capabilities of the ontology. Queries answering the

competency questions were executed with support and without the support of the proposed ontology to draw a comparison.

Appendix B: SSI Invitation EMail

Dear <Interview-Participant>!

I am Klemens, a computer science master's student at TU Wien, and currently working on my master thesis "A process and tool support for human-centred ontology validation". The first part of the research of my thesis is concerned with *understanding the process of preparing and conducting human-centred ontology evaluations*. To address this aspect I am conducting expert interviews and a follow-up focus group to collect information about such processes. For more information about the current phase of my thesis please consider the attached one-page document.

As you have been referred to me as knowledgeable in this area, I would kindly like to request you to join the interviews and the focus group. The **individual interview time slots** are planned in the **timeframe from 16.03.2022 - 22.02.2022 and should last about 1 hour**. The **focus group** workshop is planned to take place **between 25.03.2022 and 29.03.2022 and is planned to last about 1.5 to 2 hours**.

If you are willing to participate in my study (1) please respond to this email, (2) book an **interview time** slot on (note that only one time can be selected and the interview will take place that date and time; the place is to be announced - probably at TU Wien):

- <https://calendly.com/klemens-kaeszner/https-lettucemeet-com-l-jjdj8>

(3) and provide **your availability for the follow-up focus** group on:

- <https://doodle.com/meeting/participate/id/Le3jrkAd>

If none of the proposed time slots works for you please let me know!

I am looking forward to your response!
Kind regards,
Klemens Käsznar

Appendix C: SSI One Pager

A Process and Tool Support for Human-Centred Ontology Validation

Semi-Structured Interviews (SSIs) and Focus Group: Background

Thesis Overview

Ontologies in computer science allow modelling and capturing domain information explicitly, thus enabling the emergence of modern information systems that rely on rich, high-quality information. These ontologies are typically either built by human domain experts or computer algorithms, thus errors cannot be avoided. However, identifying and correcting such mistakes in ontologies is crucial for correct functioning ontology-based systems.

Due to the machine-readable formats and roots in formal logics of ontologies, certain errors, such as logical inconsistencies, can be easily detected using an automated process/tool, while other errors need human knowledge to be detected. In the context of my thesis, the latter type of evaluation referred to as human-centred ontology evaluation, shall be supported by using human computation techniques (e.g. crowdsourcing).

The overall contributions of the thesis include (1) a *process model for human-centred ontology evaluation*, (2) a *reference architecture and requirements specification* of an end-to-end process support platform for such evaluations and (3) a *prototypical implementation of such a platform* to evaluate it. Currently, support for understanding the processes is needed to form the foundation for further contributions.

Process Understandings: SSIs & Focus Group

As a first step towards building a generic process model for human-centred ontology evaluation, a systematic literature review (SLR) was conducted to identify important activities and supporting tools. An initial set of activities and tools, grouped by the three processes phases preparation, execution and follow-up, of these evaluations, has been identified using the results of the review. However, as most of the *assessed literature* typically does not focus on providing evaluation aspects in great detail, *further information based on expert knowledge collected through semi-structured interviews and a follow-up focus group is needed in that regard*.

For the *semi-structured interviews*, each participant will be asked to elaborate on their experience within the domain of human-centred ontology evaluation. The core questions of the interviews will be *about collecting information on the activities and steps involved when preparing, executing and concluding human-centred ontology evaluations*. Further, if the experts are aware of tools supporting the process, information with regards to them is highly appreciated as well.

Using the set of activities identified by the SLR and each of the experts' information, a unified view of the process shall be created. Thus in a follow-up focus group with all interview participants, *all identified activities will be discussed in a group and a final process model shall be created* by defining an ordering of the relevant steps.

Appendix D: SSI Interview Guide

Interview Guide SSI: Eliciting the process of human-centred ontology evaluation

P ... Priority of instruction / question

- 1 ... Must be included
- 2 ... If time allows include the question or the question is needed for probing
- 3 ... Non-essential question, only should be included if time allows and the participant touches on the topic; can be left out

Sentences in “...” indicate scripted parts of the interview while bullet points and enumerations are used to outline the content to be discussed.

Instructions for the interviewer are given *in italic font*.

Introductory Questions (~ 10 min)

#	P	Instruction / Question
1	1	Greet participants and thank them for taking their time!
2	1	“To allow detailed analysis and to make sure no details are left out during analysis, I would like to record the interview, is this okay for you?”
3	1	<i>Turn on recording</i> “Ok, <Name> thanks for letting me record the interview. Now that everything is set up, before we actually start with the interview part, I will just briefly introduce my thesis and outline today's agenda.”
4	1	Thesis introduction: <ol style="list-style-type: none"> 1. Context: ontology evaluation & human computation 2. Two types of errors → automatically detectable VS. human knowledge 3. Focus on human-centred evaluation to be approaches by human computation 4. Not yet well understood and tool support not existing 5. Thus aims / contributions <ol style="list-style-type: none"> a. Understanding the process of such evaluations by designing a process model b. Proposing a reference architecture and requirements specification c. Implementing a prototype tool / backend that support ontology engineers
5	1	Introduce current state of thesis (keep rather short!) <ol style="list-style-type: none"> 1. In the beginning, trying to understand the process 2. SLR with about 45 papers conducted 3. Results are shallow do not go into detail 4. Need for expert view on the activities 5. Thus the interview today 6. And once all experts are interviewed, a follow-up focus group will be

#	P	Instruction / Question
		used to design the final process model

Participant Involvement Questions (~10 min)

#	P	Instruction / Question
6	1	"Can you please provide a short summary on your background with ontologies, ontology engineering etc. in computer science?"
7	1	<i>If not already mentioned then</i> "To get into more detail on your experience with regards to today's interview, what is your experience with ontology evaluation?"
8	2	<i>Additional probes for information about ontology evaluation?</i> "What ontologies did you evaluate? Can you elaborate on their characteristics?" "What aspects of the ontology did you evaluate?"
9	3	"When comparing all steps of ontology engineering, how would you rate the importance of ontology evaluation on a scale from 1 to 5."
10	1	<i>If not already mentioned then</i> "How or in which context have you already encountered the need for human-centred ontology evaluation?"
11	3	<i>If not already mentioned and the participants mentions crowdsourcing tools</i> "Can you elaborate into more detail (e.g. what ontology was evaluated, how many participants, results) on your experience with conducting ontology evaluations on crowdsourcing platforms?"

Process Elicitation / Core Questions (~ 30 min)

#	P	Instruction / Question
12	1	"To guide the following questions and to be in line with the work already performed, I would like to propose that human-centred ontology evaluation on a high-level can be divided into the following-phases: (1) preparation, (2) execution and (3) follow-up."
13	1	"Further for the following questions / remaining part of the interview consider you have built an ontology or you found an ontology you want to use and there is the need to verify certain aspects of the ontology. Thus you decide to conduct an ontology evaluation where humans should judge that certain aspect. In the best case this reflects one of your earlier projects / experiences with human-centred ontology evaluation. Is there a concrete project in mind where you have done this?"

#	P	Instruction / Question
14	1	"Starting with the preparation phase of the evaluation, can you list specific activities / steps that you would perform before you show the ontology and the task to your ontology evaluators?" <i>Collect all the mentioned steps in a list on paper the participant can see.</i>
15	1	"Now suppose that all of the activities associated with the preparation phase are completed, what are the steps that can be expected during the execution of the evaluation task with humans?" <i>Collect all the mentioned steps in a list on paper the participant can see.</i>
16	1	"Finally all the data is collected from the evaluators. What are the steps you are performing to conclude the ontology evaluation?" <i>Collect all the mentioned steps in a list on paper the participant can see.</i>
17	2	<i>If a step is unclear or more clarification is needed</i> "Can you please elaborate step <X> into more detail?"
18	2	<i>If the participant has some reference project</i> "Considering your previous experience with human-centred ontology evaluations, is there anything you would do differently now?"
19	2	<i>If time allows</i> <i>Use second sheet of paper to draw the process together</i> "Now that I have gathered some more understanding of the process I would like to draw a high-level process model with you. For that I have collected a list of steps you mentioned and now it would be great if we could order them together."
20	3	"Did you make use or implement any concrete tools to support the steps you performed?"
21	3	<i>If question answered with yes</i> "Can you name the tools you used?" "Can you elaborate on the implemented tool? Is it publicly accessible?"
22	3	"Could you assign the mentioned tools to the steps / activities you proposed earlier?"

Wrap Up / Review of Questions (~5min)

#	P	Instruction / Question
23	1	"Now this brings me to the last question of my interview: As an outcome of my thesis it is expected to implement a process model and tool that supports ontology engineers like you when preparing for such evaluations. How do you think your future work would benefit from it?"
24	1	Thank the participants for their time and say that you are looking forward to the focus group where a final process model is agreed on among you and other experts and the literature review.

Appendix E: Focus Group Discussion Guide

Discussion Guide Focus-Group: Align the process activities of human-centred ontology evaluation

P ... Priority of instruction / question

- 1 ... Must be included
- 2 ... If time allows include the question or the question is needed for probing
- 3 ... Non-essential question, only should be included if time allows and the participants touch on the topic; can be left out

Sentences in “...” indicate scripted parts of the discussion while bullet points and enumerations are used to outline the content to be discussed.

Instructions for the interviewer are given *in italic font*.

Introductory Questions (~ 15 min)

#	P	Instruction / Question
1	1	Greet participants and thank them for taking their time!
2	1	“To allow detailed analysis and to make sure no details are left out during analysis, I would like to record the discussion group, is this okay for you?”
3	1	<i>Turn on recording</i> “Ok, thanks for letting me record the discussion group.. Now that everything is set up, before we actually start with the core part of the focus group, I will just briefly introduce the goals and today's agenda using some slides.” <i>Present slide-deck</i>
4	1	<i>Show Slide 2</i> Outline the goal of the session <ul style="list-style-type: none"> • Two data sources: <ul style="list-style-type: none"> ○ Process activities collected from SLR <ul style="list-style-type: none"> ■ Not shown to any participants before the interview to avoid bias ○ Process activities elicited by the experts • Merged set of activities diverge from each other • Ordering of the activities is unclear • Clear goal: collect inputs <i>for generic process model</i> of human-centred ontology verification <ul style="list-style-type: none"> ○ Generic process model: includes many activities which should act as a baseline. <ul style="list-style-type: none"> ■ Users should be able to apply it to several different verification tasks ■ Steps can be left out / be considered optional etc.

#	P	Instruction / Question
5	1	<p><i>Show Slide 3</i></p> <p>Outline the structure of the discussion session</p> <ul style="list-style-type: none"> ● Expected duration at most two hours ● Structured in three main blocks as already familiar from the interview: <ul style="list-style-type: none"> ○ Preparation activities ○ Execution activities ○ Follow-up activities ● For each of the phase a digital whiteboard will be used <ul style="list-style-type: none"> ○ Sketch the ordering of the activities together ○ Be able to remember what was discussed earlier ● It should be an interactive active discussion <ul style="list-style-type: none"> ○ Any idea / feedback etc is welcomed ○ Just speak up ○ If any activity seems to be superfluous or etc. also just interrupt the process and mention it

Block: Preparation Activities (~ 45 min)

#	P	Instruction / Question
6	1	<p><i>Open whiteboard for preparation</i></p> <p>Explain the structure of the whiteboard:</p> <ul style="list-style-type: none"> ● Consists of the activities sent out with the preparation email ● Based on the data collected from the interview, a heuristic approach was selected to define an ordering <ul style="list-style-type: none"> ○ Based on common activities ○ Other activities not commonly mentioned by all experts are inserted based on their respective position to the common activity ○ Does not include activities from the SLR because no ordering could be defined → in the box above ● Explain the process groups <ul style="list-style-type: none"> ○ E.g. Task design ○ Includes other activities to be expected during this activity ○ Mentioned during the experts ○ Provide a way to align the granularity of the tasks ● Activities are also expected to happen in parallel
7	1	<p>Explain what the next brainstorming block is about</p> <ul style="list-style-type: none"> ● For certain activities during the interviews a ordering based on the heuristics could not be derived or experts considered them to be part of a activity group or not ● Thus following questions are about the placement of certain activities or about the relevance of the tasks <ul style="list-style-type: none"> ○ Questions are rather short and concise as they are already based on the data from the SSIs
8	1	<p>"How does "Prepare instructions" differ from "Prepare training questions". Can these two activities be merged?" (Colored: RED)</p>

#	P	Instruction / Question
8	1	"Where should the activity "Specify evaluation environment" be placed? The interviews suggest the following placements: before, after or inside "Task Design"." (Colored: ORANGE)
9	1	"Is the task "Extract relevant ontology elements" part of the preparation phase or part of the execution phase?" (Colored: YELLOW) <ul style="list-style-type: none"> Explain that some of the experts mentioned this activity also to be referred to as scoping the ontology or as splitting the ontology
10	1	<i>If experts consider "Extract relevant ontology elements" to be part of the preparation phase?</i> "Where should the activity "Extract relevant ontological elements" be placed? The interviews suggest a placement either inside "Task Design" or before "Task Design"." (Colored: YELLOW)
11	1	"Is the "Prepare self assessment" test needed? If yes, where should it be placed? The interviews suggest a placement either inside "Task Design" or parallel to "Task Design"." (Colored: GREEN)
12	1	"Where should the activity "Find a crowd" be placed? The interviews suggest the following placements: before, after or inside "Task Design"." (Colored: BLUE)
13	1	"Where should the activity "Seed in control questions" be placed? The interviews suggest a placement either after "Task Design" or parallel to it." (Colored: PURPLE)
14	1	"Should "Prepare feedback form" be part of the "Task Design" or should it be placed after "Task Design"?" (Colored: PINK)
15	1	<i>Ask participants to now consider the activities in the box above "Move to other phase?"</i> "Are the activities "HIT: Populate Template" and "Submit to crowdsourcing platform" part of the preparation or part of the execution?"
16	1	<i>Ask participants to now consider the activities in the box above "Activities from SLR"</i> "Should the following activities "Create Survey", "Prepare presentation", "Collect competency questions" and "Translate to natural language" be included to the process model? If yes for any of them, where to place them."
17	1	"Can any ordering for the activities inside "Task Design" be defined?"
18	2	<i>If time allows and not mentioned earlier.</i> "If there are any activities you suggest removing or moving to another position in the process model or any other comment with regards to the process model, please write them down on a sheet of paper" <i>Let participants write down feedback.</i> Ask each individual participant for her feedback / points she wrote down.

#	P	Instruction / Question
19	1	Give the participants a moment to review the inputs for the process model.

Block: Execution Activities (~ 30 min)

#	P	Instruction / Question
20	1	<p><i>Open whiteboard for execution</i></p> <p>Explain the structure of the whiteboard:</p> <ul style="list-style-type: none"> • Consists of the activities sent out with the preparation email • In contrast to preparation activities far less activities were mentioned <ul style="list-style-type: none"> ◦ No heuristic for ordering the process activities applied as no activity was mentioned by all experts • 10 activities
21	1	<p>"Is the task "Collect results" part of the execution phase or part of the follow-up phase?"</p> <ul style="list-style-type: none"> • Mention that three experts said execution while one said follow-up
22	1	<p>"Is the task "Aggregate results" part of the execution phase or part of the follow-up phase?"</p> <ul style="list-style-type: none"> • Mention that three experts said execution while one said follow-up
23	1	<p>"Now an ordering of the execution activities needs to be defined and to do that I will hand out small sheets of paper. The ordering is going to be structured in several rounds of two steps:</p> <ol style="list-style-type: none"> 1. Each one of you writes down the first (second, third ...) activity. 2. Then each will read out her activity and we will discuss it." <p><i>Do this for each of the ten activities</i></p>
24	2	<p><i>If time allows and not mentioned earlier.</i></p> <p>"If there are any activities you suggest removing or moving to another position in the process model or any other comment with regards to the process model, please write them down on a sheet of paper"</p> <p><i>Let participants write down feedback.</i></p> <p>Ask each individual participant for her feedback / points she wrote down.</p>
25	1	Give the participants a moment to review the inputs for the process model.

Block: Follow-Up Activities (~ 25 min)

#	P	Instruction / Question
26	1	<p><i>Open whiteboard for follow-up</i> Explain the structure of the whiteboard:</p> <ul style="list-style-type: none"> • Consists of the activities sent out with the preparation email • Same as with the execution activities, there is no ordering identified as no common activities are observed • 8 activities
27	1	<p>"Is it beneficial for the final process to create an activity group "Data analysis"? If yes, which of the follow-up activities should be added to it?"</p> <ul style="list-style-type: none"> • Mention that the granularity of the activities mentioned during the interviews were different and therefore this pattern could potentially be identified
28	1	<p>"Now an ordering of the follow-up activities needs to be defined and to do that I will hand out small sheets of paper. The ordering is going to be structured in several rounds of two steps:</p> <ol style="list-style-type: none"> 1. Each one of you writes down the first (second, third ...) activity. 2. Then each will read out her activity and we will discuss it." <p><i>Do this for each of the ten activities</i></p>
29	2	<p><i>If experts agreed on defining data analysis as a group</i> "Is there any possibility to define an ordering for the activities inside data analysis?"</p>
30	1	Give the participants a moment to review the inputs for the process model.

Wrap Up (~ 5min)

#	P	Instruction / Question
31	1	<ul style="list-style-type: none"> • Explain that the three phases will be merged together to one process model and translated as an UML activity diagram • The UML activity diagram will then enable the implementation of a process support platform
32	2	<p><i>If time allows</i> "To guide the implementation also requirements will be identified, to that end I would like to conclude today's session with collecting inputs with regards to a support tool?" (like. REST interface etc)</p>
33	1	Thank the participants for their time!

Appendix F: Individual Process Models

Please turn over to the next page.

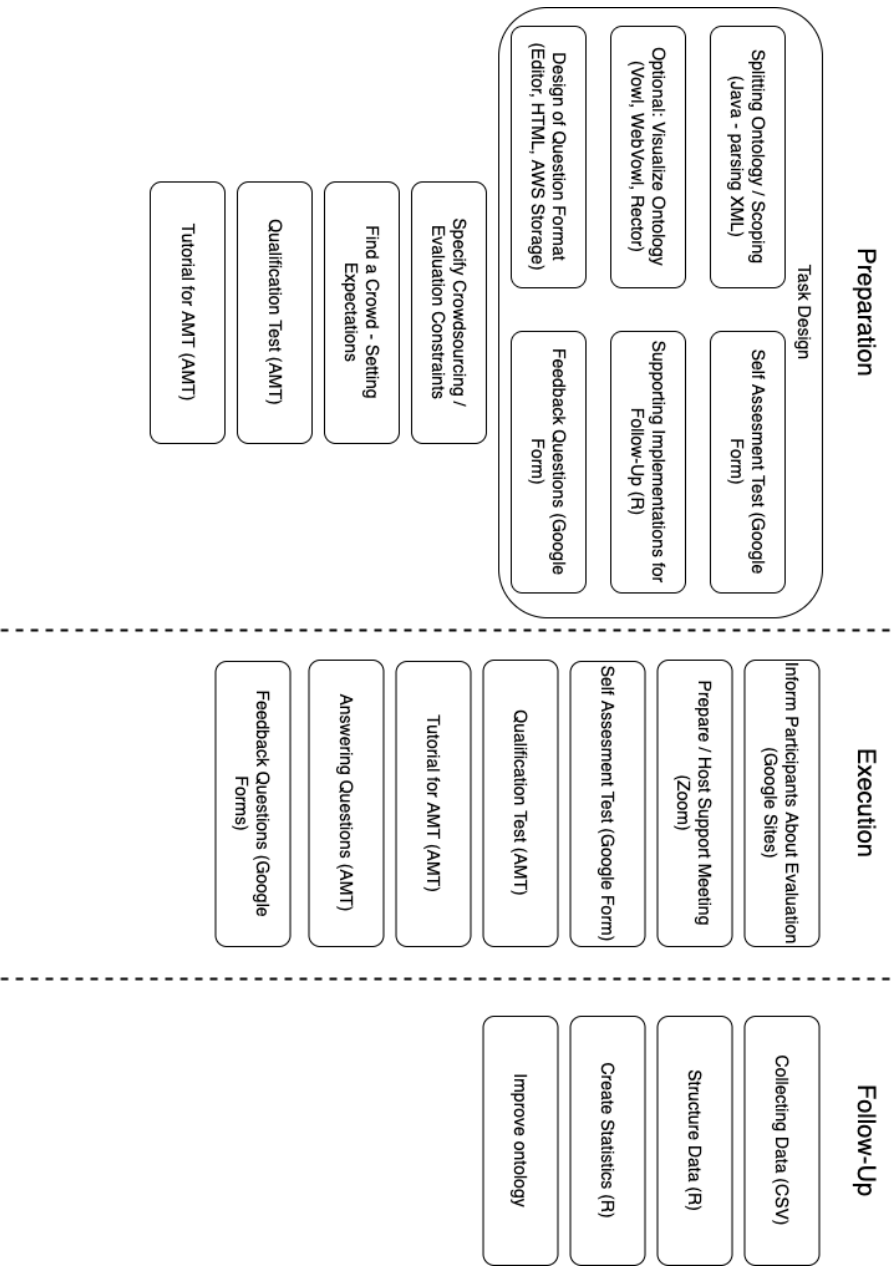


Figure 1: Individual Process Model of Expert 1

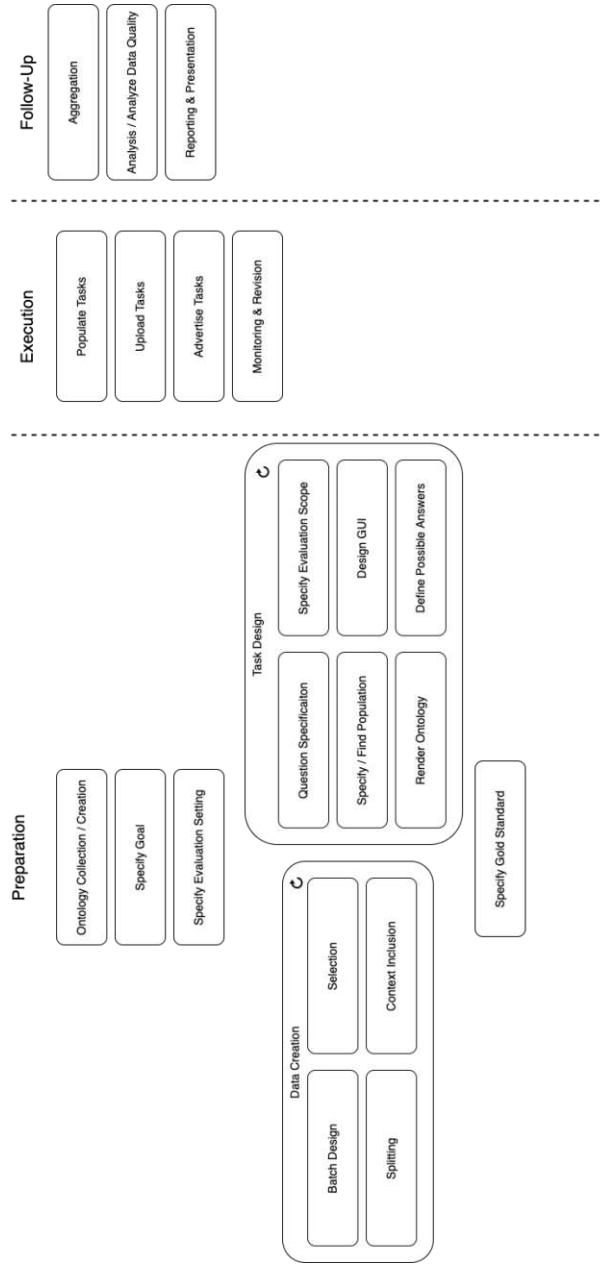


Figure 2: Individual Process Model of Expert 2

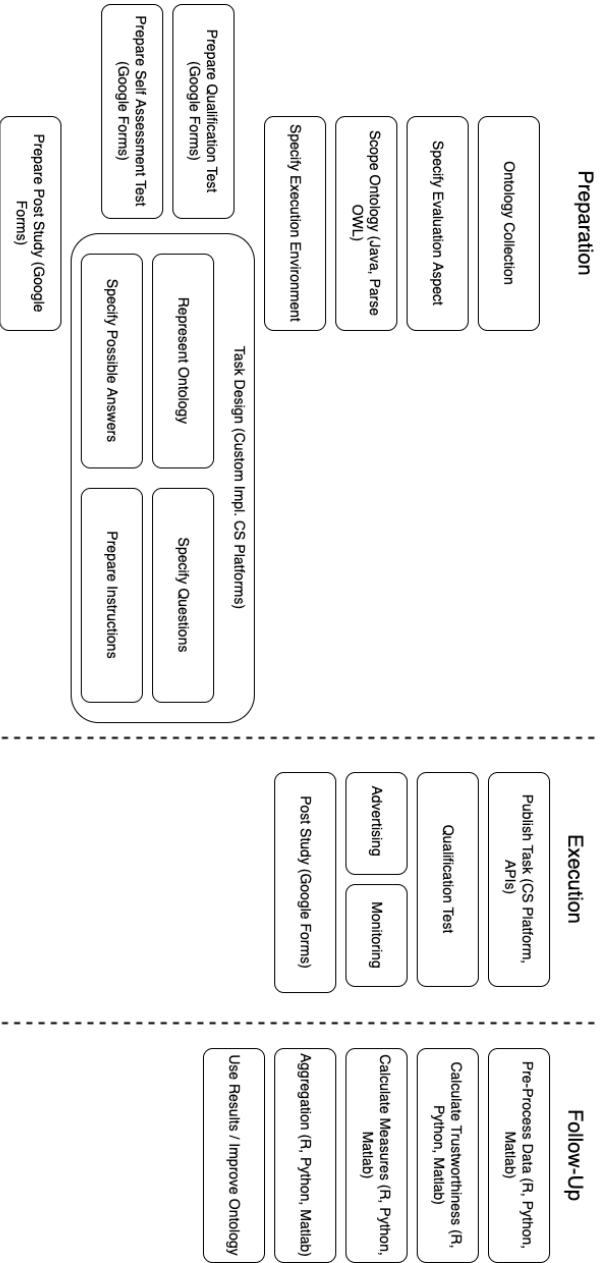


Figure 3: Individual Process Model of Expert 3

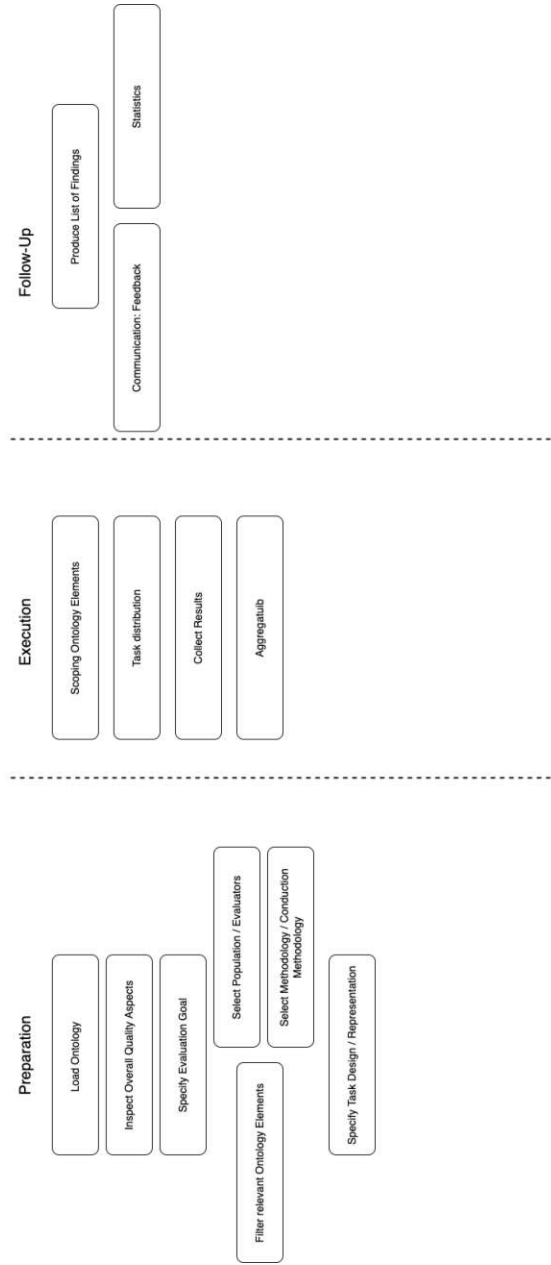


Figure 4: Individual Process Model of Expert 4