Human-Centric Ontology Evaluation: Process and Tool Support

Stefani Tsaneva[1,2(✉)], Klemens Käsznar[2], and Marta Sabou[1,2]

[1] Vienna University of Economics and Business, Vienna, Austria
{stefani.tsaneva,marta.sabou}@wu.ac.at
[2] Vienna University of Technology, Vienna, Austria
klemens.kaesznar@student.tuwien.ac.at

Abstract. As ontologies enable advanced intelligent applications, ensuring their correctness is crucial. While many quality aspects can be automatically verified, some evaluation tasks can only be solved with human intervention. Nevertheless, there is currently no generic methodology or tool support available for human-centric evaluation of ontologies. This leads to high efforts for organizing such evaluation campaigns as ontology engineers are neither guided in terms of the activities to follow nor do they benefit from tool support. To address this gap, we propose HERO - a **H**uman-Centric Ontology **E**valuation P**RO**cess, capturing all preparation, execution and follow-up activities involved in such verifications. We further propose a reference architecture of a support platform, based on HERO. We perform a case-study-centric evaluation of HERO and its reference architecture and observe a decrease in the manual effort up to 88% when ontology engineers are supported by the proposed artifacts versus a manual preparation of the evaluation.

Keywords: Ontology evaluation · Process model · Human-in-the-loop

1 Introduction

Semantic resources such as ontologies, taxonomies and knowledge graphs are increasingly used to enable an ever-growing array of intelligent systems. This raises the need of ensuring that these resources are of high quality because incorrectly represented information or controversial concepts modeled from a single viewpoint can lead to invalid application outputs and biased systems.

While several ontology issues can be automatically detected, such as logical inconsistencies and hierarchy cycles, some aspects require a human-centric evaluation. Examples are the identification of concepts not compliant with how humans think and the detection of incorrectly represented facts or controversial statements modeled from a single viewpoint [2,8]. For example, Poveda-Villalon et al. [12] identified 41 frequent ontology pitfalls, out of which 33 can be automatically detected while the remaining 8 require human judgment to be identified, e.g., P09 - Missing domain information, P14 - Misusing "owl:allValuesFrom", P15 - Using "some not" in place of "not some", or P16 - Using a primitive class in

© The Author(s) 2022
O. Corcho et al. (Eds.): EKAW 2022, LNAI 13514, pp. 182–197, 2022.
https://doi.org/10.1007/978-3-031-17105-5_14

place of a defined one. In particular, P14 covers modeling mistakes related to universal (∀) and existential (∃) quantifiers stemming from the incorrect assumptions that either (1) missing information is incorrect or that (2) the universal restriction implies also the existential restriction. An example from the Pizza ontology would be a pizza *Margherita* with the two toppings *Tomato* and *Mozzarella*. Modeling these two toppings using either (1) only existential restrictions or (2) only universal restrictions, would lead to (1) pizza instances having tomato and mozzarella topping *and other toppings* or (2) pizza instances without any toppings being classified as *Margherita* pizzas.

There is a large body of literature in which ontology evaluation tasks, such as P14 above, are evaluated by humans. Indeed, a recent Systematic Mapping Study (SMS) in the field of human-centric evaluation of semantic resources identified 100 papers published on this topic in the last decade (2010-2020) [16]. A large portion of these papers (over 40%) relies on Human Computation and Crowdsourcing (HC&C) techniques [7,15], for example, to evaluate large biomedical ontologies [9] or to ensure the quality of Linked Data as a collaborative effort between experts and the crowd [1]. In [19], we applied HC&C for supporting the evaluation of P14 through Human Intelligence Tasks (HITs) such as those shown in Fig. 1 where evaluators see a concrete entity (1, a menu item for a pizza), an axiom that models the class to which the entity belongs (2) as well as four options of (potential) errors in the axiom (3).

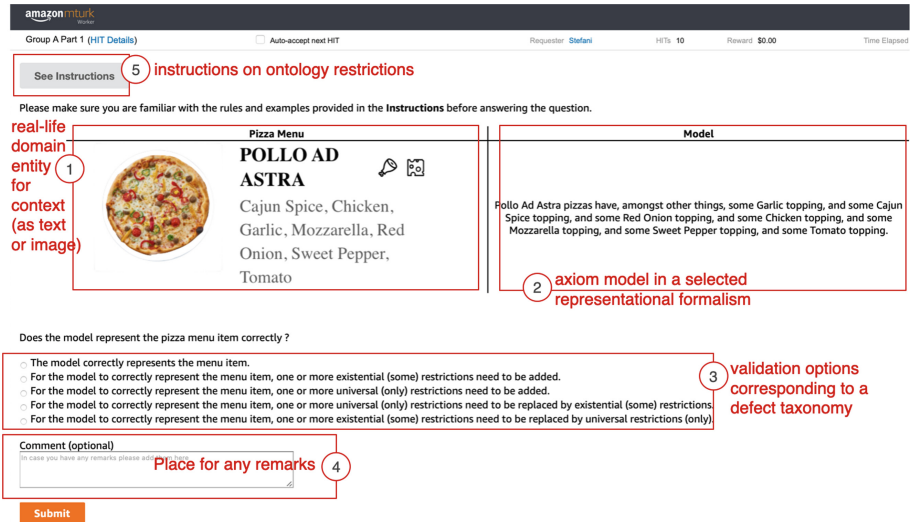

Fig. 1. HIT interface for verifying the correct use of quantifiers (from [19]).

An analysis of the 100 papers from the SMS [16] revealed two major gaps. First, there is limited understanding of the followed process by ontology engineers performing human-centric ontology evaluations: not even half of the papers

outline their methodology explicitly. Some of the available methodologies are tailored to one particular evaluation aspect [11,21] or focus on other conceptual structures than ontologies [3,5,17]. Second, a lack of appropriate tool support dovetails the lack of an accepted process model: indeed, less than 15% of the 100 papers mention the use of tools or libraries when preparing the evaluation.

This lack of a generalized methodology and tool support, considerably hampers the development of human-centric ontology evaluation campaigns, with each ontology engineer "reinventing the wheel" when planning such evaluations. In our own work [19], in order to prepare a human-centric ontology evaluation campaign, we could not rely on any pre-existing process or tool support and spent approximately 195 h for its realization.

In this paper, we address ontology engineers that similarly wish to prepare a human-centric ontology evaluation and aim to reduce the effort and time they need to spend on this process. To that end, we adopt a Design Science methodology [4], that leads to the following contributions in terms of concrete information artifacts and their evaluation:

- *A process model* capturing the main stages of human-centric ontology evaluation (HERO - a **H**uman-Centric Ontology **E**valuation P**RO**cess) which aims to support ontology engineers in the preparation, execution and follow-up activities of such evaluations. We focus on evaluations performed with HC&C techniques as these are currently the most frequently used. HERO was derived based on a literature review, expert interviews and a focus group.
- *A reference architecture* which supports HERO and consists of a core, which implements the general activities (such as loading an ontology), while task-specific evaluation implementations are captured as plugins and can be further extended to cater to the individual needs of evaluation tasks.
- The *evaluation of HERO and its reference architecture* by replicating the use case in [19] shows that with the support of the developed framework manual effort for preparing a human-centric ontology evaluation campaign could decrease from 30% to 88%, depending on the level of artifact reuse.

We continue with a discussion of our methodology (Sect. 2) and its main results in terms of the HERO process model (Sect. 3), a corresponding reference architecture (Sect. 4) and the use-case-based evaluation thereof (Sect. 5). Lastly, we present related work (Sect. 6) and discuss concluding remarks in Sect. 7.

2 Methodology

As our goal is to establish two information artifacts (a process model and a supporting reference architecture), we apply the *Design Science methodology for information systems research* [4] realized in three steps as illustrated in Fig. 2: Step 1 and Step 2 cover the *development phase* of the two artifacts, while Step 3 represents the *evaluation phase* based on a concrete case study. In Step 1, we incorporate knowledge from existing literature into the design of the artifacts (*rigor cycle*) while also involving key stakeholders in need of such artifacts (*relevance cycle*). The details of each methodological step are described next:

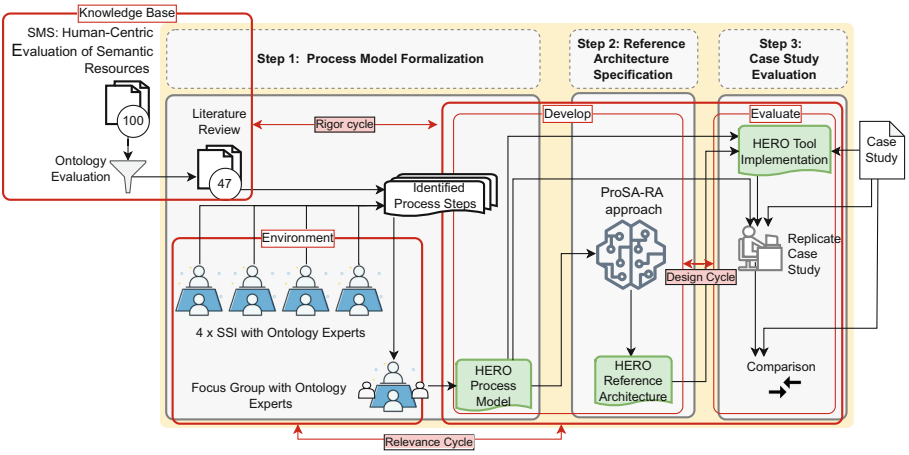

Fig. 2. Design-Science-based methodology.

Step 1: Process Model Formalization relies on three diverse methods for deriving the HERO process model. First, we review existing literature discussing ontology evaluation relying on human involvement. To that end, we leverage ongoing work on a *Systematic Mapping Study (SMS)* about human-centric evaluation of semantic resources [16]. From the set of 100 papers collected by the SMS, we identified 47 papers that discuss *ontology* evaluation and review them to identify typical activities followed when performing human-centric ontology evaluation and the tools used in that process. As a result, we collect a set of activities to be included in HERO and group them into three stages: preparation, execution and follow-up. We structure the data collected from these 47 papers in a Knowledge Graph, published at our git repository[1], making it available to other researchers.

Second, we perform a set of *semi-structured interviews (SSI) with experts in Ontology Engineering* to uncover missing aspects not described in the papers from the literature, discuss order of activities and required tools. Interviewees were selected from the Vienna University of Technology and included a senior researcher, a Ph.D. student, a graduate student, and a master's student, each conducting work in the area of human-centric ontology evaluation. During the interviews, a set of activities, part of human-centric ontology evaluations, are identified from the perspective of each expert as well as tools they used when conducting past evaluations. The interviews aim at strengthening and supporting the findings from the literature corpus and both approaches are designed independently to ensure that the experts are not biased and their personal views on the process are captured. More details on the SSI and a comparison of the steps found in literature vs. those identified during the SSI can be found in [6].

Third, we conduct a *focus group with the experts* that participated in the interviews to combine the literature analysis results with the insights gathered

[1] github.com/k-klemens/hc-ov-process-models/tree/main/slr.

from the interviews. During the discussions open aspects are clarified, activities are ordered and the final process model is agreed on (Sect. 3).

Step 2: Reference Architecture Specification. We follow the ProSA-RA [10] approach for establishing a reference architecture and rely on a Microkernel architecture, which features (1) a core, capturing the general logic and (2) plugins, which extend the platform functionalities [14], as detailed in Sect. 4.

Step 3: Case Study Evaluation. We focus on evaluating how HERO and the corresponding reference architecture can support ontology engineers when conducting human-centric ontology verifications and follow the methodology proposed by Wohlin et al. [22]. We first implement a platform prototype based on HERO and the reference architecture to support as many activities in the use case described in [19] as possible. Subsequently, we compare the effort required to prepare the evaluation campaign with HERO and tool support against the effort of manually creating the evaluation in the original use case (Sect. 5).

3 HERO - A Human-Centric Ontology Evaluation Process

HERO is a process model for human-centric ontology evaluation, targeted toward micro-tasking environments such as crowdsourcing platforms and focusing on batch-style evaluations. At a high-level the process and its activities can be structured into the stages of *preparation*, *execution* and *follow-up*, as detailed in the next sections and exemplified by the use case from [19] introduced in Sect. 1. The *preparation* stage consists of the design and definition of the evaluation, while during the execution stage the evaluation is conducted, followed by the *follow-up* stage where the evaluation data is collected and analyzed. Note that HERO aims at being broadly applicable and as a result includes activities that might not be needed in every human-centric ontology evaluation.

3.1 Preparation Stage

The activities part of the HERO preparation stage are visualized in Fig. 3. 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. Parallel activities are situated between two vertical lines, while connected activities are placed into activity groups (e.g., "Task design").

As a starting point, the *ontology to be verified needs to be loaded* (1 in Fig. 3) and to get an overview of the ontology, *standard metrics and quality aspects should be inspected* (2; e.g., in Protégé, among other things, the number of axioms, classes and data properties can be explored). A crucial preparation activity is the *specification of the aspect for evaluation* (3) and the overall goal of the verification. In the specified use case, the correct usage of ontology restrictions is the aspect to be verified. *Specifying the evaluation environment* (4) is the

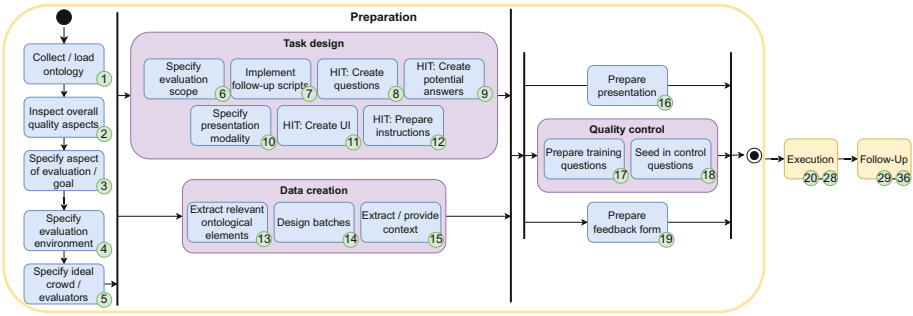

Fig. 3. HERO preparation stage.

next activity which refers to deciding on a crowdsourcing platform (e.g., in [19] Amazon Mechanical Turk (mTurk, mturk.com) was used) or another medium (e.g., games with a purpose, pen&paper, custom interface) that enables conducting the verification. In addition, the *ideal crowd's characteristics* (5; e.g., demographics, expertise in the domain) should be specified at an early stage of the process as these could have an impact on the verification task design. Nevertheless, special consideration should be taken with regard to avoiding creating a potential bias through crowd selection. In the evaluation performed in [19] we decided on an internal student crowd rather than a layman crowd since general modeling understanding of the evaluators and a more controlled environment were prerequisites. We further asked the students to complete a self-assessment to gain a better overview of their background in several areas (e.g., English skills, modeling experience). Evaluators' skills were also tested by implementing a qualification test, which offered an objective assessment of the evaluators' prior knowledge of the quantifiers. Further details on these assets can be found in [18].

Next, the verification *task design* follows. Several activities are to be expected, which have no particular order, since the task design process is iterative. *Specifying the evaluation scope* (6) can include specifying what subset of the ontology to show to the evaluators. In [19] all restrictions on a single relation are grouped together forming ontology restriction axioms that fully describe the specific relation and can be evaluated independently from the rest of the axioms. However, to verify a subclass relation it might be sufficient to only present the ontology triple, while for judging the relevance of a concept, more ontological elements might be needed to ensure a correct judgment.

As the task design might impose the structure of the final data and implications for analysis arise, *follow-up scripts* (7) for data processing can be implemented. In [19] analysis scripts are implemented in R (r-project.org) and tested in the preparation stage to avoid unexpected issues at the final process stage. The *specification of presentation modality* (10) implies deciding on the representation of ontology elements that evaluators will see. In the specified use case, we considered 3 representations- two plain text axiom translations, proposed by the authors of [13] and [20] as alternatives to showing OWL to novice ontology engi-

neers, and the graphical representation VOWL. Next, the Human Intelligence Tasks (HITs) are designed which includes *creating questions* (8), such that the required information can be collected, *creating potential answers* (9) (e.g., in [19] we created answer possibilities based on a predefined defect catalog), *creating the user interface* (11) and *preparing HIT instructions* (12).

In parallel, the ontology should be prepared for the evaluation (*data preparation*) by *extracting relevant ontological elements* (13), which in [19] we accomplished using Apache Jena (jena.apache.org), *designing batches* (14) so that relevant tasks are grouped together, and *extracting context* (15) to be presented to the evaluators (e.g., in [19] pizza menu items are manually created) to be provided to the evaluator.

In some evaluations it is beneficial to *create a presentation* (16) to inform the evaluators what the verification is about and what their assignment is (e.g., expert evaluation). In [19] we prepared a presentation for the student crowd, which included the main goals of the evaluation, instructions and tips on using mTurk, and organizational aspects.

Another important activity group is the *quality control*. One approach is to *prepare training questions* (17) to be completed by the evaluators prior to the actual verification, similar to the qualification tests which can ensure that the crowd acquires a particular skill. For the evaluation performed in [19] several tutorial questions were available to ensure the students are familiar with the mTurk interface and tasks format prior to the actual verifications and as aforementioned qualification tests were also completed by the participants. Another option is to *seed in control questions* (18) based on a (partial) gold standard without the evaluators' knowledge, which allows for assessing the intention or trustworthiness of the workers later on when the judgments are aggregated (e.g., filtering out spammers or malicious workers). In the described use case [19] it was not necessary to seed in control questions since a gold standard was available and the evaluation only had experimental aims.

Lastly, a *feedback form preparation* (19) might be especially useful in evaluation cases, where the verification should be repeated and the process should be improved based on comments from the evaluators. In [19] we collect feedback to analyze the students' experience when performing the verifications and outline confusing aspects that could be improved.

3.2 Execution Stage

Once the preparatory activities are completed, the verification tasks need to be performed following the activities depicted in Fig. 4. First, the *HIT templates are populated* (20 in Fig. 4) with data. At this point, the tasks are not yet publicly accessible and can be refined if needed before the evaluators can start working on them. Next, the *HITs are published* (21) and if needed *a presentation is shown* (22) to ensure the evaluators are familiar with the verification tasks.

To ensure high-quality judgments, *qualification tests* (23) can be made a requirement for working on a HIT. In [19], a high score of the developed qualification test for the crowd's skills in ontology modeling was not a prerequisite

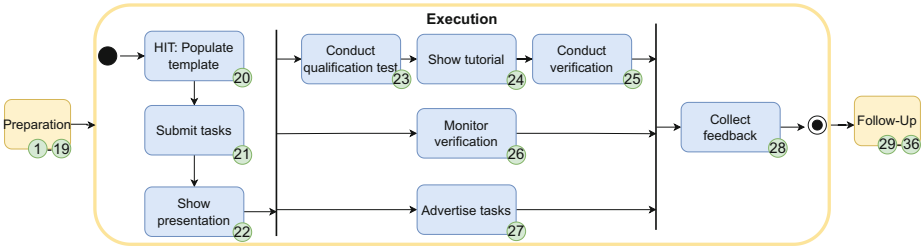

Fig. 4. HERO execution stage.

since the performed experiments aimed at analyzing how prior knowledge affects the verification results. Next, *the tutorial tasks are shown* (24), which consists in showing the previously prepared training questions. Afterward, the *verification is conducted* (25), which is the main activity in this stage where the verification tasks are completed by the evaluators. Typically the evaluation environment (e.g., mTurk) is responsible for showing open batches of questions and collecting the answers from the evaluators.

In parallel to the previously described activities, the *verification should be monitored* (26), which ensures potential problems are identified and corrected early on. To that end, crowdsourcing platforms typically provide management interfaces that can be used. In some cases, it might be required to stop the process and go back to a previous activity for revision. During the evaluations performed in [19] a Zoom Meeting was active, where evaluators could ask questions and technical problems were solved. Another parallel activity is the *advertisement of tasks* (27), which can be achieved through different means such as newsletters, web pages, or any other communication means. This activity is of particular importance if not enough evaluators are engaging in the tasks.

Finally, *feedback is collected* (28) from the evaluators using the prepared feedback form, so that potential problems with the workflow can be identified.

3.3 Follow-Up Stage

Follow-up activities conclude the ontology evaluation process as depicted in Fig. 5. Initially, the *crowdsourced data is to be collected* (29 in Fig. 5) and *pre-processed (30)* to be compatible with the prepared data analysis scripts. To gain an overview of the collected judgments, *data quality statistics are calculated*, which can include (but is not limited to) *calculating trustworthiness* (31) based on the control questions and other measures provided by the evaluation environment and *calculating inter-rater agreement* (32).

In micro-tasking environments typically redundant judgments are collected for each task. To obtain a conclusion on a task these answers need to be aggregated (33; e.g., using majority voting as in [19]). Afterward, the *data needs to be analyzed* (34) in order to obtain the final results of the verification.

Once a final set of results is obtained through analysis, this can be used to *improve the verified ontology* (35). This activity is tightly linked to the goal

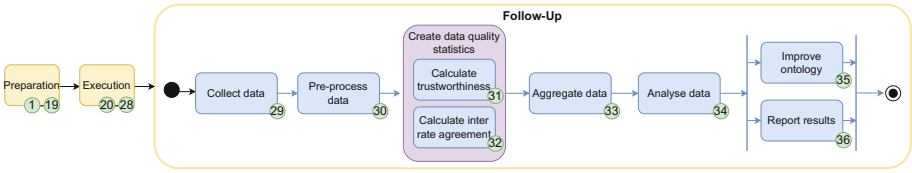

Fig. 5. HERO follow-up stage.

specified during the preparation and depending on it, the results can be used to improve certain aspects of the ontology. In [19] the participants were provided with their verification scores to enable the learning process and results were analyzed to test the experiment hypotheses. At the same time, the *results are to be reported* (36).

4 HERO Reference Architecture

The HERO process provides an in-depth understanding of the activities typically performed during human-centric ontology evaluation, and as such enables the design of a *reference architecture* that can be used as a basis for creating platforms that (partially) automate the activities of such evaluation processes.

HERO contains both activities that are relevant for a wide range of evaluation campaigns (e.g., loading and inspecting the ontology) as well as activities that are specific to certain evaluations (e.g., task design). To that end, we relied on a Microkernel Architecture which features (1) a core, where the general logic is captured and (2) the plugins, which extend the platform functionalities [14] (Fig. 6). Accordingly, the general functionalities of the platform are included as core components (i.e., an *Ontology Loader, Ontology Metrics, Data Provider, Triple Store, Verification Task Creator, Quality Control, Crowdsourcing Manager, Meta Data Store* and *Data Processor*), while plugins allow for customization to specific use cases. For instance, different *Context Provider* plugins can be developed to extract relevant context to be presented to the evaluators in the HITs and a separate *Crowdsourcing Connector* is needed for each crowdsourcing platform. Further information on the reference architecture (i.e., crosscutting, deployment and run-time viewpoints) can be found in [6].

5 Case-Study-Based Evaluation

The evaluation of the created artifacts investigates: *To what extent can the HERO process model and the corresponding reference architecture support the preparation of a human-centric ontology evaluation (i.e., the HERO preparation stage)*? We focus the evaluation on the *preparation stage of HERO* as it is the most effort intensive and can be most reliably replicated. Our evaluation goal translates into the following sub-questions:

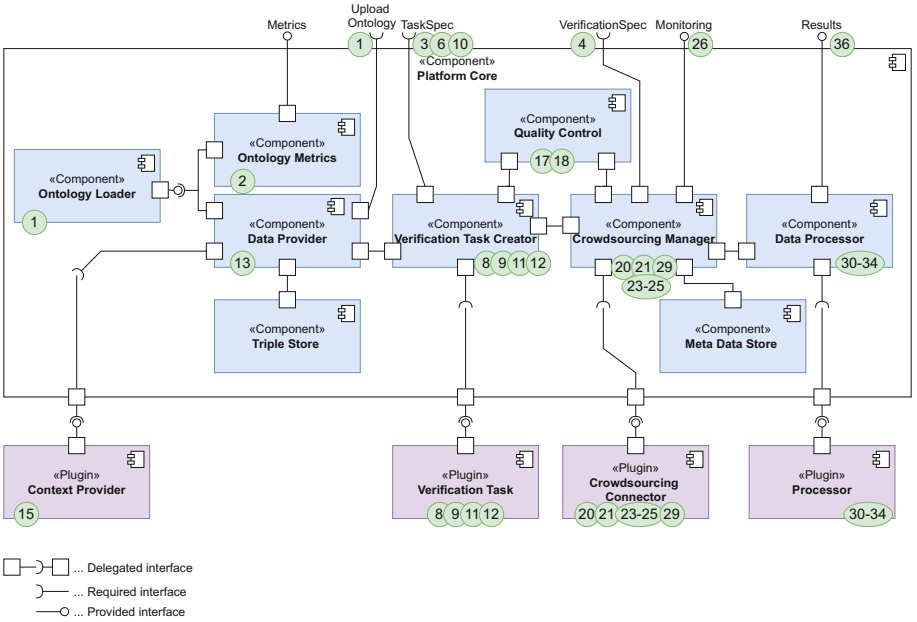

Fig. 6. Source code viewpoint of the HERO reference architecture including numbers of the connected process activities (see also Figs. 3, 4 and 5).

- *RQ1: Can the HERO process model be used to better structure the activities of a concrete evaluation campaign?*
- *RQ2: Is it feasible to implement a supporting platform based on the reference architecture?*
- *RQ3: How many preparation activities can be automated by a HERO-based platform?*
- *RQ4: What is the effort reduction when using the platform as opposed to a manual preparation process?*

To answer these research questions, we adopt a Case Study methodology [22] consisting of replicating the use case described in [19] by making use of the artifacts we developed. We started by representing the activities we followed during the preparation of the evaluation campaign from the use case in terms of the HERO process model in Fig. 7. We found that HERO can contribute to more clearly structuring how and through which activities the preparation of the evaluation campaign was performed (RQ1). It can also highlight potential weaknesses, for example, that the original preparation did not cover three activities: inspecting the ontology quality, designing batches and seeding control questions (which could be beneficial additions). Subsequently, as per RQ2, we used the reference architecture as a basis to implement a prototype platform to support the use case activities (Sect. 5.1). The prototype platform allowed the

tool-supported replication of the use case and enabled a comparison with the effort spent during the manual execution of the use case (RQ3, RQ4, Sect. 5.2).

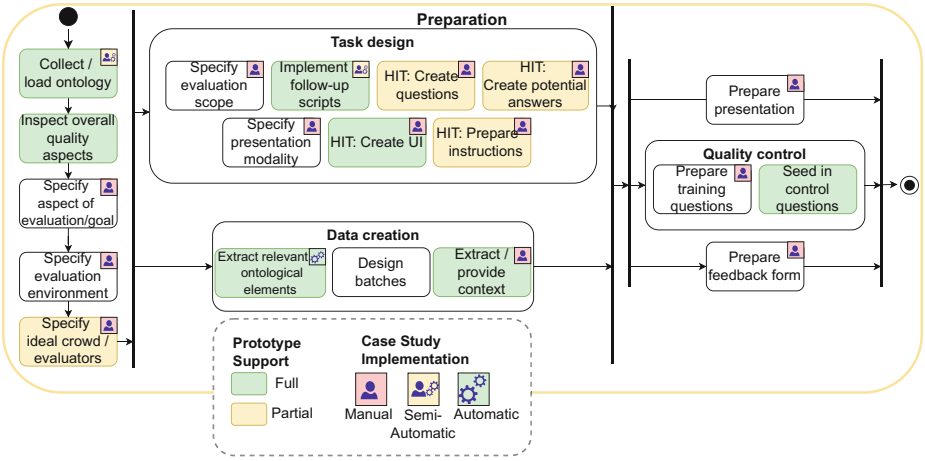

Fig. 7. Structured representation of the original use case (from [19]) in terms of the HERO activities (with indication of whether they were performed manually, automatically or semi-automatically). Indication of which of the activities of the process can be fully/partially supported by the HERO-based prototype platform.

5.1 Feasibility Study: Prototype Implementation

Following Fig. 6 of the reference architecture, we implement all core components[2] except the *Ontology metrics* (since this component is not required for replication of the use case [19]). To replicate the use case from [19], we develop several customized plugins[3], as follows:

- *RestrictionVerificationPlugin (VT)*, responsible for defining how the universal and existential quantification axioms are extracted from a given ontology. Further, we specify an HTML template and a method on how to extract values from the ontology for each variable in a template to define the GUI of the HITs. By using a configuration property the axioms can be rendered in the representational formalism proposed in [13] and [20].
- *PizzaMenuContextProviderPlugin (CP)*, responsible for creating a restaurant-menu-styled-item for each pizza ontology axiom.
- *AMTCrowdsourcingConnector (CC)*, responsible for publishing tasks on mTruk, retrieving the current status of the published verification and also obtaining the raw results from the platform.

[2] github.com/k-klemens/hc-ov-core.
[3] github.com/k-klemens/hc-ov-pizza-verification-plugins, ../hc-ov-amt-connector.

We conclude that the reference architecture was sufficiently detailed to make the implementation of a concrete supporting platform *feasible (RQ2)*. The implementation of the platform core took 55 h while the use-case-specific plugins required 28.5 h. We expect that similar implementation efforts will be required for other technology stacks or evaluation use cases.

5.2 Evaluation Results

Automated Activities (RQ3). As color-coded in Fig. 7, with the implemented platform prototype we can fully support 7 out of 19 (37%) and partially support 4 out of 19 (21%) of the HERO preparation activities. In the context of our use case [19], which only covered 16 activities, over 55% of the conducted preparation activities can be (fully or partially) supported by the platform. 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. Further, going beyond preparation activities, the platform and the implemented plugins allow publishing, monitoring and retrieving the results of created HITs on mTurk (although these activities are not subject to the current evaluation).

Reduction of Time Effort (RQ4). Our baseline is the time effort that was spent to prepare the ontology evaluation campaign in the original use case [19], that is 195 h. As part of this case study, we replicated the use case with the tool support based on HERO. The total effort spent on this replication amounts to 137 h (Fig. 8) and includes: specifying the reference architecture (48 h), implementing the platform core (55 h), implementing specific plugins (28.5 h), and miscellaneous activities, e.g., meetings (5 h).

Comparisons of these two effort categories can be performed in two ways. First, assuming that this replication case study is a one-of-activity, the effort for the preparation stage of the use case could be reduced by 30% by adopting the principled HERO-based approach and relying on the corresponding tool support. Second, our aim is that the artifacts created so far can be re-used in follow-up projects. In that case, assuming that in this case study we would have reused the reference architecture and core platform implementations, the effort for replication consists only in the adaptation of the plugins (28.5), thus leading to an effort reduction of 88%. More details are available in [6].

Improved Aspects. Besides time effort reduction, the platform offers *centralized orchestration and storage* by implementing end-to-end process support for human-centric ontology verification. It also allows for *extensibility and reusability* via the plugin architecture. Once a plugin is implemented, it can be reused for future verifications, thus, overall implementation efforts are expected to be reduced as the availability of plugins grows. Since the platform allows for the automation of manual activities (e.g., the extraction of context), *data scalability* will be ensured, especially for larger ontologies.

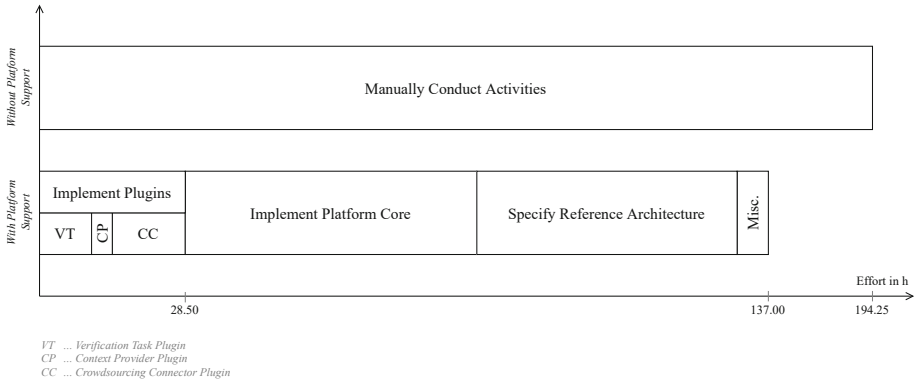

Fig. 8. Comparison of efforts between the use case evaluation approach *without platform support* and *with platform support*.

6 Related Work

Several works have already explored the conceptualization of a verification process model with a human-in-the-loop for evaluations of Semantic Web artifacts. In [5] the authors propose a high-level methodology for the crowdsourcing-based assessment of the quality of Linked Data resources, partly supported by the TripleCheckMate tool. However the methodology and tool are very much geared towards *assessing linked data triples*. A new method for *task-based ontology evaluation* is proposed by the authors of [11]. The presented methodology is tailored towards application ontologies and how fitted they are for a certain application task. In [21] the authors describe *a plugin for Protégé* supporting ontology modelers in the Ontology Engineering process, by outsourcing a set of human-centric tasks to games with a purpose or a crowdsourcing platform. However, this approach is dependent of the Protégé editor, which reduces its reusability.

Additionally to the Semantic Web research, in the Software Engineering domain, we are aware of related work such as (1) an approach for the *verification of Enhanced Entity-Relationship diagrams* based on textual requirement specifications relying on HC&C techniques [17] and (2) a process and framework for the human-centirc *validation of OntoUML (*ontouml.org*) models* [3].

Only two [11,21] of the process models above focus on the evaluation of ontologies. Most of the process models are lacking key details and have been derived ad-hoc as opposed to following a principled approach. Therefore, a need arises for a human-centric ontology evaluation method with more details and which is derived in a methodologically principled way.

7 Summary and Future Work

While human-centric ontology evaluation is often performed in order to verify ontology quality aspects that cannot be identified automatically, this area currently lacks detailed methodologies and suitable tool support. In this paper we

address this gap by providing (1) HERO- a detailed process for preparing and conducting human-centric ontology evaluation; (2) a reference architecture that supports this process; (3) a case-study-based evaluation exemplifying the use and benefits of these artifacts during the reproduction of a concrete use case. The case study indicates that, when supported by the HERO process and platform, ontology experts require, depending on the level of the artifact reuse, between 30% and 88% less time to prepare an ontology verification compared to a manual setting. As more plugins become available for reuse, we expect further reductions of effort, especially in use cases dealing with large ontologies. All artifacts were derived in a methodologically principled way by covering all three Design Science cycles and shared in the GitHub repository together with additional information.

In *future work*, we aim to address some of the current *limitations*. While we carefully followed a Design Science methodology, the core cycle of this method was only performed once. Therefore, we wish to conduct more design-evaluation cycles to further improve the current artifacts. Along the *evaluation* axis, the reference architecture was only indirectly evaluated through the case study's evaluation. A more sophisticated evaluation approach could involve interviews with domain experts and software architects. The case study focused on a use case with a small ontology thus giving only partial insights into efficiency gains, especially for larger ontologies. We plan a number of follow-up replication studies with larger ontologies and different verification problems to further test our artifacts. Along the *design* axis, further evaluations as described above will lead to iterative improvements of the artifacts such as (1) formalizing the HERO process using standards such as Business Process Model and Notation (BPMN) in order to create a richer model including also information about roles and activity results; (2) extending the implementation of the platform core and creating additional plugins for other types of ontology verification problems as well.

Acknowledgments. We thank all interview and focus group participants. This work was supported by the FWF HOnEst project (V 754-N).

References

1. Acosta, M., Zaveri, A., Simperl, E., Kontokostas, D., Auer, S., Lehmann, J.: Crowdsourcing linked data quality assessment. In: ISWC, pp. 260–276 (2013)
2. Erez, E.S., Zhitomirsky-Geffet, M., Bar-Ilan, J.: Subjective vs. objective evaluation of ontological statements with crowdsourcing. In: Proceedings of the Association for Information Science and Technology, vol. 52, no. 1, pp. 1–4 (2015)
3. Fumagalli, Mattia, Sales, Tiago Prince, Guizzardi, Giancarlo: Mind the Gap!: learning missing constraints from annotated conceptual model simulations. In: Serral, Estefanía, Stirna, Janis, Ralyté, Jolita, Grabis, J.ānis (eds.) PoEM 2021. LNBIP, vol. 432, pp. 64–79. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91279-6_5

4. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. 75–105 (2004)
5. Kontokostas, D., Zaveri, A., Auer, S., Lehmann, J.: TripleCheckMate: a tool for crowdsourcing the quality assessment of linked data. Commun. Comput. Inf. Sci. **394**, 265–272 (2013)
6. Käsznar, K.: A process and tool support for human-centred ontology verification. Master's thesis, Technische Universität Wien (2022). https://repositum.tuwien.at/handle/20.500.12708/20577
7. Law, E., Ahn, L.V.: Human computation. Synth. Lect. Artif. Intell. Mach. Learn. **5**(3), 1–121 (2011)
8. McDaniel, M., Storey, V.C.: Evaluating domain ontologies: clarification, classification, and challenges. ACM Comput. Surv. (CSUR) **52**(4), 1–44 (2019)
9. Mortensen, J.M., et al.: Using the wisdom of the crowds to find critical errors in biomedical ontologies: a study of SNOMED CT. J. Am. Med. Inform. Assoc. **22**(3), 640–648 (2015)
10. Nakagawa, E.Y., Guessi, M., Maldonado, J.C., Feitosa, D., Oquendo, F.: Consolidating a process for the design, representation, and evaluation of reference architectures. In: 2014 IEEE/IFIP Conf. on Software Architecture,pp. 143–152 (2014)
11. Pittet, P., Barthélémy, J.: Exploiting users' feedbacks: towards a task-based evaluation of application ontologies throughout their lifecycle. In: IC3K 2015 - Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, vol. 2, pp. 263–268 (2015)
12. Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa, M.C.: Oops!(ontology pitfall scanner!): an on-line tool for ontology evaluation. Int. J. Semant. Web Inf. Syst. (IJSWIS) **10**(2), 7–34 (2014)
13. Rector, A., et al.: OWL Pizzas: practical experience of teaching OWL-DL: common errors & common patterns. In: Motta, Enrico, Shadbolt, Nigel R., Stutt, Arthur, Gibbins, Nick (eds.) EKAW 2004. LNCS (LNAI), vol. 3257, pp. 63–81. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30202-5_5
14. Richards, M.: Software Architecture Patterns, vol. 4. O'Reilly Media, Incorporated 1005 Gravenstein Highway North, Sebastopol, CA (2015)
15. Sabou, M., Aroyo, L., Bontcheva, K., Bozzon, A., Qarout, R.: Semantic web and human computation: the status of an emerging field. Semant. Web J. **9**(3), 291–302 (2018)
16. Sabou, M., Fernandez, M., Poveda-Villalón, M., Suárez-Figueroa, M.C., Tsaneva, S.: Human-centric evaluation of semantic resources: a systematic mapping study, In preparation
17. Sabou, M., Winkler, D., Penzerstadler, P., Biffl, S.: Verifying conceptual domain models with human computation: A case study in software engineering. In: Proceedings of the AAAI Conference on Human Computing and Crowdsourcing, vol. 6, pp. 164–173 (2018)
18. Tsaneva, S.: Human-Centric Ontology Evaluation. Master's thesis, Technische Universität Wien (2021). https://repositum.tuwien.at/handle/20.500.12708/17249
19. Tsaneva, S., Sabou, M.: A human computation approach for ontology restrictions verification. In: Proceedings of the AAAI Conf. on Human Computation and Crowdsourcing (2021). www.humancomputation.com/2021/assets/wips_demos/HCOMP_2021_paper_90.pdf
20. Warren, P., Mulholland, P., Collins, T., Motta, E.: Improving comprehension of knowledge representation languages: a case study with description logics. Int. J. Hum.-Comput. Stud. **122**, 145–167 (2019)

21. Wohlgenannt, G., Sabou, M., Hanika, F.: Crowd-based ontology engineering with the uComp Protégé plugin. Semant. Web **7**(4), 379–398 (2016)
22. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29044-2

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.