2023 European Conference on Computing in Construction
40th International CIB W78 Conference
Heraklion, Crete, Greece
July 10-12, 2023

SUPPORTING APPRAISAL COST ESTIMATION BY LINKED DATA

Sebastian Seiß[1], Jan Niklas Lünig[2] and Jürgen Melzner[1]
[1]Bauhaus-Unversität Weimar, Weimar, Germany
[2]Technische Universität Braunschweig, Braunschweig, Germany

Abstract

The construction of a building is a unique and project-specific process associated with a multitude of risks. In particular, execution risks and associated defect costs have risen recently. Deadlines, cost pressure, growing process, and material complexity justify these cost increases. Therefore, achieving a balance between the optimum level of quality and the associated costs requires knowledge of the composition of quality-related costs. Despite steadily increasing amounts of quality data from digital building models, sensors, and standards, calculation models in the construction industry predicting appraisal costs remain non-existent. Therefore, this paper aims to determine expected appraisal costs from building information modeling (BIM) objects via automated semantic web technologies. The ontology for construction quality assurance (OCQA), which is used for the automatic planning of quality inspections, serves as the initial ontology in this paper. The expected quality-related appraisal costs are determined automatically via Shapes Constraint Language (SHACL) rules based on the OCQA, BIM data, and historical quality data. The inferred appraisal costs are stored in the OCQA extension ontology for construction quality assurance appraisal cost (OCQA-AC). To demonstrate the rules' functionality, leveling methods and associated defect-type unevenness are used as examples.

Introduction

As described by the proverb, "You can't make an omelet without breaking eggs", failures are integral to daily construction. Thus, the ability to deal with failures, risks, and consequences has become a central success factor for companies in recent years. While other industries have made steady progress in dealing with failures, the consideration of failures in the construction industry, according to a survey by the Fraunhofer Information Center for Space and Building, plays only a subordinate role. Despite annual failure elimination costs of 16.5 billion euros (as of 2021), only 51% of construction companies in Germany consider the expected calculable costs (Rauh and Weitz, 2014). As a result of the low consideration for failures, the number of cost overruns, delays, and quality defects in construction projects is increasing.

Simultaneously, due to the building information modeling (BIM) method and the use of sensors and digital construction site tools, the amount of data related to construction quality has steadily increased. Quality data is derived from the specification and demonstration of process and system conformity (Schmitt et al., 2016). However, the actual value of the data remains limited due to the lack of machine readability and data networking. Linked Open Data (LOD) technologies, part of the semantic web, provide a proven solution to convert unstructured data into machine-readable formats. In addition to improved data exchange, the technologies support users in finding and formalizing relevant data. LOD technologies have been used for these purposes in various fields; for example, in medicine, semantic web applications help make faster and better medical decisions (Liyanage et al., 2015).

Modern quality management systems must use all available quality data to detect potential failures in their early stages and initiate preventive measures (Schmitt et al., 2016). Hence, knowledge of appraisal costs of the component in question is necessary to assess the effects of failures and determine appropriate preventive measures . Therefore, this research aims to investigate how predictable inspection costs can be automatically retrieved by a given workflow preparation that includes BIM and scheduling data.

The paper is structured as follows. First, Section 2 reviews the related work on ontologies regarding construction inspection costs, while Section 3 describes the methodology used to develop the proposed ontology. Next, Section 4 discusses the conceptualization and implementation of the methodology, evaluated using a fictitious example in Section 5. Finally, Section 6 discusses the limitations and contributions of the research and provides conclusions.

Background on appraisal cost estimation

The classic quality cost model follows an activity-oriented structure according to the quality-related cost concept from DIN 55350 concepts for quality management. According to the DIN definition, quality-related costs include prevention-, appraisal- and nonconformity costs. Prevention costs (PCs) result from analyzing and eliminating error causes, while appraisal costs (ACs) are caused by routine inspections without specific error reasons. Nonconformity costs (NCs) result from failures. The quality cost categories may be delimited and recorded according to organization specific criteria. In the research conducted by Shafiei et al. (2020), the complex interaction of quality cost elements in building projects was examined and the resulting costs varied with the extent of the quality inspection (100% inspection vs. the sample), quality standards, and as the respective inspection method. Indeed, higher requirements for safety, serviceability, durability and aesthetics require greater testing expenditures, leading to higher ACs. However, according to Kiani et al. (2009), the associated costs of increased prevention and appraisal efforts reduce NCs.

$$AC_X = C_{X,O} + n * (C_{X,L} + C_{X,M}) \quad (1)$$

According to DIN 55350, the ACs calculation does not have to follow a predefined calculation model. ACs include labour ($C_{X,L}$) and material costs ($C_{X,M}$) for quality inspection incurred within and beyond the quality system (Hering et al., 2013). The cost types include maintenance costs, calibration, measurement, and equipment testing (Roden and Dale, 2001; Kazaz et al., 2005). In contrast to the stationary industry, ACs in the construction industry have an overhead cost component ($C_{X,O}$) and a variable part depending on the number of inspections n, according to Simon and Müller (2014). Overhead costs include the travel to the inspection location, the installation and dismantling of inspection equipment, and external acceptance costs (Simon and Müller, 2014; Hering et al., 2013). For recording ACs, the following steps are required per Brüggemann and Bremer (2015): (1) a status quo recording of the actual costs from quality assurance must be undertaken, (2) identified cost items must be allocated to individual cost types (labour, equipment, material), and (3) an evaluation of ACs and, if necessary, an initiation of measures to reduce the individual cost components must be made.

Related works

This section discusses previous research on ACs and the related ontologies to acquire proper knowledge, background and references to develop the proposed ontology. The review is divided into a general review of models to estimate ACs and ontological approaches.

Related works on quality inspection

Chen and Luo (2014) developed a BIM-based construction quality management model where an algorithm uses 4D BIM to identify quality inspection criteria and derive responsibilities (Chen and Luo, 2014). Moreover Won and Lee (2011) also derived inspection tasks using an IFC model and hard-programmed algorithms and the presented method for scheduling inspection tasks reduced planning time measurably. This work showed that automated inspection planning can significantly improve the work preparation phase (Won and Lee, 2011). Ma et al. (2018) published the paper "Construction quality management based on a collaborative system using BIM and indoor positioning" to describe developing of a quality-related collaboration system combined with indoor positioning. Algorithms generate inspection tasks, and inspections are visually supported by a positioning system, followed by a visualization of inspection results by bubbles, manual checks, and the interpretation of inspection results. (Ma et al., 2018) Xu et al. (2018) developed an an IFC-based database for evaluating quality-related data, extended by quality data and a neural network. The evaluation was not performed on a single defect but on the entire project. Finally, Cheng. (2018) focused on checking mobile recorded quality data, by developing a tool in Revit for quality recording and compliance checking by using Revit API access to store and check quality.

Ontological works on appraisal cost estimation

Zhong, Luo, Hu, and Sun (2012) an ontology for recording construction qualities, leaving quality control planning to the end user, and the planning support of the work was limited to the requirement that quality control must occur for individual construction tasks Zhong, Luo, Hu, and Sun (2012). Martinez et al. (2019) developed an ontology identifying relevant regulations for quality controls based on a BIM via queries in SPARQL (Martinez et al., 2019). (Chen and Luo, 2014) developed a 4D BIM-based quality management system: automatic selection tool of quality control specifications for of-site construction manufacturing products, a BIM-based ontology model approach. The ontology automatically selected company or organization quality control specifications according to the requirements defined in the BIM, while SPARQL queries derived the standards.

The quality costing ontology (OCQ) developed by Eldridge et al. (2006) supports tracking and analyzing quality-related costs in the stationary industry. The OCQ includes a standard form for recording costs, allowing users to assign costs to specific categories, such as avoidance, inspection, or defect costs, and includes fields for recording location, department, and cost center . Compared to traditional static forms, the OCQ allows a more structured and efficient process for collecting, storing, and evaluating quality-related cost data.

Temporal workflow condition ontology (TWCO) by Lee et al. (2014) defines the conditions that must be met to execute certain tasks within a workflow. These conditions include temporal constraints, such as the order in which tasks must be executed, and other constraints, such as the availability of resources or the completion of specific prerequisite tasks. TWCO also defines the relationships between different conditions and how they interact. In comparison, temporal workflow item ontology (TWIO) by Lee et al. (2014) defines tasks, or "work items," comprising the workflow and the resources required to execute them. It also defines the relationships between different tasks and resources and how they interact (Lee et al., 2014).

These studies demonstrated the potential benefits of ontological application in inspection planning. However, many limitations prevent the ontology-based estimation of ACs in incubation periods since other researchers and industries have not widely used or adopted the illustrated ontologies. Moreover, most of the ontologies don't meet the requirements from DIN 55350, and the applications of the QCO are limited to the execution period. Additionally, an extension of the presented ontologies is impossible due to the lack of publications.

The present study aims to develop a set of higher-level ontologies to support the CW information integration within digital construction contexts to address these limitations that may (1) represent the CW in detail and connect activities and flow entities, (2) integrate heterogeneous flow-related information sources from ICT-based systems, and (3) convey multi-context representational data.

Methodology

In research, various methodologies have been developed to define ontologies. The most common definitions were developed by Grüninger and Fox, Uschold and Grüninger, the ontology development guideline 101, and the On-To-Knowledge methodology(Zheng et al., 2021).This paper follows a hybrid model combining the LOT methodology with that of Uschold and Grüninger and Grüninger and Fox. The hybrid model enables the compensation of disadvantages between these methodologies. According to LOT, the process of ontology engineering is divided as follows:

1. specification,
2. implementation,
3. publication,
4. maintenance phases (Poveda-Villalón et al., 2022).

The detailed process of the ontology's development and engineering methodology is illustrated in Figure 1. The specification phase defines the scope, purpose, use cases, users, and requirements for the inspection cost ontology. Notably, knowledge acquisition is an ongoing iterative process throughout the ontology engineering process initially undertaken in step 1.3(Fernández-López, M. et al., 1997). Based on the specification, a conceptualization, including existing ontologies and encoding, is conducted, and the evaluation of the developed ontology finalizes the implementation. Therefore, different evaluation methods are used to judge the developed ontology. Afterward, the ontology is documented via a Hypertext Markup Language (HTML) sheet and published on GitHub to provide access to a wide range of users and developers. GitHub enables proper ontology maintenance due to collaboration and version control based on the repository (GitHub, 2020).

Specification

This specification aims to produce a formal document in natural language utilizing competency questions (CQs) as requirements (Fernández-López, M. et al., 1997). These CQs are derived from previously defined use cases based on the purpose and scope of the corresponding ontology. The specification document is provided on GitHub (Poveda-Villalón et al., 2022). In the following discussion, the different specification steps are formalized.

Purpose: The purpose is an ontology focused on quality inspection costs in construction execution to provide a structured representation of cost information that civil engineers can utilize to better understand quality inspection costs. Furthermore, the ontology can be used to support (semi-)automated inspection estimation, which can aid in cost forecasting and decision-making. The proposed ontology should :

1. allow an allocation of costs to individual cost units,
2. reveal the cost-effectiveness of quality inspection,
3. enable the estimation of quality-related ACs,
4. support tactical and operational decisions in quality assurance, and
5. strengthen the quality awareness of employees (Wendehals, 2000)

Scope: The ontology aims to describe and estimate quality-related costs based on ACs according to DIN 55350:2021-10.144 by capturing ACs as a part of quality costs but does not cover NCs. ACs in the proposed ontology include overhead, labor, and material costs. However, these costs are not included due to the difficulty of measuring PCs from employee training and inspection planning.

Use-Cases:

1. Description of quality-related ACs
2. Supporting manual appraisal cost estimation
3. Automated estimation of ACs

End users: The end users of the ontology are defined according to the use cases of the ontology. Therefore, construction project management will mostly use the ontology, including construction managers, estimators, and quality inspectors. However, regarding closely-defined use cases, the ontology will not be used by construction workers or other stakeholders not included in the management of construction sites.

Non-functional requirements (NFRs): NFRs define how a system should perform, behave, and operate rather than what it should do. Various studies have defined NFRs for ontologies in the field of construction that can be summarized as follows: (1) coverage/sufficiency, (2) consistency, (3) usability, (4) extendibility/reusability, and (5) clarity and conciseness (Costin, A. and Eastman, C., 2017; Zhou et al., 2016; El-Gohary and El-Diraby, 2010; Zheng et al., 2021). The defined NFRs are used as evaluation criterium and covered by different evaluation methods.

Functional requirements: The functional requirements are described as CQs according to the previously defined use cases. Table 1 lists the CQs to be answered by the proposed ontology.

Table 1: Competency questions related to use cases

Description of quality quality-related costs	Supporting manual inspection cost estimation	Automated estimation of inspection costs
1. What are the total ACs of an inspection?	3. How can cost parameters be restored in an ontology?	4. How can reasoning support cost parameter assignment?
2. What are the partial ACs of an inspection?		5. How can inspection cost types be estimated

Conceptualization and Implementation

The proposed ontology system is divided into three modules according to the project's aim: (1) overall OCQA, (2) inspection cost estimation, and (3) catalog.

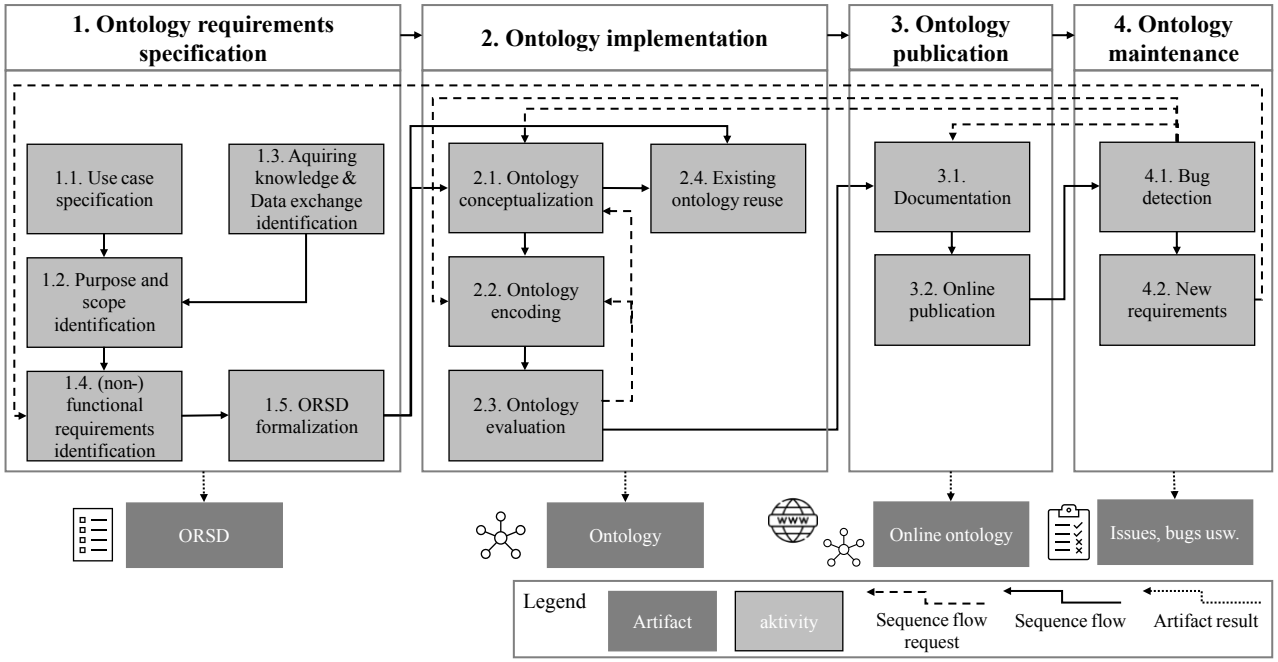

Figure 1: Ontology development/engineering methodology (Poveda-Villalón et al., 2022)

OCQA environment

The developed OCQA-AC ontology is an extension of the OCQA ontology designed to provide information about quality inspections in construction to support inspection planning. The OCQA also contains rules for automated inspection planning. Therefore, the OCQA is an ideal reference for developing the OCQA-AC, designed as an extension of the OCQA. In turn, the OCQA is an extension of the DiCon ontology, providing basic concepts for construction workflows, including agents, processes, and equipment (Zheng et al., 2021). The alignment between the different ontologies can be seen in Figure 2 and is implemented as hard reuse via owl:imports, which reuses the imported ontology as it is and as a whole (Poveda-Villalón et al., 2022). Moreover, trade-specific modules of the OCQA ontology cover task-specific inspection planning. For example, the extension OCQA screed covers all inspections regarding the trade screed.

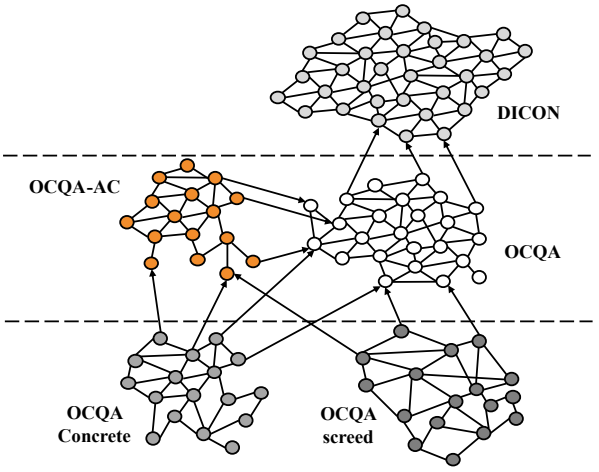

Figure 2: OCQA-AC as an extension/alignment/reuse of the OCQA and DiCon ontology

Ontology for appraisal costs (OCQA-AC)

This section presents a more detailed overview of the OCQA-AC and the associated ontologies. Therefore, the classes, relations, and properties needed to estimate inspection costs are shown in Figure 3. To estimate ACs, the inspections and the associated inspection equipment, persons, inspection objects, and inspection methods must be represented in detail directly via DiCon classes and OCQA classes, defined as subclasses of the DiCon ontology. The class *ocqa:Inspection* is a subclass of *dicp:Activity*. The required cost indices for cost calculations are linked to the inspection description. Cost values are stored as *ocqa:characteristic* in a data catalog and are linked to *ocqa:inspection* or other classes via *ocqa:hasCostCharacteristic*. To provide fitting cost value relations, *ocqa:hasCostCharacteristic* can be specialist by sub-properties like *ocqa:hasHourlyRateCostCharacteristic* to describe the hourly rate of an inspector. The *ocqa:Characteristic* represents a subclass of opm:property and can therefore be updated with actual costing values according to the OPM ontology. The catalogs are designed according to the DCAT ontology and can be differentiated into a main catalog and a sub-catalog (Riccardo et al., 2020). The cost estimation results are represented as data values linked to the inspection via Datatype properties.

OCQA-AC appraisal cost estimation

The proposed ontology OCQA-AC was designed to enable the automated estimation of appraisal costs. Hence,

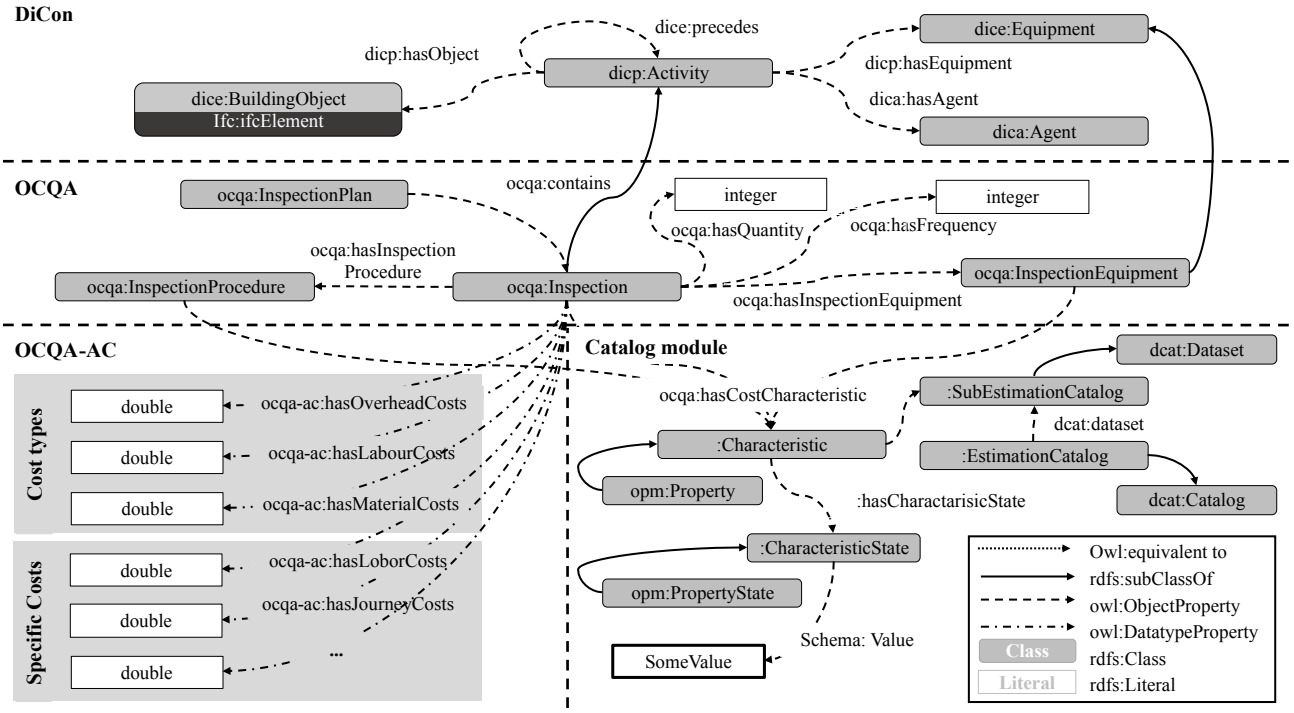

Figure 3: Overview of classes, relations, and data properties of OCQA-AC and aligned ontologies

the ontology had to infer and estimate various types of ACs, so the inference process was divided into two distinct steps. In the first step, inspections are linked to cost characteristics in a predefined cost catalog. The relation between cost characteristics and inspections, equipment, and staff is done by reasoning via OWL class restrictions. All instances of relevant classes – in this case, subclasses of agent, equipment, and inspection – are automatically assigned to cost characteristics via the OWL restriction *owl:hasValue* (Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L., 2004).

In Step 2, ACs are calculated using the SHACL rule in Listing 1 divided into two *sh:nodeShapes*. The first part estimates the costs based on the cost value characteristic times the count of inspections. The cost value for equipment cost is defined as cost per inspection. The multiplication of inspection quantity and cost value characteristic is done by *ocqa_rule:multiply*, which is predefined as a SPARQL function. Moreover, a *sh:condition* is added to enforce the execution of the rule only under the condition of predefined *ocqa:characteristics*. The second *sh:NodeShape* sums up the different ACs to the total ACs of an inspection. The implementation is done via a SPARQL construct function using *sh:construct* and a subquery, to sum up the values.

```
ocqa_rules:CalculateAppraisalCost_EquipmentCost
a sh:NodeShape;
sh:targetClass ocqa:Inspection;
sh:rule [
  a sh:TripleRule ;
  sh:subject sh:this ;
  sh:predicate ocqa:hasEquipmentCost;
  sh:object [
      ocqa_rule:multiply ([ sh:path ocqa:
    hasQuantity ] [ sh:path (oqca:
    hasInspectionEquipment ... schema:value)]);
  ]
    sh:condition ocqa_rules:
    costEquipmentCondition;
].
ocqa_rules:CalculateAppraisalCost_TotalCost
a sh:NodeShape;
sh:targetClass ocqa:Inspection;
sh:rule [
  a sh:SPARQLRule ;
    sh:construct """
       [Prefixes]
       CONSTRUCT {$this ocqa:hasAppraisalCost ?
    sum.}
       WHERE { [Subquery to get all assigned
    costs on inspection]}
       """
].
```

Listing 1: SHACL rule for appraisal cost inferencing

Evaluation

The evaluation judges the predefined requirements and the use cases to determine if the specification process is satisfied by the ontology (Fernández-López, M. et al., 1997). According to Zheng et al. (2021) different evaluation methods can be used to check compliance with different criteria (Zheng et al., 2021; El-Gohary and El-Diraby, 2010). The proposed ontology is evaluated by automated consistency checking CQs in combination with a task-based evaluation and by a workshop consisting of focus-group interviews.

Automated consistency checking

The evaluation based on automated consistency checking enables consistency verification of the proposed ontology. A consistency check for OCQA-AC ontology was performed with Pellet Reasoner, an open OWL-DL reasoner used for ontology inference and consistency checking (Sirin et al., 2007), and the Pellet check showed that the created ontology was consistent and coherent. Furthermore, the ontology was checked for possible modeling errors by the OntOlogy Pitfall Scanner (OOPS), developed to automatically detect certain failures in ontologies based on object-oriented and ontological principles (Poveda-Villalón et al., 2014). This check was also successfully completed.

Task-based evaluation including CQs

Task-based evaluation assesses whether an ontology can solve a specific task. Furthermore, the instance data can answer the CQs defined in the specification section and illustrate how the ontology can be used in practice (Zheng et al., 2021). Therefore, applying the OCQA-AC and the corresponding ontologies was done using the example case of ACs for leveling inspections, which are measurements to check the evenness of building components, for example, on a gravel layer or finished screed. The illustrated example case was based on practical data provided by our partners. Nevertheless, the cost values could not be provided by our partners due to missing data and were assumed by ourselves.

In the example case, the ACs were structured into *ocqa-ac:haslabourCost* and *ocqa-ac:hasMaterialCost* costs (see Figure 4). To estimate material costs of a leveling inspection, the cost values of *ocqa:hasEquipmentCostsCharacteristic* and *ocqa:hasMaintainanceCostsCharacteristic* were linked to the instance *ex:AutomaticLevel_p3sb42d_ka3* of class *ocqa:AutomaticLevel_NAK2*. The cost values were stored as *ocqa:CharacteristicState* and are linked by *ocqa:hasCostCharacteristic* to corresponding classes according to figure 3. The linking between cost values and instances of corresponding classes was automatically done by reasoning using OWL restriction *owl:hasValue* as shown in the previous section. Based on linked cost values, the specific cost was calculated.

To calculate the specific cost of *ocqa-ac:hasEquipmentCost* the SHACL-rules were used to multiply the inspection quantity *ocqa:hasQuantity* with cost value *ocqa:hasEquipmentCostsCharacteristic*. The calculation of *ocqa:hasMaintainanceCosts* and *ocqa-ac:hasHourlyRate* was analogous to the previous one. Finally, the cost types were calculated, resulting from the sum of the assigned specific costs. In this example, the *ocqa-ac:hasEquipment Cost* and *ocqa-ac:hasMaintainanceCosts* were used to estimate cost type *ocqa-ac:hasMaterialCosts*. The *ocqa-ac:hasAppraisalCosts* resulted from the sum over all cost types.

The described example contained a total ACs of 235 EUR CQ(1) for leveling measurements. Partial cost components from CQ(2) were broken down in the ontology using 1. The modeled catalog provided quality engineers with cost values to estimate ACs CQ(3). Based on catalog values, the associated costs for inspection were determined automatically using the reasoner from CQ(4). Finally, total ACs from CQ(1) and individual costs from CQ(5) were determined using SHACL rules.

Focus group interviews

The coverage/sufficiency, usability, clarity, and concise evaluation criteria were covered during online focus-group interviews and included a minimum of four and maximum seven domain experts for quality inspections. During the expert interviews, current company-specific calculation procedures for estimating the costs of quality inspections were discussed. Based on the insights gained from these interviews, the implementation of the proposed ontology for quality inspection cost estimation was presented and evaluated by the experts. Additionally, the experts formulated new requirements for the ontology as CQs, which were subsequently incorporated into the study's CQs to ensure comprehensive topic coverage. Finally, the experts' feedback and new requirements were considered in further developing and refining the ontology.

Discussion and Conclusion

This paper presented the OCQA extension OCQA-AC to reduce quality-related costs and support quality engineers in planning and estimating ACs. The OCQA-based ontology suite enables estimators, construction managers, and other stakeholders to store ACs, support manual AC estimation, and automate the estimation of ACs. The knowledge about ACs enables engineers to compare NCs with ACs. The comparison prevents uneconomical inspections while avoiding building elements with high NCs to ACs ratios. Therefore, a balance between AC and NC costs can be achieved by the application OCQA-AC. In addition, quality engineers can integrate heterogeneous information, such as BOQs, schedules, and standards, from different software solutions to plan inspections and estimate ACs. Finally, the modular design of OCQA offers the flexibility to expand the ontology to include trades not yet covered.

This study demonstrated the application of semantic web technologies to estimate ACs in construction. Therefore this paper is a crucial part of estimating quality-related costs according to DIN 55350. The significant contributions of the proposed OCQA-AC are 1) providing terminology to represent ACs information and knowledge, 2) enabling the storage and knowledge-based linking of ACs indices with inspection or other classes via catalogs, and 3) calculating sub and total ACs based on SHACL rules. In addition, the ontology suite was evaluated by 1) automated consistency checks, 2) task-based evaluation using CQs, and 3) focus-group interviews. The evaluation showed the requirements were fulfilled, so the developed

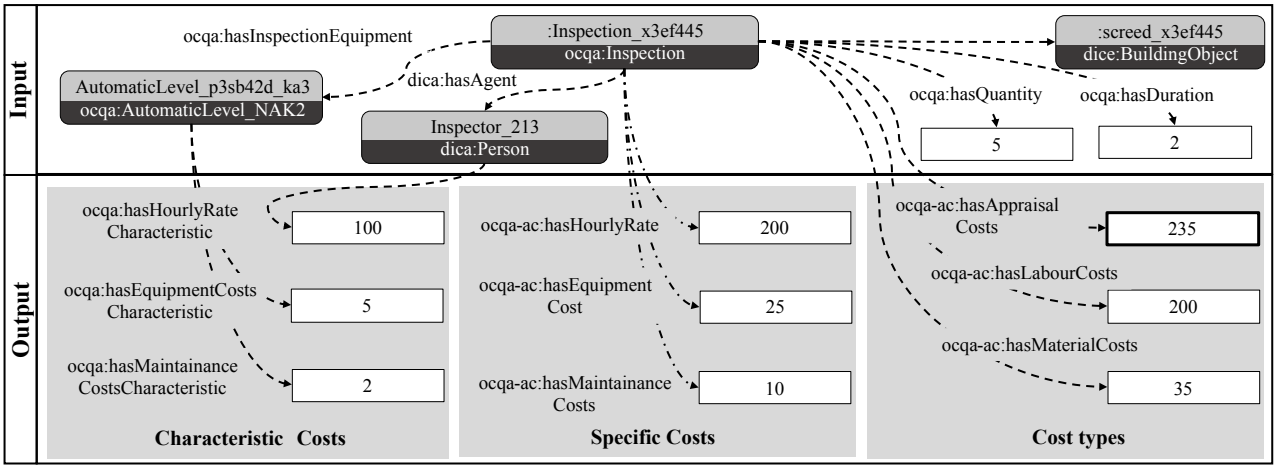

Figure 4: OCQA-AC validation on sample of levelling inspection

ontology suite could support the described use cases.
According to our results, further research is needed to encompass all quality-related costs within the proposed ontology. Estimating NCs and PCs will enhance the ability to determine the economic efficiency of quality inspections. Integrating probabilities to calculate the risks of NCs will significantly impact future inspection planning, moving from the current code- and experience-based inspection planning in construction to a failure analysis-based approach. Therefore, further OCQA extensions are required to cover NCs and PCs.
Moreover, integrating SHACL rules into the characteristics should be considered to run the SHACL rules based on the assigned characteristics while abandoning the condition checks. A methodology similar to the one presented in this work can also be applied to estimation software.

References

Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L. (2004). Owl web ontology language reference: W3c recommendation 10 february 2004.

Brüggemann, H. and Bremer, P. (2015). Grundlagen Qualitätsmanagement - Von den Werkzeugen über Methoden zum TQM. Springer-Verlag, Berlin Heidelberg New York.

Chen, L. and Luo, H. (2014). A bim-based construction quality management model and its applications. Automation in Construction, 46:64–73.

Cheng., Y. (2018). Building information modeling for quality management. In Proceedings of the 20th International Conference on Enterprise Information Systems - Volume 2: ICEIS,, pages 351–358. INSTICC, SciTePress.

Costin, A. and Eastman, C. (2017). Requirements for ontology development in the aeco industry. Lean and Computing in Construction Congress (LC3): Volume I Ð Proceedings of the Joint Conference on Computing in Construction (JC3), July 4-7, 2017, (Volume I).

El-Gohary, N. M. and El-Diraby, T. E. (2010). Domain ontology for processes in infrastructure and construction. Journal of Construction Engineering and Management, 136(7):730–744.

Eldridge, S., Balubaid, M., and Barber, K. (2006). Using a knowledge management approach to support quality costing. International Journal of Quality Reliability Management, 23:81–101.

Fernández-López, M., Gómez-Pérez, A., and Juristo, N. (1997). Methontology: From ontological art towards ontological engineering. undefined.

GitHub (2020). Github.com.

Hering, E., Triemel, J., and Blank, H.-P. (2013). Qualitätsmanagement für Ingenieure. Springer-Verlag, Berlin Heidelberg New York.

Kazaz, A., Birgonul, M., and Ulubeyli, S. (2005). Cost-based analysis of quality in developing countries: A case study of building projects. Building and Environment - BLDG ENVIRON, 40:1356–1365.

Kiani, B., Shirouyehzad, H., Bafti, F., and Fouladgar, H. (2009). System dynamics approach to analysing the cost factors effects on cost of quality. International Journal of Quality and Reliability Management, 26:685–698.

Lee, S.-K., Kim, K.-R., and Yu, J.-H. (2014). Bim and ontology-based approach for building cost estimation. Automation in Construction, 41:96–105.

Liyanage, H., Krause, P., and de Lusignan, S. (2015). Using ontologies to improve semantic interoperability in health data. BMJ Health & Care Informatics, 22(2):309–315.

Ma, Z., Cai, S., na, M., Yang, Q., Feng, J., and Wang, P. (2018). Construction quality management based on a collaborative system using bim and indoor positioning. Automation in Construction, 92:35–45.

Martinez, P., Ahmad, D. R., and Al-Hussein, M. (2019). Automatic selection tool of quality control specifications for off-site construction manufacturing products: A bim-based ontology model approach. pages 141–148.

Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., and García-Castro, R. (2022). Lot: An industrial oriented ontology engineering framework. Engineering Applications of Artificial Intelligence, 111:1–22.

Poveda-Villalón, M., Gómez-Pérez, A., and Suárez-Figueroa, M. C. (2014). Oops! (ontology pitfall scanner!): An on-line tool for ontology evaluation. International Journal on Semantic Web and Information Systems (IJSWIS), 10(2):7–34.

Rauh, R., F. M. G.-S. N. K. P. and Weitz, G. (2014). Organisationsmodelle und vertragliche Anreizsysteme zur Verbesserung der Bauqualität bei der Ausführung schlüsselfertiger Baumaßnahmen. Fraunhofer-IRB-Verlag, Stuttgart.

Riccardo, A., David, B., Simon, C., Alejandra, Gonzalez, B., Andrea, P., and Peter, W. (03.02.2020). Data catalog vocabulary (dcat) - version 2.

Roden, S. and Dale, B. G. (2001). Quality costing in a small engineering company: issues and difficulties. The Tqm Magazine, 13:388–399.

Schmitt, R. H., Ngo, Q. H., Groggert, S., and Elser, H. (2016). Datenbasierte Qualitätsregelung. volume 6 of Kasseler Schriftenreihe Qualitätsmanagement, pages 23–42, Kassel. Kassel University Press.

Shafiei, I., Eshtehardian, E., Nasirzadeh, F., and Arabi, S. (2020). Dynamic modeling to reduce the cost of quality in construction projects. International Journal of Construction Management, pages 1–14.

Simon, A. and Müller, A. (2014). Die bauwerksprüfung geschützter holzbrücken – handnah und wirtschaftlich?

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical owl-dl reasoner. Journal of Web Semantics, 5(2):51–53.

Wendehals, M. (2000). Kostenorientiertes Qualitätscontrolling: Planung – Steuerung – Beurteilung. Gabler Edition Wissenschaft Ser. Deutscher Universitäts Verlag, Wiesbaden.

Won, J. and Lee, G. (2011). Algorithm for efficiently extracting ifc building elements from an ifc building model. pages 713–719.

Xu, Z., Huang, T., Li, B., Li, H., and Li, Q. (2018). Developing an ifc-based database for construction quality evaluation. Advances in Civil Engineering, 2018:1–22.

Zheng, Y., Törmä, S., and Seppänen, O. (2021). A shared ontology suite for digital construction workflow. Automation in Construction, 132:1–24.

Zhong, Luo, Hu, and Sun (2012). Ontology-based approach for automated quality compliance checking against regulation in metro construction project: China. In Ni, Y.-Q. and Ye, X.-W., editors, Proceedings of the 1st International Workshop on High-Speed and Intercity Railways, Lecture Notes in Electrical Engineering, 148, pages 385–396. Springer-Verlag, s.l.

Zhou, Z., Goh, Y. M., and Shen, L. (2016). Overview and analysis of ontology studies supporting development of the construction industry. Journal of Computing in Civil Engineering, 30(6):04016026.