Automation in Construction 28 (2012) 58–70

Contents lists available at SciVerse ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking

B.T. Zhong, L.Y. Ding *, H.B. Luo, Y. Zhou, Y.Z. Hu, H.M. Hu

Institution of Construction Engineering, Huazhong Univ. of Science and Technology, Wuhan 430074, P.R. China
Hubei Key Laboratory of Control Structure, Huazhong Univ. of Science and Technology, Wuhan 430074, P.R. China

ARTICLE INFO

Article history:
Accepted 26 June 2012
Available online 4 August 2012

Keywords:
Construction quality
Compliance checking
Regulation constraint
Ontology
Semantic modeling

ABSTRACT

Regulations play an important role in assuring the construction quality. However, due to the large amount of regulation needs considered, the construction quality compliance checking against regulations can be cumbersome/time-consuming and error prone. In order to give more computerized support to the construction quality compliance checking against regulations, an ontology-based semantic modeling approach of regulation constraints is explored. A meta model for construction quality inspection and evaluation i.e. CQIEOntology is proposed in this paper. Based on CQIEontology, the regulation constraints are modeled into OWL axioms and SWRL rules. By these OWL axioms and SWRL rules, the regulation provisions imposed on construction quality inspection can be translated into a set of inspection tasks, and get associated with the specific construction tasks. Once the construction starts, the applicable inspection tasks, including a series of quality checking and evaluation, will be reminded and recommended. Obviously, the proposed approach facilitates taking the construction quality compliance checking as a paralleling activity to the construction rather than an afterthought, and helps the inspectors in quality inspection. Finally, the approach is demonstrated in Protégé 3.4.6 through case studies based on regulation examples taken from "Code for Acceptance of Construction Quality of Building Foundation (GB50202-2002) and the validation and discussion are given for it."

1. Introduction

The construction phase of a project is governed by lots of regulations. It is important to inspect the construction process according to regulations to assure the construction quality, which is called regulation-based construction quality compliance checking or inspection. The quality inspection involves many regulations, however, due to the reasons, such as unfamiliarity with or even lack of the regulation knowledge, or being overwhelmed of the amount of regulation text, from which the applicable regulation provisions should be referenced (this is a common case especially to the new hand), during quality inspection, the inspectors may do their quality decision-making relying on the their own experience, such as which construction activities should not be neglected, which construction tasks should be inspected, what quality data need to be collected and checked, what are the quality acceptance criteria, and so on, which may vary from inspector to inspector. Therefore, manual construction quality compliance checking has been a time-consuming and error-prone task [11,14,15]. Regulation-based automated construction quality compliance checking would reduce quality inspection errors, consequently improve quality compliance and reduce violations to the regulations that govern the construction process.

In terms of construction quality compliance checking, to inspect the whole construction process is more important than only to inspect the outcome of the process. Therefore, how to support the inspectors to inspect the whole construction process under the guidance of the regulation knowledge becomes the starting point of this paper.

Nowadays, in order to improve the efficiency and reduce the cost of compliance checking, the ontology and semantic web technology have been applied to model building of code-related knowledge and implement compliance checking in the construction industry (the related research review is given in the following Section 2). Most of the researches are mainly in the architectural and structural design domains, also, some efforts have been put into the construction domain.

In order to make the construction quality compliance checking an easier and more efficient process for the inspectors, an ontology-based semantic modeling of regulation constraints is explored. A meta model for construction quality inspection and evaluation i.e. CQIEOntology is proposed, based on which, regulation constraints can be modeled into OWL axioms and SWRL rules. The proposed approach will facilitate integrating the regulations with the construction process and improve the construction quality inspection during the construction stage, from the definition and execution of the construction process to construction quality acceptance evaluation, by enabling treating the construction

* Corresponding author at: Institution of Construction Engineering, Huazhong Univ. of Science and Technology, Wuhan 430074, P.R. China.
E-mail address: dingly_wuhan@yahoo.com.cn (L.Y. Ding).

0926-5805/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.autcon.2012.06.006

quality compliance checking as a parallel activity to the construction, rather than an afterthought.

The remainder of this paper is organized as follows. The research background is sketched in Section 1. Section 2 provides a brief review of related work. A construction quality inspection and evaluation ontology i.e. CQIEOntology is proposed in Section 3. Section 4, based on CQIEOntology , discusses how to model the relevant regulation constraints into OWL axioms and SWRL rules to support the automated construction quality compliance checking during the construction stage, from the definition and execution of the construction process to construction quality acceptance check. In Section 5, the proposed approach is illustrated with regulation examples, and the validation is discussed. Finally, the conclusion is drawn in Section 6.

2. Related work

2.1. Computer-based system for construction quality management and inspection

Quality management and inspection is a complex process. Significant work has been done in computer-based quality management and inspection system in construction. Battikha, M.G. [8] developed the computer-based system for construction quality management (QUALICON), which can deal with a diverse range of information related to requirements/criteria, inspections/tests, actual results, inspection and test plans, etc. and can integrate them with the project physical aspect. Young et al. [9] proposed a computerized Quality Inspection and Defect Management System (QIDMS), which can collect defect data at a site in real time using Personal Digital Assistant (PDA) and wireless internet, and effectively manage statuses and results of the corrective works performed.

These researches play an important role in the way to automated quality inspection. However, these systems are developed to facilitate the quality data collection, storage, and comparison between these data and the standard data extracted from the regulation. They are not developed as an assistance tool for regulatory compliance checking. In addition, to assure the construction products' high quality, it is very important to monitor the quality of (semi-) finished products as well as the construction activities and procedures. Nevertheless, these systems only focus on the final quality data checking rather than the construction process.

2.2. Regulation knowledge modeling and compliance checking

The study of regulation knowledge modeling and automated compliance checking in construction has received much attention from the academic community and industry over years. As Demir et al. pointed out, most regulations are presented as texts, whether paper or electronic, in order to give more computerized support to the users of regulations, it is necessary to represent the regulation knowledge in a formal and computer-interpretable way [12].

The review of regulation knowledge modeling and automatic rule-based checking (although dedicated in building design domain) can be obtained in refs. [12–14].

Previous research efforts mainly focused on the procedural implementation approach. Recently, attention has been directed towards the study of rule-based approach. Many researches, such as CORENET project [16], embeds the logic within the programming code, using parameterization and branching (namely, rules are hard coded in computer programming language). It requires high-level expertise to define, write and maintain the codes. It is time-consuming to update the computer programming code, because the building codes tend to change frequently by their nature. In addition, rules written in computer codes can be used only in dedicated applications, they are not likely to support widespread use. Evidently, rules, which reflect technological constraints, national regulations, etc., are an intuitive way of implementing the logics in building codes. However, in many rule-driven approaches, the actual implementation method is seldom based on a logic theory [23].

Ontology is defined as the conceptualization of terms and relations in a domain [1]. Ontology and semantic web technology offer a means to structurally represent and reuse domain knowledge [2–5]. Ontology and semantic web technology have been applied to model building code-related knowledge and implement compliance checking for the construction industry. Kim and Grobler [17] employed an ontology reasoning mechanism to detect conflicts between diverging participants' requirements in collaborative design scenarios. Yurchyshyna et al. [24] conducted the research in which the norms are extracted from the electronic regulations and formalized as SPARQL queries in terms of the IFC model. The compliance checking process is based on matching an RDF representation of a project to a SPARQL conformity query. Pauwels et al. [23] explored how to establish a semantic rule checking environment for building performance checking, the N3-logic was selected as rule language to represent the logic in regulations. Han-Hsiang et al. [28] use ontologies to structure the knowledge about activities, job steps, and hazards, for improving access to a company's JHA knowledge, and discuss an ontological reasoning mechanism for identifying safety rules applicable to given activities. These researches provide useful declarative implementation formalism.

Some researches focus on integrating building regulations with BIM/IFC and making the automated checking against constraints [21]. Ding et al. [21] implemented the Australian disabled access code on the basis of IFC models. These approaches are based on IFC (Industry Foundation Classes). The International Code Council (ICC) has created the SmartCodes initiative in this direction (Nisbet et al. 2009). These constraints are applied only to the IFC schema level; it is hard to set constraints on the instance level so as to affect only specific objects. Nguyen [18] and H.M. Satti et al. [19] integrated building code compliance checking into CAD System. However, it is not economically viable for the major CAD vendors to develop multiple local flavors of their product, since the constraints come from the national, provincial, local governments or the corresponding industry administration.

Undoubtedly, these current research efforts have paved the way for automated compliance checking in construction industry. However, from the literature review given above, it can be concluded that most of the research efforts mainly focus on the architectural and structural design domains while little efforts have been put into the construction domain. In addition, the existing automated compliance checking efforts lay more emphasis on the constraints such as wall thickness, the overlaps and intersections of building components and so on, namely, these constraints usually are about the quality requirements for the (semi-) finished products. Nevertheless, the regulation knowledge not only includes the quality requirements for the (semi-) finished products, but also includes the requirements/constraints about the construction activities and procedures of the (semi-) finished products, and the constraints about quality inspection tasks.

The existing research efforts closely related to the construction quality compliance checking in construction industry are made by Jirapon Sunkpho, Garrett, J. H., et al. (2001, 2005), Boukamp, F., Akinci, B., et al. (2007) and [12], from Carnegie Mellon University. Jirapon Sunkpho, J.H. Garrett Jr., et al. [20,22] (2001, 2005) developed a Java-based Inspection Framework (JIF), in which, an XML-based inspection modeling for developing field inspection support systems is proposed by capturing the common components and their relationships found in several field inspection support systems. Their study has given us much inspiration. The obvious differences lie in that, in our paper, the model is for the regulation constraint knowledge representation and reasoning, and used in ontological and semantic approach. Akinci et al. [10] provided an approach for automating the processing of construction specifications to support inspection and quality control tasks in construction projects. Demir et al. [12] proposed a semantic web-based approach for representing and reasoning with vocabulary for computer-based standards processing. Their research efforts have paved the way for construction regulation modeling in construction industry. The proposed approach in this paper is based on the ontology and semantic technology, which is different from their object-oriented specification modeling. Moreover, the starting point and technology roadmap of this research are also different from theirs.

3. The quality inspection and evaluation ontology (CQIEOntology) for compliance checking

Construction quality must meet the relevant quality constraints or requirements. These quality constraints come from the regulations (including national, provincial, local or from the corresponding industrial administrative departments), customer requirements, experts' experience, engineering practice or work environment, etc. Here, all these constraints are called as regulation constraints. In this paper, the quality specification "Code for Acceptance of Construction Quality of Building Foundation (GB50202-2002) [6]" is selected as an example for illustration.

The quality regulation GB50202-2002 defines objectives that have to be fulfilled during construction to assure the construction quality. These objectives are guaranteed by a set of supporting quality requirements about the resource used, the construction process (the activities and procedures), the (semi-) finished products, the inspection task and the inspection responsibility, etc.

In this paper, the following two types of regulation constraints will be mainly focused on: the quality evaluation criteria about the (semi-) finished products and the deontic constraints about the construction and inspection process.

The quality evaluation criteria define the quality objectives that the construction objects need to achieve. For example, the constraint "The permissible variation of the underground-diaphragm-wall trench's verticality degree should be less than 1/300 (or 1/150), if it acts as a permanent structure (or makeshift structure)" (extracted from the provision 7.6.12 in GB 50202–2002)".

The deontic constraints correspond to the supporting quality requirements mentioned above and regulate the quality assuring measures for achieving the quality objectives. Obviously, it is important to effectively enforce the deontic constraints so as to assure that the construction quality meets the quality evaluation criteria. Furthermore, the deontic constraints are categorized into two subtypes of constraints: the construction process constraints and the inspection task constraints.

The construction process plays a role in guiding the construction activities. The construction process constraints regulate which construction activities should (or not) be done or what procedures should be complied by. For example, during the construction of the underground-diaphragm-wall, "the joints must be cleaned before the pouring of concrete (extracted from the provision 7.6.4 in GB 50202–2002)". However, some construction activities are often neglected while some construction procedures are often violated during construction. This is why these construction process constraints are included in the regulation documents as provisions.

During construction, the proper implementation of inspection tasks is critical for assuring the construction process quality. The inspection task constraints regulate when and which inspecting task to take during construction. For example, "In the process of underground-diaphragm-wall construction, it is necessary to examine the verticality of finished trench… slump of commercial concrete, extracting time and speed of locking-in pipe or collar box (extracted from the provision 7.6.7 in GB 50202–2002)".

Now, in order to improve the efficiency of compliance checking and reduce its cost, ontology and semantic web technology have been applied to modeling of code-related knowledge and implementation of compliance checking in the construction industry (the related research review is given in the following Section 2). Our research focuses on using these technologies for semantic modeling of regulation knowledge, facilitating their integration with the definition and execution of construction processes and making the construction quality compliance checking act as a paralleling activity to the construction, rather than an afterthought.

In terms of the quality inspection, typical constraints in the regulation can be considered as being composed of the set of rules. In order to enable the automated assessment of each rule, the rule and all terms used in defining it, should be capable of being correctly "understood" and interpreted by the machine. Ontology, as formal and explicit specification of a shared conceptualization [1], provides the sharing understanding foundation. Ontological modeling is a proper approach to model regulation constraint knowledge. First, regulation concepts and their semantic relationships can be easily represented in the form of classes and properties of ontology. Second, just as Han-Hsiang et al. [28],[29] use the ontology concepts to model the construction context and the applicability safety rules, the regulation concepts and the semantic relations can imply the applicability or inapplicability of one rule. The ontology-based semantic representation of the regulation constraint knowledge will help to improve the precision, completeness and correctness of the regulation knowledge retrieval, since the ontology-based matchmaker performs semantic reasoning, referring to ontology, instead of syntax matching.

In order to make the ontology knowledge understandable to the machine, the ontology knowledge needs being described in ontology web language. In this paper, the semantic web technology was chosen to represent construction quality information, for following reasons:

One is that, as pointed out by Boukamp and Akinci [11], some construction quality information for quality inspection currently is not and/or cannot be represented in project models, even though the building information modeling (BIM) is one of the most notable efforts in recent years regarding project information management in the AEC industry. Another is that the semantic web technology allows one to easily describe and link information from the different knowledge domains together. This feature facilitates the construction quality information representation and management. In addition, semantic web technology is based on logic theory, and there are several rule languages, e.g. SWRL [7], RIF [31] and N3Logic [32], available in the semantic web domain to express logic into explicit rules. These rule languages enable the definition of the regulations without the need to write an explicit, procedural programming code.

In this section, a meta model for construction quality inspection and evaluation i.e. CQIEOntology is proposed, as shown in Fig. 1. Based on CQIEOntology, together with the support of the construction process ontology, the regulation constraints can be modeled in OWL axioms and SWRL rules for construction quality inspection and evaluation.

This meta model is built partially under the guidance of the Java Inspection Framework (JIF) as described by Jirapon Sunkpho, J.H. Garrett Jr., et al. [22] (2001). In JIF, an inspection application is composed of a collection of tasks and each task is composed of a set of inspection domain objects, including inspection data, inspection elements, inspection knowledge, and instruments, which indicate, respectively, what is to be collected, what is to be inspected, what is known about a particular inspection task, and what tools are used in inspection. In CQIEOntology, the similar concepts and relations are defined to model the respective facet in JIF.

CQIEOntology serves as a meta model, defining the generic terms and relations related to the construction quality compliance checking.

According to the structure of CQIEOntology, the *Inspection-Task* class is the heart of the CQIEOntology. An *Inspection-Task* is set according to the specific deontic regulation constraint.

An *Inspection-Task* can be related to the *Inspection-Object* through the "*hasInspectionObject*" property, which indicates that the *Inspection-Object* will be inspected to make sure their compliance to the relevant regulation constraints through the execution of the *Inspection-Task*.

The *Inspection-Object* refers to any entities governed by regulations and indicates what is to be inspected, in the case of construction

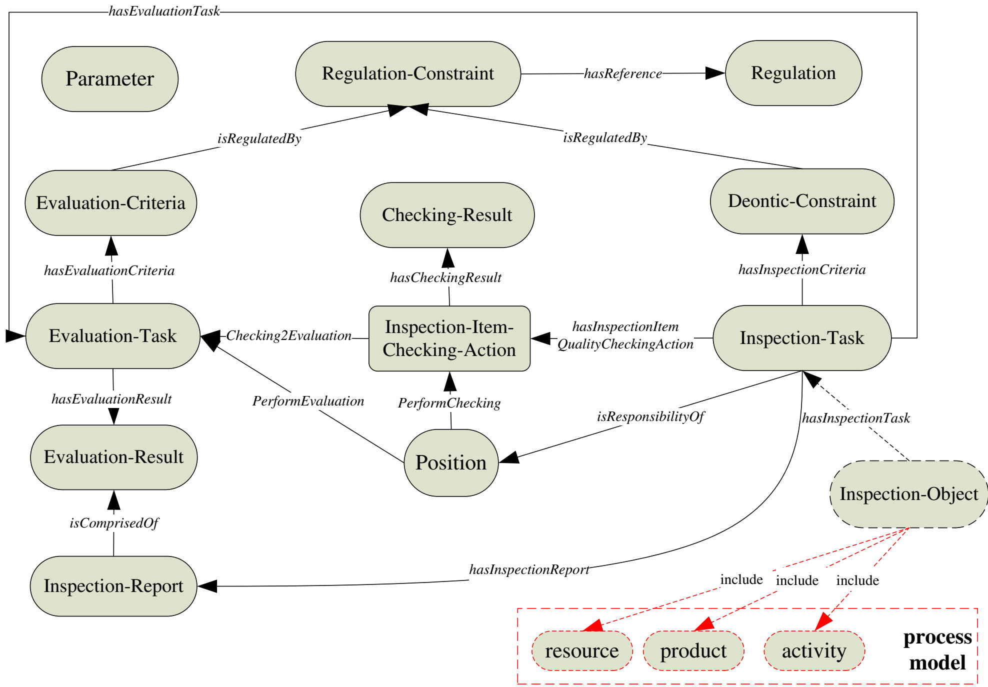

Fig. 1. The CQIEOntology for construction quality compliance checking.

quality inspection domain, the entities include construction process (the activities and the procedures), the construction products, and resources used in construction.

An *Inspection-Object* may include a set of quality inspection items. These inspection items can be identified from the regulation provisions. For example, the provision 7.6.7 in GB 50202–2002, "In the process of underground-diaphragm-wall construction, it is necessary to examine the verticality of finished trench… slump of commercial concrete, extracting time and speed of locking-in pipe or collar box", the inspection items include the verticality of finished trench, … slump of commercial concrete, extracting time and speed of locking-in pipe, and so on.

Furthermore, an *Inspection-Task* needs a set of *Inspection-Item-Checking-Action* to test and collect the quality information/data for the inspection items. Each *Inspection-Item-Checking-Action* has a *Checking-Result*, which represents the actual construction quality information collected.

Similarly, an *Inspection-Task* needs a set of *Evaluation-Task* to evaluate the quality of those inspection items in accordance with the *Evaluation-Criteria*. The *Evaluation-Criteria* is imposed by the regulation provisions or set by the domain experts.

Basing on the *Checking-Result* and the *Evaluation-Criteria*, the *Evaluation-Task* can be done to judge whether the inspection items are compliant with the regulation constraints. Each *Evaluation-Task* has an *Evaluation-Result*, which all together are constituted the *Inspection-Report*. The *Inspection-Report* of a particular *Inspection-Task* for the corresponding *Inspection-Object* can be documented, based on the *Evaluation-Result* of all the inspection items.

In CQIEOntology, the *Regulation-Constraint* constitutes the main the inspection knowledge, since the focus is the regulation-based quality inspection. Each constraint comes from the corresponding provision text in regulations. The relation "*hasRegulation*" associates the constraint with the provision text from which the constraint is extracted.

Meanwhile, an *Inspection-Task* must be assigned to a *Position* as its responsibility, who performs the *Inspection-Item-Checking-Action* and the *Evaluation-Task* to accomplish the *Inspection-Task*.

In addition, many parameters, such as engineering parameters, geometric parameters and so on, are used to depict the quality features/situation, in the quality inspection domain. These parameters mainly include the measurement concepts such as concrete slump, temperature, concentration, etc., and the measurement units such as milliliter, centigrade degree, etc.

As shown in Fig. 1, the *Inspection-Object* can be the resource, building product or construction activity and so on. Here, each main concept indicates one facet of the inspection objects, and can be modeled as the construction process ontology. Now, some efforts are made for construction process modeling. N.M. El Gohary et al. [25] (2010) proposed the Domain Ontology for Processes in Infrastructure and Construction called "IC-PRO-Onto". Katranuschkov et al. [26] presented the idea of reusing process knowledge through conceptual process patterns. Benevolenskiy et al. [27] proposed an ontological methodology for the configuration of construction process, using process patterns. According to these researches, the concrete construction process for a specific construction task can be represented as an instantiation of the construction process ontology. In CQIEOntology, the *Inspection-Object* concepts (enveloped with the dashed line, as shown in Fig. 1) are also the concepts of the construction process model. In order to reuse the existing research results, the structure and definition of these concepts remain the same as the original concepts defined in the construction process model. Through the *Inspection-Object* concept, the CQIEOntology for compliance checking can interact with the construction process model. In this paper, the aim is mainly to model the regulation constraints, and the development of construction process ontology is beyond the scope of this paper.

The meta mode provides general and common terms and relations common to the construction quality compliance checking domain. Basing on the meta model, the specific domain model for the construction quality inspection can be obtained via specializing and instantiating the generic concepts and relations in the meta model. Since the meta-model is not limited to any specific construction domain, the meta-model can be reused independently of concrete construction object domains. Basing on the meta model and the ontology, the constraints knowledge imposed by the regulations can be clearly and unambiguously defined such that they may potentially be interpreted by a machine.

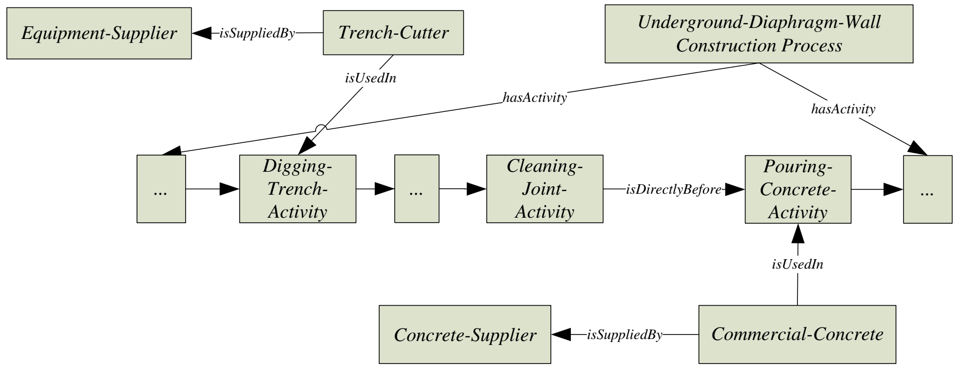

Fig. 2. Part of process flow of the underground-diaphragm-wall construction.

4. Ontology-based regulation constraint semantic modeling and application in construction quality compliance checking

Basing on the CQIEOntology meta model, along with above-mentioned ontology, the real construction information can be represented as the ontology instances in OWL, and the regulation constraints can be modeled into the OWL axioms and SWRL rules. Since the construction quality information and the regulation constraint knowledge are all based on logic theory, we can implement semantic reasoning to support the construction quality compliance checking during the whole construction stage, from the construction process definition, execution, to the final quality acceptance and evaluation.

4.1. An illustrative example

Here, the underground-diaphragm-wall construction in the building foundation pit engineering is selected as an example to demonstrate. A part of process flow of the *underground-diaphragm-wall* construction in RDF is given in Fig. 2.

Based on CQIEOntology and the construction process, each construction quality inspection task can be modeled as an ontology instance. Fig. 3 shows the CQIEOntology instance for *Underground-Diaphragm-Wall* construction quality compliance checking.

In order to make the ontology knowledge understandable to both machines and human beings, the ontology knowledge is described in OWL. OWL is a W3C recommended language for ontology representation on the semantic web. It offers a relatively high level of expressivity while still being decidable. In addition, OWL, as a formal language with description logic based semantics, enables automatic reasoning about inconsistencies of concepts, and provides RDF/XML syntax to represent ontology knowledge.

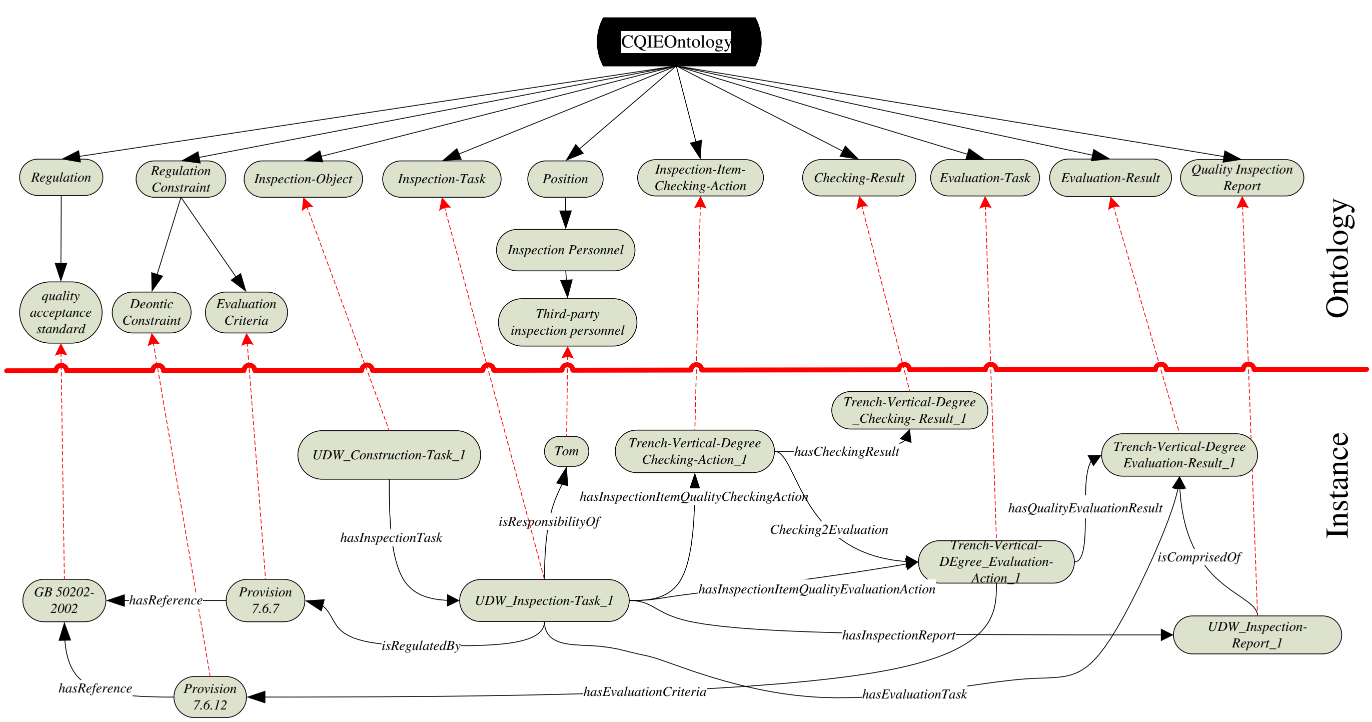

Fig. 3. Underground-Diaphragm-Wall construction quality compliance checking instance.

Table 1
OWL built-in constructs.

OWL built-in constructs	The meaning
rdfs:subClassOf (A, B)	Class A is a subclass of class B.
rdfs:subPropertyOf (A, B)	Property A is a sub-property of property B.
OWL: allValueFrom (C)	All values of property P belong to class C.
OWL: someValueFrom (C)	Some value of property P belongs to class C.
OWL: miniCardinality (n)	The number of values that property P can take must be greater than or equal to n.
OWL: maxCardinality (n)	The number of values that property P can take must not exceed n.
OWL: Cardinality (n)	The number of values that property P takes is exactly n.

For example, in the quality inspection domain, there are three types of quality inspection: 1) site office inspection, 2) third-party inspection, and, 3) customer-oriented inspection. Respectively, there exist three types of inspection personnel: self-inspection personnel, third-party inspection personnel and customer-inspection personnel. These concepts can be encoded in the following RDF/XML fragment.

```
<owl: Class rdf: about="# third-party inspection personnel">
    <rdfs: subClassOf rdf: resource="# inspection personnel"/>
    <owl: disjointWith rdf: resource="# self-inspection personnel"/>
    <owl: disjointWith rdf: resource="# customer inspection personnel"/>
</owl: Class>
```

The fragment shows that inspection personnel are classified into three types, and they are disjoint with each other. The third-party inspection personnel is the subclass of inspection personnel class through the use of OWL construct rdfs: subClassOf.

Similarly, both object properties and data properties are also encoded in OWL RDF/XML syntax. For example, the relation *hasInspectionItemQualityCheckingAction* is written in OWL RDF/XML syntax:

```
<owl:ObjectProperty rdf:about="# hasInspectionItemQualityCheckingAction">
    <rdfs:domain rdf:resource="# Inspection-Task"/>
    <rdfs: range rdf:resource="# Inspection-Item-Checking-Action"/>
    </owl:ObjectProperty>
```

It indicates that *hasInspectionItemQualityCheckingAction* is an object property and the domain and range are restricted to the class *Inspection-Task* and *Inspection-Item-Checking-Action*, respectively.

In addition, OWL provides some useful built-in constructs for restrictions representation such as existential, universal and cardinality restrictions, as shown in Table 1.

Fig. 4 shows the part of RDF/XML file that encodes the existential restriction: one *Inspection-Task* has at least one *Inspection-Item-Checking-Action*. At the same time, this restriction can also be represented in the description logic axioms, as shown in Axiom A1-2.

Axiom A1. Inspection-Task hasInspectionItemQualityCheckingAction only Inspection-Item-Checking-Action

```
<owl:Class rdf:ID="Inspection_Task">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:allValuesFrom>
          <owl:Class rdf:ID=" Inspection-Item-Checking-Action"/>
        </owl:allValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="hasInspectionItemQualityCheckingAction"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int" >1</owl:minCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#hasInspectionItemQualityCheckingAction"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
```

Fig. 4. Encoding classes in OWL RDF/XML.

Axiom A2. Inspection-Task hasInspectionItemQualityCheckingAction min 1

In the following section, for convenience of illustration, all restrictions are represented in description logic axioms.

4.2. Regulation constraint semantic modeling and application

The regulation constraints can be extracted from the regulation provisions. The semantics of concepts and a set of linguistic clues such as deontic verbs, “must, should\shall, ought to, is obliged to, is subject to” often used in regulation provisions are employed. Here, typical provisions are selected as examples for the aforementioned three types of constraints.

4.2.1. Construction process constraint modeling

Some constraints, imposed on the construction process by regulations, can be modeled in axioms. Through these axioms, the definition of the construction process can be checked, and assure that the construction process is compliant with the relevant constraints.

For example, the constraint: “the commercial concrete shall be used for the underground-diaphragm-wall, and the supplier must be qualified (from the provision 7.6.6 in GB 50202–2002)” can be modeled in the following Axiom B:

Axiom B. *Commercial-Concrete isSuppliedBy* some (*Concrete-Supplier* and (*isQualified* has *Qualified*))

Since the OWL axiom is at the class level, any instance of the class must obey it. When defining the construction process of the underground-diaphragm-wall, if there is any concrete instance of the class *Commercial-Concrete* in construction activity “*Pouring-Concrete-Activity”*, whose concrete supplier has been left empty or assigned a not-qualified supplier instance, the violation of Axiom B will be detected.

Typical constraints in the regulation occur in the form of rules. However, OWL only provides a basic, standard level of reasoning, limited to a certain level of complexity. When a more complex logical reasoning is necessary, one may need to build his or her rules in a more dedicated rule language. Several rule languages, such as Semantic Web Rule Language (SWRL), the Rule Interchange Format (RIF) and the N3Logic language, have been developed to express such logic. In our approach, SWRL is selected to represent the constraints rules, since SWRL rule language is tightly integrated with OWL and the predicates in SWRL rules may be OWL-based classes or properties. The use of SWRL rules along with OWL axioms results in more powerful constraints and intuitive inferring capability, which could not be achieved through the use of axioms alone.

In addition, the SWRL rule representation enables that the quality inspection information is separated from regulation constraint knowledge, and provides the level of flexibility needed so that the user can add or modify the set of governing rules and regulations. This feature is useful, since the regulation often changes. Another benefit is that SWRL is a descriptive language that is independent of any rule language internal to rule engines, which decouples the rules from the technical implementation of the rules engine.

For example, the constraint “the joints must be cleaned before the pouring of concrete” (extracted from the provision 7.6.4 in GB 50202–2002), can be modeled in Axiom C1 & C2 and SWRL *Rule 1*:

Rule 1. Pouring-Concrete-Activity(?CT_pc) ∧ Cleaning-Joint-Activity (?CT_cj) → isDirectlyBefore(?CT_pc, ?CT_cj)

Axiom C1. Pouring-Concrete-Activity isDirectlyBefore only Cleaning- Joint-Activity

Axiom C2. Pouring-Concrete-Activity isDirectlyBefore exactly 1

Here, *Rule 1* is written in terms/concepts from the construction process model. *Rule 1* indicates that the existence of one instance of the *Pouring-Concrete-Activity* implies the existence of one corresponding instance of the *Cleaning- Joint-Activity*, and the constraints represented by object property *isDirectlyBefore* should be met.

In the construction process definition, people may mistakenly configure the instance of one construction process (for convenience, called construction process A) with the instance of another construction process (for convenience, called construction process B). In the instance level, the computer will think that the relationship *isDirectlyBefore*, between the instance *Ins_Pouring-Concrete-Activity_a* of class *Pouring-Concrete-Activity* and the instance *Ins_Cleaning-Joint-Activity_b* of class *Cleaning- Joint-Activity*, is reasonable. Obviously, this is not logical, as shown in Fig. 5. Given that two construction processes A and B occur at the same time, the property *isDirectlyBefore* will be followed with two instances: *Ins_Cleaning-Joint-Activity_a* and *Ins_Cleaning-Joint-Activity_b*. However, this will violate Axiom C1 & C2. Therefore, we can preclude the mismatch.

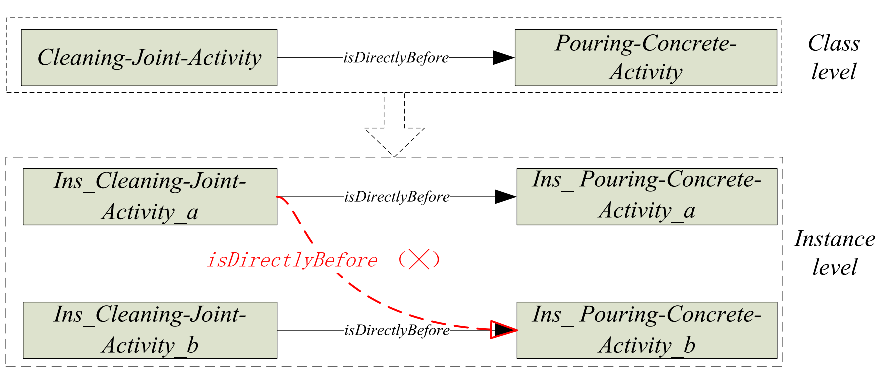

Fig. 5. *isDirectlyBefore* on class/ instance level.

As a result, these constraints can be modeled to guide the definition of the construction process and assure its compliance with the regulation constraints.

4.2.2. Inspection task constraint modeling

During construction stage, namely, during the execution of the construction process, it is necessary to assure the corresponding inspection actions and evaluation are done, and any quality defect should be detected in time to prevent rework and cost increase.

Based on the ontology, we can divide the inspection task constraints into a set of SWRL rules. For example, the deontic provision: "In the process of construction, it is necessary to examine the verticality of finished trench… slump of commercial concrete, extracting time and speed of locking-in pipe or collar box. (extracted the provision 7.6.7 in GB 50202–2002)", can be modeled in the following two SWRL rules.

Underground-Diaphragm-Wall_Construction-Task (UDW_Construction-Task_1) → needsInspection (UDW_Construction-Task_1, true) ***(Rule 2-1)***

Underground-Diaphragm-Wall_Construction-Task (UDW_Construction-Task_1) ∧ needsInspection (UDW_Construction-Task_1, true)

→Underground-Diaphragm-Wall_Inspection-Task (UDW_Inspection-Task_1) ∧ hasInspectionTask (UDW_Construction-Task_1, UDW_Inspection-Task_1) ***(Rule 2-2)***

Rule 2-1 indicates that once the construction of the underground diaphragm wall starts, the inspector will be reminded that the *UDW_Construction-Task_1* should be inspected.

When executing *Rule 2-2*, the instance *UDW_Inspection-Task_1* is assigned to the class *Underground-Diaphragm-Wall_Construction-Task via the property hasInspectionTask*.

Similarly, the constraint(s) "the slump of concrete used should be checked" can be modeled in *Rule 3-1/2, which* indicates that the inspection task for *Underground Diaphragm Wall* includes the checking action of the inspection item.

Underground-Diaphragm-Wall_Inspection-Task(UDW_ Inspection-Task_1) → Inspection-Item-Checking-Action

(UDW_Con-Slu_Checking-Action_1) ∧ hasInspectionItemQualityCheckingAction(UDW_ Inspection-Task_1,

UDW_Con-Slu_Checking-Action_1) ***(Rule 3–1)***

Underground-Diaphragm-Wall_Inspection-Task(UDW_ Inspection-Task_1) → Evaluation- Task (UDW_Con-Slu_Evaluation-Action_1) ∧

hasInspectionItemQualityEvaluationAction (UDW_Inspection-Task_1, UDW_Con-Slu_Evaluation-Action_1) ***(Rule 3–2)***

Furthermore, based on *Rule 4-1/2*, the checking action and evaluation for some specific quality inspection items, which comprise the inspection task, can be assigned to the corresponding construction task.

Construction-Task (?ct) ∧ Inspection-Task(?it) ∧ hasInspectionTask(?ct, ?it) ∧ Checking-Action(?cha) ∧

hasInspectionItemQualityCheckingAction(?it, ?cha) → QualityCheck(?cha, ?ct) ***(Rule 4–1)***

Construction-Task (?ct) ∧ Inspection-Task(?it) ∧ hasInspectionTask(?ct, ?it) ∧ Evaluation-Task (?et) ∧ hasInspectionItemQualityEvaluationAction

(?it, ? et) → QualityEvaluate(?et, ?ct) ***(Rule 4–2)***

Once *Rule 4–1* is fired, the property "*QualityCheck*" of the construction task *UDW_Construction-Task_1 is* automatically deduced and filled in.

Obviously, based on these rules, the applicable quality requirements can be translated into a set of inspection tasks to be performed by inspectors. This enables the regulatory compliance checking to be a paralleling activity to the construction, rather than an afterthought. Meanwhile, it also facilitates the integration of the regulation deontic knowledge and the construction process.

4.2.3. Quality acceptance criteria constraints modeling and quality evaluation

Once the inspection task of one construction task is determined, the checking actions are performed to collect quality data. The quality data should be evaluated according to the acceptance criteria imposed by regulations so as to decide whether the inspection objects are compliant with the quality acceptance criteria constraints.

In the quality inspection, if the difference between the actual inspection results and the acceptance criteria exceeds a certain range, the inspection objects are identified as quality defects and need to be investigated or reworked.

Here, the general rules are defined as following:

Inspection-Item-Checking-Action(?cha) ∧ hasActualDeviation(?cha, ?ad) ∧ Evaluation-Criteria(?ec) ∧ hasPermissionlDeviation(?ec, ?pd) ∧

swrlb:lessThan(?ad, ?pd) ∧ Evaluation-Task(?et) ∧ hasEvaluationCriteria(?et, ?ec) ∧ Checking2Evaluation(?cha, ?et) →

hasInspectionItemQualityEvaluationResult(?et, "isAccepted ") ∧ Accepted-Entity(?cha) **(Rule 5–1)**

Inspection-Item-Checking-Action(?cha) ∧ hasActualDeviation(?cha, ?ad) ∧ Evaluation-Criteria(?ec) ∧ hasPermissionlDeviation(?ec, ?pd) ∧

swrlb:greaterThan(?ad, ?pd) ∧ Evaluation-Task(?et) ∧ hasEvaluationCriteria(?et, ?ec) ∧ Checking2Evaluation(?cha, ?ec) →

hasInspectionItemQualityEvaluationResult(?et, "isDefected ")∧ Defected-Entity(?cha) **(Rule 5–2)**

Note that the object property *hasPermissibleDeviation* is a sub-property of *hasEvaluationCriteria*. The object property *hasActualDeviation* is a sub-property of *hasCheckingResult*.

After executing *Rule 5-1/2*, the inspection items, whose *ActualDeviation* is less than *PermissibleDeviation*, will be classified into the *Accepted-Entity*, which means that the inspection quality satisfies the requirements. Otherwise, the quality inspection items will be classified as *Defected-Entity*, which means that further measures (investigation or rework) need be taken.

5. Implementation of construction quality compliance checking

5.1. Implementing soft environment

The implementing soft environment is shown in Fig. 6.

Protégé-OWL v3.4.6 [30] enables the users to load and save OWL and RDF ontologies, edits and visualizes classes, properties, and SWRL rules, defines logical class characteristics as OWL expressions, executes reasoners such as description logic classifiers and edits OWL individuals.

Pellet v.1.5.2 is selected as the reasoner due to its convenient interface with Java, and more importantly, its axiom pinpointing feature that tracks the exact source axioms and instances in the case of a logical inconsistency. Furthermore, it provides all the standard inference services that are traditionally provided by DL reasoners: consistency checking, classification and realization.

In this research, actual reasoning process is conducted through the JESS rule engine. The JESS rule engine converts a combination of OWL + SWRL into jess facts (i.e. new facts). The inferences are carried out in JESS inference engine by matching facts in working memories in accordance with the rules in rule base. Also, if the inference engine infers knowledge using forward chaining, the new knowledge can be used for further inference or querying stored or inferred knowledge.

The modeling process for ontology knowledge is facilitated using Protégé developed by Stanford University, which provides an environment of creating, editing and saving ontologies in a visual way. Protégé also offers the support of modeling ontologies in the OWL. It is very convenient to create the sub-classes of a class and to specify restrictions on class properties using the protégé. As shown in Fig. 7, the classes, properties and their restrictions are defined in Protégé 3.4.6; all constraint rules in regulations are modeled using the SWRL rules editor, namely SWRLTab, a plug-in and editor integrated in Protégé.

Based on the definition of classes and properties, Protégé automatically generates graphical user interface (GUI) forms that can be used to create instances of classes.

Thus, the construction quality inspection data can be entered, using these Protégé generated forms, as shown in Fig. 8.

5.2. Implementation of the construction quality compliance checking based on the rule engine

5.2.1. Translating the OWL axioms and SWRL rules into JESS

Since SWRL is a descriptive language that is independent of any rule language internal to rule engines, OWL and SWRL-based regulation knowledge is required to be transformed into the rules expressed in the rule language of some rule engine.

This can be done by using the "SWRLJessTab" plug-in for Protégé-OWL that supports the interpretation and execution of SWRL rules using the Jess rule engine.

By transforming the ontology knowledge in OWL into JESS facts, and constraint knowledge in SWRL into JESS rules, actual quality inspection is implemented by JESS inference engine. Details about this transformation are described below.

The classes in OWL knowledge base are mapped onto the JESS templates that are used to define the classes and hierarchy of classes, similar to an object-oriented way.

For example, among *Quality-Inspection-Domain, Inspection-Item-Checking-Action* and *Underground-Diaphragm-Wall_Inspection-Item_*

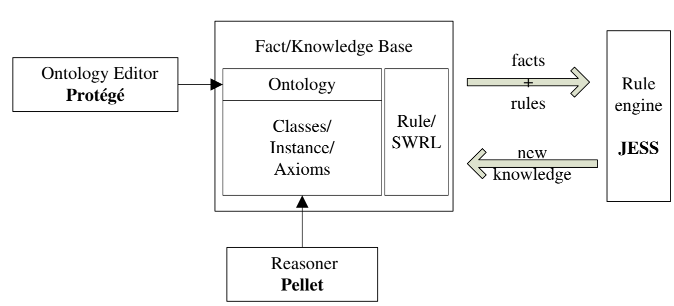

Fig. 6. Implementing soft environment in Protégé.

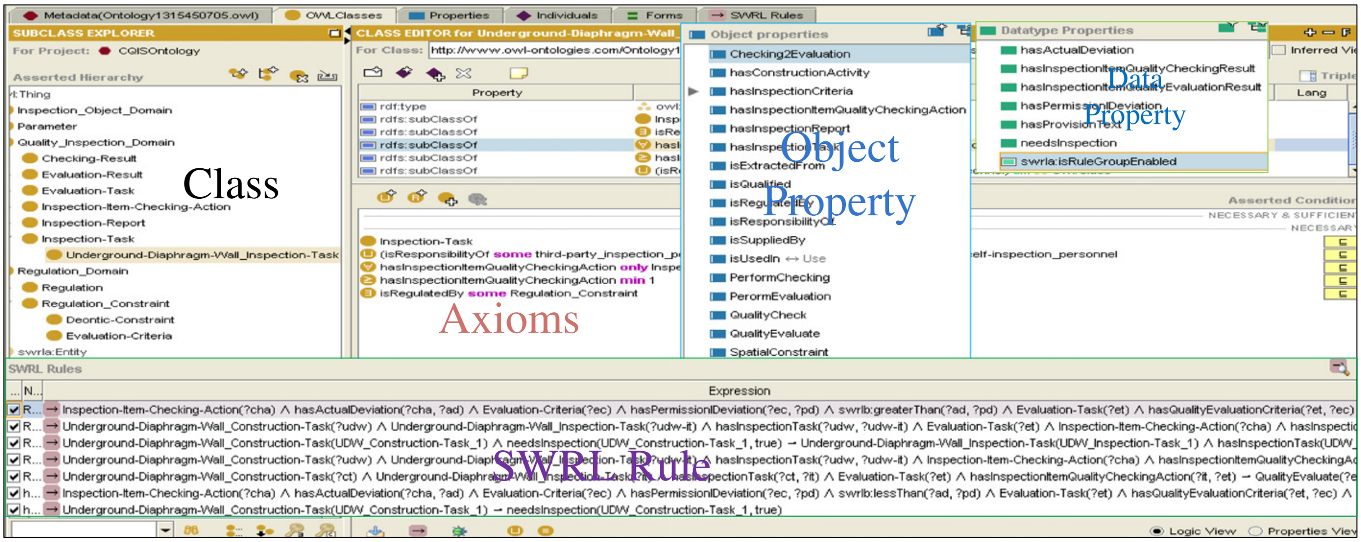

Fig. 7. Ontology in Protégé-OWL 3.4.6 screenshot.

Checking-Action, some parent–child relationships are built and transformed into the following JESS templates.

(deftemplate owl:Thing(slot name))

(deftemplate Quality-Inspection-Domain extends owl:Thing)

(deftemplate Inspection-Item-Checking-Action extends Quality-Inspection-Domain)

(deftemplate Underground-Diaphragm-Wall_Inspection-Item_Checking-Action extends Inspection-Item-Checking-Action)

Here, the *deftemplate* construct in JESS is employed to define the type of *slots* in a fact. The *extends* construct indicates the inheritance relationship between two templates defining facts. OWL:Thing is a most top level class in OWL from which all other classes are directly or indirectly subclassified.

The instances of the OWL classes, which have been declared as facts and modeled in OWL-Protégé, should be transformed into JESS facts. For example the fact *UDW_Inspection-Task_1* is an instance of the *Inspection-Task* class, thus it is transformed into the JESS fact.

(assert (Inspection-Task (name UDW_Inspection-Task_1)))

Here the assert *construct* in JESS is used to declare a fact. The declared facts will be saved in the JESS fact base.

Furthermore, some facts representing relationships between individuals or between individuals and data values, namely object properties or data properties in OWL, also need to be transformed into JESS facts. For instance, the fact "the instance *UDW_Inspection-Task_1* has the instance *Trench-Vertical-Degree_Checking-Action_1 as its* inspection item quality checking action" is transformed into the JESS fact.

(assert (hasInspectionItemQualityCheckingAction UDW_Inspection-Task_1 Trench-Vertical-Degree_Checking-Action_1))

The SWRL-rules are transformed into JESS rules by the SWRL2JESS transformer. For example, *Rule 2–2* shown in Section 5.2.2 is transformed into the JESS rule, as described below.

(defrule Rule 2–2 (needsInspection UDW_Construction-Task_1 "true") (Underground-Diaphragm-Wall_Construction-Task (name UDW_Construction-Task_1)) =>(assert (hasInspectionTask UDW_Construction-Task_1 UDW_Inspection-Task_1)) (assertOWLProperty hasInspectionTask UDW_Construction-Task_1 UDW_Inspection-Task_1))

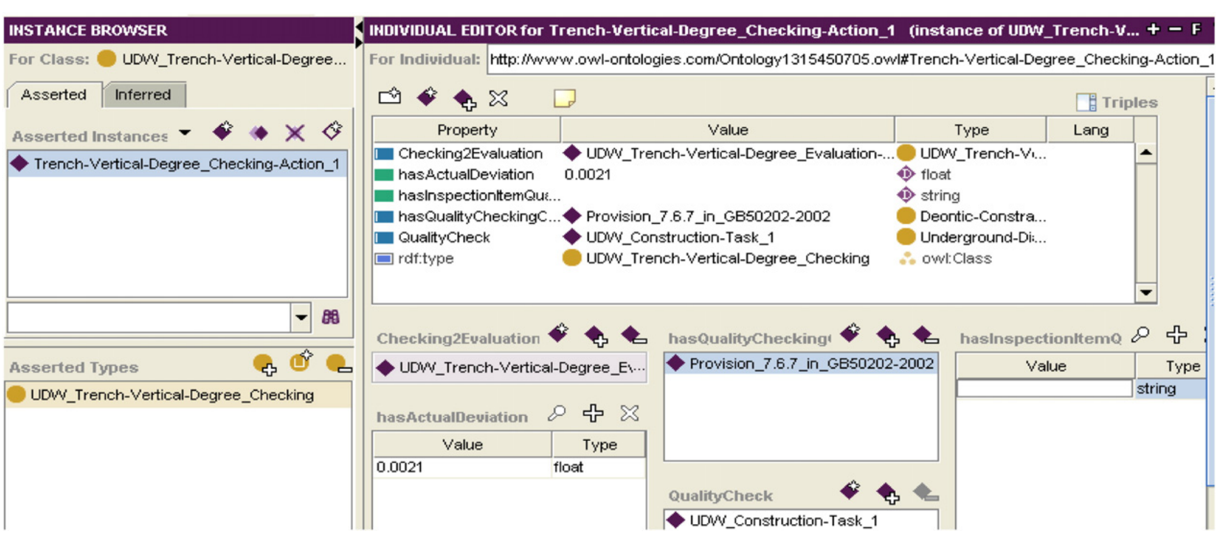

Fig. 8. Protégé-OWL screenshot of quality data input UGI.

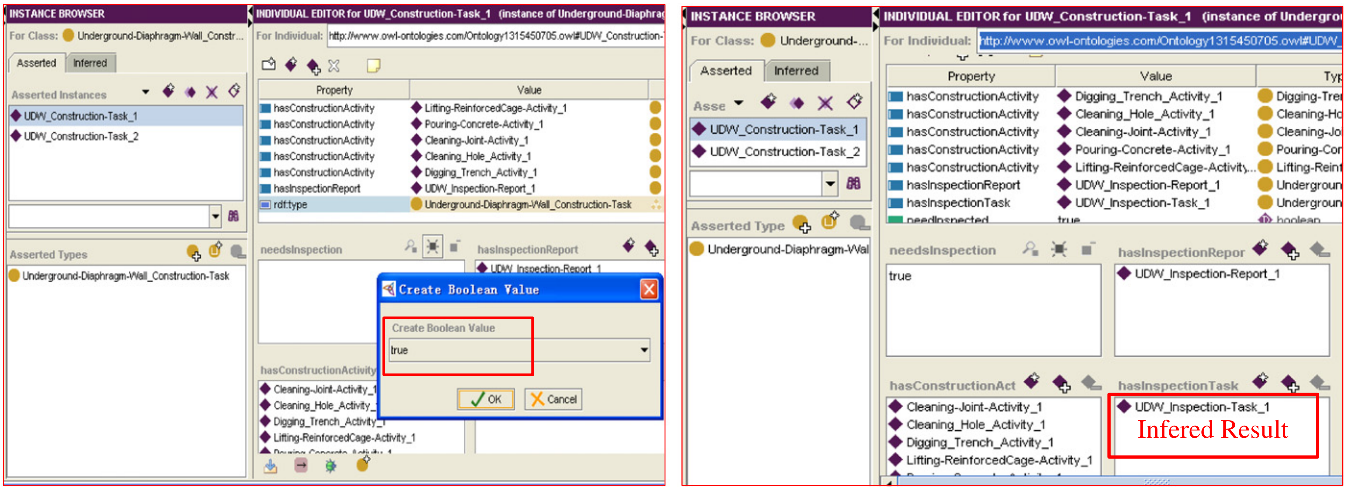

Fig. 9. Protege-OWL screenshot of the *UDW_Construction-Task_1* individual.

5.2.2. Execution of JESS inference engine

Here, the underground-diaphragm-wall construction quality inspection, depicted in Section 5.1, is selected to show how to implement the construction quality compliance checking.

Fig. 9 depicts how the applicable regulation provisions imposed on the underground-diaphragm-wall construction are translated into a set of inspection tasks to be performed.

According to the Provision 7.6.12 in GB 50202–2002, in the process of construction, it is necessary to examine the verticality degree of the finished trench.

Once determining that the instance *UDW_Construction-Task_1* needs inspection, namely, the Boolean property *needsInspection* is given the value "true", *Rule 2–2* (given in Section 5.2.2) is fired. When running *Rule 2–2*, the object property *hasInspectionTask* is filled with the instance *UDW_Inspection-Task_1*, which means the inspection task instance *UDW_Inspection-Task_1* is assigned to the instance *UDW_Construction-Task_1*. Furthermore, when *Rule 4–1* is fired, the engine automatically deduces and fills in the "*QualityCheck*" property of the individual *UDW_Construction-Task_1*, which means the checking action is assigned to the instance *UDW_Construction-Task_1*.

From this scenario, we can see that these rules enable the regulatory provisions to be integrated with the construction process, and the regulatory compliance checking is regarded as a paralleling activity to construction process rather than as an afterthought.

The scenario about the quality evaluation of the underground-diaphragm-wall trench's verticality degree is shown in Fig. 10.

The permissible deviation for the underground-diaphragm-wall trench's verticality degree should be less than 1/300 (acceptance criteria), if the underground diaphragm wall acts as permanent structure (from Provision 7.6.12 in GB 50202–2002).

Once the actual deviation (quality inspection data) is got, *Rule 5-1/2* is fired to compare the actual deviation (0.0021, ① in Fig. 10) with the permissible deviation (0.0033, ② in Fig. 10), thus the evaluation result is determined (③ in Fig. 10), and the quality evaluation value *isAccepted* is assigned to the property *hasInspectionItemQualityEvaluationResult*. At the same time, the quality inspection items with the result "*isAccepted*"

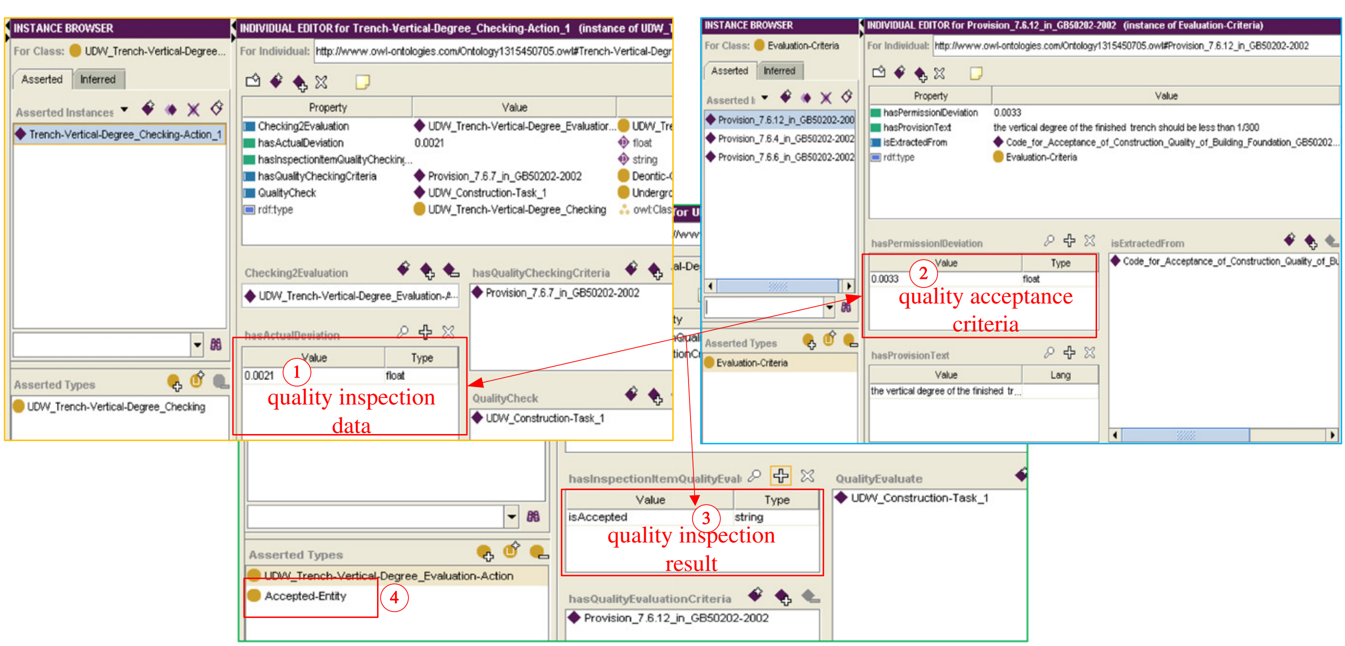

Fig. 10. Protégé-OWL screenshot of quality evaluation for the trench's verticality degree.

are classified as the *Accepted-Entity* (④ in Fig. 10), which means that the quality of the inspection items is compliant with the relevant acceptance criteria.

In addition, these SWRL rules can be used in conjunction with SPARQL and SQWRL to query OWL knowledge, for example, to query the good construction process definition for reusing.

Moreover, the expression of the SWRL body part, with the ontological concepts as operands, can be seen as the indication of the rule applicability condition, which can be used to filter the irrelevant rules. Furthermore, these rules can be annotated semantically, with the concepts defined in the ontology mentioned above. Basing on these annotated information, the constraint rule base can be classified into rule subset base, this will facilitate managing and maintaining the rule base, especially when the rule base contain large numbers of constraint rules.

5.2.3. Validation and discussion

As we know, the constraints for the construction quality inspection not only come from the codes for the acceptance of construction quality, but also from other codes for design and technical regulations. In addition, the regulation provisions usually are referenced to each other. No matter where these constraints come from, we can classify them into the group by the construction product or process domain, and further group them into the two types of constraints: the quality evaluation criteria and the deontic constraints about the construction and inspection process.

In this paper, we mainly investigate a serial of codes for the acceptance of construction quality, from GB50202 to GB50210.

First, we look at the relevant provisions in the code for one construction process quality inspection, and then the referenced provisions are also investigated. For example, the provision 7.6.11 about the underground-diaphragm-wall, in GB 50202-2002: "the reinforced concrete base slab shall comply with the national standard of Code for the Construction and Acceptance of Concrete Structures(GB-50204-2002) that is currently in force and effective." This indicates that the relevant provisions in GB 50204 need to be investigated for the quality inspection of underground-diaphragm-wall. Finally, all the provisions for one specific construction process quality inspection are selected altogether and the two types of constraints are extracted and formalized.

As to the deontic constraints, including the construction procedure constraints and inspection process constraints, etc., in this paper, we mainly discussed their embedding into the construction process, via encapsulating the process and constraint knowledge in OWL ontologies, and using OWL axioms and SWRL rules as means to ensure their compliance to the regulation, which can be seen as an indirect efforts to improve the construction quality.

The quality evaluation criteria define the quality objectives that the construction objects need to achieve or meet. The criteria may be the requirement about the properties of the inspection items, such as geometric feature and engineering parameters, or the requirement about the spatial relationship between the inspection items, and so on.

As to the quality evaluation criteria, some constraints can be represented as the combination of the math built-ins provided by SWRL Built-in base. The constraints about the spatial relationships cannot be modeled with the SWRL built-ins, since the SWRL does not provide the spatial relation built-ins. The modeling capability of the spatial constraints depends on the concepts and the semantic relations defined in the process ontology, since the SWRL provides an extension mechanism to add user-defined predicates.

Since the meta-model and approach are not limited to any specific construction domain, the meta-model can be reused independently of concrete construction object domains. Although, the two types of constraints only about the underground-diaphragm-wall are illustrated in this paper, I am convinced that the constraints from other regulations can be represented and reasoned about as well, as long as they belong to the constraint types discussed here, no matter whether they are about the underground-diaphragm-wall, or other building components, such as all kinds of piles.

In our approach, the semantic representation of the information for construction compliance checking is assumed as an underlying prerequisites. Currently, the Protégé-OWL editor is used to establish the construction quality semantic information. Based on the class and property definitions, Protégé automatically generates graphical user interface (GUI) forms that can be used to create instances of classes. Thus, users can enter the instance information by using these Protégé generated forms. Further research will be done on developing a prototypical user-friendly interface which can simply receive input data and generate output results.

Nowadays, IFC-based project model is getting more focus, and IFC as an semantic standard has been supported by many CAD tools. As we know, some design information is need for the quality inspection, so the design information in these tools need exported into an IFC representation, and further be converted into an IFC/RDF [33]. For example, P. [23] has developed the IFC-to-RDF converter. In this way, the design information required by the specification can be provided by the project model.

Now, formalizing the provisions into rules is done manually at present, which is time-consuming and bored. In addition, since the SWRL does not support disjunction relationship representation, in some case, some constraints need to be divided in multiple specifications only targeting one single requirement. For example, "the permissible deviation for the underground-diaphragm-wall trench's verticality degree should be less than 1/300 (1/150), if the underground diaphragm wall acts as permanent structure (makeshift structure)". In the future, the natural language processing techniques (NLP) can be used to parse the regulation provision from text into OWL axioms and SWRL rules.

6. Conclusion

The construction phase of a project is governed by lots of regulations. The regulation-based construction quality compliance checking is necessary and remains a challenge.

This study tries to prove that an ontological and semantic approach can be applied to the building construction regulation compliance checking domain, for improving the support to the construction quality inspection and management. In this paper, an approach for ontology-based regulation semantic modeling is explored and the ontology for construction quality inspection and evaluation, i.e. CQIEOntology is proposed, based on which the regulation constraints can be modeled into OWL axioms and SWRL rules. The proposed approach is demonstrated in Protégé 3.4.6, through case studies based on regulation examples taken from "Code for Acceptance of Construction Quality of Building Foundation (GB50202-2002)".

The proposed approach facilitates the integration of the regulation constraints with the construction process, which can improve the automated construction quality compliance checking during construction from the definition and execution of the construction process to the final construction quality acceptance evaluation.

Regulation-based automated construction quality compliance checking will improve efficiency, reduce errors and save human resources. Furthermore, the ontology-based approaches can also be applied, with minor modifications, to other regulation-based compliance checking domains, for example, safety and environment regulation in construction industry. As an initial exploratory study, this work is an interesting step towards exploring in greater detail the possibilities of implementing the semantic web technologies in the building construction regulation compliance checking domain.

Acknowledgements

The research is supported by the Fundamental Research Funds for the Central Universities (grant no. 2010QN067). The authors also would like to thank anonymous reviewers who gave valuable suggestion that has helped to improve the quality of the manuscript.

References

[1] T.R. Gruber, Towards Principles for the Design of Ontologies Used for Knowledge Sharing, Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic Publishers, Deventer, the Netherlands, 1993.

[2] C.J. Anumba, R.R.A. Issa, J.Y. Pan, I. Mutis, Ontology-based information and knowledge management in construction, Construction Innovation 8 (3) (2008) 218–239.

[3] T.E. El-Diraby, J. Zhang, A semantic framework to support corporate memory management in building construction, Automation in Construction 15 (2006) 504–521.

[4] I. Mutis, R. Raja, Semantic tags for interoperable construction workflow, In: ASCE Conf. Proc. Computing in Civil Engineering, 2009, pp. 623–632.

[5] I. Svetel, M. Pejanović, The role of the semantic web for knowledge management in the construction industry, Information 34 (2010) 331–336.

[6] P.R.C. Ministry of Construction, Building Code for Acceptance of Construction Quality of Building Foundation: GB50202_2002, Beijing: China Architecture & Building Press, China, 2002.

[7] World Wide Web Consortium, SWRL: a Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/2004/SUBM-SWRL-20040521 (last accessed on 20th. January 2010).

[8] M.G. Battikha, QUALICON: computer-based system for construction quality management, Journal of Construction Engineering and Management 128 (2) (2002) 164–173.

[9] Young S. Kim, Oh. SeW, Yong K. Cho, Jong W. Seo, A PDA and wireless web-integrated system for quality inspection and defect management of apartment housing projects, Automation in Construction 17 (2008) 163–179.

[10] B. Akinci, F. Boukamp, C. Gordon, D. Huber, C. Lyons, K. Park, A formalism for utilization of sensor systems and integrated project models for active construction quality control, Automation in Construction 15 (2) (2006) 124–138.

[11] F. Boukamp, B. Akinci, Automated processing of construction specifications to support inspection and quality control, Automation in Construction 17 (1) (2007) 90–106.

[12] S. Demir, J.H. Garrett Jr., B. Akinci, O. Akin, M. Palmer, A semantic web-based approach for representing and reasoning with vocabulary for computer based standards processing, In: The Proceedings of the 2010 International Conference on Computing in Civil and Building Engineering (ICCCBE'10), Nottingham, UK, June 30- July 2, 2010, 2010.

[13] John de Gelder, Conceptional modeling of building regulation knowledge, Artificial Intelligence in Engineering 11 (1997) 273–284.

[14] C. Eastman, J. Lee, Y. Jeong, J. Lee, Automatic rule-based checking of building designs, Automation in Construction 18 (8) (2010) 1011–1033.

[15] X. Tan, A. Hammad, P. Fazio, Automated code compliance checking for building envelope design, Journal of Computing in Civil Engineering 24 (2) (2010) 203–211.

[16] L. Khemlani, CORENET e-PlanCheck: Singapore's automated code checking system AECBytes[EB/OL], http://www.aecbytes.com/buildingthefuture/2005/CORENETePlanCheck.html 2005 (accessed September 20, 2009.).

[17] H. Kim, F. Grobler, Design coordination in building information modeling using ontological consistency checking, In: Proc. of the ASCE International Workshop on Computing in Civil Engineering, 2009, pp. 410–421.

[18] T.H. Nguyen, Integrating building code compliance checking into a 3D CAD System, Computing in Civil Engineering (2005) 1–12.

[19] H.M. Satti, R.J. Krawczyk, Issues of integrating building codes in CAD, In: 1st ASCAAD International Conference e-Design in Architecture, Dhahran, Saudi Arabia, 2004.

[20] Jirapon Sunkpho, J.H. Garrett Jr., S McNeil, XML-based inspection modeling for developing field inspection support systems, Journal of infrastructure systems 11 (3) (2005) 190–200.

[21] L. Ding, R. Drogemuller, M. Rosenman, D. Marchant, J. Gero, Automating code checking for building designs, In: The Second International Conference–Clients Driving Innovation: Moving Ideas into Practice, Australia, 2006.

[22] Jirapon Sunkpho. A framework for developing field inspection support systems, Ph.D. thesis, Carnegie Institute of Technology, Department of Civil and Environmental Engineering, Carnegie Mellon University Pittsburgh, 2001.

[23] P. Pauwels, D. Van Deursen, R. Verstraeten, J. De Roo, R. De Meyer, A semantic rule checking environment for building performance checking, In: Automation in Construction, 2009, pp. 1–13.

[24] A. Yurchyshyna, A. Zarli, An ontology-based approach for formalization and semantic organization of conformance requirements in construction, Automation in Construction 18 (8) (2009) 1084–1098.

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

[26] P. Katranuschkov, A. Gehre, R.J. Scherer, Reusable process patterns for collaborative work environments in AEC, In: *ICE 2007 Proceedings of the 13th International Conference on Concurrent Enterprising*, Nottingham, UK, 2007.

[27] A. Benevolenskiy, P. Katranuschkov, R.J. Scherer, Ontology-Based Configuration of Construction Processes Using Process Patterns, In: Proceedings of EG-ICE Workshop 2011, Twente, Netherlands, 2011.

[28] Han-Hsiang Wang, Frank Boukamp, Tarek Elghamrawy, An ontology-based approach to context representation and reasoning for managing context-sensitive construction information, Journal of Computing in Civil Engineering 25 (5) (2009) 331–346.

[29] Han-Hsiang Wang, Frank Boukamp, Ontology-based representation and reasoning framework for supporting job hazard analysis, Journal of Computing in Civil Engineering 25 (4) (2011) 442–456.

[30] Stanford Center for Biomedical Informatics Research, The Protégé Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/.

[31] Rule Interchange Format (RIF) Working Group, http://www.w3.org/2005/rules/wiki/RIFWorkingGroup(last accessed on 6th. May 2010).

[32] T. Berners-Lee, D. Connolly, L. Kagal, Y. Scharf, J. Hendler, N3Logic: a logical framework for the world wide web, Theory and Practice of Logic Programming 8 (3) (2008) 249–269.

[33] UGent MultiMediaLab, IFC-to-RDF Converter Service, http://ninsuna.elis.ugent.be/IfcRDFService/8(last accessed on 9th. July 2011).