The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479247.022
The program can be downloaded here: http://homepages.wmich.edu/~jyb5534/resources/ZE_BIM_FOL_Converter.zip

# Automated Extraction of Information from Building Information Models into a Semantic Logic-Based Representation

J. Zhang[1] and N. M. El-Gohary[2]

[1]Graduate Student, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Ave., Urbana, IL 61801; PH (217) 607-6006; FAX (217) 265-8039; email: jzhang70@illinois.edu
[2]Assistant Professor, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Ave., Urbana, IL 61801; PH (217) 333-6620; FAX (217) 265-8039; email: gohary@illinois.edu

## ABSTRACT

One of the major goals of building information modeling (BIM) is to support automated compliance checking (ACC). To support ACC, project information (i.e., building design information) needs to be extracted from BIM models and transformed into a representation that would allow for automated reasoning about those project/design information in combination with information from regulatory documents. However, existing BIM information extraction (IE) efforts are limited in supporting complete automation of ACC. Complete automation of ACC requires (1) automating both the extraction of information from BIM models and the extraction of regulatory information from regulatory documents and (2) aligning the instances of information concepts and relations extracted from a BIM model with those extracted from regulatory documents, in order to facilitate direct automated reasoning about both information for compliance assessment. To address this gap, this paper proposes an automated BIM IE method for extracting project information from industry foundation classes (IFC)-based building information models (BIMs) into a semantic-based logic representation that is aligned with a matching semantic-based logic representation of regulatory information. The proposed BIM IE method utilizes semantic natural language processing (NLP) techniques and java standard data access interface (JSDAI) techniques to automatically extract project information from IFC-based BIMs and transform it into a logic format (logic facts) that is ready to be automatically checked against logic-represented regulatory rules. The BIM IE method was tested on extracting project/design information from a Duplex Apartment BIM model. Compared to a manually developed gold standard, the testing results showed a 100% precision and a short time of 15.02 seconds for processing 38898 lines of data.

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479247.022
The program can be downloaded here: http://homepages.wmich.edu/~jyb5534/resources/ZE_BIM_FOL_Converter.zip

## INTRODUCTION

Construction projects are governed by various regulations. The manual process of checking compliance with regulations is time consuming, costly, and error-prone (Zhang and El-Gohary 2013). Automated compliance checking (ACC) of construction projects against various regulations could save time, cost, and reduce human errors (Zhong et al. 2012). To facilitate ACC, a computer-interpretable, user-understandable, and unambiguous representation is needed for construction regulations (Garrett and Palmer 2014). To address that, a semantic logic-based information representation (IRep) and compliance reasoning (CR) schema for regulatory information and project information was proposed (Zhang and El-Gohary 2014b). A challenge is then how to extract project information from building information models (BIMs) – the most popular and promising digital representation of project information, and how to transform the extracted project information into the semantic logic-based representation. This paper aims to address this challenge by proposing a new BIM information extraction method. The proposed method utilizes java standard data access interface (JSDAI) and semantic natural language processing (NLP) techniques. The remaining sections of this paper present the details of the proposed method and its preliminary testing on extracting information from a Duplex Apartment BIM model.

## BACKGROUND

**Semantic NLP.** NLP targets to enable computers to process natural language texts and speeches in a human-like manner (Cherpas 1992). NLP has many application domains such as automated natural language translation (Marquez 2000), text classification (Zhou and El-Gohary 2014), and information extraction (Zhang and El-Gohary 2013). Techniques in NLP utilize two main types of features - syntactic features and semantic features. Syntactic features are related to the grammatical structure of text, such as part of speech (POS) tags that depict lexical and functional categories of words, and phrasal tags that depict lexical and functional categories of phrases. Semantic features are related to the meaning of text, such as concepts and relations from a semantic model. Ontology is a widely used semantic model which captures domain knowledge through concepts, relations, and axioms in a structured manner (El-Gohary and El-Diraby 2010). Semantic NLP utilizes both syntactic features and semantic features to fulfill NLP tasks.

**JSDAI.** JSDAI is a standard data access interface (SDAI) application programming interface (API) for accessing and processing information in EXPRESS written models. EXPRESS is an ISO standard product data modeling language (ISO 2004). The industry foundation classes (IFC) specification, as the most popular data schema for BIMs, is written in EXPRESS language. There are two types of information access methods in JSDAI: early binding and late binding. Early binding requires the availability of the EXPRESS model at the program compiling time, and

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479247.022
The program can be downloaded here: http://homepages.wmich.edu/~jyb5534/resources/ZE_BIM_FOL_Converter.zip

accesses each entity of the known EXPRESS model with standard access methods such as "set" and "get". Late binding does not require the availability of the EXPRESS model at the program compiling time, and accesses each attribute and entity using standard access methods such as "set" and "get". Late binding is more complex than early binding, but is independent of specific EXPRESS models.

**Proposed Information Representation (IRep) Schema.** The proposed semantic logic-based IRep schema (Zhang and El-Gohary 2014a) is based on first order logic (FOL) – a formally defined logic that is expressive and effective in supporting automated reasoning. Among the different varieties of FOL types, horn clause (HC) was selected because it is most effective in supporting automated reasoning. HC is a type of FOL which is composed of a disjunction of literals of which at most one is positive. Prolog logic programming language and reasoner were selected to represent the HC-based project information, because it is the most widely used logic programming language. There are three types of clauses in Prolog – rules, facts, and queries. A rule has the form: "H :- B1, B2, …, Bn. (n>0)." H is called the head and B1 to Bn are called the body, where all are atomic formulas. The rule means "if B1, B2, …, and Bn, then H." A fact is a special type of rule whose body is always true (Zhou 2012).

## PROPOSED METHOD

The authors propose a two-step method (Figure 1) to automatically extract project information from IFC-based BIMs, and transform the extracted information into logic facts following their previously proposed semantic logic-based IRep schema. The information extraction (IE) step utilizes JSDAI techniques to access and extract the entities and attributes in the IFC-based BIMs. The information transformation (ITr) step utilizes semantic NLP techniques to map and transform the entities and attributes into logic facts.

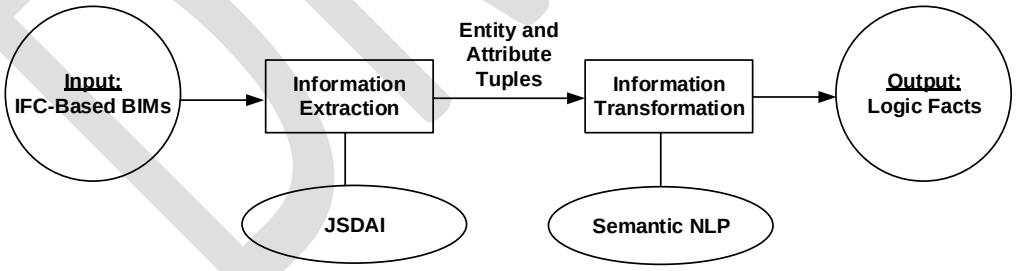



**Figure 1. Proposed method.**

**Input: IFC-Based BIMS.** The proposed method aims to process BIMs in IFC format (i.e., files having the extension name of ".ifc", referred to as IFC files hereafter) based on the IFC schema. The IFC schema is the main data model schema to describe data in the building and construction industry, which is registered with ISO as ISO 16739. The IFC format is neutral and platform independent (BuildingSMART 2014). The BIMs in IFC format use "STEP physical file" format

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479247.022
The program can be downloaded here: http://homepages.wmich.edu/~jyb5534/resources/ZE_BIM_FOL_Converter.zip

defined as ISO10303-21 (BuildingSMART 2014). In a "STEP physical file", each line is assigned a line number. Each line represents an entity. An entity in IFC format represents either a concept or relation. For example, "IFCBUILDINGSTOREY" is an entity representing a concept "building storey", and "IFCRELVOIDSELEMENT" is an entity representing a relation "voids element" which defines the relation between an "opening element" and the "void" made by the "opening element".

**Output: Logic Facts.** The proposed method outputs text files carrying processed information represented as logic facts, following a previously proposed IRep schema. In the proposed IRep schema, two types of facts are used: concept facts and relation facts. A concept fact defines a constant as an instance of a certain concept. For example, "door(door6652)" defines the constant "door6652" as an instance of the "door" concept. A relation fact defines a relationship between an instance of a concept and an instance of another concept or a value. For example, has(project34, site38274) defines the association relation between an instance of project "project34" and an instance of site "site38274." The number in an instance is the line number of the instance in its source IFC file. The use of these line numbers satisfies three purposes: (1) identifying instances, (2) distinguishing instances, and (3) establishing links between the logic facts and lines in their IFC source file. Both concept facts and relation facts are represented as predicates. A predicate is the building block of a logic clause. A predicate consists of a predicate symbol and one or more arguments (i.e., constants or variables) in parenthesis following the predicate symbol [e.g., the predicate "door(door6652)" has one predicate symbol "door" and one argument "door6652", where "door6652" is a constant].

**Information Extraction.** In the IE step, JSDAI is used to access the entities and attributes in the IFC file. Between the two access types of early binding and late binding, the late binding is selected. Because the proposed method is designed to support BIMs based on different versions of the IFC schema (e.g., IFC 2x3, IFC2x3-TC1, IFC4). In this manner, even future BIMs based on future IFC schema releases (which are not published and known now) could still be supported by this proposed method, with the only premise that future IFC releases continue using EXPRESS language. The proposed IE step processes information according to metadata at the EXPRESS data schema level (i.e., schema of schema of BIMs). The algorithm for this IE step is shown in Figure 2. The algorithm first initializes all variables to be used and compiles the version of IFC schema to use, then processes the lines in an IFC file one by one. The processing of each line uses a subroutine S1. In S1, if the entity represented by the line being processed is of aggregate type, then S1 is recursively called on each sub-entity of the aggregate entity. Otherwise, the names of all attributes of the entity being processed are looked up in the compiled IFC schema, and the values of all attributes of the entity being processed are then accessed. At the end of S1, the following information for the entity processed are stored into a tuple: (1) the name of the entity; (2) the line number of the entity; (3) the list of attribute names of the entity; and (4) the values of the attributes of the entity. Figure 3 shows

DRAFT

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479247.022
The program can be downloaded here: http://homepages.wmich.edu/~jyb5534/resources/ZE_BIM_FOL_Converter.zip

an example processing, where the entity in line "#36686" from Part I generates the bold-highlighted tuple in Part II.

**Information Transformation.** In the ITr step, semantic NLP is used to transform the extracted tuple-represented entities and attributes into logic facts. This step includes two main processing subtasks: (1) semantic look up of entity names and attribute names; and (2) transformation of entities and attributes into concept facts and relation facts.

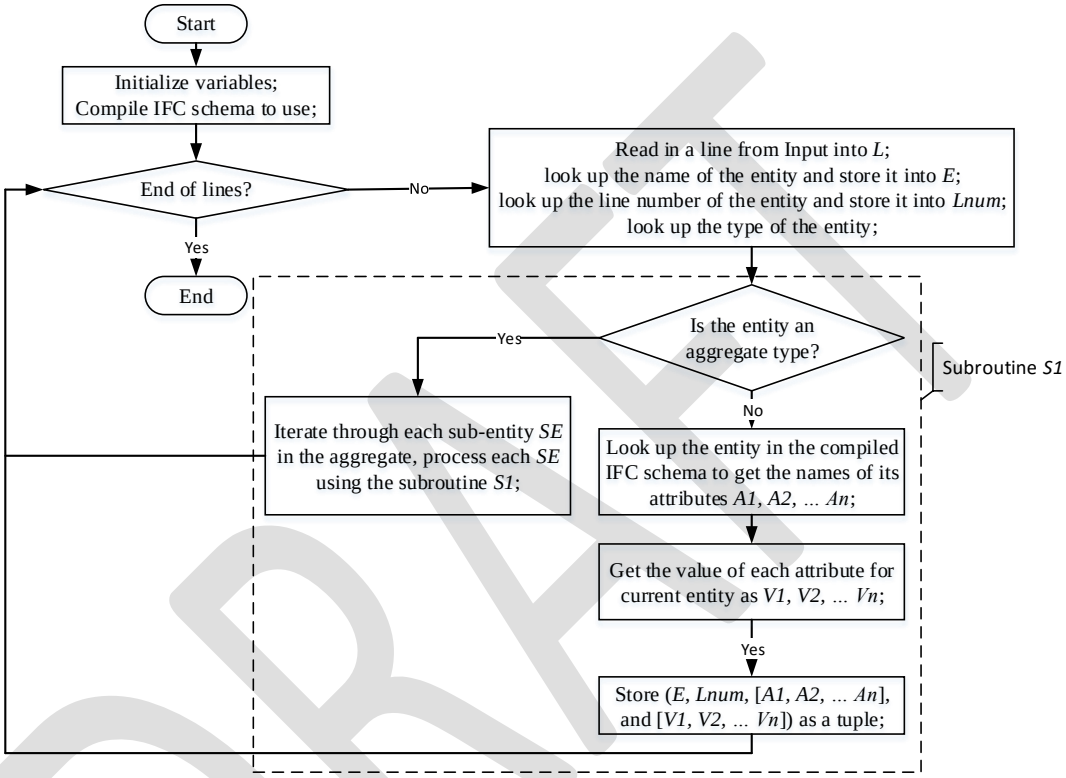
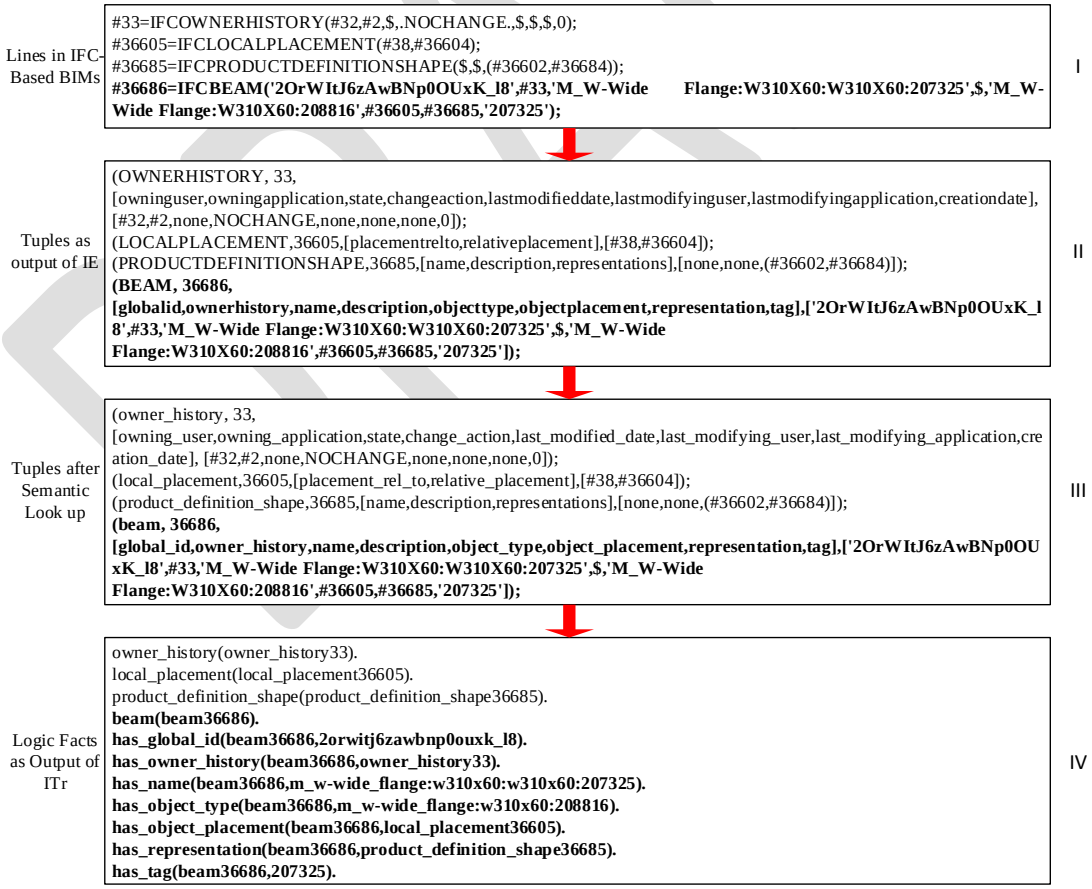



**Figure 2. Proposed IE algorithm.**

In the semantic look up subtask, each extracted entity name, attribute name, and attribute value (i.e., if the attribute value is of entity type or enumeration type) is looked up in the used IFC schema version. The matched name or enumeration type value in the IFC schema is then used to convert the extracted name/value into underscore-connected terms. For example in Figure 3, an entity "OWNERHISTORY" is looked up in the IFC schema to find the matched entity name "IfcOwnerHistory". Then the term boundary information in "IfcOwnerHistory" (i.e., represented by capitalization) is used to convert the extracted "OWNERHISTORY" to "owner_history." This is needed because the output of this ITr step is to instantiate logic rules based on the semantic logic-based representation. To enable that instantiation, the semantic information of each term in an entity or attribute name needs to be semantically matched with terms in concepts and relations from regulatory requirements (represented in logic rules). In the entities/attributes

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479247.022
The program can be downloaded here: http://homepages.wmich.edu/~jyb5534/resources/ZE_BIM_FOL_Converter.zip

transformation subtask, a rule-based NLP approach is selected (Zhang and El-Gohary 2014a). Three main NLP-based transformation rules are used: (1) an entity is transformed into a concept fact (i.e., a predicate) by using the name of the entity as the name of the predicate, and using the name of the entity concatenated with the line number as the argument (i.e., an entity constant) of the predicate. For example, in Figure 3, the beam entity is transformed into a concept fact "beam(beam36686)," with the name of the entity "beam" being the predicate name and concatenation of the entity name and the line number "beam36686" as the predicate argument; (2) an attribute of an entity is transformed into a relation fact (i.e., a predicate), using the name of the attribute preceded by "has_" as the name of the predicate, using the corresponding entity constant as the first argument of the predicate, and using the value of the attribute as the second argument of the predicate (if the value is not a reference to another entity). For example, in Figure 3, the attribute "global_id" for the beam entity is transformed into a relation fact "has_global_id(beam36686, 2OrWItJ6zAwBNp0OUxK_l8);" and (3) if the value of an attribute is a reference to another entity, then the referred entity constant is used as the second argument of the predicate. For example, in Figure 3, the attribute "owner_history" for the beam entity is transformed into a relation fact with the referred entity constant owner_history33 as the second argument - "has_owner_history(beam36686,owner_history33)."



The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479247.022
The program can be downloaded here: http://homepages.wmich.edu/~jyb5534/resources/ZE_BIM_FOL_Converter.zip

## PRELIMINARY EXPERIMENTAL RESULTS AND ANALYSIS

For experimental purpose, the proposed IE and ITr algorithms were implemented in JAVA Standard Edition Development Kit jdk1.7.0_40. The JSDAI v4 was used in the experiment to access IFC-based BIMs. The Duplex Apartment Project from buildingSMARTalliance of the National Institute of Building Sciences was selected as the source of data. The IFC file "Duplex_A_20110907.ifc" was selected for testing which includes 38898 lines of data. Out of the 38898 lines of data, 100 lines were randomly selected as a testing sample. A gold standard was developed by manually interpreting the entities and attributes in these 100 lines and generating the target logic facts for them. The proposed automated IE and ITr algorithms were applied to the "Duplex_A_20110907.ifc" file, and the output results were compared with a manually developed gold standard. The experimental results are summarized in Table 1. For the 100 concept facts and 328 relation facts corresponding to the 100 lines of data, 100% precision was achieved. In addition, it took only 15.02 seconds to process all the 38898 lines of data. Yet, two limitations of the proposed method are identified: (1) some output logic facts are not interpretation-friendly. For example, the universal pre-fix for relation facts (i.e., "has_") does not fit in cases like the following predicate *P1*; (2) the relations represented in the IFC file are not perfectly aligned with the authors' proposed semantic logic-based representation. For example, an explicit relation entity in IFC is typically represented by two predicates (e.g., *P2* and *P3*) whereas in the semantic logic-based representation it is represented by only one predicate (e.g., *P4*).

*P1: has_for_layer_set(material_layer_set_usage21369,material_layer_set21320).*
*P2: has_relating_space(rel_space_boundary38711,space514).*
*P3: has_related_building_element(rel_space_boundary38711,covering23992).*
*P4: has_space_boundary(space514, covering23992).*

**Table 1. Preliminary Experimental Results.**

| Number/Measure | Concept Facts | Relation Facts |
|---|---|---|
| In Gold Standard | 100 | 328 |
| Extracted | 100 | 328 |
| Correctly Extracted | 100 | 328 |
| Precision | 100% | 100% |

## CONCLUSION AND FUTURE WORK

This paper presents a new BIM IE method to extract project information from IFC-based BIMs and transform the extracted information into logic facts, automatically. The proposed method is intended to support automated reasoning for automated compliance checking. But, the proposed method could be further used to

DRAFT

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479247.022
The program can be downloaded here: http://homepages.wmich.edu/~jyb5534/resources/ZE_BIM_FOL_Converter.zip

assist other analysis and reasoning applications that use IFC-based BIM models, because it provides logic facts-represented information that are human-interpretable and computer-processable. The method utilizes JSDAI and semantic NLP techniques. It could process BIMs based on different versions of IFC schema. The method was tested on processing information in the Duplex Apartment Project from buildingSMARTalliance of the National Institute of Building Sciences. Comparing to a manually developed gold standard, the experimental results showed a 100% precision. In addition, the processing of 38898 lines of data only took 15.02 seconds. Two limitations of the proposed method were identified. In their future work, the authors will further refine the proposed method to make its output logic facts better aligned with regulatory requirements represented as logic rules.

## ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant No. 1201170. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

ISO. (2004). "ISO 10303-11:2004 - Part 11: Description methods: The EXPRESS language reference manual." <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38047> (Dec. 05, 2014).

BuildingSMART. (2014). "Industry Foundation Classes (IFC) data model." < http://www.buildingsmart-tech.org/specifications/ifc-overview> (Dec. 06, 2014).

Cherpas, C. (1992). "Natural language processing, pragmatics, and verbal behavior." *Anal. Verbal Behav*., 10(1992), 135–147.

El-Gohary, N. M., and El-Diraby, T. E. (2010). "Domain ontology for processes in infrastructure and construction." *J. Constr. Eng. Manage*., 136(7), 730–744.

Garrett, Jr., J.H., and Palmer, M.E. (2014). "Delivering the infrastructure for digital building regulations." *J. Comput. in Civ. Eng*., 28, 167-169.

Marquez, L. (2000). "Machine learning and natural language processing." *Proc., "Aprendizaje* 869 *automatico aplicado al procesamiento del lenguaje natural".*

Zhang, J., and El-Gohary, N.M. (2014a). "Automated information transformation for automated regulatory compliance checking in construction." *J. Comput. in Civ. Eng.*, accepted.

Zhang, J., and El-Gohary, N.M. (2014b). "Logic-based automated reasoning for construction regulatory compliance checking." *J. Comput. in Civ. Eng*., submitted.

Zhang, J., and El-Gohary, N.M. (2013). "Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking." *J. Comput. in Civ. Eng.*, published ahead of print.

DRAFT

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479247.022
The program can be downloaded here: http://homepages.wmich.edu/~jyb5534/resources/ZE_BIM_FOL_Converter.zip

Zhong, B. T., Ding, L. Y., Luo, H. B., Zhou, Y., Hu, Y. Z., and Hu, H. M. (2012). "Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking." *Autom. Constr*., 28(2012), 58–70.

Zhou, N. (2012). "B-Prolog user's manual (version 7.8): Prolog, agent, and constraint programming." Afany Software. <http://www.probp.com/manual/manual.html> (Dec. 28, 2013).

Zhou, P., and El-Gohary, N. (2014). "Ontology-based multi-label text classification for enhanced information retrieval for supporting automated environmental compliance checking." *Proc., 2014 ASCE Constr. Res. Congress (CRC)*, ASCE, Reston, VA, 2238-2245.

DRAFT