Proceedings of the IASS Annual Symposium 2016
"Spatial Structures in the 21st Century"
26–30 September, 2016, Tokyo, Japan
K. Kawaguchi, M. Ohsaki, T. Takeuchi (eds.)

Feasibility of a Knowledge-Based Engineering framework for the AEC industries

Maarten MATHOT*, Jeroen COENDERS[a], Anke ROLVINK[b]

*Delft University of Technology, White Lioness technologies
Stevinweg 1, 2628 CN Delft
maartenmathot@gmail.com

[a] Delft University of Technology, White Lioness technologies
[b] White Lioness technologies

Abstract

This paper presents the development of a novel computational framework with which expert analysis knowledge can be captured, stored and dynamically retrieved when required to determine the performance of a building design with respect to user-defined requirements.

By subdividing the engineer's and designer's knowledge into discrete steps, and storing these steps in a database along with a description of the context in which the knowledge is to be applied, a searchable knowledge base is created. Given a context consisting of a BIM model and one or more building requirements for which metrics are to be provided, the applicable knowledge can be iteratively retrieved from the knowledge base. Through computational reasoning an analysis is acquired as a chain of logically connected analysis steps.

Foreseen benefits of use of the framework include safekeeping and disclosure of AEC expert's knowledge and automation of analyses without loss of analysis transparency.

Keywords: computational analysis, knowledge-based engineering, requirements engineering, BIM, model checking, knowledge sharing

1. Introduction

Knowledge is a peculiar commodity. Although it is critical to the daily operations of all parties involved in the design and engineering of buildings, knowledge is most commonly stored in the minds of a small set of experts. This property makes it both fragile, as an expert may leave the company or just forget it, and difficult to retrieve, as there is no apparent way to know what knowledge someone else holds. Furthermore, not all or all the best knowledge present in companies is available to all design teams, as it relies on the knowledge that exists in combinatinos of individual team members.

Knowledge has no intrinsic value: only through its application or dissemination it becomes valuable (Dalkir [6]). An important way knowledge is applied by engineers and designers during building design is in the form of analyses and assessments; applying knowledge to the information of a design at hand to acquire required metrics, for instance on the structural performance-, thermal behaviour-, energy usage- or projected construction cost of the design. The selection of which knowledge to apply at what stage in which analysis can be based on two pieces of information: the information available on the current iteration of the design and the information required to execute the steps of the analysis.

This paper proposes a computational framework which mimics the aforementioned way of constructing an analysis out of available knowledge given design information and metrics to be determined, along with the required facilities to submit and store analysis knowledge and describe performance metrics. As the iterative selection of knowledge steps to be applied is based on human logic, given the building design at hand and the performance metric(s) to be acquired, the resulting analysis can be easily interpreted by any (semi-)expert in the field. In this way a designer or engineer can automatically retrieve and apply knowledge he or she did not enter him- or herself, and still have confidence in the correctness of the resulting analysis because it can be checked against the person's own reasoning. The knowledge within a building design- or engineering firm is not only stored, but easily disclosed for use throughout the organisation.

Section 2 provides a brief overview of previous research in various fields on which the development is based. Section 3 described the concept of the solution proposed, and Section 4 introduces the different developed components of the framework and their role in the accomplishment of the research objectives.

2. Literature review

2.1 Knowledge-Based Engineering

In Knowledge-Based Engineering (La Rocca [9]), or KBE, a central knowledge repository called a knowledge base contains the discretized engineering knowledge required to design an object. The case-specific context of the design at hand, consisting of both a current state of the design and design objectives, is introduced to the system by the end-user. When the KBE application is run, an inference- or reasoning engine iteratively executes a set of predefined rules to assess the state of design information available, select appropriate knowledge to apply from the knowledge base and then execute the knowledge, thereby changing the state of information.

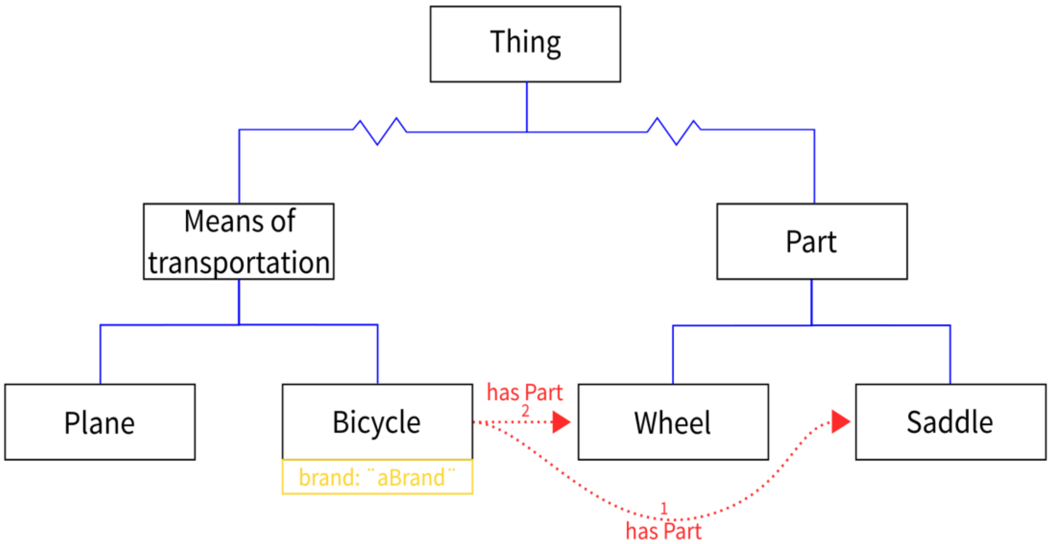

Figure 1. Example ontology of a bicycle of brand `aBrand`, which is a means of transportation, as indicated by the blue angular connections, and has Wheel(s) and Saddle(s) as parts, as indicated by the red dotted arrows.

2.2 Ontologies

An ontology is a specification of how the knowledge of a domain can be represented, which can be used in IT systems to provide additional information to other parts of the system (Milton [11]). In KBE, ontologies can be used to assign a unique definition to user input and enhance the input with properties commonly associated with the input provided (Stokes [15]). Given the example ontology in figure 1, an input of "bicycle" would be expected to have wheels, a saddle and a brand associated with it.

Ontologies are often represented as a graph consisting of subject-predicate-object *triples,* in which concepts in the domain form the nodes of the graph and their interrelations form the edges between them (Cyganiak et al. [5]). Through this basic representation a large portion of static knowledge can be represented.

2.3 Natural Language Processing

Natural Language Processing, or NLP, aims to translate input in the form of human language into a computer-interpretable form (Cambria and White [4]), as for instance translating into database queries (Bates [1]). Through a series of processing steps and models input text or speech is classified. Several clues can be used to perform this classification; the *syntax* of the words in the sentence, possibly in combination with the grammatical structure of the sentence as a whole, can assess whether a text is well formed. Interpreting the meaning of the input, or *semantic* analysis, however requires more knowledge on the context of the sentence.

Even more complex forms of NLP transcend the level of the sentence and can for instance resolve coreferences, in which multiple words reference the same thing, or classify entire texts based on writer intent.

2.4 Semantic annotation

A manual way of interpreting the semantics of human language is semantic annotation (Hjelseth and Nisbet [8]). Given a small set of annotators for which the meaning is strictly defined, a use is asked to assign the most appropriate annotator to a (set of) words in order to convey the semantic role of that (set of) words to the system. The process of annotating text is widely used in linguistic studies (Stenetorp et al. [14]) and research in the Architecture, Engineering and Construction (AEC) industries has shown that it has a relatively shallow learning curve (Niemeijer et al. [12]).

3. Concept

For this research project a prototype framework has been developed which applies aspects of all the research fields listed in Section 2 to the AEC industries. The framework aims to capture expert's analysis knowledge and the context in which it is applied in a knowledge base and, given a desired outcome of the analysis and a Building Information Model (BIM) containing the design information, chain together the applicable knowledge to form an analysis. Critical aspects include the way an expert can convey under what conditions his or her analysis step is applicable, and the way a desired analysis outcome is conveyed to the system.

As most design performance metrics to be determined are formulated as sentences in for instance building codes and contracts, being able to enter these in their original form is considered beneficial. NLP techniques are employed to allow for this direct entry. Describing the validity constraints of an analysis step in a sentence as well is expected to lower the threshold for experts to commit their knowledge. To assure the sentences are interpreted by the system in the way intended by the submitter, he or she can review and change the interpreted semantics in a semantic annotation process.

Another hurdle in interpreting an AEC experts' language is that it is often based on implicit knowledge of the domain. He or she for instance knows that a girder is a type of beam, which in turn is part of a vertical load bearing system, and thus knows that any assessment of the vertical load bearing system should include all objects marked as girder in the building model. Furthermore, different experts may use different terms for the same concept. It is very hard to develop a computer system capable of dealing with these ambiguities and implications, without containing some model of the implicit knowledge used. An ontology is used to represent the implicit, static domain knowledge and provide unique definitions to user input. The building models, analysis step contexts and performance metrics to be provided are all normalized with the use of an ontology.

By assessing the information of the design, contained in the building model, and the required output of the analysis, applicable analysis steps can be iteratively retrieved from the knowledge base to form an analysis.

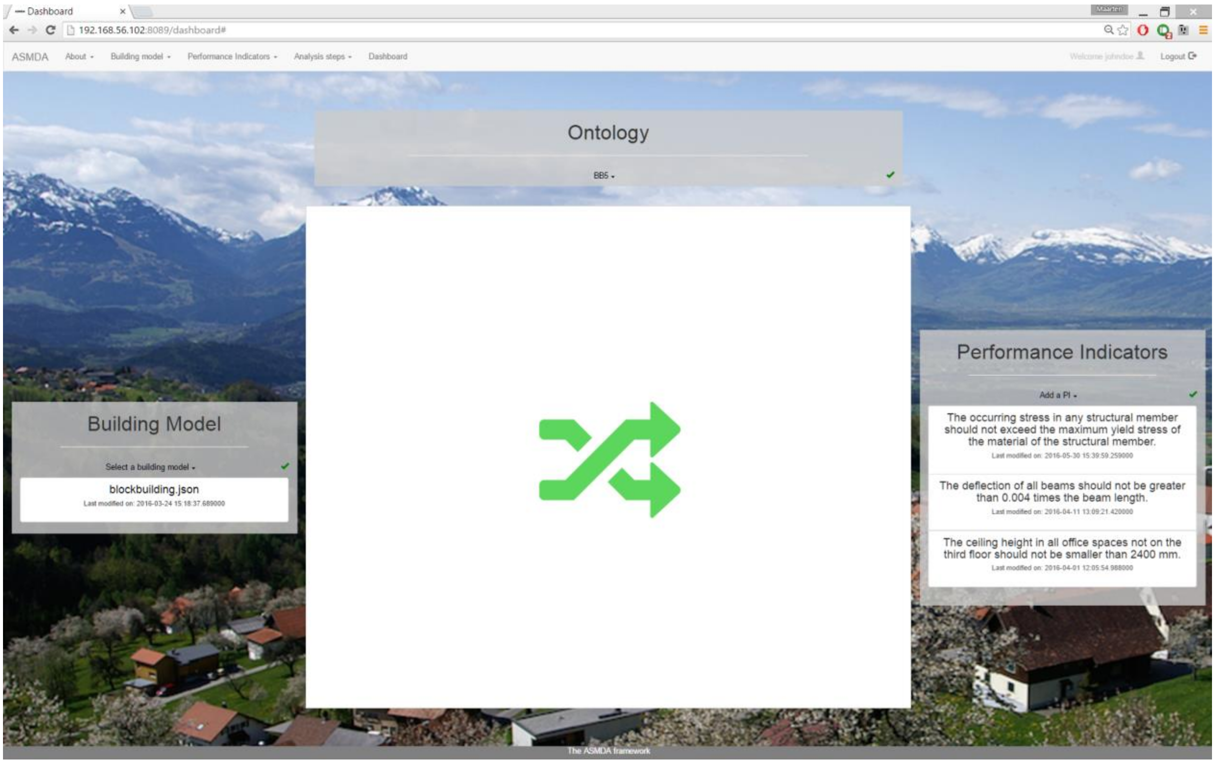

Figure 2. Dashboard page of the user interface, coordinating the inputs provided in the different framework components

4. Framework architecture

Based on the conceptual research described in the previous sections, a prototype of the framework has been developed. To maximize the accessibility the prototype has been developed as a webservice, so that the interfaces are accessible through any modern internet browser for registered users. Figure 2 depicts the dashboard page of the user interface, at which different inputs from the framework components are coordinated. The prototype consists of the following five components.

- an ontology editor module to model the background knowledge of concepts and their relations within the AEC domain,
- an analysis step definition module to enter expert knowledge to the knowledge base,
- a building model entry module to supply the project specific information of the design,
- a performance indicator definition module to define the required output of the analysis, and
- the analysis assembly module which iteratively selects knowledge from the knowledge base to form the analysis, given building model and performance indicators to be provided.

4.1 Ontology editor

As the framework will be using computational reasoning to chain together knowledge to form an analysis, it is useful if not necessary for the system to have some understanding of the AEC industry terms and concepts. When designers apply knowledge, they implicitly use their understanding of the domain to assess what knowledge they can use in the given situation. Additionally, they use a common understanding of the domain to facilitate communication with other experts in the domain. For example, if one structural engineer mentions a Vierendeel beam to another, they both share an understanding of some general properties of the beam. Furthermore, they both know what set of formulae might be applicable and what behaviour is likely to be governing the design.

Mimicking this background knowledge of the domain can greatly increase the efficiency of the user interaction with the framework and effectivity of the platform in retrieving the right knowledge in a specific circumstance. Within the framework the background knowledge is modelled in the form of an ontology.

Because the definitions of the concepts in the AEC domain and the way they are connected are stored in the ontology in a computer-interpretable way, the framework can interpret the terms used by an expert when he or she is submitting knowledge or requesting performance metrics by traversing and selecting parts of the graph. The connection between human language as entered by an expert and formal, computer-interpretable definitions is achieved through minimal Natural Language Processing as introduced in Section 2.3.

Ideally, one central ontology containing all concepts and relations in the domain would be used to reference all user interaction against. Unfortunately, no such universally accepted interpretation of the domain is currently in use in the AEC industries, and since different experts are likely to have different perspectives on the domain it is deemed unlikely that such a universally accepted standard ontology will soon be realized. Therefore, the framework provides both (draft) standard ontologies of the AEC domain that can be used directly as well as the means to edit ontologies to make custom versions. It is essential however for all parties involved in a project to agree on the use of one ontology to assure correct interpretation of concepts and relations.

Within the framework, all nodes and properties are assigned (a set of) corresponding codified word definitions, WordNet (Fellbaum and Miller [7]) synsets, as selected by the expert editing the ontology. In this way words in an English sentence are related to their counterparts in the ontology.

4.2 Analysis step definition

The analysis will be assembled from modular knowledge steps entered by a domain expert, i.e. a mechanical engineer. During analysis assembly, as described in Section 3.5, the knowledge will be selected based on:

- the information application of the step renders, or step *output,*
- the information required to apply the step, or step *input,*
- the circumstance in which the step is valid, or step *applicability.*

These three selection parameters are to be provided by the expert along with the step itself, in the form of a descriptive sentence. The step itself is stored as a separate file, currently in the form of a spreadsheet, Python script, or regular text, containing a description of an action, a formula, piece of logic or other manner of transforming the input of the step to the output of the step, given the applicability. Execution of the step is considered out of scope for the research, but incorporation of a parametric engine to execute the steps after analysis assembly is considered a valuable additional research direction.

The sentence describing the input, output and applicability of the step is related to the nodes and properties in the ontology by means of syntactic NLP. WordNet and the Natural Language Toolkit (Bird et al. [3]) are used to stem the words in the sentence and lookup potential definitions. These are then matched to the definitions that are part of the ontology and if a match is found the word is related to the corresponding node or property. If no match is found this is conveyed to the end-user, and he or she can either accept that the word is ignored in further processing (in the case of non-essential words in the sentence), change the phrasing of the sentence or adjust the ontology. In case of multiple matches against the ontology the end-user can manually choose the definition closest to his or her intended meaning.

With the definitions provided, the semantics of the sentence as a whole need to be assessed. To this end the framework applies a combination of semantic NLP and semantic annotation. Semantic NLP, based on Stanford CoreNLP (Manning et al. [10]) dependency analysis and custom developed models, is used to provide an estimate initial annotation which the domain expert can adjust to assure correctness using a customized version of BRAT (Stenetorp et al. [14]). The semantic marks used are *input, output* and *applicability*, corresponding to the selection parameters, *selection* and *negation*, indicating that certain parts of the sentence are used to specify others or that certain parts are negated, and *void*, used to remove incorrect estimated annotations.

The output of the annotation process is translated into an index for the step according to which the step can be retrieved during analysis assembly.

4.3 Building model entry

As the information in the building model is to be considered during analysis assembly to assess both whether the analysis has reached an end condition and whether a certain analysis step in the knowledge base can be applied, its contents need to be related to the ontology. Whilst efforts are being made to translate current building model format standards into ontology compliant formats, as for example the IFC to IfcOWL translation by the buildingSMART Linked Data working group (Beetz et al. [2]**,** Terkaj and Sojic [16]), the state of these developments at the time of research was not deemed sufficient to support the custom ontologies present in the framework.

In order to still be able to link the information in the building model to the ontology awaiting the arrival of industry-wide accepted ontology standards, a temporary solution has been adopted to support testing of the platform. The test building is modelled in Grasshopper (Rutten [13]) and custom components are used to add non-geometrical information to the design and export the compounded information as objects in JavaScript Object Notation (.json) files. In this file the object types are directly related to the node names of the ontology, and the property names directly align with the names of the properties in the ontology. The implementation of standard ontology compatible formats is recommended for further research.

4.4 Performance indicator definition

Performance indicators are defined in a similar way as analysis steps: the user defines the desired outcome of the analysis in the form of an English comparative sentence. This approach is adopted because in current practice most requirements stem from building codes or contracts, which are text-based documents.

These sentences in these texts describing metrics to be determined can be entered directly, after which syntactic NLP is used to assign the WordNet synsets and relate the words to the ontology nodes and properties. Semantic NLP with a custom model is used to provide an initial estimate for the annotation which can be altered by the end-user to assure correct interpretation.

The main differences between the way analysis steps and performance indicators are entered are as follows:

- Analysis steps are stored as index to a file, performance indicators are stored as initial queries to start the analysis assembly process
- The content of the sentences must be comparative, as e.g. $A > B$
- Performance indicators are annotated with *subject*, *comparator* (>, <, =!=, etc) and *object* marks, instead of the *input*, *output* and *applicability* annotators of analysis steps

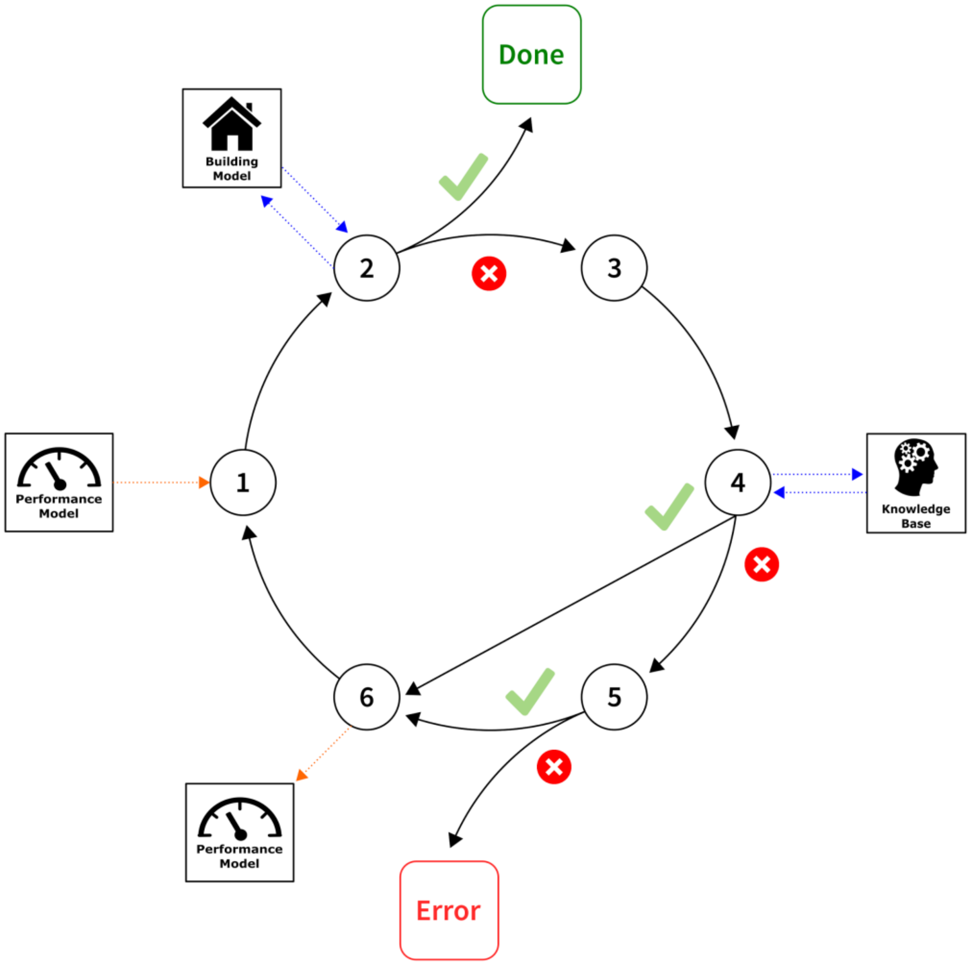

Figure 3. Abstract representation of the analysis assembly process

4.5 Analysis assembly

With an ontology, building model and performance indicator selected and a knowledge base filled with a certain amount of analysis steps, the analysis can be assembled. A custom backward-chaining reasoning engine is developed that works on a basis of iterative querying. The process, abstractly depicted in figure 3, is as follows:

1) A query is formulated based on the output metrics required, as formulated in the performance indicator entry module
2) The query is run against the building model to assess whether the required information is already present, for the model as a whole or certain parts of the model:
    - If all required information is retrieved for a certain object in the building model, the analysis is considered finished for this object and stored as a closed chain,
    - otherwise steps 3 to 6 are executed
3) The query is updated, removing the information that was found in the building model in step 2
4) The query is run against the knowledge base, looking for analysis steps that have the required information as output and are applicable to the objects in the building model that are specified in the performance indicator sentence:
    - If one applicable step is found, it is added to the analysis in the new analysis generation, and the process continues at step 6
    - If multiple applicable steps are found, one is selected at random and added to the analysis, whilst all found alternative steps are added to the new generation. The

process continues at step 6. Non-random selection of steps is expected to increase the efficiency of analysis assembly, and is recommended for further research
    - If no applicable steps are found, the path is considered a 'dead end' and step 5 is executed
5) Backtracking is applied, tracing the chain of steps through previous generations looking for the most recent generation in which there was an alternative step to the one selected:
    - If an alternative step is found, all generations that followed are removed from the stored analysis and the found alternative is added to the analysis. The process continues at step 6.
    - If no alternative step is found, the knowledge base is considered not to contain enough knowledge to provide the performance metrics. The analysis assembly is terminated and an error message is presented to the end-user
6) The state of the required performance indicators is updated, adding the inputs of the analysis steps retrieved from the knowledge base in step 4 and/or 5. The process restarts at step 1.

5. Conclusions and recommended further research

This paper has explored the potential of knowledge-based engineering principles to promote knowledge retainment and -sharing, as well as enable transparent automated analysis in the AEC industries. Natural language processing techniques and ontologies have been studied as means to provide intuitive interaction with- and increased performance of KBE frameworks. A concept and architecture for a knowledge-based automated analysis framework has been proposed and a prototype framework is developed for use in building design and engineering.

At the time of writing, initial tests have been performed to demonstrate the feasibility of the framework. The coherence between the ontology and the user inputs as well as the correct use of semantic models to translate sentences into knowledge base indexes and queries have shown to be critical to the correct assembly of an analysis. Preliminary results show that, given the correct ontology adherence and semantic interpretation, meaningful analyses can be assembled.

Further studies are recommended to be performed to validate the performance and stability of the framework in its current form. As the balance between the different framework components is already showing to be delicate, the formulation of standards and standard formats for AEC ontologies, ontology-complicit building models and annotators is considered beneficial to the framework performance. Implementation of e.g. IfcOWL is considered an interesting initial area of further research.

Additionally exploration of alternatives to WordNet for assigning unique definitions to words, ontology nodes and ontology properties is deemed essential to the scalability of the framework. Use of dereferencable URLs, as used in f.i. Linked Data initiatives, may greatly increase the amount of concepts that can be defined and can allow for more subtle disambiguation.

Different possibilities of marketing the framework need to be considered; as the submittance of knowledge may not seem in the interest of the expert certain incentives need to be provided. The nature of these incentives can greatly influence the way the framework can be brought to market or for instance be open-sourced.

Acknowledgements

This research was performed at the BEMNext Lab of the Civil Engineering faculty of Delft University of Technology, with financial support as well as expertise was kindly and amply provided by White Lioness technologies in Amsterdam. Sincere gratitude is hereby extended to prof. dr. ir. J.G. Rots and dr. ir. K.C. Terwel for their council during the research project, and all friends, family and colleagues for their support throughout the research process.

References

[1] Bates M., Models of natural language understanding, *Proceedings of the National Academy of Sciences*, 1995; **92.22**; 9977-9982.

[2] Beetz J., van Leeuwen J. and de Vries B., IfcOWL: a case of transforming EXPRESS schemas into ontologies, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2008; **23**; 89-101

[3] Bird S., Loper E. and Klein E., *Natural Language Processing with Python*, O'Reilly Media Inc., 2009

[4] Cambria E. and White B., Jumping NLP Curves: A Review of Natural Language Processing Research, 2014; **9.2**; 48-57

[5] Cyganiak R., Wood D., Lanthaler M., Klyne G., Carroll J.J. and McBride B., *RDF 1.1. Concepts and Abstract Syntax*, W3C, 2014

[6] Dalkir K. and Liebowitz J., *Knowledge Management in Theory and Practice* (2nd rev.), MIT Press, 2011

[7] Fellbaum C. and Miller G., *WordNet: An Electronic Lexical Database*, MIT Press, 1998

[8] Hjelseth E. and Nisbet N., Exploring semantic based model checking, *Proceedings of the CIB W78*, 2010; **27**

[9] La Rocca G., Knowledge based engineering techniques to support aircraft design and optimization, Delft University of Technology, 2011

[10] Manning C.D., Bauer J., Finkel J., Bethard S.J., Surdeanu M. and McClocky D., The Stanford CoreNLP Natural Language Processing Toolkit, *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014; **52**; 55-60

[11] Milton N.R., *Knowledge Technologies*, Polimetrica s.a.s., 2008

[12] Niemeijer R., de Vries B. and Beetz J., Freedom through constraints: User-oriented architectural design, *Advanced Engineering Informatics*, 2014; **28.1**; 28-36

[13] Rutten D., Grasshopper 3D Rhinoceros plug-in, www.*grasshopper3d.com*, Robert McNeel & Associates

[14] Stenetorp P., Pyysalo S., Topi G., Ohta T., Ananiadou S. and Tsujii T., BRAT: a Web-based Tool for NLP-Assisted Text Annotation, *Proceedings of the Demonstrations at the 13th Conference of the Association for Computational Linguistics*, 2012; **13;**102-107

[15] Stokes M., Managing Engineering Knowledge: MOKA, Methodology for Knowledge Based Engineering Applications, Professional Engineering Publishing, 2001

[16] Terkaj W. and Sojic A., Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology, *Automation in Construction*, 2015; **5**