Information system support in construction industry with semantic web technologies and/or autonomous reasoning agents

P. Pauwels & R. De Meyer
Department of Architecture and Urban Planning
Ghent University, Ghent, Belgium

J. Van Campenhout
Department of Electronics and Information Systems
Ghent University, Ghent, Belgium.

ABSTRACT: Information technology support is hard to find for the early design phases of the architectural design process. Many of the existing issues in such design decision support tools appear to be caused by a mismatch between the ways in which designers think and the ways in which information systems aim to give support. We therefore started an investigation of existing theories of design thinking, compared to the way in which design decision support systems provide information to the designer. We identify two main strategies towards information system support in the early design phase: (1) applications for making design try-outs, and (2) applications as autonomous reasoning agents. We outline preview implementations for both approaches and indicate to what extent these strategies can be used to improve information system support for the architectural designer.

1 DESIGN DECISION SUPPORT TOOLS FOR THE ARCHITECTURAL DESIGNER

1.1 *Current information system support for the designer*

Many design decision support tools have been designed and implemented for the domain of architecture, engineering and construction (AEC). These tools can be categorized in archive tools, modeling tools, calculation tools, and visualization tools. Applications in these categories are diverse in their design and implementation approaches, yet they often show similar shortcomings when they are evaluated in a real-world context: *a malfunctioning information flow*. The information that can be described within modeling applications is either ‘not enough and too simple’, or ‘too much and too complex’. The functionality provided by simulation software is ‘not correct’ or ‘irrelevant’. The visualization communicated by visualization software is ‘not clear’ to the end user. And archive applications typically contain only the information one ‘does not need’. Additionally, none of the applications effectively reuses information from any of the other applications. Notwithstanding the significant amount of effort put into the design and implementation of applications for the AEC domain, these issues return time and again in the evaluation of software applications and their usage in the AEC domain.

As a result, many experts involved in architectural design and construction often fall back on traditional support tools, such as paper-based sketching, simplified simulation models, and so forth. Even when experts decide to use available computer applications for design and decision support, they often use these applications in rather pragmatic and traditional ways. Computer-aided design (CAD) applications, for instance, are more often used as ‘computer-aided drafting’ environments. Instead of building complete information models for obtaining exact simulation results, information models are completed when project requirements are already met and simulations can only alter details in the design. On-topic databases with detailed information and sophisticated search functionalities are largely unused, and instead, architectural designers rely on simple keyword searches and the images obtained through these searches.

1.2 *The power of interpretation*

The amount and diversity of information is one of the most notable characteristics of a project in the AEC domain. Many domain experts with different back-

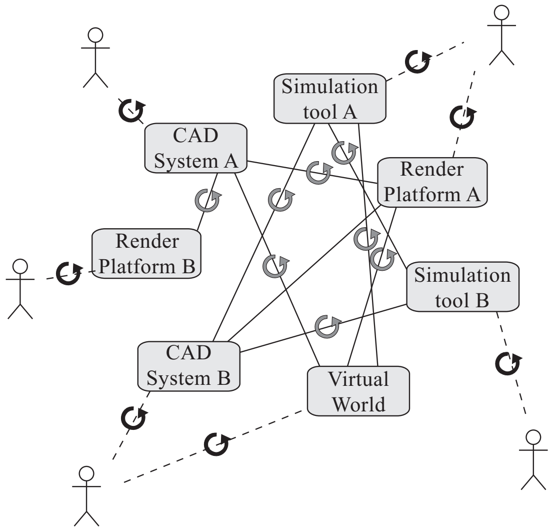


Figure 1: Interface points are points where information is interpreted from one schema into the other, both between human users and information systems (in black) and between information systems (in grey).

grounds typically meet within the context of a building project, each of them composing a personal understanding of the building design and providing with this personal understanding a specific contribution to the project. Additionally, each of these experts relies on diverse software tools. This situation causes a multiplication of the number of information schemas at play in a project. Since these information schemas are all part of one and the same project, a lot of *information flows* with an equal number of *interface points* emerge between these information schemas. This results in a situation as depicted in Figure 1. Information flows connect the diverse 'information managers' of the project, which are both human users and information systems. Crucial in this context of continuous information flows are the *interface points* where two understandings come together. Information is interpreted from one understanding or information schema into another in these interface points, thereby making them sensitive to misconceptions or 'mistakes' because of the possible misunderstanding. These interface points thus supposedly lie at the basis of the 'malfunctioning information flow' identified above.

Diverse strategies can be imagined to address this situation. For information exchange between information systems, several strategies are outlined by Pauwels et al. (2011). These strategies can be compared to the ways in which human users exchange information among each other. Suggested strategies are:

- Sharing information in the wild: one-to-one conversion for each interface point in Figure 1.

- Back to the roots: exchange information in a more general format or schema, such as the schema of the 3D kernel or schema used throughout the software suite (Pauwels et al. 2011)

- Central information structure: provide a central information structure that is to be used by every partner, as it happens in the building information modeling (BIM) approach (Eastman et al. 2008)).

- Linked data approach: provide a web of interlinked information that explicitly and unambiguously combines all information structures.

What remains missing in these strategies, however, is the element of *interpretation*. In each of the environments or 'information managers' depicted in Figure 1, namely, resides a unique description or conceptual model of the same building design. The four strategies outlined above suggest to address this situation by somehow assigning a human user with the task of describing a mapping between these conceptual models. The kind of 'knowledge' embedded in information systems and their mappings are static interpretations of information. If these interpretations are not updated to one's changing understanding on a regular basis, corresponding applications are rapidly outdated.

We argue that only the element of interpretation can enable information managers to keep up with such a changing understanding. Therefore, the outlined strategies can only provide limited help. Different continuously changing conceptual models about the same physical object(s) remain present in diverse environments. The lack of exact overlaps between concepts in these models or interpretations makes it near to impossible to efficiently combine or relate these models.

1.3 *How to improve information system support for the designer*

As was indicated in Pauwels et al. (2012), Peirce's process of inquiry and his understanding of abductive reasoning (Peirce 1958) provides a possible explanation for the element of interpretation. His theory is an important theory of human cognition that appears to suggest a reconciliation of the context of discovery with the context of justification (Pauwels et al. 2012, Peirce 1958, Aliseda 2004), a distinction that was earlier made by Reichenbach (1938). A good discourse and illustration of Peirce's process of inquiry is documented by Flach and Kakas (2000). Nowadays, it is more or less accepted that Peirce distinguishes three types of reasoning, namely abductive, deductive, and inductive reasoning. According to Peirce, reasoning should not be limited to a 'correct' or 'rational' kind of reasoning solely (deduction and induction), but instead it should reflect a combination of all possible thought processes of the human mind (including abduction). In comparison with traditional viewpoints, Peirce thus suggests the addition of a third, abductive kind of reasoning, which encompasses one's ability

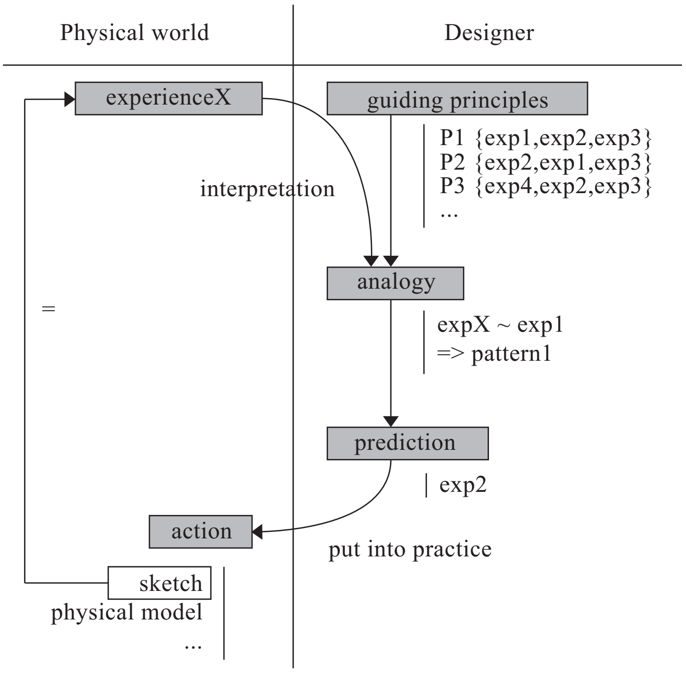


Figure 2: Possible outline of the design process.

to generate hypotheses about the world and to choose one of them as a possible explanation. This explanatory or hypothetical reasoning can be considered as the cornerstone of interpretation.

Recent studies of Peirce's theory try to reassess to what extent the diverse aspects or stages in Peirce's process of inquiry can be subject to a formal treatment (Flach and Kakas 2000, Aliseda 2006, Fischer 2001, Paavola 2006). If this is the case, an alternative strategy that includes 'interpretation' might be within reach for the issue of a malfunctioning information flow outlined above (Fig. 1).

2 DESIGN DECISION SUPPORT: THE RIGHT INFORMATION AT THE RIGHT TIME?

Peirce's process of inquiry was compared with diverse theories in design thinking in Pauwels et al. (2012). Theories in design thinking are first summarized into the schema depicted in Figure 2 and the following description of the design process.

"*The design process proceeds by making analogies between encountered situations in the physical world and guiding principles in the human mind. The resulting analogies can be considered the designer's interpretations of encountered situations. By making an analogy, the designer hypothesizes and predicts that the rest of the familiar pattern also applies to the encountered situation. In other words, new knowledge is created by the analogy. The designer finally tests the prediction made, thereby creating a new situation or experience. This either confirms or refutes the original analogy. When refuted, an alternative analogy is sought. When confirmed, the pattern is added to the background knowledge, thereby indirectly changing the guiding principles of the designer.*" (Pauwels et al. 2012).

The concept of guiding principles relates to Lawson's theory of how designers think (Lawson 2005a). Guiding principles are in his theory understood as the background knowledge or the knowledge by experience of a designer. This is obviously a lot broader than mere objective or factual information. It also contains subjective information, including typically intangible or tacit concepts (Polanyi 1958, Polanyi 1966) such as beliefs, values and attitudes. The concept of analogical reasoning relates, in the context of design, to Goldschmidt's theory of the dialectics of sketching (Goldschmidt 1991). In this theory, Goldschmidt indicates how designers typically use sketches for making new interpretations or analogies on the considered matter, and thus for creatively producing new ideas and knowledge.

The schema in Figure 2 is compared with Peirce's process of inquiry in Pauwels et al. (2012). This results in the schema depicted in Figure 3, which is the same schema as in Figure 2, but with an additional overlay. One can see how guiding principles remain central to the overall reasoning process. These guiding principles steer the overall design thinking process through abductive, deductive and inductive reasoning.

Two valid strategies towards supporting the architectural design process with information systems are indicated in Pauwels et al. (2012):

1. information systems as environments for making design tryouts;

2. information systems as autonomous reasoning agents that function as advisors.

The first approach extends the set of tools available to designers for producing design tryouts. As such, this approach can be considered similar to the kind of tools typically used by designers nowadays. The second approach is a radically different approach, in which an autonomous reasoning agent gains or constructs knowledge independently by following the process of inquiry as outlined in Figure 3. This approach aligns with the third situation outlined in Lawson (2005b), in which information systems function as agents or personal advisors of the designer. This approach assumes that Peirce's process of inquiry is a valid theory for architectural design thinking and that this approach can indeed be formalized, as was suggested before (Flach and Kakas 2000, Aliseda 2006, Fischer 2001, Paavola 2006). These assumptions are considered in Pauwels et al. (2012), resulting in the anticipation of important barriers for this second approach.

In the remainder of this paper, we want to document both approaches in more detail, including their respective implementation strategies and example applications. By doing so, we hope to give an indication of their capacity to improve decision support for architectural design thinking. We indicate in

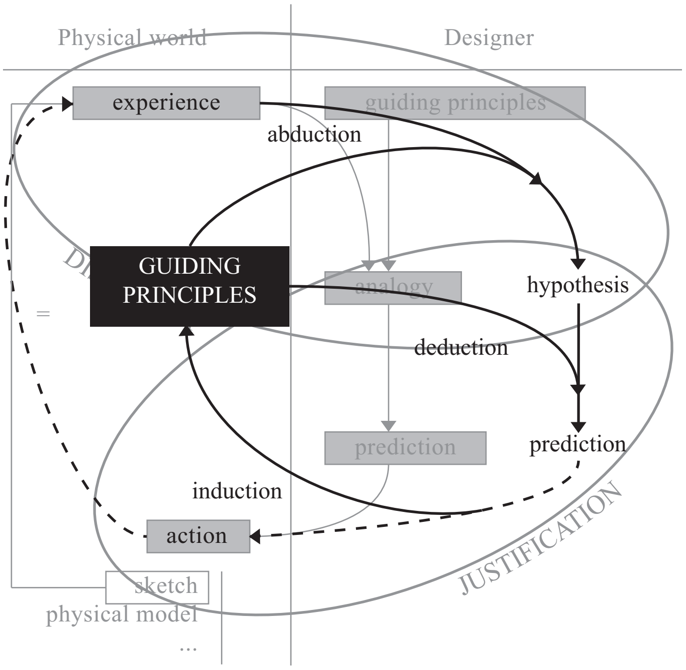


Figure 3: The intertwining of abductive, deductive and inductive reasoning in the context of design thinking.

which ways newly emerging semantic web technologies (Berners-Lee et al. 2001) could further enhance such improvements.

3 THE ROLE OF SEMANTIC WEB TECHNOLOGIES

3.1 *What are semantic web technologies*

Extensive documentation of semantic web technologies exists elsewhere (Berners-Lee et al. 2001, Brickley and Guha 2004, Grant and Beckett 2004, Manola and Miller 2004, McGuinness and van Harmelen 2009, W3C 2012, Bizer et al. 2009, Cyganiak and Jentzsch 2011). Therefore, we will only give a brief outline of these technologies, in which the main principles and ideas are explained.

The semantic web was conceived and suggested by Berners-Lee et al. (2001) as the successor of the existing World Wide Web (WWW). In this successor, all information would supposedly be described in a language that can be ‘understood’ by both humans and computer applications. Because the WWW contains information about almost any possible concept in the world, the language describing this information cannot follow one domain-specific schema. Instead, a flexible and generic language is needed that allows to describe and easily link information from different knowledge domains together.

Therefore, the semantic web was conceived as a semantic network that describes the meaning of its concepts through a directed labeled graph (Fig. 4) based on description logic (DL) (Baader and Nutt 2003). Each node in this graph represents a concept or object in the world and each arc in this graph represents the logical relation between two of these concepts or objects. In this way, the graph represents a set of logic-based declarative sentences. By describing information in a single directed labeled graph, a uniform description of information is targeted, both syntactically and semantically, making information reusable by both humans and computer applications. Today, the development of the semantic web is mainly led by the World Wide Web Consortium (W3C), significantly supported by the actors stemming from various corners, including both research institutes and industrial partners (W3C 2012). This is resulting in a web of Linked (Open) Data (LOD) (Cyganiak and Jentzsch 2011) that is supposedly superseding the borders of individual applications.

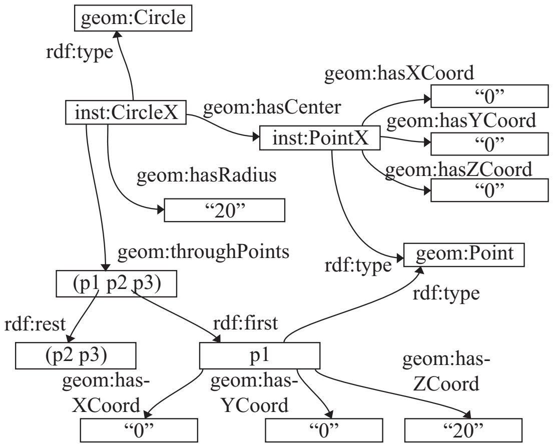
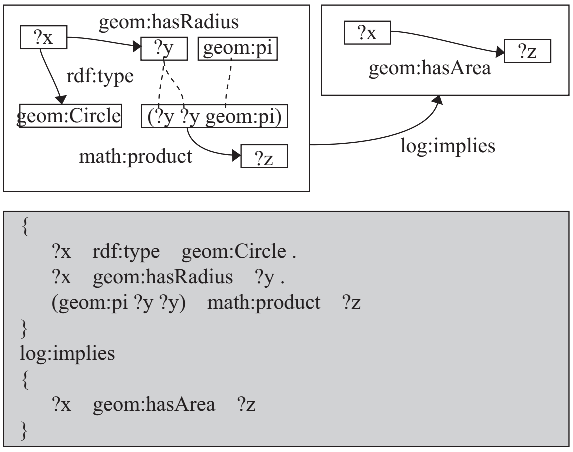


Figure 4: A directed labeled graph that describes basic three-dimensional information.

Semantic web technologies, and by extent also the LOD cloud, use the Resource Description Framework (RDF) as a language to represent graph structures (Grant and Beckett 2004, Manola and Miller 2004). These graph structures are generally referred to as RDF graphs. RDF graphs can be given an improved semantic structure using RDF vocabularies or ontologies. The most basic elements to describe such ontologies are available in the RDF Schema (RDFS) vocabulary (Brickley and Guha 2004). More expressive elements for describing ontologies are available within the Web Ontology Language (OWL) (McGuinness and van Harmelen 2009), which uses RDFS as a subset. The RDF graphs constructed with OWL concepts are called OWL ontologies, and they can be used as an available vocabulary when making other, more complex RDF statements.

The available RDFS and OWL concepts enable only a basic, standard reasoning, limited to a certain level of complexity. When more complex reasoning is necessary, one should describe rules with a more dedicated rule language. Using a specific rule language, one is able to define custom rules and subsequently use them in a rule-based reasoning process. Several rule languages have been developed to express such rules. Three of the most notable initiatives in the semantic web domain are the Semantic Web Rule Language (SWRL) (Horrocks et al. 2004), the Rule Interchange Format (RIF) (Kifer and Boley 2010) and N3Logic (Berners-Lee et al. 2008). These rule lan-



```
{
    ?x   rdf:type   geom:Circle .
    ?x   geom:hasRadius   ?y .
    (geom:pi ?y ?y)   math:product   ?z
}
log:implies
{
    ?x   geom:hasArea   ?z
}
```

Figure 5: N3Logic rule in its normative N3 notation, describing how the area of a circle can be inferred from the radius property of this circle.

guages aim to do for rules what RDF does for data: to provide a common data model so that this information becomes globally sharable. In other words, any reasoning engine in the semantic web (e.g. CWM (Berners-Lee 2009) or EYE (De Roo 2012)) should be able to understand a rule described in such a rule language and apply it within its proper situation and environment. An example rule described in N3Logic can be found in Figure 5.

3.2 *Semantic web technologies for the design tryout tools strategy*

When following the first application development approach outlined above, applications are developed as tools which can be used by architectural designers as tools for making design tryouts. In this case, the actual reasoning takes place in the mind of the designer and the designer can rely on additional tools for design tryouts. In addition to the traditional paper and pencil, for instance, the designer can build CAD models, simulation models, real-time visualizations, and so forth.

We argue that the usage of semantic web technologies in this approach reduces the loss of information as explained in Figure 1, because of the following reason. Semantic web technologies enable combining the descriptions of information used in the outlined applications (modeling, calculation, visualization and archive applications). These descriptions relate to their application-specific description schemas, which are described in OWL ontologies, and are mutually interconnected by specific semantic relations (Figure 6). As such, these technologies enable one to describe with only one language (RDF) a web in which distinct information structures describing the same building model can co-exist with respect for the inherent semantics and syntax of each of these subgraphs. This allows a better information management for the designer, consequently leading to an improved environment for making design tryouts.

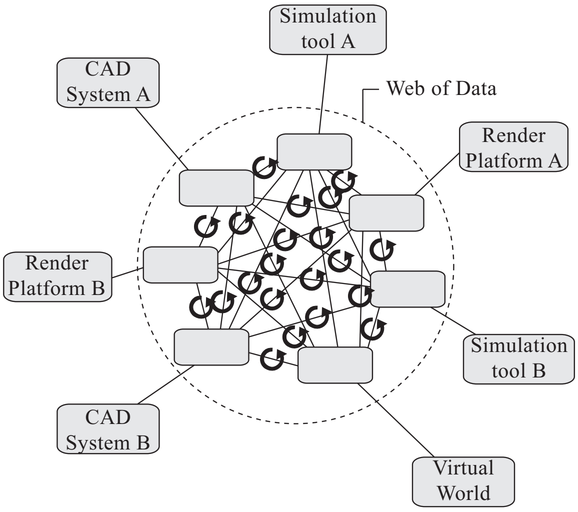


Figure 6: Semantic web technologies allow information to be linked on a data level, making them accessible from within any application.

The element of interpretation, however, is in this approach not addressed differently from existing approaches. As is indicated in Figure 6, the fundamental issue of how to translate between object syntax and semantics is merely shifted to another level, namely the data level. Instead of having to worry about how data is imported to and/or exported from an application, one now needs to worry about how to link the diverse information schemas together into a comprehensible linked data web. In conclusion, information still needs to be translated from one schema into the other, resulting in a loss of information, but an improved information management can be provided to designers, supposedly leading to more efficient design processes.

3.3 *Semantic web technologies for the reasoning agent strategy*

The usage of semantic web technologies, alternatively, may give access to reasoning engines that are not only capable of deductive reasoning, but also of inductive and abductive reasoning. Research in the semantic web domain is, namely, turning to these reasoning processes as well (Elsenbroich et al. 2006, Sensoy et al. 2011, d’Amato et al. 2010). Consequently, semantic web technologies might allow one to let an autonomous reasoning agent run through the three reasoning processes outlined by Peirce and accordingly make modifications to its ‘knowledge’, which is described as an RDF graph. In this setting, the information structure evolves step by step through every single observation made by the reasoning agent. By combining the three reasoning modes in a continuously ongoing cyclic process instead of focusing on each of these reasoning modes separately, one might theoretically be able to develop an information system that is able to make hypotheses, make predictions, devise design tryouts and learn, all based on the observations the system continuously goes through. A similar approach is suggested and used in the ‘Robot

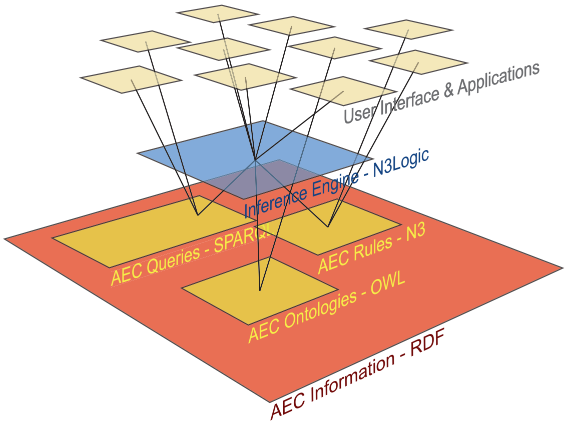


Figure 7: Diverse layers of additional functionality can be built on a common layer of information. As such, the same source of information can be used in various contexts and applications.

Scientist' project documented in Ray (2007, Ray et al. (2009, King et al. (2009, King (2011). Since Peirce's process of inquiry is supposedly also at play in other application domains, such as design thinking, a similar approach could theoretically also be developed for design thinking support.

This kind of support would be completely different from any traditional kind of support by information systems. Similar to how it happens in the robot scientist project, the reasoning system would evolve into a relatively independent agent capable of reasoning about a design situation, and it would thus not directly interfere with reasoning processes of the human designer. The main support it could give to a designer would presumably be similar to how any designer gives support to any other designer, namely by simple dialogue and discussion of design alternatives, from which both make their own interpretations and start their own reasoning processes.

4 APPLICATIONS FOR DESIGN TRYOUTS: IMPLEMENTATION RESULTS

For the first implementation strategy, we suggest to rely on one central web of linked data, and build diverse applications on top of this web with which the designer can interact. When relying on semantic web technologies, this results in a situation as depicted in Figure 7. This setting includes the following layers:

- a layer of basic AEC information in RDF graphs;
- a layer of more complex information, such as OWL ontologies, rules in N3Logic, and SPARQL constructs;
- a layer of information that can be inferred by inference engines;
- a layer of the actual end user applications.

To test this approach, we started to combine diverse RDF graphs describing information from different sources or application domains in the AEC sector. One of these sources is the IFC-to-RDF service (UGent Multimedialab 2012), which converts building models described with the Industry Foundation Classes (IFC) (Liebich et al. 2012), into an IFC/RDF graph. Instances from this graph were connected to instances from within the LOD cloud and to new instances created with custom ontologies (Fig. 8).

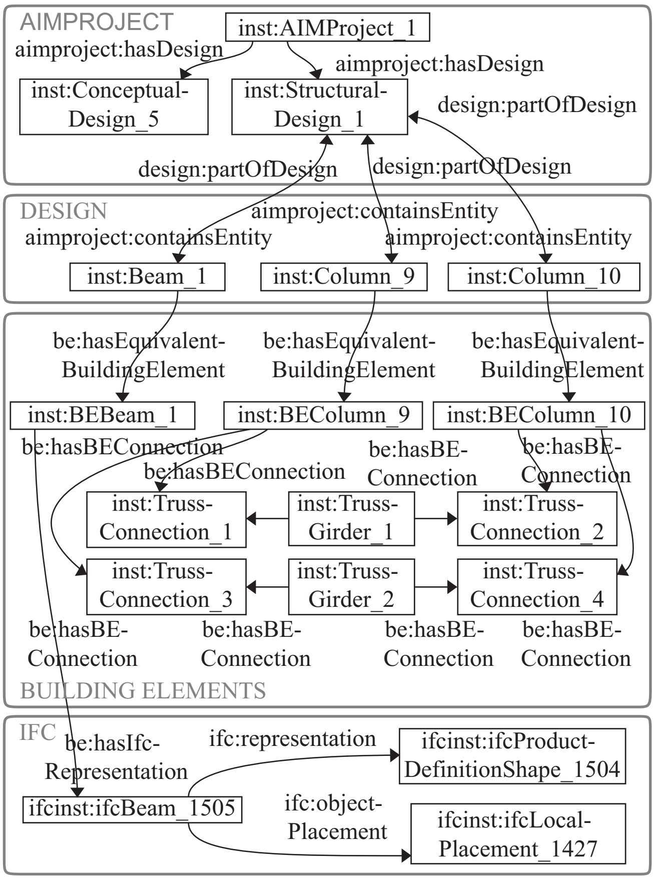


Figure 8: Connecting multiple domains together using semantic web technologies.

Numerous applications can be built on top of this web of linked data. This includes archive applications, modeling applications, calculation applications and visualization applications. Precisely because of the ability to manage and connect data explicitly on a data level, possibilities for information management and exchange notably improve. In the context of Peirce's theory, the resulting applications are tools for making *design tryouts*, based on which a human mind can do the deductive, inductive and abductive reasoning. A design tryout in such an application serves as nothing more than a new experience, similar to sketches, discussions, physical models or even complete buildings. From this experience, a new reasoning cycle of abductive, deductive and inductive reasoning starts anew.

5 AUTONOMOUS REASONING AGENTS: IMPLEMENTATION RESULTS

5.1 *Experimental setup*

The second implementation strategy outlined above suggests building autonomous reasoning agents that construct information from scratch by continuously going through the reasoning cycle by themselves. In this approach, all three reasoning modes outlined by Peirce are implemented and combined in a dynamic information system. We have done an exploratory experiment to find out where this approach may lead to when relying on semantic web technologies. This experiment targets a simulation of the reasoning process behind an observation. This topic is, namely, often chosen to indicate the role of this reasoning cycle in enabling interpretation.

Note that, because of the exploratory character of the theory and the experiment, we are not able to present here a knowledge base that somehow represents all kinds of experiences in an intricate web of data and rules. We can only make presumptions about how a very small part of this knowledge base might look like and how it may be used by autonomous reasoning agents. We nonetheless try to make the experiment resemble a realistic situation as much as possible.

In the experiment, we consider an observing reasoning agent who is located in an unfamiliar space that is bounded by a wall. In the observation experiment, the observing reasoning agent notices that this space feels warm, and the agent tries to find out the main cause of this notice. In other words, the agent tries to explain or interpret her notice of a warm space, using personal knowledge or guiding principles. Considering our usage of semantic web technologies, our implementation of this context includes the three following elements:

- an RDF graph that describes the background knowledge of the reasoning agent;
- an RDF graph that describes the newly encountered situation;
- a reasoning engine capable of simulating all three reasoning modes.

The first two elements can initially be handled with simple RDF instance graphs, and for the third element, we are relying on the EYE reasoning engine (De Roo 2012). Note that the way in which experiences are recorded in a personal background knowledge will most probably have to be reconsidered in future stages of research.

For this experiment, we have described a set of 100 disjoint experiences of similar spaces (Fig. 9). Each experience is hereby described by a specific combination of the attributes `cold` / `warm`, `windows` / `no windows`, `heater` / `no heater`, `insulation` / `no insulation` (Fig. 9).

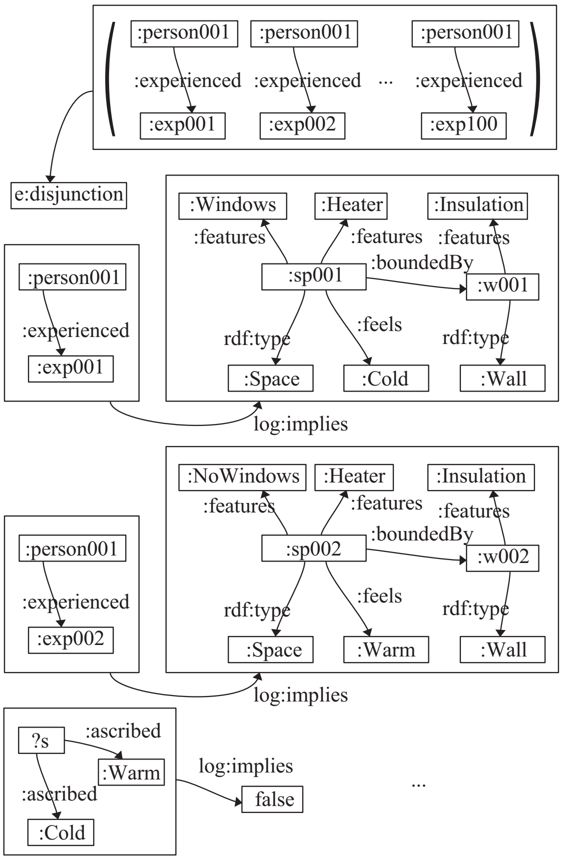


Figure 9: The set of experiences as it is stored and used by the reasoning agent.

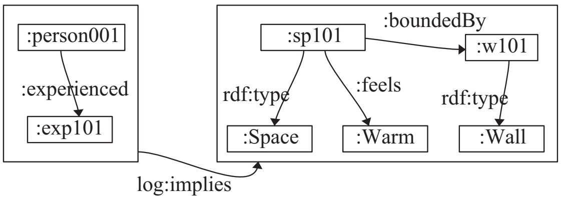
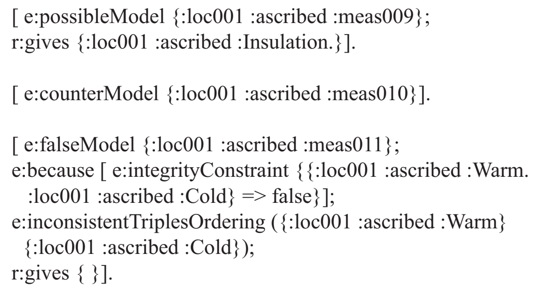


Figure 10: The experience of the observing reasoning agent of being in a `space` enclosed by a `wall` and feeling `warm`.

5.2 *The reasoning process*

Following the schema depicted in Figure 3, the reasoning agent starts with an abductive reasoning step. This abductive reasoning step combines the personal background knowledge, memory or guiding principles of the reasoning agent (Fig. 9), and the experience that is to be interpreted. As indicated above, this new experience is in this case a `space` that is enclosed by a `wall` and that feels `warm`. This experience is described in an RDF graph as shown in Figure 10. Both information sources are passed to the EYE reasoning engine when starting the abductive reasoning process that is supposed to generate the possible explanations for the current experience.

The abductive reasoning process is started by sending a query to the reasoning engine, in which we manually pass the explanations to be considered, e.g. the presence of `insulation`. In real world situations,



```
[ e:possibleModel {:loc001 :ascribed :meas009};
r:gives {:loc001 :ascribed :Insulation.}].

[ e:counterModel {:loc001 :ascribed :meas010}].

[ e:falseModel {:loc001 :ascribed :meas011};
e:because [ e:integrityConstraint {{:loc001 :ascribed :Warm.
  :loc001 :ascribed :Cold} => false}];
e:inconsistentTriplesOrdering ({:loc001 :ascribed :Warm}
  {:loc001 :ascribed :Cold});
r:gives { }].
```

Figure 11: Three of the 100 models generated by the EYE reasoning engine after querying whether or not `insulation` could be a valid explanation for experiencing a `warm space`. The examples include a `possibleModel`, a `falseModel` and a `counterModel`.

these explanations appear to pop up automatically from the background knowledge of the observer. Ideally, we should thus eventually generate these queries automatically from the memory of experiences of the observer. From this query, the reasoning engine starts an analysis of the available knowledge (Fig. 9) and matches the new experience (Fig. 10) with the previous experiences in this knowledge. By doing so, the reasoning engine infers to what extent the explanation suggested in the query can really form a valid explanation for the current observation.

Documenting the full details of this reasoning process would lead us too far in this paper, but essentially, the reasoning engine eventually produces as much ‘models’ as there are previous experiences, in this case 100. In each of these models, an experience is compared with the current situation and an analysis is made of whether or not the model in question may provide for a good explanation or interpretation of the current experience. Three of these models are displayed in Figure 11, illustrating the three kinds of models that can be found in the output: `possibleModels`, `falseModels` and `counterModels`. A `possibleModel` refers to an experience that is (1) similar to the current experience (a `warm space` - Fig. 10) and (2) that confirms the explanation in the query (`insulation`). A `falseModel` refers to an experience that is not similar to the current experience (a `cold space` is inconsistent with a `warm space` because of our understanding that something cannot be `cold` and `warm` at the same time - Fig. 9, bottom). Such `falseModels` are immediately disregarded in the further abductive reasoning process. A `counterModel` refers to an experience that (1) is similar to the current experience (a `warm space` - Fig. 10), but (2) that does not confirm the explanation in the query (no `insulation`).

By manually passing several possible explanations to the reasoning engine, e.g. there is a `heater`, there are `windows`, and so forth, each time 100 explanatory models are produced. By analyzing these models, one can obtain the probability of each of these explanations for the perceived experience:

$$belief(Q_i|E) = \frac{P_i}{P_i + C_i} \qquad (1)$$

where Q_i = the considered query or explanation; E = the perceived experience; P_i = the number of produced `possibleModels`; and C_i = the number of produced `counterModels`.

In our example, this led to a belief of 0.84 in `insulation` as a possible explanation, a belief of 0.58 in `windows` as a possible explanation, and a belief of 0.54 in `heater` as a possible explanation. From this overall process, the reasoning engine may now conclude that `insulation` is the most probable interpretation of the current experience.

This simplified abductive reasoning process resembles a human mind that relies on previous experiences to come up with a reason behind a certain new observation. After this reasoning process, one of the models is chosen as an explanation or interpretation of the observation, and the corresponding consequences are reflected on the situation at hand: there is most probably `insulation` in the `wall` surrounding the observed `space`. As such, the observing reasoning agent temporarily brings a whole set of extra knowledge into consideration about the behavior expected from the observed world. This extra knowledge is similar to the extra knowledge typically emerging from analogical reasoning (Goldschmidt 1991). With this extra knowledge, the reasoning agent is able to deduce diverse statements about the observed world. For instance, a rule might be brought in, stating that any `insulated wall` has a `thickness` of at least `25 centimeters`. With the addition of this rule, the agent can infer that the `wall` bounding the `space` should at least be `25 centimeters` thick.

The prediction that was obtained in the deductive reasoning step can be tested in a separate experiment to confirm the original hypothesis. By comparing the predicted result with the observed result, the reasoning agent further confirms or rejects the original hypothesis that there is `insulation` in the `wall` and that this makes the `space` feel `warm`. From this inductive reasoning step, a rule is concluded and added to the observer’s background knowledge. We think that in reality, a less explicit, smoother modification of this background knowledge takes place. A possible approach would be to assign probability values to the induced rule, similar to what happens in the abductive reasoning process.

5.3 *Evaluation*

This first experiment appears promising in that it gives an idea of how the element of interpretation might find its way into information systems. It might consequently lead to design decision support systems that can provide very specific and specialized support, in contrast to existing applications, which give only limited external support in the form of environments for making design tryouts. However, the following significant question marks remain present as well, and need to be addressed in future work, when more realistic test cases in a semantic web context are within reach.

- In realistic environments, the three reasoning modes are not deployed independently from each other. Instead, a reasoning line in any reasoning mode is supposed to start from the conclusions resulting from a previous reasoning line in another reasoning mode, thereby creating a never ending reasoning cycle or learning cycle. From our initial discussion of these reasoning processes, it is somewhat clear *how the three reasoning modes should be combined into one cyclic process*. However, it remains to be seen to what extent this can be achieved in an implementation. How does the result of the abductive reasoning process start a deductive reasoning line? Should we consider multiple deductive reasoning lines to start from the accepted abductive hypothesis? Should we rather focus on smaller and faster reasoning cycles or more intense cycles, in which several reasoning lines are processed next to each other?

- In the complete reasoning cycle, *the role of design tryouts is extremely important*, because they provide the mechanism through which original interpretations are confirmed or refuted, and through which new experiences emerge. How these design tryouts, which typically include real-world interactions, are to take place in a computer environment, remains at question.

- In our experiment, we showed how, in the abductive reasoning process, a probability value can be calculated for certain hypotheses or explanations behind an observation / experience. *This probability value is very important, because it is the main parameter on which the rest of the reasoning cycle relies*. We also showed how, by making a design tryout, an observation / experience is essentially added to the memory of experiences of the reasoning agent. The next time a similar abductive reasoning process is started, the agent thus relies on an extra experience in this memory, hence changing the probability values calculated in following abductive reasoning lines. In other words, the probability value of the original hypothesis is changed indirectly, only by adding the observed design tryout to the memory of experiences. This implies that no rules are added explicitly to a memory after the inductive reasoning part, but only experiences as described in Figure 9 and 10. So then, where do the rules deployed in the deductive reasoning mode come from? Might they be inferred just-in-time from our memory of experiences when starting the deductive reasoning process? Maybe a completely different process is followed, in which every rule that is induced after a design tryout, is stored explicitly, but with an assigned probability value?

6 CONCLUSION

Research documented in this paper started from the observation that currently existing information technology support for the architectural design process suffers from a malfunctioning information flow. The information provided by the diverse applications typically does not align with the kind of information used and required by the architectural designer. We therefore investigated existing theories of design thinking and compared these to the way in which design decision support systems provide information to the designer. This pointed towards Peirce's process of inquiry as a possible explanatory theory for human design thinking.

With Peirce's theory and main theories in design thinking, one can see how the element of interpretation is missing in existing information system support and traditionally deployed implementation strategies. From this conclusion, we outlined two strategies towards information system support: (1) applications for making design tryouts, and (2) applications as autonomous reasoning agents. Whereas the first strategy is largely similar to the existing strategy for information system support, the second strategy is a radically different strategy. We gave a brief indication of how both strategies can be implemented with semantic web technologies, and how they could benefit from this implementation approach.

The strategy in which applications are considered as environments fit for making design tryouts can only provide limited support to the architectural designer. The improvements generated by the usage of semantic web technologies are therefore limited to an improved information management. The strategy in which autonomous reasoning agents provide support theoretically allows a far more interesting decision making support for the architectural designer. This promising character is mainly caused by the interpretative power of such reasoning agents. And this approach might become feasible because of the availability of semantic web technologies. However, even assuming that Peirce's theory is a good explanatory framework and that it is understood correctly, there are several significant barriers towards building such a system.

REFERENCES

Aliseda, A. (2004). Logics in scientific discovery. *Found. Sci. 9*, 339–363.

Aliseda, A. (2006). *Abductive reasoning: logical investigations into discovery and explanation.* Dordrecht: Springer.

Baader, F. & W. Nutt (2003). Basic description logics. In *Description logic hand-book: theory, implementation, and applications*, pp. 47–100. Cambridge: Cambridge University Press.

Berners-Lee, T. (2009). CWM - a general purpose data processor for the semantic web. Available online: http://www.w3.org/2000/10/swap/doc/cwm. Last accessed on 22 February 2012.

Berners-Lee, T., D. Connolly, L. Kagal, & Y. Scharf (2008). N3logic: a logical framework for the world wide web. *Theor. Pract. Log. Prog. 8*, 249–269.

Berners-Lee, T., J. Hendler, & O. Lasilla (2001). The semantic web. *Sci. Am. 284*, 35–43.

Bizer, C., T. Heath, & T. Berners-Lee (2009). Linked data - the story so far. *Int. J. Semant. Web. Inf. 5*, 1–22.

Brickley, D. & R. Guha (2004). RDF vocabulary description language 1.0: RDF schema. W3C recommendation. Available online: http://www.w3.org/TR/rdf-schema/. Last accessed on 22 February 2012.

Cyganiak, R. & A. Jentzsch (2011). The linking open data cloud diagram. Available online: http://lod-cloud.net/. Last accessed on 22 February 2012.

d'Amato, C., N. Fanizzi, & F. Esposito (2010). Inductive learning for the semantic web: what does it buy? *Semantic Web 1*, 53–59.

De Roo, J. (2012). Eulersharp - euler proof mechanism. Available online: http://eulersharp.sourceforge.net/. Last accessed on 22 February 2012.

Eastman, C., P. Teicholz, R. Sacks, & K. Liston (2008). *BIM handbook: a guide to building information modeling for owners, managers, architects, engineers, contractors, and fabricators.* New York: Wiley.

Elsenbroich, C., O. Kutz, & U. Sattler (2006). A case for abductive reasoning over ontologies. In *Proceedings of the OWLED*06 workshop on OWL: experiences and directions*.

Fischer, H. (2001). Abductive reasoning as a way of world-making. *Found. Sci. 6*, 361–383.

Flach, P. & A. Kakas (2000). Abductive and inductive reasoning: background and issues. In *Abduction and induction: essays on their relation and integration*, pp. 1–27. Dordrecht: Kluwer Academic Press.

Goldschmidt, G. (1991). The dialectics of sketching. *Creativity Res. J. 4*, 123–143.

Grant, J. & D. Beckett (2004). RDF test cases. W3C recommendation. Available online: http://www.w3.org/TR/rdf-testcases/. Last accessed on 22 February 2012.

Horrocks, I., P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, & M. Dean (2004). SWRL: a semantic web rule language combining OWL and RuleML. W3C member submission. Available online: http://www.w3.org/Submission/-SWRL/. Last accessed on 22 February 2012.

Kifer, M. & H. Boley (2010). RIF overview. W3C working group note. Available online: http://www.w3.org/TR/rif-overview/. Last accessed on 22 February 2012.

King, R. (2011). Rise of the robo scientists. *Sci. Am. 304*, 72–77.

King, R., J. Rowland, S. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L. Soldatova, A. Sparkes, K. Whelan, & A. Clare (2009). The automation of science. *Science 324*, 85–89.

Lawson, B. (2005a). *How designers think - the design process demystified.* London: Architectural Press.

Lawson, B. (2005b). Oracles, draughtsmen, and agents: the nature of knowledge and creativity in design and the role of it. *Automation in Construction 14*, 383–391.

Liebich, T., Y. Adachi, J. Forester, J. Hyvarinen, K. Karstila, K. Reed, S. Richter, & J. Wix (2012). Industry Foundation Classes IFC2x edition 3 technical corrigendum 1. Available online: http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/-html/index.htm. Last accessed on 22 February 2012.

Manola, F. & E. Miller (2004). RDF Primer. W3C Recommendation. Available online: http://www.w3.org/TR/rdf-primer/. Last accessed on 22 February 2012.

McGuinness, D. & F. van Harmelen (2009). OWL 2 Web Ontology Language document overview. W3C recommendation. Available online: http://www.w3.org/TR/owl2-overview/. Last accessed on 22 February 2012.

Paavola, S. (2006). *On the origin of ideas: an abductivist approach to discovery.* Ph. D. thesis, University of Helsinki.

Pauwels, P., R. De Meyer, & J. Van Campenhout (2012). A critical evaluation of information system support for architectural design thinking. *Des. Issues*. in press.

Pauwels, P., D. Van Deursen, J. De Roo, T. Van Ackere, R. De Meyer, R. Van de Walle, & J. Van Campenhout (2011). Three-dimensional information exchange over the semantic web for the domain of architecture, engineering and construction. *Ai. Edam. 25*, 317–332.

Peirce, C. (1958). *Collected papers of Charles Sanders Peirce, vols. 1-8.* Cambridge: Harvard University Press.

Polanyi, M. (1958). *Personal knowledge: towards a post-critical philosophy.* London: Routledge.

Polanyi, M. (1966). *The tacit dimension.* New York: Doubleday & company.

Ray, O. (2007). Automated abduction in scientific discovery. In *Model-Based Reasoning in Science, Technology and Medicine*, pp. 103–112. Berlin: Springer.

Ray, O., A. Clare, M. Liakata, L. Soldatova, K. Whelan, & R. King (2009). Towards the automation of scientific method. In *Proceedings of the international joint conference on artificial intelligence: workshop on abductive and inductive knowledge development*, pp. 27–33.

Reichenbach, H. (1938). *Experience and prediction: an analysis of the foundations and the structure of knowledge.* Chicago: University of Chicago Press.

Sensoy, M., A. Fokoue, M. Srivatsa, & J. Pan (2011). Probabilistic abox abduction in description logics. In *7th international workshop on scalable semantic web knowledge base systems (SSWS)*.

UGent Multimedialab (2012). IFC-to-RDF service. Available online: http://ninsuna.elis.ugent.be/IfcRDFService/. Last accessed on 22 February 2012.

W3C (2012). W3C semantic web activity. Available online: http://www.w3.org/2001/sw/. Last accessed on 22 February 2012.