Enabling formfinding design through online systems

Philipp JUREWICZ*, Roman IVANCSITSa

*Vienna University of Technology, Inst. of Interdisciplinary Construction Process Mgmt.
Josefstaedter Str. 29 / 63, A-1080 Wien, AUSTRIA
Philipp.Jurewicz@gmx.net

a Formfinder Software GmbH, Schlossgasse 22, 1050 Vienna, AUSTRIA

Abstract

This paper outlines patterns how to prepare existing architectural databases that are focused on lightweight structure for their integration in design tools like Formfinder, so architects and engineers can benefit from each other. An analysis phase shows six key factors and the first two are covered. For "Handling of domain disagreement in architecture" we describe some patterns based on Semantic Web technologies. For "Fuzziness of description" we describe our experience with a prototype influenced by Fuzzy Set theory. We finish with showing how the findings can be translated into a user friendly web interface.

Keywords: conceptual design, morphology, form finding, design software, web application, semantic web, fuzzy sets, classification, building shape, database.

1. Introduction

Form-active structure systems are structure systems of flexible, non-rigid matter, in which the redirection of forces is effected through particular form design and characteristic form stabilization (Engel [5]). They are determined by a relative limited set of attributes but they can evolve into a vast variety of architectural building shapes. The desktop software tool Formfinder has an online typology panel which connects the designer of form-active structures with a web based project database that can assists in the design process.

The goal of this paper is to show patterns how to prepare existing architectural databases that are focused on lightweight structure, so users can benefit from each other and be integrated in design tools like Formfinder. We finish with showing how the findings about the databases can be translated into a user friendly web interface.

One of the driving ideas behind the Formfinder project is to build a design feedback loop between the digital sketch created by the user in the Formfinder desktop application/viewport Figure 1 (A), a typology hosted on a web server that tries to interpret the digital sketch Figure 1 (B) and a project database that can be connected from the typology (C) (Figure 1). To achieve a connection between typology (B) and project database (C) a highly annotated system must be set up.

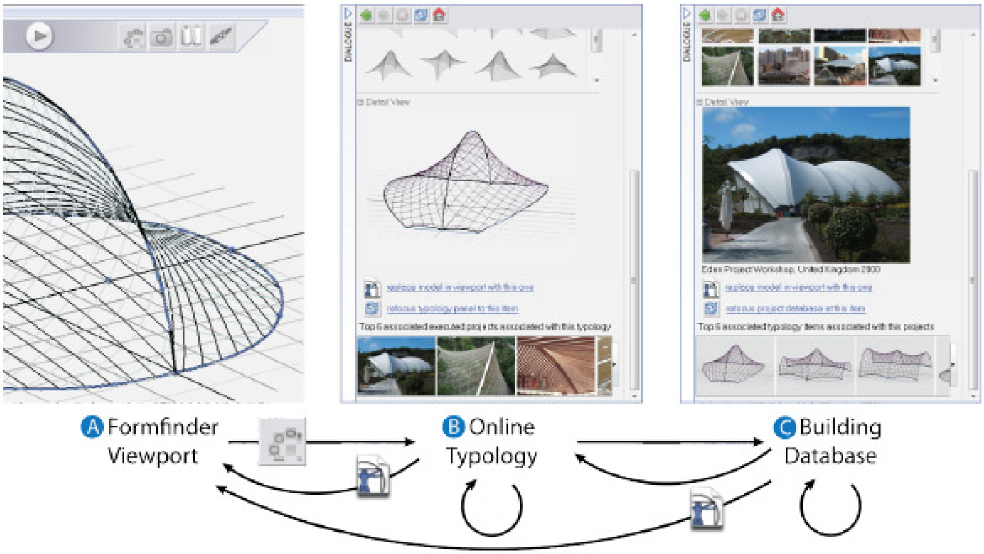

Figure 1: Feedback loop between desktop application (left) & online tools (middle & right)

1.1. Analysis phase

In a review and analysis phase we identified the following key factors which should be addressed in advance of the integration:

- Handling of domain disagreement in architecture
- Fuzziness of description
- Use of controlled vocabularies
- Significance of statements
- Breaking up of executed projects into parts
- Guided user interaction

Most of the identified factors are actually "knowledge management" issues, still we want to outline some information technologies in this paper to show techniques from other disciplines with some potentials for architecture and built environment researchers. Basedon the results of the analysis phase this paper will outline the first two key factors:

To address "Handling of domain disagreement in architecture" we looked into other disciplines and transferred a pattern from business knowledge management to architecture.

This approach includes ideas from the computer science discipline “Semantic Web”. Avery brief introduction to Semantic Web technologies and an outline of the pattern can be found in Section 2.

To address “Fuzziness of description”, we decided to prototype an extension to the Structural Design Aid database (Sedlak [15]) and have internal staff input real data with some quality assurance iterations. This approach includes ideas from the mathematics discipline of “Fuzzy Set”. Some quantitative analysis and interpretation of the data is discussed in Section 3.

1.2. More then one truth / point-of-view?

Many design topics in architecture can't be handled in a similar way like in mathematics or language science. In mathematics problem statements can be broken down according to axioms, then solved at this atomic granularity and the solution is the sum of the atomic parts. In language science the process of breaking down according to axioms and grammar rules can be used only to some degree. Some problem statements can be solved at this atomic language level (f.i. subject, predicate, object), though the solution is not necessary the sum of the atomic parts as one reassembles the parts again. Even the most structured text types like law codes and building codes are up for interpretation.

Descriptions of buildings seem to be even more complex. The root cause might be that an executed building is the product of a complex process that involve many aspects, like actors (owners, architects, engineers, construction teams, urban planer, architectural theoreticians, financial institutions, etc...), the environment (ground type, climate, etc...) and social context (public opinion, access to technology, political situation, economic situation, etc...). As a result decisions are often based on compromises driven by multiple aspects, and classification becomes subjective. The influence of discourse, architectural theory and interpretation of real world buildings often lead to diffuse, competing or even contradicting assertions.

Researcher often need to classify executed projects that they have not been involved with during its construction process. Time and traveling constrains don't allow them to visit a lot of projects in person. Access and communication with original decision makers like architects, civil engineers and owners are time consuming and often out of scope of a research project. Access to meta information like ground plans, elevations, system sketches, engineering data is also often limited due to commercial interests and the time it takes for contributors to assemble this information and transfer it to academic partners. If a building has a perfect square plan shape or if the side ratio is uneven can't be determined by a distant researcher. They still need this information and infer it based on the information available (often photographs of the projects, publications, and personal knowledge). This behavior driven by “best intention” of content contributors is hard to translate into digital models and a lot of knowledge is lost when a researcher moves on to a different project.

Traditional architecture databases hold "one-truth" or one "point-of-view" and are limited by their underlying technologies. They are often build around dichotomy (true, false) (Pedrycz and Gomide [14]) or three-value-concepts (true, false, unknown) (Korzybski [10]). Content contributors try to overcome these shortcomings, by using comma separated string values or mnemonic codes (see Loh [11], Sedlak [16]) which can be interpreted by human experts but for a computer the only access point is a full text search.

2. Handling of domain disagreement in architecture

2.1. Enabling multiple point-of-views with traditional information technology

To integrate two or more databases, technical staff of traditionally designed architecture databases need rules how to handle contradicting values, especially when the merged system can again handle only "one truth"/"point-of-view". This leads to very time consuming decision making by subject matter expert, and sometimes valuable information is being dropped. In academic scenarios contributions from students that require quality assurance by reviewers are hold back and time consuming preparations happen in spreadsheet/office documents which can't provide context.

It is possible to preserve more then one point-of-view with traditional database systems. The technical process involves switching many parts of the database schema from "one toone relationships" to "one-to-many relationships". The software redesign is a well established pattern for computer science staff.

The resulting relational database can handle the new requirements but has an increased complexity and becomes harder to maintain by none technical domain experts. Simple desktop database programs need workarounds to present the new complexity to the user (f. i. hierarchical data, foreign-keys-constrains, cascading deletes).

2.2. Semantic Web technologies

Trying to describe buildings, building shapes or structural systems of buildings in hierarchies or tables often leads to contradictions. F.i. an elevator shaft might be connected to the ground floor and the fifth floor. We have a direct connection without involving the floors in between. This interconnected thinking is in the mindset of architects and engineers and is reflected in expert systems like structural analysis software which is based on data graphs and networks to enable calculation. There are however limitations that are imposed by our daily work with office application software and database which are linear, tabular or sometimes hierarchical.

We will briefly introduce some "Semantic Web" technologies from an architectural point of view. We believe that thinking in graphs and networks can solve some traditional limitation we face in the architecture domain.

The term "Semantic Web" was coined and described by Berners-Lee et al. [3] (also Berners-Lee [4]). It describes a technology stack and usage patterns to bring the traditional

Internet to a higher level by making it better interpretable for machines. Part of this vision is the use of advanced and standardized classification and inference models based on the Web Ontology Language "OWL" (World Wide Web Consortium [19]). The biotech and life science disciplines are currently the driving forces of these classification technologies, as they are confronted with highly interconnected data like proteins, molecules and medical symptoms.

We will skip the term "ontology" (Gruber [6]) as the patterns we introduce further below reuse some smaller concepts to achieve "distributed classification" and can be integrated into a full ontology at later stage. Main elements are:

Individuals [/**items**], represent objects in the domain that we are interested in, also known as the domain of discourse (Horridge [7]). F.i. "buildingX", "personY", "rectangle". As the main audience for our projects is not likely to be familiar with knowledge base terminology the term „individual" will often be substituted it with the more accessible word „item".

Properties are binary relations on individuals [/items]. A binary relation is a relation between two things (Horridge [7]). Properties are named relations. F.i.

"isPlanedBy", "hasGroundPlan", "accordingTo", "withConfidenceOf"

Classes are interpreted as sets that contain individuals [/items]. They are described using formal (mathematical) descriptions that state precisely the requirements for membership of the class (Horridge [7]). F.i. "form active systems", "architects", "engineers", "polygonal shapes with four edge points"

Literals are strings, numeric data or other simple data types. F.i. "Japanese Pavilion on the Expo 2000", "Japanischer Pavilion auf der Expo 2000", "Pabellón japonés", "Shigeru Ban", "Frei Otto", "90%"

An individual [/item] can be member of **multiple classes**. F.i. the item "Frei Otto" can by a member of the class "architects" and the class "engineers"

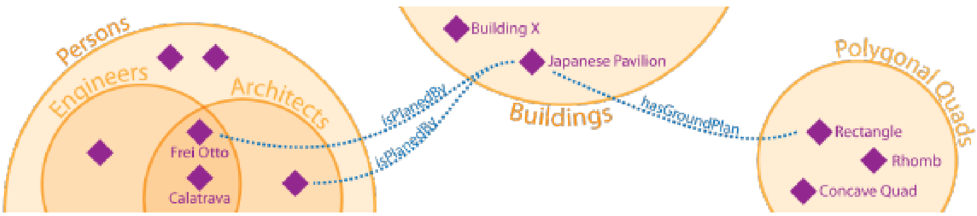

Figure 2: Basic elements of Semantic Web systems

2.3. Semantic Web macro level: Separating classification and data

Of course domain disagreement is not unique to architecture. Many other discipline and commercial practices face similar challenges. In business knowledge management of bigger organizations there are often different stakeholder (marketing, sales, procurement, upper management, information technology) that fulfill their tasks with different speed and focus but still need to exchange information. Below pattern was adjusted and abstracted to match the needs of architecture and built environment scenarios.

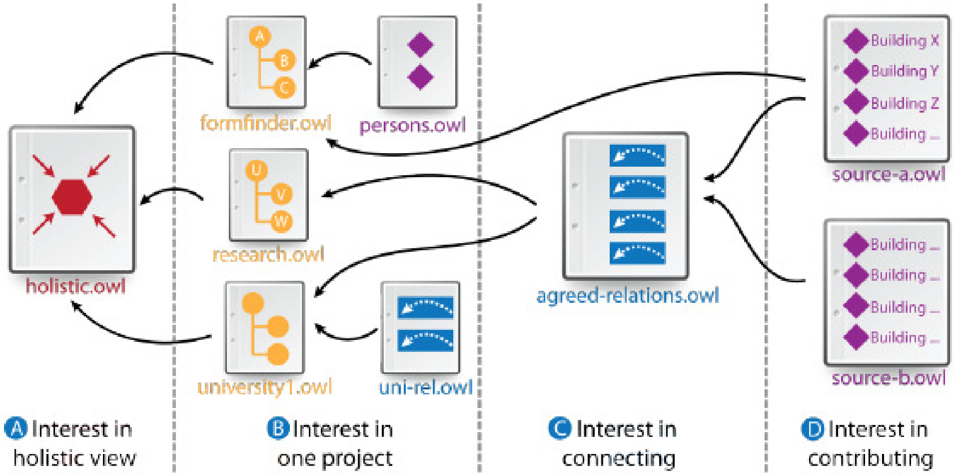

Figure 3: Managing distributed knowledge for actors with different interests

The pattern is mainly based on separation of concerns. Some actors need to focus on their direct task Figure 3 (B) & (D) (f.i. "release a new version of Formfinder"), while other have an interest in a holistic picture Figure 3 (A) & (C) (f.i. academic researchers). There is often more then one group with direct tasks, and the organization as a whole tries to encourage them to have as much overlap as possible. Still they need certain freedom to extend the system to achieve their goals quickly. The pattern relies on the import of different sets of OWL files into combined states. Current open source (f.i. Stanford Protégé 3 & Protégé 4 [17]) and commercial (f.i. TopBraid Composer) Semantic Web desktop application can handle these cascading imports.

One advantage of this pattern is that we can not only switch item sets on and off but also handle agreement and disagreement in the classes themselves and evolve them. Actors interested in a holistic (Figure 3 (A) & (C)) view can use advanced features to identify overlap. F.i. they can use standard properties like "owl:sameAs". This allows statements like "buildingX sameAs building654321" and the items will merge in a combined state.

2.4. Semantic Web micro level: Allowing multiple values with reification

We would also like to introduce a second pattern which can help solving disagreement on a very fine grained level. To better understand it we will briefly introduce "Reification".

Current Semantic Web infrastructure is based on the idea of triples in the pattern of "Subject Predicate Objects". F.i. "buildingX isPlanedBy freiOtto" or "buildingX has GroundPlan rectangle". (Antoniou and Van Harmelen [2], Allemang and Hendler [1])

There are active discussions about "Quads" and "Reification" in the computer science community (Allemang and Hendler [1]). This would allow provenance statements like "buildingX hasGroundPlan rectangle **accordingTo SourceY**" or numerical statements like "buildingX hasGroundPlan rectangle **withConfidenceOf 0.9".** We can also semantically combine these statements to form:

· buildingX hasGroundPlan rectangle **withConfidenceOf 0.6 accordingTo studentA**
· buildingX hasGroundPlan rhomb **withConfidenceOf 0.9 accordingTo researcherB**

One advantage of this approach is that we can store only the small bits where we have disagreement with little duplication of other data. Most recent open source (Protégé 4) and commercial (TopBraid Composer 3) Semantic Web desktop application can handle reification (Knublauch [9]).

Some similar result can also be achieved by using name spaces and property hierarchies, with the disadvantage that mixing more then one aspect would lead to very deep and repetitive property hierarchies. F.i. Mixing provenance and confidence concepts. Also similar results can be achieved with relational data bases, most requirements must be known upfront and database normalization becomes an issue.

3. Fuzziness of descriptions

3.1. Prototyping

During review and classification of historic lightweight structures data, we ran in the problem, that certain parameters of the simplified Formfinder attributes (Wehdorn-Roithmayr and Jurewicz [18]) were to broad, or projects features were to ambiguous with the information available to us. We decided to build a prototype extension to the Structural Design Aid (SDA) database (Sedlak [15]), to allow multiple classification with percent values. (See Figure 4).

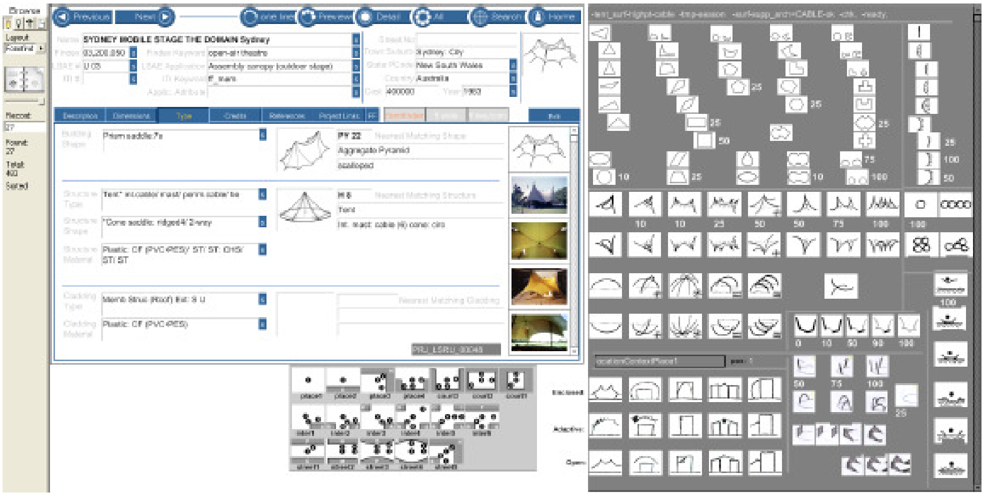

Figure 4: Screenshot of the SDA database with the prototyped extension (right)

The objective for data input in the prototype was: "When a person would browse certain attributes, should this project appear as a design inspiration?" Initially only one 100 percent value was allowed per group of attributes. The data input process was manual and 83 projects have been classified sufficiently in this manner.

3.2. Concepts from Fuzzy Sets

To prepare for later automation, simplified communication and semantic mapping the allowed values have been limited to a discrete set of "10, 25, 50, 75, 90, 100". These values can also be mapped with function approximation to some established membership function from the discpline of "Fuzzy Sets" (Zadeh [20]) (Pedrycz and Gomide [14]). The semantic intention of the 90 and 10 values was to allow statement like "nearly 100%" and "nearly 0%". We needed to push the linear membership curve before we did fuzzy arithmetic, to emphasize the 90% values. From the three typical "S" shaped functions (Sine, Gauss, SFunction) we currently use an adapted sine function.

3.3. Quantitative analysis and interpretation

To better understand how different attribute groups behave with applied percentage values, we have made a simple quantitative analysis (see Figure 5). As expected due to the manual data input process the amount of positive results declines steadily the higher the threshold. For the attribute groups (A), (B) and (E) we had enough data to apply alpha-cuts. Here empty and seldom used data field are skipped and the graphs became easier to interpret.

The graphs show some regions that are more horizontal then other. This can often be observed at the 10% and especially 90% values (Figure 5 (B),(D),(E) &(G)). This means that the users did not make much use of the 90% value. In this cases it seems that for future content contributors we could simplify the data input and reduce the discrete set to “25, 50, 75, 100”. For very small attribute groups like edge fixity (C) and sag (D) we see steps in the graph. Again we could reduce the discrete set to “10, 50, 90, 100” or even “50, 100”.

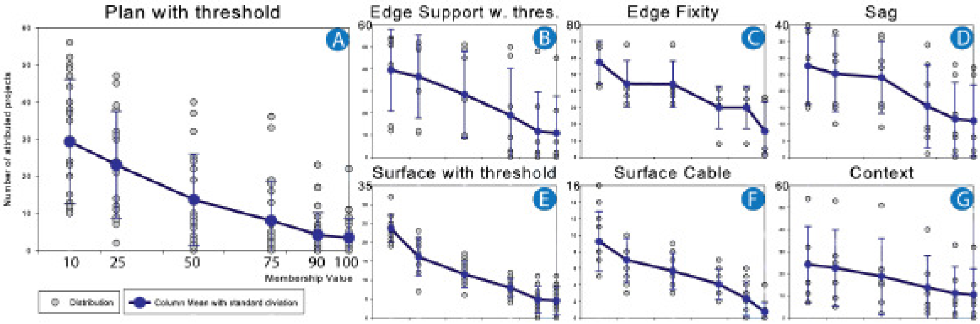

Figure 5: Usage of percentage values in various attribute groups

After additional interviews and review, we believe that our original rule to allow only one 100% value for contributors was to limiting. The attributes should have more internal hierarchies and groupings. These hierarchy levels and smaller groups must then be defined as “exclusive” or “inclusive”. Within a block of exclusive selection only one attribute can be determined as 100% accurate. Within a block of inclusive selection multiple properties can be 100% applicable. Attribute groups with exclusive selection are ground plan, edge fixity and simple arrangement. Attribute groups with inclusive selection are edge style, edge support, surface support and location contact.

The major exclusive groups tend to describe the project as a whole, inclusive groups parts or features. Within these inclusive groups it makes sense to create smaller exclusive subgroups. For instance a membrane structure might have two high points and one arch as its surface support. The cardinality of high points and arches are discrete and therefore “exclusive within its subgroups” while the architectural significance of the high points and arches is very subjective and fuzzy, depending on proportion, arrangement, symmetry, etc.

4. Proposed extensions to the user interface

The discussed key factors “disagreement” and “fuzziness” are complex in their very nature. Exposing this complexity to the end user would negatively impact the user experience for design focused application like Formfinder. We propose to extend the Formfinder online panels with simple “Top X” lists and two simple horizontal sliders. (Figure 1, lower part of screenshot (B) and (C)) The ranking of the results is inspired by Fuzzy Sets concepts. Because “Fuzziness” can increase the result set significantly, it should be handled on the server / database side, before filtering. We suggest to display one initial slider, and hide the five detailed sliders in expert settings (Figure 6(A)). We believe guiding messages (yellow boxes) are important to let the user understand which impact the slider will have.

The comparison of provenance and different point-of-views of sources should be assisted with direct visual feedback. So this should happen in the client software after database query and filtering (Figure 6(B)) . Changes to the slider trigger a resorting of the local result set and we suggest strong visual indication which items have been affected. F.i. green overlay symbols could indicate if an item was positively influenced (Figure 6(C)). Again the calculation is inspired by Fuzzy Set concepts.

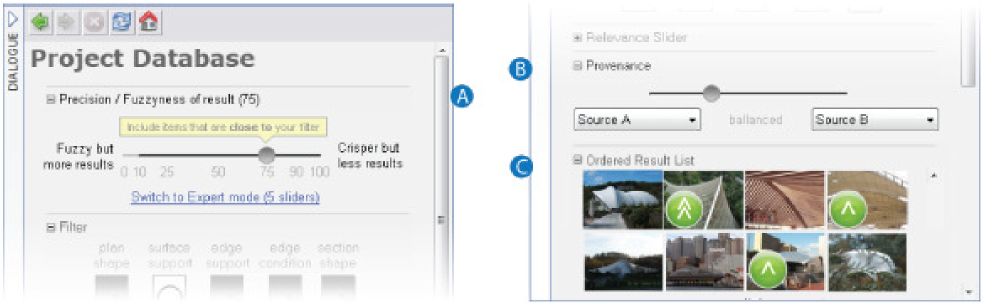

Figure 6: Proposed extension for the Formfinder web application

5. Conclusions & Prospects

The analysis phase showed that collecting and presenting lightweight structures in bigger quantities / databases is limited by traditional techniques. Buildings as the items of interest have many facets and sometimes multiple valid interpretations. These different point ofviews must be able to coexist digitally to achieve progress with integration.

From a structural engineering standpoint we could have solved some of the problems by breaking the projects up into smaller parts (f.i. nets). It is a valid approach for detailed case studies, but especially for architects during the creative design process where the search criteria are softer and the goal is to get overview of multiple buildings this is out of scope. Though automated classification where Formfinder .mem files are available for executed project show a good way forward to automate some attribution. Breaking up patterns have been identified as a key factor in our analysis phase and is under investigation. Two other disciplines appealed to us: Fuzzy Set theory and Semantic Web.

Fuzzy Sets theory can be seen as a foundation part of high level trends which are sometimes known as “human centric computing”, “soft computing” or “granular computing” (Pedrycz and Gomide [14]). Essentially they accept that people don't always have precious questions and don't expect exactly one precious answer but want to get an overview from a set of vague data. This seems to be particular true for architecture and the construction of new buildings as every construction site and situation is unique and existing solutions need to be customized to match the project requirements all the time.

Semantic Web technologies, seem to go to the opposite direction, but only at first sight. Experts should determine with mathematical precision and logic what the necessary requirements for members in a class are. But beside of this precision parts the technology stack is designed to be very integration friendly, and handle knowledge gaps. The "open world assumption" in OWL says, that just because something is not stated to be true, we can't assume it is false, the information might simply not reachable at the moment (Allemang and Hendler [1]). While designed to support reasoning and inference this axiom and the flexibility of triples with its simple "subject predicate object" pattern follow a "complexity accepted" mantra rather then over simplifying. Again this is something which seems to be very valid to architecture, as buildings are complex blurry objects with certain islands of crisp facts.

And indeed both disciplines use the term "semantic" a lot. Languages have this duality of precision and imprecision. A common formal (partly visual) classification language would allow better communication and enable in-context inspiration and access to niche expertise knowledge transfer.

As outlined in Section 1.1 the next research step will be the "use of controlled vocabularies". We plan to build up new visual attribute groups (F.i. Sections), mature the existing one (Jurewicz and Sedlak [8]), connect to emerging like MACE [12] and integrate historic (Otto [13]). We also want to look into interconnections and patterns that might connect these vocabularies.

Acknowledgement

The authors like to acknowledge the support from Prof. Achammer and Prof. Sedlak (both Vienna University of Technology) and the Formfinder GmbH, together they enabled this research. We also like to acknowledge the support from the SVA-BizSphere AG for knowledge transfer and research support especially for the findings in Section 2.3

References

[1] Allemang D. and Hendler J.A., Semantic Web for the Working Ontologist, Morgan Kaufmann, 2008, ISBN 0123735564

[2] Antoniou G. and Van Harmelen F., A semantic Web primer MIT Press, 2004, ISBN 0262012103

[3] Berners-Lee T., Hendler J., Lassila O., The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities in: Scientific American, 284 (5), May 2001, 34–43 (Online) http://www.scientificamerican.com/article.cfm?id=the-semantic-web

[4] Berners-Lee T., Sir Tim Berners-Lee: He Created the Web. Now He's Working on Internet 2 (Tim Berners-Lee interviewed by Mark Frauenfelder), in MIT Technology Review, October 2004, 40-45

[5] Engel H., Tragsysteme / Structure System (2nd ed), Verlag Gerd Hatje, 1997

[6] Gruber. T. R., A translation approach to portable ontologies in Knowledge Acquisition, 5(2):199-220. (Online)
http://tomgruber.org/writing/ontolingua-kaj-1993.htm
http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

[7] Horridge M., Protégé OWL Tutorial, by Collaborative Open Ontology Development Environment (CO-ODE), August, 2004 (Online)
http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/

[8] Jurewicz P. and Sedlak V., Building Shape Online in Lightweight Structures in Civil Engineering, IASS Warsaw , 2005, 314-318

[9] Knublauch H., Editing Reified Statements, Oct 2006 (Online) http://composing thesemantic- web.blogspot.com/2006/08/editing-reified-statements.html

[10] Korzybski A., Science and Sanity,1933 republished by Institute of General Semantics, 1994, ISBN 0937298018

[11] Loh S. K. Vocabulary of Shape and Structure Type (research report) LSRU, Faculty of Architecture, University of New South Wales, 1990

[12] Metadata for Architectural Contents in Europa (MACE). (Online) http://www.maceproject.eu/index.php

[13] Otto F., IL 22 Form, Stuttgart, Karl Krämer Verlag, 1988

[14] Pedrycz W. and Gomide F.,Fuzzy Systems Engineering, Toward Human-Centric Computing, Wiley IEEE Press, 2007, ISBN 9780471788577

[15] Sedlak V., Structures Design Aid Database, by Lightweight Structures Research Unit 1996-2001, Faculty of the Built Environment 2001-2005, University of New South Wales, Sydney, 1996-2005

[16] Sedlak V., A Computer-Aided Conceptual Structural Design Aid in *Proc. 1997 IASS Symposium, Singapore*, CI-Premier Pte Ltd., 1997, 745-754

[17] Stanford Protégé ontology editor and knowledge-base framework, by Stanford Medical Informatics, Stanford University, May, 2009 (Online)
http://protege.stanford.edu/

[18] Wehdorn-Roithmayr R. and Jurewicz P, Formfinder Professional Advanced Typology Classiciation, in *Textile Composites and Inflatable Structures III*, CIMNE, 2007, 93- 96

[19] World Wide Web Consortium, *Web Ontology Language (OWL)*, 2004 (Online) http://www.w3.org/2004/OWL/

[20] Zadeh, L. A. *Fuzzy sets, Inf. Control 8*, 1965, 338-353