COMPUTATION:
THE NEW REALM OF ARCHITECTURAL DESIGN

eCAADe 27
istanbul 2009

edited by Gülen Çağdaş and Birgül Çolakoğlu

A CD-ROM containing the digital version of the proceedings is enclosed with this book

Prof.Dr. Gülen Çağdaş
(cagdas@itu.edu.tr)
İstanbul Technical University Faculty of Architecture
Inst. of Science & Tech., Architectural Design Computing Graduate Programme
Istanbul, Turkey

Assoc.Prof.Dr. Birgül Çolakoğlu
(colak@yildiz.edu.tr)
Yıldız Technical University Faculty of Architecture
Inst. of Science & Tech., Computational Design Graduate Programme
Istanbul, Turkey

1st Edition, September 2009

www.ecaade.org

Printed by
Cenkler Printing, İbrahim Karaoğlanoğlu Cad. Civan Sok. No:7
4.Levent, Beşiktaş, İstanbul

ISBN 978-0-9541183-8-9

eCAADe 2009

COMPUTATION
the new realm of architectural design

Proceedings of the 27th Conference on
Education and Research in Computer Aided Architectural Design in Europe

September 16-19, 2009
Istanbul Technical University and Yıldız Technical University
Istanbul, Turkey

www.ecaade09.itu.edu.tr

Edited by
Gülen Çağdaş and Birgül Çolakoğlu

Contents

Formalizing and Computing Ontologies to Speed Up the Construction of Knowledge-based Collaborative Systems

Three Different Approaches

Antonio Fioravanti[1], Gianluigi Loffreda[2]
[1,2]Dept. Architecture and Urban Planning – Sapienza University of Rome
http://www.dau.uniroma1.it
[1]antonio.fioravanti@uniroma1.it, [2]gianluigi.loffreda@uniroma1.it

Abstract: *Architectural design is a purpose-oriented collective process defined in time, split up into phases, carried out directly or indirectly by numerous professional profiles and characterized by the co-presence of numerous disciplines and specialist skills. The efficiency of a shared design among multiple designers depends on how much the actor's semantics of the used terms are interpreted correctly by all the other actors involved in the design process without misunderstandings. The chosen way to find an answer to these questions is the development of a Collaborative Architectural Design system based on Knowledge formalized by Ontologies. An ontology has been implemented using three different approaches to define an entity based on Lisp, Protégé and Altova.*

Keywords: *Collaborative architectural design: knowledge-based systems; ontologies; knowledge structure.*

Computational design

Computational design endeavours to quantify the desired properties, characteristics and behaviours (as humans always want to 'quantify' countable AND uncountable entities) of a building organism, which may be considered as a system – structured set of Spaces and Building Components designed to satisfy certain goals. This is a systemic conception of buildings: the only one that allows to formalize and compute function and behaviour. These goals may vary widely: from an epic achievement for one's era to which to dedicate one's entire life – the pyramids; or more prosaically, the construction of a sludge containment tank.

The importance of calculation in a society has always been fundamental (for instance, the incunabula used by the Sumerians for accounting purposes marked the birth of the alphabet) (Diamond, 1998). It has increased in complexity in an exponential fashion, from Pascal's machines to those of Babbage (Losano, 1973) to transistors and to the present-day networked computers. Likewise the subject of calculation was first the data, then structured data and lastly highly abstract concepts.

This exponential quantitative increase entrains a quantum leap: the increase in the memory element from 8-16-32-64-128 bits paralleled the transition from word processing to A.I. processing; the memory quantum leap from KBs to PBs has allowed entire knowledge libraries to be stored (significantly, human memory may be quantified as around 10 TB). Conceptually this is the same difference as between a simple addition operation and Plato's Cratylus.

In line with this trend, we take for granted that anything that can be expressed lexically can be computed and be used for designing. However, we are aware of the limitations and ambiguities of all formal systems (Hofstadter, 1984), as well as of the fact that our hardware (our senses and our mind, at least in the basic sense, the capacity to interpret sensory data) helps us filter out the various ambiguities. In this sense, we are in agreement with the studies by Tagliasco (Manzotti e Tagliasco, 2001) that it is only by equipping the machines with sensitivity capability that they can perform intelligent tasks, i.e. to pass Turing's machine test. The sensory data (hot-cold, sight, smell, etc.) contextualize the formal expressions that describe reality. The classical example is Simon's ambiguity of the proposition "I saw the men on the hill with the telescope" (Simon, 1996, p. 79) that can be resolved by sensitivity tools interfaced with traditional A.I. tools.

Taking into consideration these limitations typical of existing design support tools, the field investigated consists of that of knowledge-based architectural design support systems, with specific reference to those referring to collaborative design.

This raises additional problems insofar as the 'entities' made up of quantities (= physical characteristics) and concepts (= non physical characteristics) on which computations are performed are 'manipulated by several hands': the actors (Wix, 1997) in the design process.

For this reason it is necessary to observe more attentively the design process as it exists today in architecture.

Architectural design process

Architectural design is a purpose-oriented collective process defined in time, split up into phases, carried out directly or indirectly by numerous professional profiles (all denoted as 'actors'), and characterized by the co-presence of numerous disciplines and specialist skills which is part of a broader process aimed at the construction, maintenance, and ultimately the recovery or demolition of buildings (Carrara et al., 2009).

Once the actors have acquired the necessary information (books, reviews, on-line libraries, codes, etc.), reprocessed their own experiences and laid down their own design solutions, they ultimately have to interact with the other actors in such a way as to reciprocally combine into the overall design solution to which they all contribute by means of their own partial solutions they gradually develop. To make it possible, actors have to correctly understand of the information associated with the entities they often concurrently manipulate that make up the overall design solutions.

The deep differences in the actors' cultural and technical background, which are the outcome of a wide range of different professional and training experiences, are reflected in the extremely varied ways in which any actor knows and considers the entities (objects and processes, properties and relations) involved in the building process, that makes it extremely difficult for the actors to understand each other.

Till now exchanging contents, even among commercial applications has been very difficult to be done. As a matter of fact the export of proprietary BIMs, from their own file formats to the correspondent IFC one, are not equivalent due to their own different primary conceptual models of the building. Moreover, even though different specialist actors use the same integrated application tool (e.g. Revit, Triforma, etc.), the entities they consider can have different meanings and behaviours as belonging to different specialist domains.

As an instance, a window assumes different meanings and representations when related to different specialist domains (such as an architect's, structural engineer's, building scientist's and so forth) as the former ones are close linked to underlain models of the considered aspects of reality. From the point of view of an architect, a window is an element of his/her knowledge domain that allows to relate each other interior with exterior of a building, how such a relationship is balanced, how much exterior is enjoyable from interior and the latter can be seen from the former, the rhythm that scans a façade, an element more or less important in comparison with a wall, etc. And its consequent representation-display, describing its form in a detailed or symbolic fashion with different scales.

For a structural engineer a window has many different functions: it is a aperture in a non load bearing wall and may be considered a lesser load, or else it is an absence of material in a shear-wall and a possible weak point of it. In the last case the representation is limited to the dimensions of the aperture and the weight of the window.

For a plant engineer, a window behaves as a 'flow tube' with reference to its energy contribution both to the whole building and to the rooms with which it is in direct contact in terms of heat, light and sound (which varies during the year, the day and each minute). Its representation involves its dimensions, its thermo-physical properties, and its colour, orientation, etc.

The physical element considered, the window, remains the same in actual reality although it is evident how its significance and function differ in the entities representing it in the three specialist domains. All these concepts particular for an actor are hard to understand for other actors.

Collaborative architectural design

With new technologies and the computers' network it's possible to think about a global work environment in which a building project can be (remotely) developed in a collaborative way so as actors of different disciplines with their own specific tools are involved in several collaborative design processes (Chen, 2004; Carrara and Fioravanti, 2007; Cheng, 2008).

The Collaborative Architectural Design can be improved by means of Knowledge Based Systems (KBSs) that allows designers to have an efficient support as they can 'capitalize' and 'managed' knowledge, expertise and experiences.

Nevertheless this promising approach has been proven successful in research programs, KBSs are remained at prototype level as the low growing process to implement them and as the ever increasing difficulties to check consistency and coherency (programmes must check every node in every chaining inheritance process). Both difficulties depend on the construction of KBSs: craft made, practically.

The low growing implementation is intensified by the rapid increase of professional skills in design process increasingly shows new specialist disciplinary fields. Moreover another element that thwarts the KBSs implementation is that in every field of knowledge, experts communicate by concepts at high level of abstraction with specific drawings, symbols and documents, according to their disciplinary field. That means usual instruments are used 'surreptitiously': not for directly carrying concepts, but for communicating symbols that other actors will 'decipher' ... In fact actors use their own terms and definitions to represent a building or component design by means of their own accustomed disciplinary jargon with which they can understand each other only within the same field (and sometimes not thoroughly).

The efficiency of a shared design among multiple designers depends on how much the actor's semantics (and successively, intentions and goals) of the used terms are interpreted correctly by all the other actors involved in the design process without any misunderstanding.

Our research aims at realizing a system that allows a mutual comprehension by means of a shared ontology among actors that realizes a true Collaborative Design as 'the ability to discuss a given topic at the same level of abstraction'. The main purpose is to create a collaborative working environment using

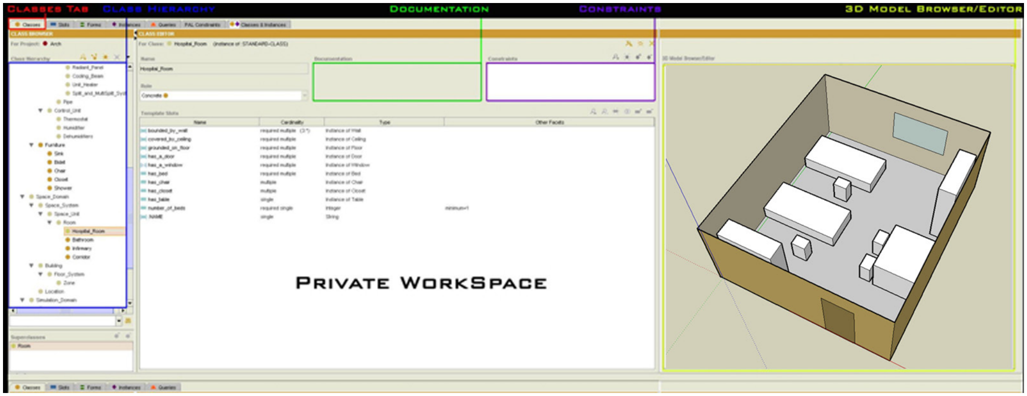

Figure 1
A screenshot of Architect's Personal Design Workspace – Arch. step1.0

usual known tools to the actors involved, the terminology they usually use and new instruments to map concepts during the entire process.

The interesting and innovative approach to deal with an effective implementation of entities for Knowledge Based Systems is the development of a multi dimensional model for ontology representation of different disciplines. Such a representation showing ontologies, constraints and design processes in a visual way helps actors to better understand the meanings of each entity according to the personal representation s/he chose. For any entity with multiple interacting actors, the representation allows an agreement between the ontologies of different disciplinary domains involved in the specific design process.

Allowing Multiple Shared Ontologies, each actor is free to map her/his own entities and meanings (his Private Knowledge) differently with the others ones according to the her/his own target within the overall project. An actor can also check all the constraints, both private and shared ones, using the implemented multi dimensional model editor and by the usual ICT tools. Hence the visual editor points out all the violated constraints implemented in ontologies and in semantics. In this way each actor is able to activate only the constraints s/he wants to check.

Sequence of a collaborative design session

At the beginning of the project the Architect develops a layout of a patient room in her/is Personal Design Workspace – PeDW using her/is Specialist Knowledge Structure – SpKS with dimensions, furniture, door, window (Fig. 1, Arch.step.1.0).

The dashboard of an actor allows to design in her/is own PeDW or directly in the Overall Design Workspace, ODW with her/is constraints, using her/is SpKS filtered entities. i.e. every entity of her/is SpKS is exported in this environment that s/he can see inside her/is dashboard by Common View of Project, and that can be conceived like a 'Test' DW as the ODW is influenced by constraints of other actors SpKS.

Then s/he filters this instance to the Common View, for ex. hiding the furniture. Notice that at this step s/he has not published yet to the ODW, so nobody can see anything nor can be influenced by her/is constraints.

After the Architect's design solution Arch. step 1.1 has been published, the Mechanical Engineer links to the server and 'privatizes' this design solution (Fig. 2, MecEng. step.1.1) in her/is PeDW, then s/he includes this design solution in her/is PeDW by

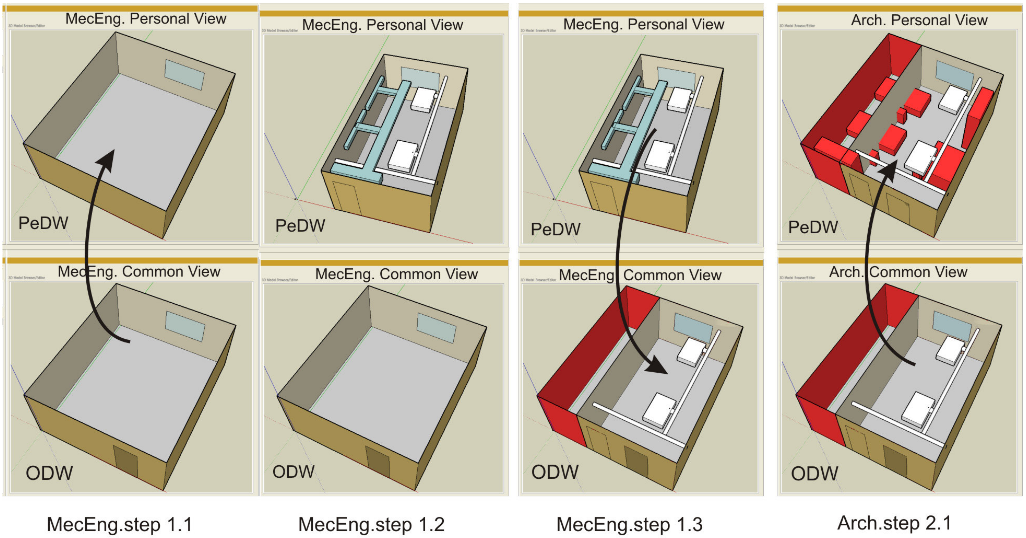

Figure 2
Sequence of a collaborative architectural design session: MECEng. and ARCH steps

enriching data through the Filter (Fioravanti, 2008), and develops the project with her/is instruments adding her/is specialist entities (A.C. equipments, A.C. ducts); and/or modifying all entities s/he can see, for instance s/he moves the wall and makes the patient room shorter (Fig. 2, MecEng.step.1.2).

When the Mechanical Engineer considers her/is design solution ready to be published on the ODW, s/he can filter the instances/properties/rules s/he wants to hide to other actors (for instance air ducts, but not A.C. equipments) and then publishes her/is design solution in the ODW. The system finds out conflicts among instances changed by different authors pointing out each change (in red) (Fig. 2, MecEng.step.1.3).

Then the Architect imports this new design proposal from the ODW into her/is PeDW, checks it by means of her/is specialist constraints/rules/requirements, and s/he is warned the interference between the wall moved by the Mechanical Engineer and the furniture (Fig. 2, Arch.step.2.1).

Three representations for an ontology based design

The chosen way to find an answer to these questions is the development of a Collaborative Architectural Design system based on Knowledge formalized by several Ontologies that can significantly improve collaboration between different specialists (Ugwu, 2005; Fioravanti, 2008).

One of the greatest difficulties in this field is how to rapidly formalize the prototype entities making up the ontology of a specialist actor.

So far we have taken into consideration three types of formalization to model Systemic Knowledge of Building and its entities (components, building parts, characteristics, constraints, relationships) using different tools like Lisp, Protégé and Altova.

The first implementation was performed using pure Lisp. In this way it was possible to manipulate the instantiation and the inference engine 'on the fly' and to modify the characteristics of the entities

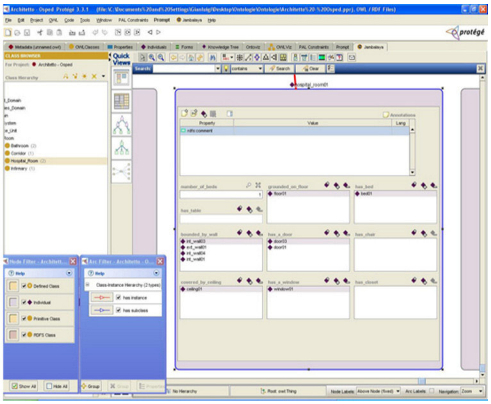

Figure 3
Main classes and an instance visual representation

relatively freely and precisely, indeed ad hoc, but at the cost of a artisanal implementation.

The main characteristic of the entities is related to the 'type' of entity: the membership 'class'. This one is formalized by means of a custom made frame structure, similar to the one investigated by McCarthy, by means of an ISA slot (Is-A).

The advantage of being able to manipulate also this level of an entity's structure (which we may term a knowledge atom) is not only being able to change the inheritance of an entity but also to mix entity assemblies. The freedom we are given by this formal logic enables us to compose an entity of a class (whole-of, or assembly of) also from entities of different classes belonging from heterogeneous domains, for ex. Space domain and Building Component domain.

In our case we implemented the System of Spaces which, together with the System of Building Components, contribute to fully defining a building so that the two systems (normally separate) can be interfaced directly through an inversion of the inheritance relationship (a slot in the frame structure) with the assembly one.

At the time of instantiation this peculiarity makes it possible to simultaneously verify the constraints that are normally found on 'orthogonal' logical planes: classes and assemblies. As a matter of facts this approach, from a logical point of view, is not rigorous, but in architectural design practice has been used from centuries.

A second approach to the formalization of the entities linked to building design was using an open-source tool: Protégé. The model of the building entities was formalized according to a the three-fold structure based on 'Meaning-Properties-Rules'.

The above-mentioned ontology editor not only allows class names to be attributed, extended meanings associated with them (descriptions) and properties (functions and fields to which values of a specific predetermined 'type' may be attributed), but also makes it possible to define, within the entity considered, a set of rules associated with them that determine relations between them and other entities, instances or attributes, in terms of constraints/specifications and/or goals of the design solution (Fig. 3).

In this case, the distinction between entity 'meaning' (name and description), 'properties' (slots and associated attributes) and 'relations' is sharp and well defined. The rules, in particular, are formalized by means of a software development kit (SDK) the PAL – Protégé Axiom Language; they operate on the instances of the ontology and establish relations, constraints and specifications associated with the entities to which they are applied.

The constraints checking and verification, as it is separated from the definition of the entity, is not contemporaneous with the instantiation of the object and so the processes of verification and control of consistency, coherence and congruence are necessarily subsequent to the completion of the instantiation of the entities involved in the design solution.

Altova SemanticWorks® is another tool used by the authors to model entities involved in building design. In this case, although using the same formal language – the Ontology Web language (OWL) - used in Protégé, as no rule editing tool is provided (if-then protocols, verification cycles and/or normal computation operations), the set of verification and control processes must necessarily be developed outside the actual modelling process using different tools and combining them later. The user is therefore asked to verify the coherence of the rules implemented externally using the ontology implemented on SemanticWorks®. The approach followed by the authors was to implement rules using the abovementioned PAL, exporting them in OWL and subsequently importing them into a SemanticWorks® formalized ontology. An inheritance type association is created between the rules and a 'remote' class denoted as PAL-CONSTRAINTS with which the concepts PAL-NAME, RANGE and STATEMENT specific to rule definition in Protégé are associated as properties of the PAL-CONSTRAINTS class.

Unlike the first approach proposed, the formalization by means of Altova and Protégé does not involve any internal 'reflexive and dynamic modification' capacity: it does not allow changes of its own structure except by means of a new edition/version of the entire or partial ontology.

Conclusions

In architectural/building design and construction the co-presence of numerous disciplines, specialist skills, actors and processes makes it very difficult to produce design syntheses of the problems pertaining to building, including architectural form, construction technology, load-bearing structures, engineering aspects, energy, and costs. These difficulties, which may be all the greater the more 'creative' the design is, are often aggravated by misunderstandings, lack of data, privacy, ownership, and the different aims pursued by the various actors, owing to the closer links that exist between actors, activities, resources and culture.

The ontology based approach can suitably support a true collaboration among actors that often have competences overlapping each other and that are jointly responsible, so as they have a mutual interest in a successful outcome of their work.

The three representations aforesaid have positive aspects and drawbacks.

The first one, pure frame approach (in Lisp), has the ability to change on the fly the inheritance structure of an ontology without any new implementation of entities and as a consequence can support 'aspects' of the same instance, can mix the path of instantiation pursued by an inference engine, but till now can be implemented only with craftsmanship, oppositely any Sensitive Language Editor (the first one came from Digital Equipment Corporation!) imposes a fixed and uniform implementation.

The second, Protégé, has an excellent user interfaces and a powerful rule editor SDK, it is easier to interface with graphic programmes than the two others, but it is impossible to change rules during the elaboration process.

The third one, Altova, is a robust ontology editor with a well defined precise structure, intuitive user interface, but it lacks SDK tools, and it does not allows a dynamic object oriented programming.

All these representations speed up the implementation of well formed knowledge, that in turn makes knowledge-based system content rich.

Therefore from these considerations and after testing a few simple design problems it was evident that to develop an effective ontology for specialist actors of architectural design process it should be used a mix of these representation tools. Altova and Protégé may fulfil the exigency of Lower-Ontology Level and an open frame structure fits Upper-Ontology Level (Fioravanti, 2008).

The so defined ontology contributes to speed up the implementation a Knowledge-based Collaborative Design system for architectural design that, on the one hand, can avoid common mistakes, support consistency, coherence and the requirements of the project; on the other hand, by sharing knowledge, can improve the mutual awareness of design choices and spread innovations.

Acknowledgements

The research was partially funded by MiUR (Ministry of the University and Research), Project of Research of National Interest 2005: "*A Model of cross-cultural collaboration for integrated design in architecture*".

References

Cheng, M.-Y.: 2008, "Cross-organization process integration in design-build team, Automation in Construction", 17(2), pp. 151-162.

Chen, P.-H., Cui L., Wan C., Yang Q., Ting S.K., Tiong R.L.K.: 2004, "Implementation of IFC-based web server for collaborative building design between architects and structural engineers", Automation in Construction, 14(1), pp. 115-128.

Diamond J., 1998, Armi, Acciaio e Malattie. Breve storia del mondo negli ultimi tredicimila anni, Giulio Einaudi Ed., Torino. Italian edition of Guns, Germs and Steel. The Fates of Human Societies, first published by W.W. Norton and Company, New York, 1997.

Carrara, G., Fioravanti, A.: 2007, c-House – a Game to Improve Collaboration in Architectural Design. How to distill a CD Based Model into an E-learning Tool, in: Joachim B. Kieferle and Karen Ehlers (eds), Predicting the Future. 24th eCAADe Conference. Frankfurt. 26-29 Sep 2007. Kanne Graphischer Betrieb GmbH, Ginsheim-Gustavsburg, pp. 141-149.

Carrara, G., Fioravanti, A., Nanni, U.: 2009, An innovative knowledge structure for supporting collaboration in building design, in: Dato, E., Beheshti, R. and Khaldoun, Z (eds), Innovation for Building and Construction. Towards a better, stronger and sustainable future using advanced ICT, EuropIA 12 Conference, Paris, 18-20 March 2009, Europia Production, pp. 331-350.

Fioravanti, A.: 2008, An e-Learning Environment to Enhance Quality in Collaborative Design. How to Build Intelligent Assistants and 'Filters' Between Them, [online]. Available from: < http://www.marhi.ru/AMIT/2008/4kvart08/Fioravanti/Article.php >, Architecture and Modern Information Technology 4(5) [Accessed on 26th May 2009]. Extended edition of paper published in M. Muylle (ed.), Architecture 'in computro' - integrating methods and techniques, 26th eCAADe Confernce, Antwerp, 17-19 September 2008, Antilope Printing, p. 829-836.

Hofstadter D.R.,: 1988, Gödel, Escher, Bach: un'Eterna Girlanda Brillante, Adelphi Edizioni, Milano. Italian edition of Gödel, Escher, Bach: an Eternal Golden Braid, 1979, Basic Books, New York.

Losano, M.G.: 1973, Babbage: la macchina analitica – Un secolo di calcolo automatic, Etas Compass Libri, Milano.

Simon, H.A.: 1996, The Sciences of the Artificial, 3rd ed., MIT Press, Cambridge, MA, USA.

Manzotti R. and Tagliasco, V.: 2001, *Coscienza e realtà: una teoria della coscienza per costruttori di menti e di cervelli*, Il Mulino, Bologna.

Turner J.A.: 1997, Some Thoughts on the Existence of a Generic Building Object, *in* R. Junge (ed.) *CAAD futures 1997* Proceedings, Munich, 4-6 August 1997, Kluwer Academic Publishers, pp. 533-552.

Ugwu, O.O., Anuba C.J. and Thorpe A.: 2005, Ontological foundation for agent support in constructability assessment of steel structure – a case study, *Automation in Construction*, 14:1, pp. 99-114.

Wix, J.: 1997, ISO 10303 Part 106, BCCM (Building Construction Core Model) /T200 draft.

Index of Authors

ISBN 978-0-9541183-8-9

COMPUTATION: THE NEW REALM OF ARCHITECTURAL DESIGN

In the field of architecture, computational design has emerged as sub-discipline having a multidisciplinary nature and using computing methods and capabilities to understand and solve architectural design problems. Computational design is based on computational thinking that includes a range of mental tools in solving problems, designing systems, and understanding human behavior. It has drawn on the concepts of mathematics and computer science.

Computational design elements are derived from both theoretical science and experimental design in such a way that its mechanism relies heavily on mathematical logic, but once built, experimentation is done by varying one parameter at a time to study individual changes. It is a design model, not design itself. Computational design involves applying appropriate computational mechanisms, algorithms, or methods to architecture inorder to solve design problems and develop design applications. This process creates systems that can be used as design tools for exploring and forming entirely new design concepts and strategies.