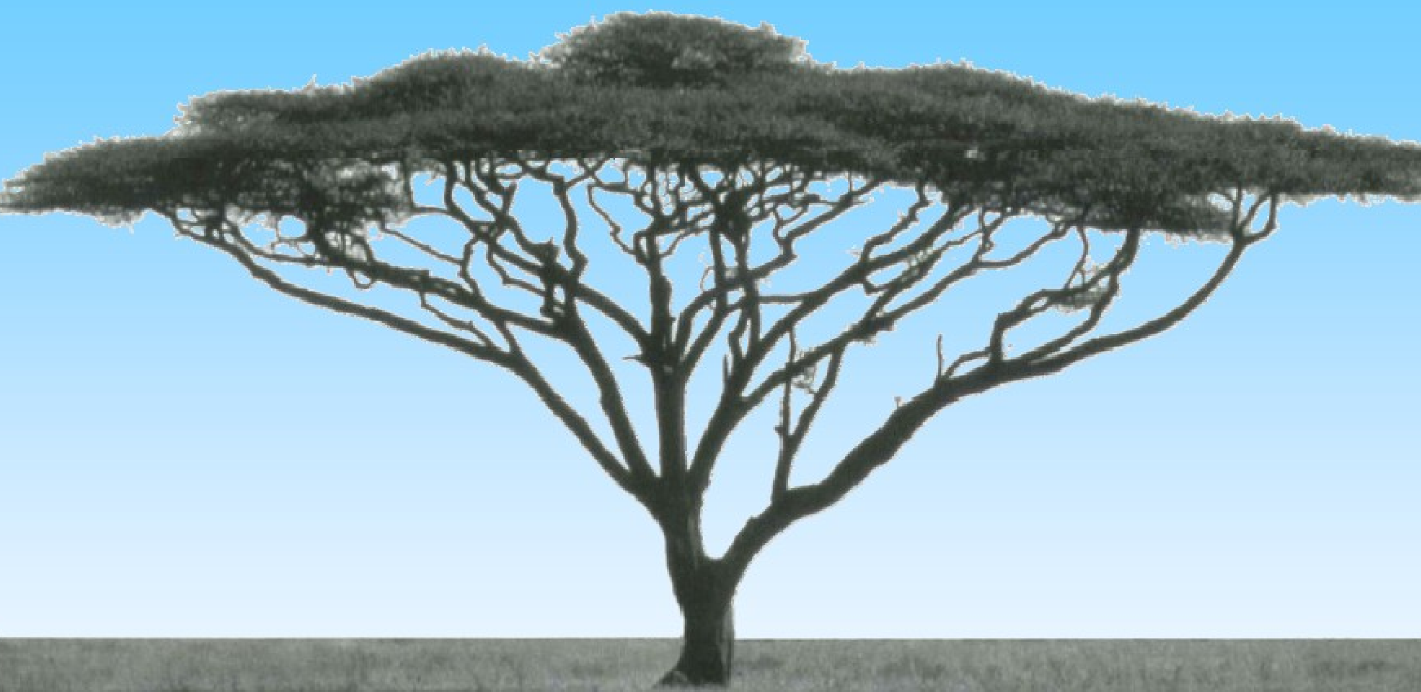

International Journal of Design Sciences & Technology

Volume 16 Number 2

ISSN 1630 - 7267

Editor-in-Chief:
Reza Beheshti
Khaldoun Zreik

Carrara, G., Fioravanti, A. and Nanni, U. (2009). Knowledge-based Collaborative Architectural Design, International Journal of Design Sciences and Technology, 16:2, pp. 1-16

europia PRODUCTIONS

ISSN 1630 - 7267

15, avenue de Ségur,
75007 Paris, France.
Tel (Fr) 01 45 51 26 07 - (Int.) +33 1 45 51 26 07
Fax (Fr) 01 45 51 26 32- (Int.) +33 1 45 51 26 32
E-mail: dst@europia.fr
http://www.europia.fr/DST

International Journal of
Design Sciences and Technology

Editor-in-Chief: **Reza Beheshti** and **Khaldoun Zreik**

Volume 16 Number 1

Issue Editor: **Edwin Dado**

Table of Contents

1630-7267

Knowledge-based Collaborative Architectural Design

Gianfranco Carrara*, **Antonio Fioravanti**** and **Umberto Nanni*****
* Sapienza University of Rome, Italy. E-mail: gianfranco.carrara@uniroma1.it
** Sapienza University of Rome, Italy. E-mail: antonio.fioravanti@uniroma1.it
** * Sapienza University of Rome, Italy. E-mail: nanni@dis.uniroma1.it

The constant increase of the complexity of the building process is generally accompanied by a parallel general reduction of product quality, commonly ascribed to the inadequacy of the routine design methods and tools. In facts these ones make their overall integration more difficult and impose serious constraints on design creativity, while they do not help design considered as 'the ability to choose from different solutions'. It is generally recognized that the solution of the problem lies in efficient forms of collaboration among all the actors involved in a project. However none of the forms and tools proposed hitherto has been found satisfactory.
The essay shows how the essential basis of all forms of collaboration lie in the representation and management of the knowledge activated along the design process. It subsequently illustrates an innovative distributed Knowledge-based system that allows effective and creative collaboration among the actors. By virtue of the interoperability established among the various semantic universes it enhances the level and the quality of the exchanged information among the actors while managing not to change their operating modes.

Keywords: architectural design, collaboration, ICT, knowledge based systems, ontologies

1 Shortcomings of present building design process

Building is one of the most widespread activities in the world. In Europe the Architecture-Engineering-Construction (AEC) sector involves a larger number of professional profiles than any other industry and, between construction and the actual running, absorbs about half global energy consumption.
Currently two critical factors are emerging. On the one hand, the *increasing complexity of the building process and product*, which makes them ever more difficult to manage, and on the other an almost imperceptible but constant reduction in the final *quality of buildings* during the process which turns out as unsatisfactory.
The *process/product complexity* seems to be a direct consequence of our technological culture, bound up as it is with continuous growth and segmentation of technical and procedural rules, codes, changes in the cultural and environmental contexts, fragmentation of activities into parallel and sequential phases, increasing size of building operations, higher performance levels required for the whole and the separate parts of the product and thus becomes an unavoidable component of the daily work of professionals.
In the present essay all the physical and virtual subjects that in any way take part in the design process (humans, firms, designers, users, clients, intelligent software agents and assistants are hereafter denoted as 'actors'.[1]
The *quality of the building* is obtained through this type of process is too often unsatisfactory in terms of the formal results, of the failure to achieve the technical and functional objectives, of excessive energy consumption, of unsustainable environmental impact, and of cost and time overruns.
These two critical factors are linked and interdependent. The inferior quality of

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

the building product is generally a direct result of inadequate design, often the result of problems of communication and understanding among the various actors in the project and is part and parcel of the process itself.
Generally speaking the concept of design extends increasingly into a large multitude of sectors as it is necessary to try and foresee the often unpredictable changes resulting from new inventions and changes in technology, tools, methods and social customs: *design is -pervasive vis-à-vis all the problems*. The present study therefore takes into account the general features of the *essence of design*.
The tendency to speed up the design process by taking other design hypotheses into low consideration often leads to raising and spreading conflicts among actors. Moreover an impoverishment of the final quality of the project is observed vis-à-vis the initial idea of specialist actors. This is the result of both the adoption of simplified, conventional or relatively non innovative design solutions that do not lend themselves to providing higher performance and the praxis that tends to simplify different actors' design solutions in order to avoid reciprocal misunderstandings related to innovative design solution.
The outlined problems therefore demand to seek other forms of process which allow improving project efficiency by cutting down the time required and by favouring the development of creative ideas.
In the quest of a solution of the aforesaid problems lies the motivation of the here presented research aimed at conceiving new means for facilitating an effective interaction among actors in the current complex cross-disciplinary building process.

2 Limits of Current Architectural IT Tools

Complex building design is nowadays a process characterized by a high degree of interdisciplinarity, delocalization and activity breakdown as well as the timely use of information.[2 to 6] This kind of process requires a high degree of collaboration.
Collaboration in building design is an inherent necessity in the AEC sector, as any building is a singular, integrated and complex system in a given context with interleaved problems. The global overall solution can be attained only by means of trade-offs among actors so that they can modify their own specialist goals and adapt their own specialist solutions. Collaboration becomes all the more effective when it takes place at all the hierarchical levels into which a design and construction process is normally subdivided.
Collaboration thus facilitates the discovery of new design solutions through the proactive contribution of each actor; it allows the various solutions by the different actors to converge towards a single overall solution; it helps the actors to reciprocally modify their own design solutions so that they can be more satisfactorily integrated into the overall solution; it allows the reciprocal dysfunctions among the solutions proposed by the individual actors to be detected; lastly, it encourages the development of creativity through interaction among the different skills.[7 to 10]
However collaboration is hard to apply in highly complex projects and processes. Moreover, the difficulties facing collaboration increase in large-scale projects as a result of the scattering of actors in space and time, the different lan-

2 **Björk, B.C.** (1999). Information Technology in construction: domain definition and research issues, Computer Integrated Design and Construction, 1:1, pp. 3-16
3 **Björk, B.C.** (1992). A unified approach for modelling construction information, Building and Environments, 27:2, pp. 173-194
4 **Carrara, G.** et al (2004). Knowledge Sharing, not MetaKnowledge. How to join a collaborative design Process and safely share one's knowledge. In: Pohl, J. (ed.) Intelligent Software System for the New Infrastructure, San Luis Obispo (CA), Cal Poly, pp. 105-118
5 **Carrara G.** & **Fioravanti A.** (2002). 'Private Space' and Shared Space'. Dialectics in Collaborative Architec-tural Design. In: Pohl, J. (ed.) Collaborative Decision-Support Systems, San Luis Obispo (CA): Cal Poly, pp. 27-44
6 **Carrara, G.** & **Kalay, Y. E.** (1994). Past, present, future: process and Knowledge in Architectural Design. In: Carrara, G. & Kalay, Y.E (eds.) Knowledge-Based Computer-Aided Architectural Design, Elsevier Science Publishers, pp. v-vii, pp. 147-201, and pp. 389-396
7 **Gross, M.D.** et al (1998). Collaboration and coordination in architecttural design: approaches to computer mediated team work, Automation in Construction, 7:6, pp. 465-473
8 **Kvan, T**. (2000). Collaborative design: what is it?. In: Martens, B. (guest ed.), Special Issue eCAADe '97, Auto- mation in Construction, 9:4,

pp. 409-415
9 Jeng, T.S. & **Eastman, C. M.** (1998). A database architecture for design collaboration, Automation in Construction, 7:6, pp. 475-483
10 Kolarevic, B. et al (2000). An experiment in design collaboration, Automation in Construction, 9:1, pp. 73-81

guages involved and, above all, due to the symmetry of ignorance whenever the shared knowledge proves to be insufficient.

Efficient collaboration among actors in the design process means that any actor must be able to propose the solution to the specific problem s/he is responsible for solving, in such a way that the other actors can understand it in order to be able to modify their own solutions to adjust to the received suggestions or to consciously object and eventually reject.

For this to be possible several fundamental criteria must be satisfied: communication and acquisition of information must be correct, efficient, secure and unambiguous; the presence of as much specialist knowledge as is required by the complexity of the planned product; the presence of knowledge shared by all the actors involved to allow them correctly to interpret the exchanged information and to understand each other; a semantically and technically correct link between shared knowledge and the universe of specialist knowledge of the various actors.

All this indicates that the fundamental bases of collaboration lie in knowledge, and in the way it is exchanged among actors, regardless of the tools used in the design process.

The introduction of ICT science has radically modified the way information is transmitted in the design process: drawings and documents are transformed into data structures that imply the use of new tools to produce and transfer them.

A wide variety of computing and representation software is available on the market, which is capable of performing even relatively complex tasks within well-defined disciplinary boundaries although designed to enhance the capacity to verify a given design approach rather than to help finding out a design conception. These software applications are actually of no help in design collaboration, and indeed make it more difficult: software specialization increases the difficulty of communication and reciprocal understanding among the various actors, as data required by the different programs differ from one actor to the next even when they refer to the same object.

This lack of mutual understanding is mainly due to the low semantic level afforded by the application programs used by the actors and by the inadequate degree of interoperability of the software used.

Moreover, each type of software demands the input of data that must generally be inferred from the interpretation of the drawing of the design solutions of the other actors involved in the process. In this way different interpretations of the meanings of a same object are cause of misunderstandings that are all the more detrimental the greater the degree to which the actors continue to develop their own specific design solution often turns out to be incongruent with the one of other actors.

Such difficulties are both due to *the lack of an overall model of the building* and of the *design process* that is representative of their complexity and to an *inadequate formalization of information* pertaining to any individual actor and exchanged among the various actors.

In fact an effective formalization of information exchanged along a design process remains an unsolved problem. Each actor makes use of low level formalized semantic information and operates on it by deploying his/her own professional skills and experience.

In AEC community several efforts have been devoted to overcome these difficulties in order to integrate competencies in a single application program and to share knowledge. Among the various initiatives, we mention *BIM* and *IFC*.

BIM (Building Information Model) are product models recently driven by several CAD system firms mainly (Autodesk, GraphiSoft, Bentley, Nemetschek, etc.) which can describe the form (e.g., geometric information and its relationships) and attributes (e.g., physical characteristics) of a building throughout its life cycle. BIM define a building with proprietary formats conceived with a top-down point of view, focused just on components and neither on the process nor on the building as a system, that are a source of intelligent information about a building.

To achieve better interoperability application software across the industry and professionals, an efficacious basis for information sharing between different BIM has become necessary and urgent. To this purpose there has been developed a second initiative with a different approach: Industry Foundation Classes (IFC). It is an open XML standard (OOP) conceived with a bottom-up point of view and non-proprietary data model specifications, proposed by International Alliance for Interoperability (AIA) that is emerging very slowly among involved industries. IFC aims at granting software interoperability while exchanging more significant project data, so that nowadays CAD applications by major software houses can import and export (with some difficulties) their proprietary formatted files from/to IFC files. Such specifications represent a data structure supporting a digital project model useful in sharing structured labelled (more understandable) data across applications, but they are neither intended for *design needs* nor for *mutual understanding* among actors, but mainly for production needs.

Till now exchanging contents 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 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 considered aspects of reality.

BIM has an important role in creating and coordinating components as parts of a building, but actually it does not provide any concept of the building as a 'system' (structured set of components with functions aiming at a goal) such as architects or engineers have had for centuries. Moreover other difficulties rise from the fact that BIM data must co-exist with a number of programs with different task-oriented models, all essential in defining detailed, but partial information of a project.

IFC is based on a central model that can be either partially or entirely shared by participants, but must be accepted as a whole, being totally coherent (it is not scalable from this point of view). Although its approach supports different visualizations of the same '*component*', it is focused on converting and updating 'components' from multiple sources, at the level of the applications, into a generalized description of the entire building.

As a matter of fact current interoperability design problems related to commercial application programs are solved within the domain they have been built for, as very often they all have a similar, but specific, point of view: the one of him who first modelled the phenomenon, probably some thirty years ago. In conclusion, the model underlain the application programs of a specialist domain allows *exchanging data*, but not *inferring concepts* from the application programs themselves, as these concepts *are implicit and tacit between the actors of that domain*.

Such a problem is not a big deal between actors of the same discipline-specific domain, but it is crucial to be solved in cross-disciplinary design in order to allow and improve collaboration. Its solution is not concerned with mere interoperability formats: it is above all concerned with how to makes concepts related to products explicit and understood by all the actors. How to overcome the symmetry of ignorance is still an open problem.

At present an export of application programs to/from the common low ontology level (IFC) for machine interoperability purposes only, is starting to be available - to the extent that software houses support new IFC specifications.

The dominant way of using IFC specifications (low-ontology level) today is still a one-direction batch translation of large data sets from an application into the common language (IFC) and vice versa. Collaboration using IFC specifications exists in the industrial practice, but it is based on ‘ad-hoc’ procedures that are agreed between single specialists for a single project.

As a matter of fact IFC model servers till now implemented provide limited collaboration support and the existing model servers do not support adequate management of the instance versions (with different meanings) of various specialists.

3 An Innovative Knowledge Structure for increasing efficiency and avoiding shortcomings

In order to develop new tools that can efficiently support actor’s design work it is necessary to reflect on *what* is required along the design process in order to: allow any actor to retrieve that part of her/his own knowledge considered relevant for the project, make all the actors correctly understand each other so as to set up an effective collaboration in order to activate all this in a complex process.

A deep and detailed examination of the actual design process shows that knowledge required to develop a project is possessed by the actors in a heterogeneous, diversified and discontinuous way; it is also found that when the actors share even a limited part of knowledge they are better able to interact and understand each other. Thus to increase the efficiency of the design process any actors have to share with all the others a bigger part of their knowledge deployed in the project that is of common interest, so that everyone can easier exchange and understand that part of information.

A fundamental pre-requirement for applying such a methodology and overcoming the above-mentioned interoperability format problems is *knowledge understanding*, namely *technical knowledge*. Technical knowledge concepts can be formalized and structured by means of the *technology of ontologies*, for defining entities and by means of *explicit semantics* for defining their meanings.

On these premises a model of the structure of knowledge used in the design process as well as of its management has been developed, based on the *formal representation of knowledge*.

In the present context this term includes both the formal structure of the entities considered in a project (and the related aspects i.e. meanings, geometry, properties, relations, etc.) and the formal models (generally mathematical) that allow simulations, verifications and reasoning to be performed.

Thus defined a *Knowledge Structure* (KS) is composed of a set of Entities each of which is related to an *Ontology* (its definition) and has a *Semantics* (its meaning). Each entity can have a set of *Properties* (geometric, physical, values) and *Attributes* (function, methods or computing programs), a set of *Belonging Relationships* with other entities (part-of / whole-of), a set of *Inheritance Relationships* (class-of / is-a), a '*situation*' (or '*Condicio*')[11] dependent set of *Rules of Compatibility* with other entities (check-list, adjacency-list, etc.), *Inference Engines* (IEs) to activate and manage constraints, all of which formalized into a syntactically coherent IT structure.

The aforementioned KS is actually a 'system'. If the structured entities present in a KS aim at a goal: e.g. the habitability, the energy saving, the constructability, etc. A goal is achieved through several objectives and sub- objectives. E.g. the habitability includes the spaces' usability, the ergonomics, the space brightness, reciprocal disposition of spaces, spaces' relationships with the outside, etc.

To make it possible entities in a domain are related each other by means of specific relationships and IE the *Relation Structures* (RS).

Any actor involved in design process manages his/her/its own entities in order to attain his/her/its own specific goal. The key element of a KS, anyhow formalized and structured in the fields of architecture, energy saving, sustainability, buildings stability, etc., is the definition of entities involved in a domain, therefore on its ontology.

For all these reasons scientific communities had a new interest in the field of *ontology* that provides a valuable support for representing and sharing terminology, concepts and relationships within a given domain, so that an increasing number of communities of experts develops ontology as an underlying base for their work, including collaboration in design.[12] Actually in the growing area of network services new approaches to composition and orchestration of services are based on ontologies for representing their definitions, i.e. for disambiguating queries.

At present most entities of a specialist ontology are typically not explicit and are inherent in the model of the phenomena they are referred to [13], so that in commercial tools part of the knowledge is implicit, hidden-coded in application programs and is neither openly available nor fully understandable to actors, so that an *Explicit semantics* ('situation' dependent) is needed to make entities understandable by humans and tractable by computers. As actors take into consideration projects' entities at different levels of abstraction, from data level to reasoning level, an *ontologies based methodology* allows the actors to use in a coherent manner different levels of abstraction and/or to exploit a conceptual interoperability.

All Knowledge required in a design process can be split into a *Common Knowledge*, which allows all the actors to communicate and essentially to understand each other, and as many sets of *Specialist Knowledge* as the disciplinary domains

11 **Carrara, G.** & **Fioravanti A.** (2004). How to construct an audience in Collaborative Design - The Relationship among which Actors in the Design Process. In: B Rudiger, B Tournay and H Orbaek, (eds), Architecture in the Network Society, 22nd eCAADe Congress, pp. 426-434

12 **Ugwu, O.O.** et al (2005). Ontological foundation for agent support in constructability assesment of steel structure – a case study, Automation in Construction, 14:1, 99-114

13 **Drakos, N.** & **Knox, R.E.** (2004). You need More Than E-Mail to Share Tacit Knowledge. Stamford Gartner Research, available at: http://www4.gartner.com/DisplayDocument?id=450075

14 Italian Standard Organization
15 Carrara, G. & **Fioravanti A.** (2004). ibid

involved in the specific design process, depending on the type and stage of the project considered. A *Specialist Knowledge* therefore includes a set of entities in a specific disciplinary domain (some of them can be the same entities of the Common Knowledge) with specific semantics, properties, attributes and relationships.

The specific tasks of a *Specialist Knowledge* are: to perform simulations and behaviour verifications related to its own goal of entities, to infer reasoning from entities by means of its own RS and IE, to notify suggestions and notifications to the specialist actor and to the other actors concerned.

4 Ontologies as a Base for Knowledge Structures

The ontologies considered in Knowledge Structures include all and only the entities retrievable in any possible object that is definable in the design process.

Referring to building design, there are *two fundamental structured ontologies*: *the one of the spaces and their aggregations*, which in a project make up the so called 'spatial system', and *the one of the physical elements (components) and their aggregations* which in a project makes up *the constructive apparatus*, defined by UNI[14] as the '*technological system*'. These two systems as a whole make up the *ontology universe of Project-Independent Knowledge*.

All the knowledge used in a project is organized into an essential, general, *Common Knowledge* defined as 'light', and a multitude of specialized, self-contained in their own field, *Distributed Knowledge*, defined as 'heavy'. In order to guarantee collaboration and the proper integration of the specialist design solutions into the overall one it is crucial to set up correspondences between the ontology of the entities included in the *Common Knowledge* and the ones in any *Specialist Knowledge*. Specialist Knowledge entities often have so many more characteristics and properties than the corresponding ones of Common Knowledge as to be defined 'heavy'. Specialist Knowledge, in addition to the (light) *ontologies* corresponding to the ones included to Common Knowledge, contains also *specialist ontologies*, which are specific to the specialist domain and are not present in the Common Knowledge, although they may be present in some other Specialist Knowledge. Private Ontologies can refer to aggregations of components that are meaningful only for the specific domain, e.g. an office suite, which is meaningful for the architect but is just a collection of rooms for other specialists.

Through this identity correspondence the entities (or their aggregations, as well as their characteristics or properties) which are specific to a disciplinary domain and formalized in *Specialist Knowledge Structure*, are related to the corresponding entities in the *Common Knowledge Structure*, which contains all and only what is requested to be known by all the actors.

Any '*situation'* (or '*Condicio*')[15] of a design process determines the set of design solutions, made of a number of well defined entities (classes), placed in reciprocal relationships, whose properties a set of values and attributes have been defined.

Inside a *specific project* – a design solution – the ontology set of *Spaces* and the one of *Components* are closely interrelated and established by a set of reciprocal relationships where the characteristics of one are determined by and in turn define the characteristics of the other one.

Any design solution (an individual), called *'instance'*, whether *specialist* or *overall*, is made up of *data* (specific values) and therefore is clearly distinct from the Knowledge (a structured set of classes) used to build it up, which is made up of *metadata* (classes).

The *overall design solution* (the overall instance), which does not necessarily demand to be congruent until the end of the process, is made up of the *specialist design solutions* of all actors (the personal instances) and of the *Common Knowledge solution (Common instance).*

As one of the assumptions of this work is that actors cannot overcome the symmetry of ignorance barrier at a data level or at a low semantic level by means of usual application programs, such a goal can only be achieved by means of:

- mapping concepts of ontologies of different domains by means of a previous agreement among actors and/or by means of rules to state same entities[16]
- managing concepts of ontologies of different abstraction levels by means of inferential engines and intelligent agents in order to have an effective support in design

In order to allow mapping among the entities belonging to different ontologies (ontology mapping), each specialist should be provided with her/his own ontology while all these ones should be partially overlapping. The resulting intersection set, defined as *Common Ontology* (Figure 1), provides the base through which actors can understand each other. In order to relate current problems with past experiences they have to be stored in an appropriate format at the correct knowledge and expertise.

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

17 Nguyen, T.H. et al (2005). Algorithms for automated deduction of topological information, Automation in Construction, 14:1, pp. 59-70

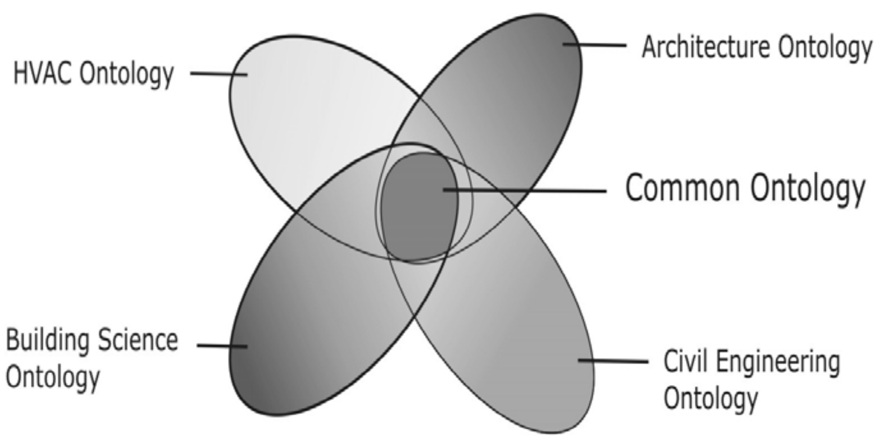

Figure 1 The Common Ontology as the intersection set of several Specialist Ontologies

It is here presented a *Structure of Knowledge* used along a design process, based on *layered levels of intelligence* (Figure 2): an IFC-based *Lower-Ontology-Level* (LOL), a rule-based *Upper-Ontology-Level* (UOL) and a logic-based *Deductive-Reasoning-Level* (DRL).[17] UOL allows its own logic/ algorithmic rules - that can adapt themselves to their '*situations*' (or 'C*ondicio*') – to explain parametric objects, constraints, etc. DRL, with its deductive capabilities, applies inference rules and intelligent agents to UOL entities. This process is facilitated by using IFC standards and is triggered whenever ontologies in an occurring 'situation' are instanced and LOL ontologies of different actors are related each other at DRL.

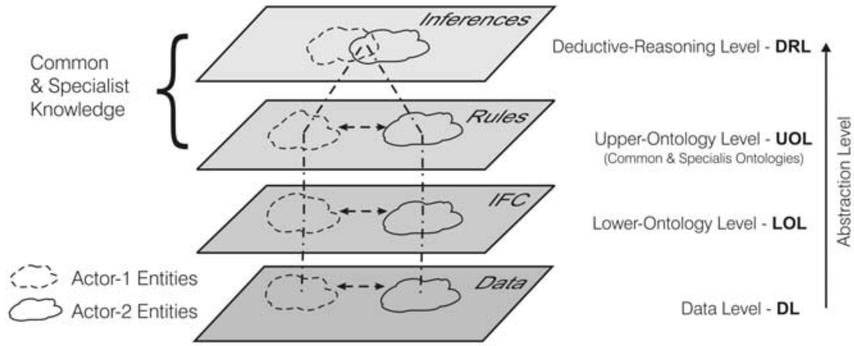

Figure 2 Structure of design Knowledge: Ontologies at Deductive-Reasoning Level map different Actor's ontologies at the Upper Ontology Level so that an inference mechanism can be applied to rules

By means of such a mechanism constraint rules can be transitively chained as much as possible.

With respect to the use of inferential engines and intelligent agents, the here described approach differs from the exhaustive and integral approach till now developed – consisting in importing and exporting all the information in an agreed and identical format such as IFC (that anyway can be a basic reference for semantic matching) – as it is based on:

- Exporting strictly necessary information among actors by means of the Common Ontology. In contrast to a centralized database model, a distributed ontologies model has been developed based over a Common Ontology Domain and several Specialist Ontology Domains.[18 to 21] Each specialist actor's domain retains its own ontologies in the most appropriate form for her/his needs and expertise, while an appropriate interface translates her/his own ontologies into/from the Common Ontology Domain.
- Leaving to an Upper-Ontology level based tool the task of linking representations of the same entities made by an actor at different levels of abstraction (vertical interfaces). The research work has defined an ontology based model that supports actor's design by linking her/his heterogeneous abstraction levels – whose formalization is oriented towards different tasks – with the data of her/his usual application programs, so that the model can point out inconsistency of data (DL), incoherency of constraints (LOL), incongruence of goals (UOL) within any actor's domain.
- Using entities at the Deductive-Reasoning level for mapping entities of different actors of the Upper-Ontology level (horizontal interfaces). Ahead of time actors can acknowledge the implications of their proposed project solutions considering other actors' points of view, constraints and goals by means of inference mechanisms at the Deductive-Reasoning Level. These ones are able to establish a mapping among the same entities present in different domains so that constraints (e.g. a constraint on a dining room surface), rules, goals among ontologies belonging to different domains can be chained.

Incoherencies are detected by an inferential mechanisms contained in the DRL as it points out (if it exists) a common entity (e.g. a pillar) in the Common Ontology by mapping the ontologies (and their structures) of different domains, checking then contradictions among these ones (e.g. different acceptability ranges of dimensions); subsequently it reports feedback information to actors involved (who

18 Carrara, G. & **Fioravanti, A.** (2007). Collaboration, New Media, Design- An Integrated Environment for Supporting Collaboration in Building Design. In: Pawlak, A., Sandkuhl, K. & Indrusiak, L.S. (eds), Coordination of Collaborative Engineering – State of the Art and Future Challenges, GI-Edition Gesellschaft fur Informatik, Köllen Druck + Verlag, Bonn, pp. 143-160

19 Carrara, G. & **Fioravanti, A**. (2001). A Theoretical Model of Shared Distributed Know-ledge Bases for Collabo-rative Architectural De-sign. In: Gero, J. & Hori, K.(eds.) Strategic Know-ledge and Concept For-mation III, Key Centre of Design Computing and Cognition, University of Sydney, pp. 129-143

20 Carrara, G. et al (2004). ibid

21 Leeuwen, van J.P. & **Zee, A. van der** (2005). Distributed object models for collaboration in the construction industry, Automation in Construction, 14:4, pp. 491-499 have pillar constraints in his/her/its own Specialist Ontology) so that they can provide the necessary action.

The dynamic and semantically-specific representation detecting incoherent/favourable situations by means of a constraint rule mechanism can allow them to be highlighted and managed in real time. At the same time it allows actors to make alternatives, more consciously reflecting on the consequences of their intents. By this way knowledge spreading throughout the networked ontology-based environment makes actors more aware of other specialist constraints allowing them to operate more participative and shared choices.

The integration of the specialist actors' design solutions (instances) translated into sub-sets of the overall design solution (instance) can give rise to inconsistencies and conflicts among instances belonging to different workspaces or to ontologies of different domains.

5 Knowledge 'Translation' by a Filter Mechanisms

To interface common knowledge with specialist knowledge a specially conceived mechanism is required. This is made of *Filter* that allow *any specialist actors design solution* (all information that is specific to each *actor)* to be directly and automatically 'simplified' and 'translated' into the '*common language*' of the overall solution. Since this one is correctly read and interpreted through the Filter, it leads to mutual understanding among all the actors involved in the design process. The representation of this 'simplified translation' is called '*Common View*'.

Common Knowledge, including all and only knowledge shared and agreed in times by *all* the actors, takes on a fundamental role required in the current complexity of the design process, too often lacking in practice, allowing the actors to interact and to mutually understand each other at a basic level, with respect to *the design solution* as it is progressively processed.

The integration of the *personal instances* into the *overall instance* is achieved through the Filter mechanism that acts at the level of classes (knowledge filter) and at the level of *instances* (data filter).

The *Knowledge Filter* works at the level of concepts (ontologies, properties, relations, values) and acts as an intermediary between each Specialist Knowledge Structure and the Common Knowledge Structure: it recognizes among the entities selected by each actor in his/her/its own Specialist Knowledge Structure those ones that are present in the Common Knowledge Structure, selects them and determines the corresponding sub-set of them inside the latter.

The second filter, the *Data Filter*, works *at the level of individual data* and acts as an intermediary between each individual data structure representing a *personal instance* and the data structure representing the *overall instance*: it is triggered by the first filter and recognizes among the data of each *specialist instance* those ones corresponding to the entities selected by the *Knowledge Filter* in a given 'Condicio' and translates them to a sub-set of the data structure representing the *overall instance*.

In order to ensure a dynamic and interactive knowledge exchange among the actors, which is an essential prerequisite for an effective collaboration, a suitable organization is required of the *Design Workspace*.

This term represents the '*place*' in which the design activity is performed and metaphorically corresponds to the 'professional office' in which actors work in the conventional design process.

The distribution and centralization of Knowledge corresponds to a similar arrangement of the *Design Workspace* in which the actors are called upon to work. This is therefore divided up into a number of *Private Design Workspaces* - specific to each individual actor and corresponding to the number of specialist actors involved - as well as into the *Overall Design Workspace*.

Design Workspace presents a distributed structure of *Private Design Workspaces*, each one referring to one of the numerous actors involved in the design process for the implementation of specialist solutions. This structure is directly linked to a *Overall Design Workplace*, shared by all the actors, so that they can visualize the merging of all the simplified partial solutions and recognise which one can be subjected to verification.

During the process each actor can therefore create or modify his/her/its own *Personal Design Instance* in his/her/its own *Personal Workspace*, using his/her/its own specific Specialist Knowledge and his/her/its own personal tools. This instance is part of the *overall design instance* constituted by the merging of all the *partial specialist instances* in the *Overall Workspace*.

During the whole design process each actor is able to verify his/her/its *own personal design instance* using his/her/its own *Specialist Knowledge* (his own ontologies and deductive capabilities) and whenever it is deemed satisfactory he/she/it can, by means of the filters translate it into the '*common language*', combine it with the *personal design instances* of other actors and verify it in the Overall Design Workspace through the *Common Knowledge.* This is still a personal 'test phase' of the *personal design instance* with respect to other actors' constraints as far as an actor does not consider to show the other actors her/his solution with related advices or warnings

When an actor deems satisfactory her/his own partial design solution (whether verifying or not constraints), s/he can '*publish*' her/his own instance in the *Overall Design Workspace* so that it becomes visible to all the actors and can be queried.

The *published* design instance is simply one version of the project's evolution, which does not have to be totally consistent. It is possible, at every stage of the process other than the final one, for the design instance (the project) to be inconsistent both internally and/or with the other instances. The process finishes when actors agree that a design solution is acceptable. Through the interaction among the personal instances worked out in the *Private Design Workspaces* and shown to all the actors through the *Overall Design Workspace,* a *flexible and continuous interaction* is established among the actors in spite of their reciprocal interdependence, thus *setting the conditions for a genuine and efficient collaboration*.

Summarizing, the described Knowledge Structure is operationally subdivided into two basic levels – that of *knowledge* and that of *data*. The upper level (*Knowledge level*) can be conceived as split into a *Project Independent Knowledge* and a *Project Dependent Knowledge*. On its turn the former is split in an *Ontology Layer* including entities, properties, relationships and rules, and, on top of this layer, in a *Deductive-Reasoning Layer*. Heretofore *Project Independent Knowledge* has been subdivided into *Specialist Knowledge* and *Common Knowledge*. Altogether it can be viewed as a sort of formalized handbooks that are directly and dynamically linked among themselves and to the ongoing design project. In the specific *Project Dependent Knowledge* any actor, through her/his own *Specialist Knowledge,* builds up on previous ones her/his own new entities (ontologies, relations, attributes and rules) relevant for the specific project. In the lower level (or the *Instance Data level*) *data structures* are defined that make up both the specialist and the common design instances, depending on the specific design project. Instance Data consist essentially of the *values* of the attributes and of the relations among the entities defined in the upper levels by the actors that are progressively specified in the course of the development of the design process together with the corresponding entities in the upper levels that can be modified at will (Figure 3).

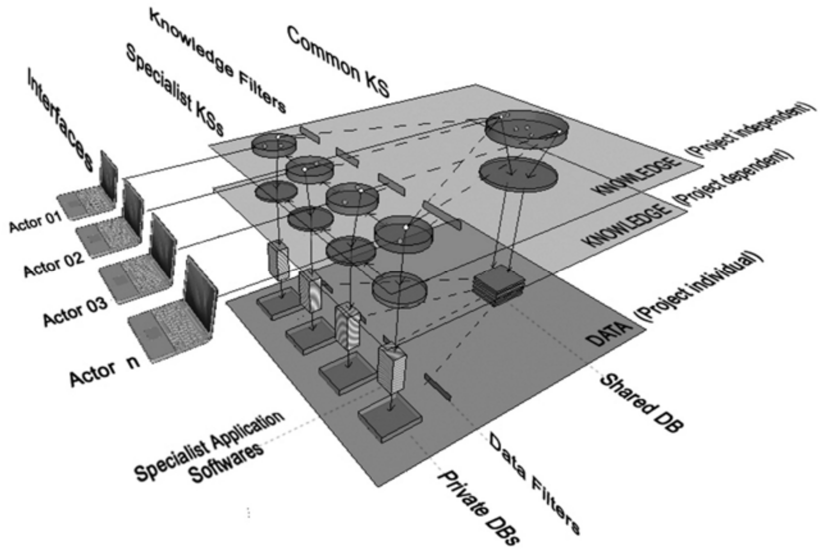

Figure 3 System for collaborative building design based on Knowledge Structures

Any actors proceed to model the behaviour and verify the internal consistency of their own specialist instance using her/his own design methods and techniques with the help of his/her/its application software.

6 Early Prototype Implementation and Future Developments

The paradigm proposed in this essay aims at improving design methodologies in AEC industry by means of a novel approach in structuring and managing knowledge: this approach requires an adequate support by IT technology. This one is mainly currently focused either on single-step or single-actor view point (standard CAD applications and suites), or on standard formats (IFC, BIM) quite complex and suitable only for machine-oriented data management. Notably, the research focused on:

- exchanging information among the actors throughout the whole design process, starting from the early stages
- adopting an ontology suitable to support the required representation of knowledge at various levels of abstractions, heterogeneous terminologies, semantic relationships

- splitting knowledge used in the design process in Common / Specialist Knowledge and Project-Independent / Project-Dependent Knowledge

The described Collaborative Working Environment has been implemented as a demonstrative prototype system, able to support a highly interactive collaborative design processes among three specialist actors in the field of Architecture, Structural and Mechanical Engineering. We have proposed framework architecture for a modular software platform supporting collaborative design. Some of the main features, partially implemented in the CoKAAD prototype system, are summarized below.[22]

22 **Nanni, U.** & **Santacaterina, A.** (2009). CoKAAD: a Framework for Collaborative Architectural Design. In Carrara, G. et al eds (2009). Collaborative Working Environments for Architectural Design, Palombi Editori, pp. 225-235

All the users are intended to make use of the same software installation, intended to be used by a set of actors during the development of a project by alternating online (connected) and offline (individual) work sessions. A client-server logical configuration has to be defined at the beginning of each online work session, designating a server; each actor has an initial option of synchronizing the offline work or importing the current state of the design activity.

The main tasks of the designated server are: accepting and managing online collaborative work session (including connection control and concurrency management), managing the Overall Workspace and broadcasting each shared action to the designated actors. A fundamental task of the server is to record the operations carried out by the various actors in the Overall Workspace, maintaining a session log recording each single action, and a version history tree.

The user interface of a CoKAAD client is a desk with a collection of tools both for the individual work and for shared activity and communication with other actors; these tools include:

- standard collaborative environments (e.g., text and voice communication, file sharing)
- a free-hand drawing tool
- an installation of a CAD application with a plug-in intercepting all the relevant operations (currently we have interfaced Autodesk Architectural Desktop) for online teamwork
- a set of tools for the client-side control of the services provided by the server: connection control, access to session log and version history, with the possibility to run back and forth the log and the version tree

Additional background tasks of the client application are: intercepting meaningful actions in the Overall Workspace (to be forwarded to the server), applying filters to the incoming and outgoing flow of information, managing the Personal Workspace, and the individual view of the Overall Workspace.

The architecture of the here presented framework includes an engine for the representation and querying of ontologies. This is required to handle and combine the modular Structure of Knowledge, split in Common Knowledge, Specialist Knowledge (a distinct one for each actor), Project-Dependent Knowledge. The key task of concept matching may be performed at various levels of abstraction to activate a rule or a constraint and to clear terminology mismatches. This problem is tackled by means of deductive rules – the Reasoning Level above the Ontology Level.

The described Knowledge Structure has been tested by means of a study use case: a meta-design of a demonstrative hospital ward. Such a case study was chosen for the following reason: it is a complex structure even in the case of re-

23 Chen, P.H. et al (2004). Implementation of IFC-based web server for collaborative building design between architects and structural engineers, Automation in Construction, 14:1, pp. 115-128
24 Calvanese D. et al (2008). Conjunctive query containment and answerring under description logic constraints, ACM Transactions on Computational Logic, 9(3):22.1–22.31, 2008

duced physical size, has similar requirements in all the EU countries and demands the contribution of numerous highly differentiated specialist skills that must be melded into an organic and balanced solution.
At present the implementation of such a demonstrative prototype system is under way, in order to support a highly interactive collaborative design processes among three specialist actors in the field of Architecture, Structural Engineering and Building Science.[23]
This implementation will make use of QuOnto (www.dis.uniroma1.it/~quonto/), developed in the past years at Sapienza University of Rome. This system, based on Description Logics, has proven to be computationally very efficient and robust enough to be used in productive environments with million instances.[24]

7 Conclusions and Expected Results in Building Collaborative Design

In this paper we propose how to organize knowledge in an integrated collaborative AEC design environment. We do not address many other issues concerning the design process and how it might be supported by a computing architecture. As an example, sharing information among actors can be done straightforwardly by using the Overall Design Workspace, or by implementing services that support query/answering among the individual installations – i.e., the Personal Workspaces.
In the early stages of the design activity, as well as any time an actor wants to open and explore a new alternative in the project, it is important that only a “draft” of a new solution may be specified. In order to avoid frustrating the creativity of an actor, it should be possible to express only some features of the alternative choice, i.e., a draft. It is important how the representation of such a Personal Design Instance, in a stage of draft, is handled:

- first it is represented only in the Personal Workspace
- then it is to be shared in the Overall Design Workspace

The existence of different abstraction levels in the Knowledge model (with ontology mapping at all levels) allows an actor to specify and share with others a new alternative: possibly this can be inconsistent, missing many details, but cheap (for the proposer) and sufficient to express an emerging new idea. Note that this would be not possible with a full-featured data model requiring strong consistency.
By means of the presented Knowledge Structure, each actor is allowed to work using his/her/its own personal methods, algorithms, software and tools to represent and manage the complexity of his/her/its own instance as a solution of her/his own design problem. Beside this activity (that is what any designer makes today with current tools), the actor is supported while pouring in the current project representation the new design steps: this support consists of the mapping among ontologies with the inferential engine, and the filtering mechanism. The actor has the added advantage that, although the other actors cannot enter his/her/its own ‘Space’ they are able to interact with the constraints/opportunities there defined. Each actor – human helped by intelligent assistants can define the instance data of his/her own design instance by explicitly mapping them with the conceptual entities they belong to, which are structured by relations and rules significant in her/his/its Specialist Knowledge. Actually

the direct link between concepts and instances and the translation of both of these ones into entities common to all the actors make it possible design collaboration, in the broad sense of the term, on the basis of a mutual comprehension and the sharing of the choices.

This evidently differs from what currently takes place in existing CAD systems in which the attribution of data to concepts is implicit, arbitrary and related to the subjective interpretative capacity of the various actors. Just for this reason there does not (and cannot) exist in these systems a knowledge or a data structure shared by all the actors, thereby ruling out any form of direct 'on-line' design collaboration. A comparison of the structure of existing CAD systems and the one here described shows how the principal innovation introduced by the latter consists in structuring, managing and sharing knowledge.

The expected results of the proposed Knowledge Structure are the following ones.

As specific to the building design process:

- a more detailed investigation of the process logic, a comparison between the latter and the pathway envisaged by current legislation and regulations, an identification of critical aspects of the process in order to improve it
- a better control and management of the design activity and the project's evolution to improve their quality/cost ratio, as far as the various phases (early conception, preliminary, detailed, constructive)
- a more competitive advantage of construction industry improving the efficacy and efficiency of the product-chain by means of an asynchronous communication and knowledge sharing and expertise
- a more coherent design logic: underlying reasons, intrinsic coherence, relations with the 'situation', iteration between actors and object, aware criteria of choice
- a deeper exploration of the nature of collaborative design processes as an on equal terms collaborative process, from early conception, through manufacturing, to construction and maintenance
- a competitive advantage to the production process, as an effective Collaborative Working Environment increases creativity and spreads innovation that play a key role in market success

As specific to educational and social outcomes:

- an e-learning tool, a 'game', that can assist university students; it can make easier to explore design solution, acquire knowledge, be aware of design constraints in a complex field as it is architectural and building design process. The 'game' could be easily applied to other educational field
- a general promotion of knowledge among professionals and workers, by the spreading of e-learning tools in industry, school, services' society

To attain these goals a strong and 'collaborative' partnership is needed with European ICT industries, software houses, design firms, engineering firms and construction industries, along with universities and professional training.

Bibliography

Björk, B.C. (1992). A unified approach for modelling construction information, Building and Environments, 27:2, pp. 173-194

Björk, B.C. (1999). Information Technology in construction: domain definition and research issues, Computer Integrated Design and Construction, 1:1, pp. 3-16

Calvanese D., De Giacomo G. and **Lenzerini M.** (2008). Conjunctive query containment and answering under description logic constraints, ACM Transactions on Computational Logic, 9(3):22.1–22.31, 2008

Carrara, G. and **Fioravanti, A.** (2007). Collaboration, New Media, Design-An Integrated Environment for Supporting Collaboration in Building Design. In: Pawlak, A., Sandkuhl, K. and Indrusiak, L.S. eds, Coordination of Collaborative Engineering – State of the Art and Future Challenges, GI-Edition Gesellschaft fur Informatik, Köllen Druck + Verlag, Bonn, pp. 143-160

Carrara, G. and **Fioravanti A.** (2004). How to construct an audience in Collaborative Design - The Relationship among which Actors in the Design Process. In: B Rudiger, B Tournay and H Orbaek, eds, Architecture in the Network Society, 22nd eCAADe Congress, pp. 426-434

Carrara, G., Fioravanti, A. and **Nanni, U.** (2004). Knowledge Sharing, not MetaKnowledge. How to join a collaborative design Process and safely share one's knowledge. In: Pohl, J. Ed, Intelligent Software System for the New Infrastructure, Cal Poly, San Luis Obispo, pp. 105-118

Carrara G. and Fioravanti A. (2002). 'Private Space' and Shared Space'. Dialectics in Collaborative Architectural Design. In: Pohl, J. ed, Collaborative Decision-Support Systems, Cal Poly, San Luis Obispo, pp. 27-44

Carrara, G. and **Fioravanti, A**. (2001). A Theoretical Model of Shared Distributed Knowledge Bases for Collaborative Architectural Design. In: Gero, J. and Hori, K. Eds, Strategic Knowledge and Concept Formation III, Key Centre of Design Computing and Cognition, University of Sydney, pp. 129-143

Carrara, G. and **Kalay, Y. E.** (1994). Past, present, future: process and Knowledge in Architectural Design. In: Carrara, G. and Kalay, Y.E. eds, Knowledge-Based Computer-Aided Architectural Design, Elsevier Science Publishers, pp. v-vii, pp. 147-201, and pp. 389-396

Chen, P.H., Cui L., Wan, C., Yang, Q., Ting, S.K. and **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

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

Drakos, N. and **Knox, R.E.** (2004). You need More Than E-Mail to Share Tacit Knowledge. Stamford (CT) Gartner Research, available at: http://www4.gartner.com/DisplayDocument?id=450075

Fioravanti A. and **Carrara, G.** (2007). Philosophy and structure of a CWE-based Model of Building Design. In: Cunningham, P. and Cunningham, M. eds, eChallenges 2007, IOS Press, pp. 1-12

Gross, M.D., Yi-Luen Do, E., McCall, R., Citrin, W.V., Hamill, P., Warmack, A. and **Kuczun, K.S.** (1998). Collaboration and coordination in architectural design: approaches to computer mediated team work, Automation in Construction, 7:6, pp. 465-473

Jeng, T. S. and **Eastman, C. M.** (1998). A database architecture for design collaboration, Automation in Construction, 7:6, pp. 475-483

Kolarevic, B., Schmitt, G., Hirschberg, U., Kurmann, D. and **Johnson, B.** (2000). An experiment in design collaboration, Automation in Construction, 9:1, pp. 73-81

Kvan, T. (2000). Collaborative design: what is it? In: Martens, B. ed, Special Issue eCAADe '97, Automation in Construction, 9:4, pp. 409-415

Leeuwen, van J.P. and **Zee, van der, A.** (2005). Distributed object models for collaboration in the construction industry, Automation in Construction, 14:4, pp. 491-499

Nanni, U. and **Santacaterina, A.** (2009). CoKAAD: a Framework for Collaborative Architectural Design. In Carrara, G., Fioravanti, A. and Kalay, Y. (eds.) Collaborative Working Environments for Architectural Design, Palombi Editori, pp. 225-235

Nguyen, T.H., Oloufa, A.A. and **Nassar, K.** (2005). Algorithms for automated deduction of topological information, Automation in Construction, 14:1, pp. 59-70

Ugwu, O.O., Anumba 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, 99-114.

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

Subscription Order Form

International Journal of ***Design Sciences & Technology***

Please use BLOCK letters

Title, Initials and Name:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Organisation:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Postal Address:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Postcode:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

City:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Country:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Telephone number:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Fax number:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

E-mail:
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

Institutional (libraries and organisations):

☐ 1 volume (2 issues per year) € 90.00 (Only in France)
☐ 1 volume (2 issues per year) € 100.00 (Outside France)

☐ 1 volume (2 issues per year) € 55.00 (Personal Subscription Only)

☐ Payment enclosed by cheque drawn on a French bank
☐ Payment to be sent in Euros via a banker's draft

Please ensure that your bank covers any transfer charges.

Signature: Date:

Please make cheques payable to: **Europia Productions**
Account N° 30002 00442 0000006991Z 58 Banque CREDIT LYONNAIS
Agence Paris Marceau 44 Avenue Marceau, 75008 Paris, France

Complete and return this Order Form to:
Europia Productions, 15, avenue de Ségur, 75007 Paris, France
Tel +33 1 45512607 - Fax +33 1 45512632 E-mail: dst@europia.fr