“Divide et Impera” to Dramatically and Consciously Simplify Design

The mental/instance path - How reasoning among spaces, components and goals

Antonio Fioravanti1, Gianluigi Loffreda2, Davide Simeone3, Armando Trento4
Sapienza University of Rome - Italy.
http://dau.uniroma1.it
1antonio.fioravanti@uniroma1.it, 2gianluigi.loffreda@uniroma1.it, 3davide.simeone@uniroma1.it, 4armando.trento@uniroma1.it.

Abstract. In our times, in a complex and universal village where problems are intertwined and pervasive beyond our imagination, we need new approaches to deal with them – appropriately. In a previous work we highlighted the importance to reason ontologies: a ‘world’ f.i. a building – as a mental image – is not a Linnaeus’s classification (structured set of entities) but a system (goals oriented set of classes) able to reasoning upon selectively chosen entities belonging to different Realms (ontology universes) (Fioravanti et al., 2011a). The general aim of our research— to be an effective aid to design – is to simulate wo/man as designer and user of designed spaces, hence how mental skill can be computably included in new tools able to tackle these problems. This paper is focused on the first role: how actor-designers approach design problems and how the inference mechanism can help them and affect the design process. A ‘Building Object’ - the dual system of Spaces and Technology elements – is inferred in several ways according to different goals and the inference mechanism can, simulating human mental shortcuts, optimize thinking.

Keywords. Design process; design operational theory; thinking optimization; inferential mechanisms; human-machine collaboration.

INTRODUCTION: ‘AIDED’ DESIGN

In the world, which has become a single global village characterized by increasingly complex relations, interdependence and now universal problems, we need tools and methods in order to ‘predict and govern future situations’ – i.e. design – that should be at the same time “simple” in order to focus the attention on possible concrete and realistic solutions.

Reflections upon these tools and methods accompany the history of humankind and these were more and more systematically developed and deeply explored from industrialization age on. Simon (1996) claims that in Industry Age there is progress (also meant in a broader sense as quality and quality control), when a certain work is freed from worker personal skills.
This statement can be considered valid also in Post-industry Age when the most distinctive activity is design and beside the original concept “to substitute human skill” new concepts rose: to support, to complement and to aid humans. Nowadays the key word could be “to enhance human capacities”.

Consequently, the basic idea pervading our CAAD community is that, by freeing the designer from tasks that can be progressively delegated to ICT, s/he can concentrate her/his efforts and creativity on higher level problems with which, for the time being, ICT has greater difficulty in coping till now. Nevertheless these limits are moving ahead: designer possibility horizon becomes wider. In this approach an explicit man-machine ‘collaboration’ is declared: the research group is thus in the mainstream of pure computed ‘aided’ design in which the designer can, at any time, turn off the design aids/constraints of application programs. At the same time, in full awareness, the designer can rely on the default mechanism which allows, at any step of the process, ICT entities to be instantiated, albeit only as regular values (namely the defaults). The research group is therefore a considerable distance from the philosophy of “automated design”.

In actual fact, the pioneering and initial phase of Computer Aided Architectural Design is over and tools, a time focused on number processing and on verifying that equations referring to physical phenomena are respected, are now cleverly directed towards solving higher level problems, but are often inadequate for this purpose.

NEW AIMS FOR CAD: FROM AIDED TO ‘ENHANCED’

In order to be effective and achieve a quantum leap in the field of Computer ‘Enhanced’ Architectural Design - CEAD -, the model of the building its definition and its behaviour - i.e. architectural design - must take into account:

- Relations between the building and “wo/man" in all his complexity, corporeality and sensitivity.

To do this it is necessary for “material humans” (like super-avatars) to be as realistic as possible in order to interact with the building - digital physicality.

- Logical processes on associated entities to define the building and the relevant design process. Reasoning procedures need to optimize the search path (for the solution, for constraint checking, for instantiation, etc.), so it is useful to imitate the designer’s mental path as experienced for centuries. The digital world needs to be brought closer to the real world through the omnicomprehensive nature of all its parameters, including both physical characteristics and human arguments – physical digitality.

This view leads us to appreciate that the current architectural design models (made up of the building and the process) have two shortcomings: on the one hand, a short-sighted view of the role of ‘material humans’ in using the building, in exploiting it culturally, in enjoying an aesthetic gratification; on the other, dull inference engines used to logically process and to explore knowledge that they ultimately populate with instances the knowledge domain on which they act. The first point is not treated in the present paper; however, research by our group is now under way. The second point is instead the subject of the present paper: Inference Engines and how they instance prototypes.

OBJECTIVE: ‘COLLABORATIVE’ COMPUTER

Traditional methodologies and tools, based on meetings and direct interaction among actors are very efficient in dealing with architectural design problems, but have shown their limits in present design process characterized by a high degree of inter-disciplinarity, delocalization of activities, subdivision of activities, timely use of information and the correct use of the more advanced methods and technologies, - in a word: complexity.

In order to manage these problems effectively it is necessary to develop new methodologies and innovative tools. At present, among the forms of actors’ interaction in the design process, the Collaborative Design paradigm (Kvan, 2000; Woo et al.,
The fundamental bases of collaboration reside in knowledge (understanding, timely, appropriate), consent (social habits, joint results) and in the way it is communicated among designers (real time, to whom, how much, selectively, device).

However, a knowledge-based system for architectural design (Carrara and Fioravanti, 2010) before ‘enhancing’ collaborations among different specialised designer teams -‘external’ collaborations- should enhance collaboration within the specialist designer team -‘internal’ collaboration-. Such new knowledge-based systems leverage collaboration between a designer and her/his specialist knowledge -her/his ontology- . To realize such an ‘internal’ and afterword, ‘external’ collaborations it is needed a ‘new’ building model able to include these characteristics.

TECHNOLOGY LACUANAE AND ‘SYSTEMIC’ BUILDING MODEL

In the CAAD community a number of efforts have been devoted to overcoming these problems in order to integrate competencies into a single application program and to store and share knowledge. Design is much more than describing a component of a building (Archea, 1987) as it is an activity aimed at helping the actor-designer to conceive of artefacts, to record expertise, to implement experience-based design rules and at “... changing existing situations into preferred ones” (Simon, 1996, pg. 111).

These aims are difficult to reach as technology and methodology lacunae of present application programs to realize and implement such objectives mainly due to the lack of an overall and unitary model of the building that is effective for actor-designers and user, representative of its complexity and even capable of introjecting aspirations and processing them.

Nowadays the formal representation of BIM and IFC does not contemplate these aspects as they consider a building as an assembly of entities of classes (class = hierarchical set of entities).

A building is instead an ‘actual’ system: several classes (ontologies) directed towards goals (e.g. habitability, energy saving, constructability, etc.) (Fioravanti et al., 2011a).

To make possible a ‘systemic’ building model, the Research Group has formalized:

- **specialist knowledge** by means of ontologies - Knowledge Structures, KSs - in the field of Architectural Design and that can be amplified during the design work so as to capitalize on the knowledge and design rules and to effectively aid a designer ‘on tap’.
- **Relational Structures and Inference Engines** that selectively relate entities, concretely instance these entities and push the instantiation process towards a goal (instantiation rules: priority, exclusion, congruency).

The above-mentioned model is based on a highly structured, formal representation of the knowledge used along the whole design process, expressed by means of Knowledge Structures.

The Knowledge Structures – KSs – are basically all structured in the same way: a set of ontology, corresponding to the ‘objects’ the final product is made of (physical elements, spaces, site, etc.).

The objects on which the design process acts are:

- Space Units (SUs), organized in Building Units (BU) the building is spatially made of.
- Functional Elements (FEs), organized in Functional SubSets (FSSs) the building is physically made of.

Any set of ontology can be linked to (already experienced) ‘good solutions’ and to ‘codes of practice’ as well to coherency rules. By assigning values (data) to a KS ‘slot’ any actor-designer defines features of an object thus activating a ‘design proposal’ of his/her solution.

A ‘NEAT’ LOGICAL FORMALIZATION FOR NEXTGEN BUILDING MODELS

As above stated the novelty of a ‘systemic’ building model mainly resides on a Relation Structure - RS -
that selectively relates ontologies and on a Inference Engines that chooses the instantiation path and rules.

To make this possible entities of one class and others of another one, are related to each other by means of specific relationships, which an Inference Engine - IE - can use to perform a goal (i.e. just a simple instantiation process!). The entities and their ontologies on which RS and IE act are very different, those can be procedures, HC plants, fire escape paths, etc.

With reference to buildings, there are two fundamental ontology classes: that of the spaces (rooms) and their aggregations, which in a project go to make up the so-called ‘Spatial Class’ domain, and that of the physical elements (components) and their aggregations, which in a project make up the constructive apparatus, defined as a ‘Technological Class’ domain. For a specialist actor (designer or user) the Building is made by her/his Spatial Class – Ω – and her/his Technology Class – Ω⁻¹ – plus her/his RS and IE (fig. 1). The two classes both have a semi-lattice structure. Correspondingly they are subdivided into Room Domain (hierarchic) and Elementary Space Domain (lattice), and Constructive Domain (hierarchic) and Material Domain (lattice).

The main characteristic of entities is related to the ‘type’ of entity: the membership “class”. This is formalized by means of a custom-made frame structure, similar to the one investigated by McCarthy (1960), by means of an ISA (Is-A) slot. Our frame has a four-tier structure: frame, slot, facet, value.

This way, the model is able to manipulate also the type of an entity’s structure so it allows a designer not only to change the inheritance of an entity but also to mix entity assemblies. The freedom a designer obtains from this formal logic enables her/him to compose an entity of a class also from

---

**Figure 1**

Optimizing thinking in architectural design – new building: not all entities of domains are involved. An example of swapping between two domains: an ‘WO-Whole-Of-swap’ from entities of a Spatial Class domain – Ω to ones of Technology Class domain – Ω⁻¹.
entities of different classes belonging from heterogeneous domains, for example, a room of a ‘Spatial Class’ domain with a pillar of the ‘Technology Class’ domain.

The Spatial Class together with the Technology Class contribute to define a building by means of the RSs that link the two domains (normally separate) through a ‘swap’ of the composition relationships WO (Whole-Of) slot allowing an assembly of mixed entities (fig. 1).

**MENTAL/INSTANTIATION PROCESS PATH**

At the time of instantiation this peculiarity makes it possible to simultaneously verify the constraints that are normally separated on ‘parallel’ logical planes: classes of different domains.

It is important how a Relation Structure - RS -, by means of an Inference Engine -IE -, explores and populates Knowledge Structures when the designer wishes to instance them.

As claimed in our previous work (Fioravanti, 2011b, pp. 181-183 and fig. 5) the architectural (or structural, or engineering, or...) concept of a Building is more than the sum of ontologies. Building is a system = goal oriented classes = RSs + ieS + ontologies. Now it is needed to take a closer look at an RS and its IE engine mechanism.

When designer wants to instance an entity it means s/he wants to populate entities of a class with value(s). We developed two implementations of instantiation process in Protégé and in Common LISP.

In Protégé implementation, as stated above, each entity consists on a structured set of meanings, properties and rules; referring to the rules associated to the specific entity that is going to be instantiated, there are mainly two kinds of relationships/rules that will be checked by the system in different ways:

- **Restrictions** - ‘internal’ to an ontology - applied to properties of a class/entity by means of its
constraints (Cardinality, Type, Value or their combination by means of Booleans operators, etc.);
• Rules - ‘external’ to ontologies - applied to classes/entities by means of Proposition Logics algorithms.

According to this duality, there are two different phases: the instantiation phase and the specific checking/control one.

In the first one, the IE will check the consistency of the entity by pointing out all the restrictions applied to the Parent Class properties asking for values, specifications, relationships and/or links to other entities or instances; depending on the specific design phase, the designer can specify all the requests and/or leave some (or all of them) filled in with default values (blank or referred to regular values). The IE will then continue pointing out the missing property specification needs, but it will also allow the inconsistent entity instantiation till the end of the overall design process when all the inconsistency, incoherence and incongruences should be solved. Restrictions can represent particular ‘conditions’ applied to the entity properties; according to the inheritance nature of the Object Oriented Ontology Structure, each Class inherits all the properties of its own SuperClass(es); and in turn SuperClass(es) inherits/inherits its own properties and their associated Restrictions; at the SubClass level, each SubClass could present different “Sub” Restrictions to that (or other one inherited) property by associating more restrictive conditions.

As stated above, Restrictions could refer to different entity characteristics:
• ‘Cardinality’ requiring a certain/minimum/maximum number of associated entities (f.i. <Room> has_wall min 3);
• ‘Value’ comparing and checking the instance with predefined values or range (f.i. <Wall> has_height min 3.5 m);
• ‘Type’ verifying associated class(es) to the considered one (f.i. <Window> has_Glass only

Figure 3
Optimizing thinking in architectural design – space metadesign: not all domains are involved. An example of not an ‘WO - Whole-Of-shift’, the design thinking is only inside the Spatial Class domain – Ω.
Referring to this limitation, the results of this check is not so clear and easily readable and understandable by involved actors: especially at the first design phases, the ongoing developed design solution are not coherent and consistent due to changing solution, needs, requirements and specifications and so the check results appears as long lists of warnings, compiling errors, ontology missing values, etc.

At present, the research team, considers this issue one of the reason that contributes to the growing sensation that to support an effective collaboration it is needed, together with actor-designers, an actor-manager, which operates as a Design Project Manager, able to handle management tools, to analyze checking results and verification processes and that owns enough expertise to set timing and communication protocols among actor-designers to individuate their reciprocal needs.

The Common LISP implementation has a more powerful capacity of expressing higher abstraction level concepts, so it is more compact and allows to
give priority to rules of IEs. In this case the instantiation process seems to be apparently simpler, as for an entity the IE sequentially checks
1. Parent entities (by means of ISA relations) and assumes values, defaults and constraints if these ones are not in contrast with its own - a leaf constraint prevails on correspondent parent constraint (OOP);
2. Then IE checks in breadth the sub-entities an entity (an assembly) is composed by (by means of WO relations);
3. In turn the latter explore their parent entities using an a) procedure.
Afterwards, in this way, the instantiation process has populated with value(s) all the parameters required - by the designer or by the default mechanism, whether verified or not.
This process has two drawbacks: it is ‘exhaustive’ for an ontology and cannot relate two or more ontologies concurrently (in the same process and at the quasi-same time).
This means an architect would have to define every space, from the building space to room space to elementary space, in the Space Class - Ω, an exhaustive and tiring process, before considering any building entities of the Technology Class - Ω-1. The same would be true for a structural engineer that can consider only the Technology Class - Ω-1, or a plant engineer and so on.

MENTAL ENERGY SAVING: SWAP FOR CUTTING OFF UNTIMELY ENTITIES
A clever solution to overcome these difficulties would be to imitate - physical digitality - what professionals, architects in particular, have done for centuries, i.e. to take into account other knowledge domain from the beginning (for the sake of example, Ω together with Ω-1) and ‘selectively’ explore the domains involved. The mind always saves and optimizes mental energy: it is a ‘thinking economy’. It is actually usual for architects, at every step of the design process, to define some different entities at different scales belonging to different ontologies. A master architect has the natural ability to effectively mix entities of different knowledge domains.
It is therefore a normal mental process to enclose spaces by means of walls, doors, windows, etc. from the beginning of architectural design without worrying at first sight about elementary space definitions and checking. That means to abandon instancing Space Class and go to Technology Class. This method has two advantages:
To intimately relate the domains involved in the mental process in order to have a comprehensive vision of problems and opportunities;
To rapidly (and roughly) estimate a bill of quantities, not mere parameter costs from the very beginning of the process: concept or preliminary design phases.
This model also clarifies what a designer does. In a refurbishment project a designer applies a different method: s/he starts from a check of the deterioration status of building components (wall, plaster, steel, reinforced concrete, woods, roof, etc.) upon building codes, then it checks rooms and space distributions to be refurbished, in respect to functional spaces requirements (fig. 2).
A third example of the use of this model is when the designer has to define a ‘metadesign’ project: s/he only works inside the Spatial Class domain, and the result will be a parameter series of functional spaces (f.i., min and max sqm of an elementary space for a clerk with her/his desk and chair, of a standard patient’s room, of a hospital ward for an infectious diseases) (fig. 3).
A fourth example of the use of this model is when the designer has to make a survey of an existing building: s/he only works inside the Technology Class domain, and the result will be document of building status and its spaces (f.i. building dimensioning, building component deterioration, georeference of building and building parts, results of material test, etc.) (fig. 4).
Referring to existing implemented prototypes, the developed ontologies have been tested by means of JessRules Inference Engine applied to Protégé ontologies, combining restrictions verification

276 | eCAADe 30 - Volume 1 - Digital Aids to Design Creativity
embedded into the Ontology Editor itself with external algorithms/rules editor included in JessRules plugin.

In this way it has been simulated the above mentioned design process, mixing entity definition (referred to Building Design) from Spatial and Technology Class domains and analyzing user feedback and computational results.

At present, research is under way to apply different inference engines and development languages to a series of ontologies in the fields of hospitals and offices.

CONCLUSIONS AND PROSPECTS

The paper affords new prospects to deal with two problems of architectural design process:

• How to define a building model that can take into account the complexity of a mental image of a real building (physical digitality);

• How to optimize an architectural design instantiation process able to follow the usual master architects thinking (digital physicality).

The first objective has been tackled by means of a ‘neat’ and sharp subdivision of building model: ontologies of spaces and components as usual, plus a Relation Structure, specific for each actor-designer that relates specific entities of two domains.

The second objective has been solved by mimicking the mental energy saving actor-designer does during the architectural design process s/he explores and defines just the essential entities s/he needs at each design process phase. The possibility to define immediately the essential information at different levels of detail during the work in progress project gives actor-designers a better control of the whole project of the time, so s/he can performs appropriate choices. It is a matter of facts that, as we can see in sketches of modern master architects like Le Corbusier (Carrara and Fioravanti, 2004, fig. 1 and pg. 428), Louis Khan, Zaha Hadid, Steven Hall, Jean Nouvel, etc., they draw at the same time the whole shape and technical details of their masterpieces - they perform ‘concurrent’ design at different levels of abstraction and detail.

The ‘systemic’ building model allows better imitating the mental path actor-designers do and the vision arisen from this study can be seminal for next generation of CAAD tools and methodologies.

REFERENCES


