Implementation of Architecture-oriented Knowledge Framework in MultiCAD Declarative Scene Modelling System

Ioanna Ravani[1,2], Dimitrios Makris[1,2], Georges Miaoulis[2], Petros Constantinides[2], Alexandros Petridis[2], Dimitri Plemenos[1]

[1] Laboratoire Méthodes et Structures Informatiques – MSI
Faculté des Sciences, Université de Limoges, 87060 Limoges cedex, France,
Email: {ravani,makris}@msi.unilim.fr, plemenos@unilim.fr

[2] Department of Compute Science
Technological Education Institute of Athens, 12210 Egaleo, Greece
Email: {jrav,demak,gmiaoul}@teiath.gr, pconst@cs.teiath.gr, apet@mycosmos.gr

Abstract. The development of suitable Computer Aided Design (CAD) environments for the early phase of architectural design is important. A declarative process could capture the nature of the early phase of architectural design. In this work we develop a software application in order to support the declarative process of developing coherent building models based on architectural knowledge. This application is built on MultiCAD software architecture that provides the characteristics of declarative modelling projected on a domain-specific knowledge framework. We use a commercial CAD system in order to apply such architecture and take advantage of CAD functionalities. Buildings sharing the same stylistic-typological principles are chosen to check the effectiveness of the proposed application. We present a new way to describe declarative buildings models through dialogs or Graphical User Interfaces, based on Architectural knowledge.

1 Introduction

An important area of Architecture/Engineer/Construction domain is the development of Computer Aided Design (CAD) environments for the early phase of architectural design. Architectural design knowledge is a challenging information domain to represent. The process deals with the combination of intuition and judgment based on a set of principles and rules, [1,2]. Furthermore, major efforts appeared regarding understanding and description of user requirements, building type, form, dimensions, architectural style, all of which play a definitive role on the final form of the building. Therefore, design encompasses the development of building's form through visual, factual representations and verbal descriptions.

A variety of commercial CAD systems, (AutoCAD^{TM}, MicroStation^{TM}, ArchiCAD^{TM}) while they help the designer in a perceptive manner, has proven too rigid, and lack verbal and even imprecise description of objects. In general, they hardly capture ambiguity, richness, fuzziness of the user requirements [3]. Although they introduce methods of building's conceptual design, the resulted models prevent flexible procedures of decision-making and design refinement.

A number of research systems for generation and evaluation design concepts are developed. OXSYS [4] develop design environments specialized for particular type of buildings. SEED [5, 6, 7, 8] provide computational support either for the whole building design or for the early design stages. Some incorporate their own design applications KAAD [9, 10], or imply expert systems ICADS [11, 12]. EDM [13] is an information model, which represents a building in different levels of abstraction and within many building technologies. OSCONCAD [14] is a model-based interactive system integrated with computer-based applications. Finally, P3 [15] is a computational environment to support design collaboration.

Architectural design can really emerge with the active participation of the designer's mind in a declarative process. The introduction of specific (building) declarative models would enrich the early phase of Architectural Design. Declarative Modelling [16,17] aims towards this direction.

Declarative scene modelling allows the description of scenes to be designed by only giving some expected properties of the scene and letting the modeller find alternative solutions, if any, verifying these properties. In advance, in order to model complex scenes, is used the *declarative modelling by hierarchical decomposition* (*DMHD*) [18,19,20], which is based on top-down designing of scenes.

Declarative scene modelling is based on the *declarative conception cycle* (fig. 1): declarative scene description, generation of solutions and scene understanding.

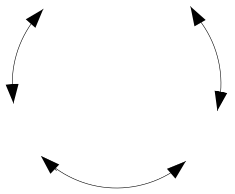


Fig. 1. Declarative conception cycle

The *description phase* comprises the introduction of the designer's mental scene in a high-level of abstraction, by describing the expected properties of the scene. For example, the user can tell the modeller "the scene A must be put on the left of scene B". There exist several possibilities to put a scene on the left of another one. The exact value of the properties of his scene is not known. In the next phase, the modeller *generates alternative scene solutions* by evaluating those properties in order to satisfy the description's constraints. In the *scene-understanding phase*, the modeller proposes to the designer the alternative scene solutions (through visualization tools). As a consequence, the designer tries to decide whether the proposed solution is satisfactory, or not. In this phase, an evaluation process is useful for the selection of the satisfactory scene solutions according to the designer's profile. All the above phases cooperate with each other, in order the designer to feed back-or forward- the output of every phase. This procedure enables the designer for example, to edit his initial description and re-input it to the generator in order to obtain new scene solutions.

Considering the above as the basic concepts, we shall develop a software application to support the declarative process of developing coherent building models based on architectural knowledge. This application is built on MultiCAD architecture that provides the characteristics of declarative modelling projected on a domain-specific knowledge framework. Furthermore, we use a commercial CAD system in order to apply such architecture and take advantage of CAD functionalities.

In section 3, MultiCAD architecture is presented and in section 4 the domain-specific knowledge framework is presented. In section 5 we explain the way MultiCAD architecture is applied to a commercial CAD system while in section 6 we implement three design paradigms on this application. In section 7, conclusions are drawn along with future research topics.

2 MultiCAD Software Architecture

MultiCAD is a software architecture framework for the development of intelligent information systems that support the declarative design process. MultiCAD system is based on *declarative modelling by hierarchical decomposition* (*DMHD*) [20, 21] of scenes. The directions of such a system are defined through a research project supported by the Laboratory MSI (Méthodes et Structures Informatiques) of the Univ. of Limoges along with the Intelligent Information Systems Engineering Research Team of Informatics Department of TEI (Technological Education Institute) of Athens [19].

MultiCAD is a multi-layered architecture that comprises the following main layers [22]:

- The *Interface layer* encompasses functions such as intelligent visualisation of scene models and documents, creation and editing of models and description, formulation of the request (tradition formulations of SQL, spatial SQL or free text search), navigation and browsing of databases acquisition and editing the different types of knowledge and information, application and interaction control.
- The *Process layer* comprises functions such as generation or understanding between the different levels of models, converting the different types of the same level's models.
- The *Information and knowledge management layer* is used for structuring, management, searching and exploitation of the different databases.

These layers are projected to the phases of the declarative conception cycle (fig. 2).

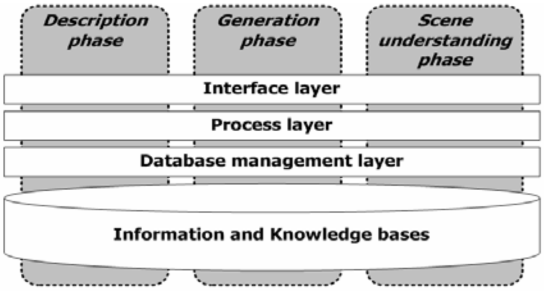

Description phase
Generation phase
Scene understanding phase
Interface layer
Process layer
Database management layer
Information and Knowledge bases

Fig. 2. MultiCAD layered architecture following the declarative conception cycle

2.1 Description Phase

The designer describes in a high-level of abstraction his desired scene by defining the scene's decomposing entities (objects), their properties and relations. The scene's description is represented in the following ways:

– Tree-formed, explicitly representing the hierarchical decomposition of the scene
– Text-formed representation, expressed in a formal Prolog-like language [20], representing the *Internal Declarative Representation – IDR* of the scene.
– Representation in a relational-object database [23]. The scene's description is stored as an assembly of *objects* having *properties*, and being related to each other through *relations*. The storage system is organized in four logical databases: The *Scene database* supports information describing the scene models. The *Project database* manipulates special data for each project. The *Multimedia library* contains all types of project's files, (geometrical models and multimedia information as pictures and videos). In the *Knowledge base* exist all necessary information about entities' types, their properties and relations.

2.2 Generation Phase

The generator results alternative solutions of models that satisfy the constraints specified from the scene's description. Various techniques (Constraint Solving Problem [24], Genetic Algorithms [16], Neural Networks [25]) are used to generate solutions.

2.3 Scene Understanding Phase

The generated scene solutions are *visualized* through their geometrical representation (bitmaps [23], VRML [16], AutoCAD designs [17]). The designer can *evaluate* the solutions according to his criteria either by selecting the best ones or by setting a score to each of them [16].

3 Declarative Knowledge Framework for Architecture-oriented Building Modelling (DKABM)

DKABM could cover all required information to describe consistently and completely any building seen from architectural point of view [17]. The adoption of alternative DKABMs in MultiCAD architecture incorporates the conceptual design of scenes with domain-specific knowledge.

The description of a building is organized in categories of objects with same properties and relations. Information is organized hierarchically forming a knowledge decomposition tree (fig. 3) influenced from the DMHD. All tree nodes are *entities*, considered as categories of objects, having *properties*, *inner* and *inter-relations*: descriptive and structural properties, information that describes the function of the entity, declarative properties, inner-relations, generalization relations, aggregation relations and further associations.

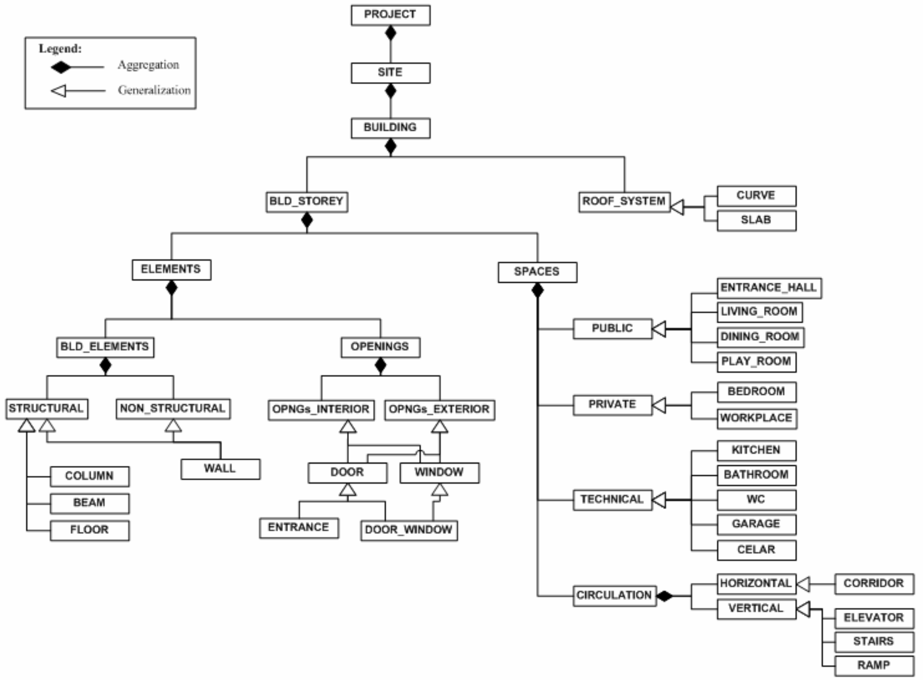


Fig. 3. Entities that participate in DKABM and their aggregation and generalization relations

3.1 Architectural Constraints

For each DKABM, several architectural constraints have to be respected, for a coherent description of a building. Constraints are interpreted to restrictions on the existence or not of specified entities, properties and relations in the building description. If these constraints are not satisfied, the mechanisms correct the building description by generating absent (or re-arranging existent) entities, properties or relations, respectively.

In this way, two categories of constraints (fig. 4) are determined:

- *Obligatory* constraints. Obliged to appear in the user's description of the building and if they do not, the system generates them.
- *Permissive* constraints. Allowed to be applied to each entity in case that the user utilizes it in the description of the building.

The adoption of DKABM produces coherent architectural building descriptions. In advance-in a higher degree of exploration of architectural knowledge-we expressed models that incorporate sets of particular constraints such as *architectural style* (fig. 5) and *building typology* principles.

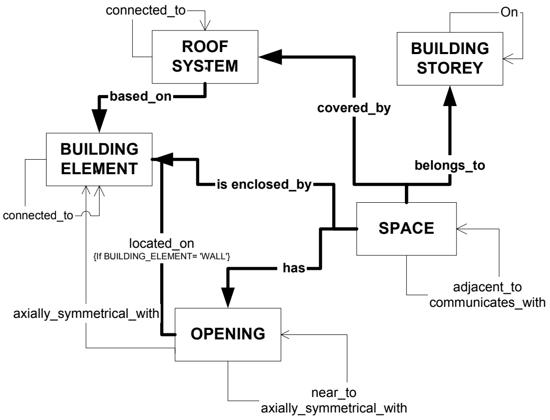
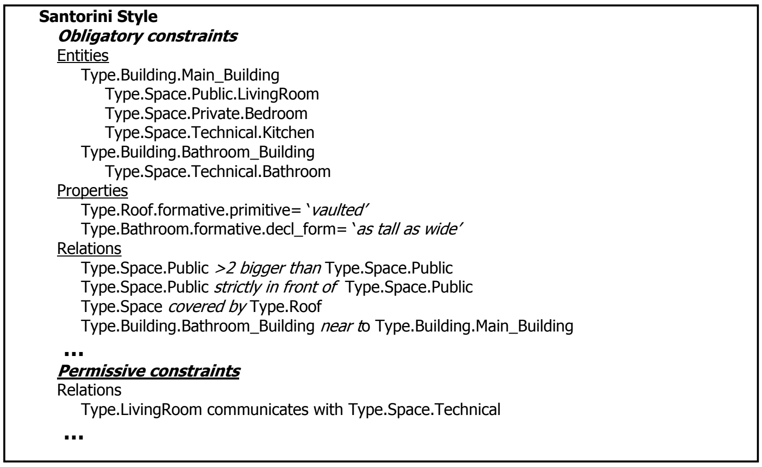


Fig. 4. Obligatory and Permissive relations (respectively, highlighted and simple arrows) between entities in DKABM.



Santorini Style

Obligatory constraints

Entities

Type.Building.Main_Building
Type.Space.Public.LivingRoom
Type.Space.Private.Bedroom
Type.Space.Technical.Kitchen
Type.Building.Bathroom_Building
Type.Space.Technical.Bathroom

Properties

Type.Roof.formative.primitive= '*vaulted'*
Type.Bathroom.formative.decl_form= '*as tall as wide'*

Relations

Type.Space.Public *>2 bigger than* Type.Space.Public
Type.Space.Public *strictly in front of* Type.Space.Public
Type.Space *covered by* Type.Roof
Type.Building.Bathroom_Building *near to* Type.Building.Main_Building

...

Permissive constraints

Relations

Type.LivingRoom communicates with Type.Space.Technical

...

Fig. 5. Specialised constraints for defining the architectural style of Santorini house

As a result, coherent building models are derived, which obey to precise rules and concepts from particular areas of architectural design. These constraints are also categorized as either obligatory or permissive. Furthermore, they define more specialized declarative frameworks to express style and building type.

When the description of a scene is incomplete or contains conflicting parameters, a mechanism is provided to correct this description-by re-arranging the scene decomposition tree and/or by generating instances of objects relying to the missing concepts. Hence, the *Normalized Declarative Building Model* (*NDBM*) of the scene is the new coherent building description enhanced with architectural knowledge through the DKABM.

3.2 Architectural Knowledge in MultiCAD Software Architecture

By incorporating the DKABM within the MultiCAD architecture, the system deals with the concepts and rules of architectural design domain. The Knowledge Base of MultiCAD follows a specific meta-model (fig. 6) with a precise structure of storing and manipulating the type of elements of the Scene Base for different knowledge domains. It contains a series of *types of objects*, *types of properties* and *types of relations* constituting the description of knowledge in MultiCAD with the architectural constraints of DKABM. This means that the obligatory-permissive constraints are stored in the above tables according to the entities they are applied on.

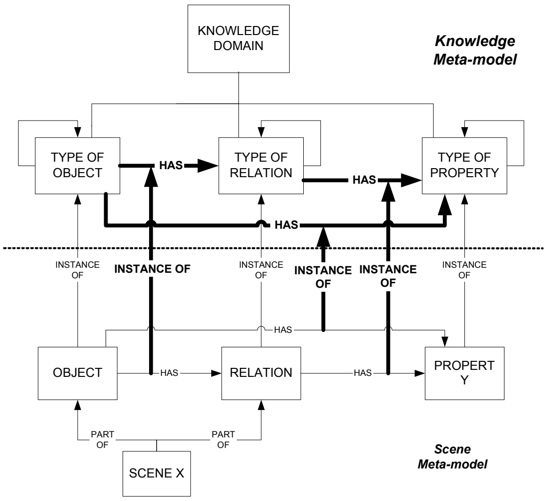


Fig. 6. Scene and Knowledge Meta-model in MultiCAD.

The resulted NDBMs are stored in the Scene Base of MultiCAD, following a meta-model respective to the knowledge meta-model. In this way, elements of the Knowledge base are instantiated in the Scene base. As a result of the above schema, we obtain a flexible environment to manipulate declarative representations of scenes enhanced with architectural concepts and rules through the DKABM. For this reason, we developed an application framework in order to combine the characteristics of MultiCAD architecture with commercial CAD functionalities.

4 Application Framework

We applied the MultiCAD software architecture in a general-purpose commercial CAD system, (AutoCAD 2002). Because MultiCAD is a multi-layered architecture, for each layer, we developed a series of components that enables the linkage within AutoCAD (fig. 7).

We implanted the components to AutoCAD in order to take advantage of its functionalities for description, geometric modelling and visualisation of scene models.

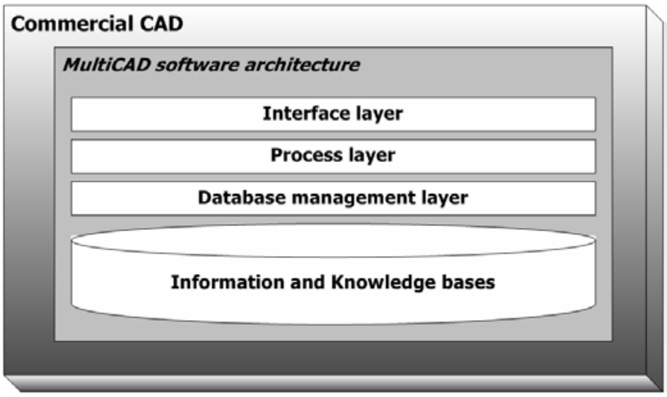

Commercial CAD
MultiCAD software architecture
Interface layer
Process layer
Database management layer
Information and Knowledge bases

Fig. 7. Application of MultiCAD software architecture in a Commercial CAD system

4.1 Components

The sets of developed components appear in Interface, Process and Database management layer and interpenetrate within the three phases of the declarative conception cycle (fig. 1). They are implemented with the use of Visual Basic for Applications supported by AutoCAD.

Interface layer. Its components support the phase of description and the scene understanding on the declarative conception cycle.
Description phase: We generate user interfaces for normalised declarative scene description. We define a new way of declarative description of NDBMs based on dialogue, and knowledge-driven interface components. They enable parametric and declarative description of scenes and they support constraints drawn from the applied DKABM. Furthermore, the designer can use stylistic and building typology knowledge.
Scene understating phase: A series of interfaces are designed in order to visualise the generated scenes in a user-friendly manner in the design environment of AutoCAD. In this way, we take full advantage of advanced CAD functionalities provided by a system such as AutoCAD (great variety of file formats, model editing, etc.).

Process layer. Its components support the first two phases of the declarative conception cycle.
Description phase: Three components are embedded during this phase:

- Component for logical error-handling. It supports the designer with error warnings. These errors prevent descriptions from containing contradictory constraints such as for example Building Area cannot be greater than Site Area.
- Component that ensure that input description satisfies the constraints drawn from the applied DKABM. The input description is corrected to match up with the decomposition hierarchy, the permissive and obligatory constraints of the DKABM by generating respective objects, relations and properties, if needed.

- Component that controls the way that architectural constraints of style and building typology affect the input description. The control follows the DKABM correction process, but now focuses on more specialized constraints.

Generation phase: A component controls the generation of valid scene solutions that verify the input description's constraints (designer constraints, applied DKABM constraints). The scene solutions are modelled geometrically through the comprehensive solid modeller of AutoCAD.

Database management layer. For data management (architectural knowledge and building scene models), we developed components that interact with MultiCAD's database (Knowledge and Scene Base, respectively).

Description phase: two components are developed: the first interacts with MultiCAD's Knowledge Base, where the architectural knowledge is already stored. The second manipulates the Scene Base of MultiCAD in order to store and handle the normalised declarative scene descriptions (NDBMs).

Generation phase: A component that stores in MultiCAD's Scene Base, the scene solutions derived during the generation phase. Solutions are stored in the form of geometric information.

5 Implementation

We apply examples from three design cases to check the effectiveness of the proposed application, two habitations and an office building. Each of the two habitations belongs in different architectural style, Santorini style [26], and Metsovo style [27], both located in Greece. The third design example belongs to a typical office building [28].

5.1 Design Cases

Santorini style. In the Greek island of Santorini, houses have a specific composition of spaces, while their shape is expressed through characteristic geometrical primitive forms (fig. 8a). A typical house, usually, consists of two buildings: (i) the main building containing a living room, a bedroom and the kitchen and (ii) the bathroom which is placed outside of the main building. Among the special characteristics of Santorini's architecture is that every room has either vaulted or flat roof.

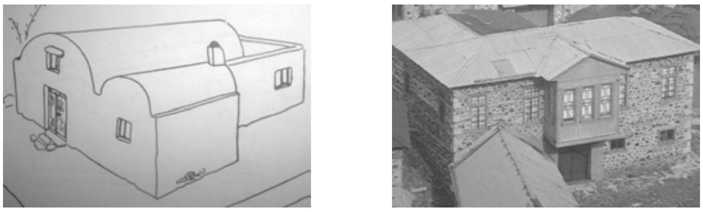


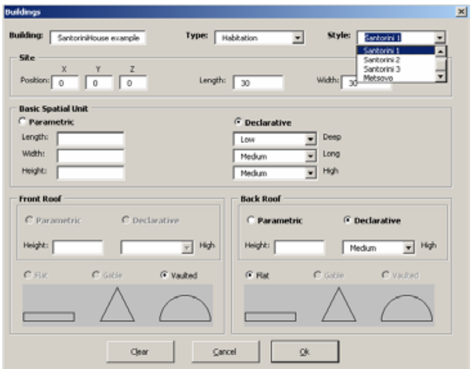
Fig. 8. (a) House of Santorini style and (b) House of Metsovo style

Metsovo style. In the west midland of Greece, (village of Metsovo), houses are built following restrictions of the mountain environment. The regional architectural style introduces a specific typology of spaces, and their shape is expressed through characteristic geometrical primitive forms (fig. 8b). A typical house consists of two buildings: (i) the main two-floor building that satisfies the principal habitation functions: reception area, living room, bedroom and kitchen (upper floor) and storage place with cellar (basement), and (ii) the bathroom which is placed outside of the main building. Among the special characteristics of this architecture is that the upper floor is covered by gable roof while the reception area' roof is placed vertically. An important particularity of this kind of building is that it offers a paradigm of space overlapping: The kitchen and the bedroom share the same space area.

Office building. A type of multi-floor office building is described. It consists of a ground floor containing the reception and vertical circulation areas (elevators, stairs). The rest of floors follow a typical space planning, where office areas, sanitary spaces, horizontal and vertical circulation areas (same as on ground floor) are located.

5.2 Interfaces

The components developed in order to apply MultiCAD software architecture in AutoCAD are underlain behind a series of interfaces that help the designer to follow the declarative conception cycle and in parallel utilize CAD functionalities in a user-friendly manner.



Buildings

Building: SantoriniHouse example
Type: Habitation
Style: Santorini 1

Site
Position: X 0 Y 0 Z 0
Length: 30
Width: 30

Basic Spatial Unit
○ Parametric
Length:
Width:
Height:
● Declarative
Low Deep
Medium Long
Medium High

Front Roof
○ Parametric ○ Declarative
Height:
High
○ Flat ○ Gable ● Vaulted

Back Roof
○ Parametric ● Declarative
Height:
Medium High
● Flat ○ Gable ○ Vaulted

Fig. 9. Interface for the description of normalized scenes

The main interface concerns the parameters of the normalized scene description, (e.g. declarative or parametric). In addition, different style or building type options are provided. Figure 9 shows the main interface, and illustrates an example of user's input description for a building according to a style (among different choices), that of Santorini. The main sections of this interface concern alternatives (e.g. different architectural styles-building types) and scene description parameters. The interface, also, provides the designer with a selection between two different types of description approaches: a *parametric* one, with exact numeric values and a *declarative* one that enables vague expression of design intentions.

In parallel with the designer's input process, the components of coherency are activated to support the description phase:

- Conflicted parameters are controlled.
- Accordingly to the type or style of building chosen, some of the properties as appeared on the interface are locked to default values. For example, in Santorini style, roofs are vaulted, by default.
- The normalized scene description is ensured, by generating the respective entities, properties and relations according to the applied DKABM. Figure 10 shows the decomposition tree of the normalised scene description for the building of Santorini style.

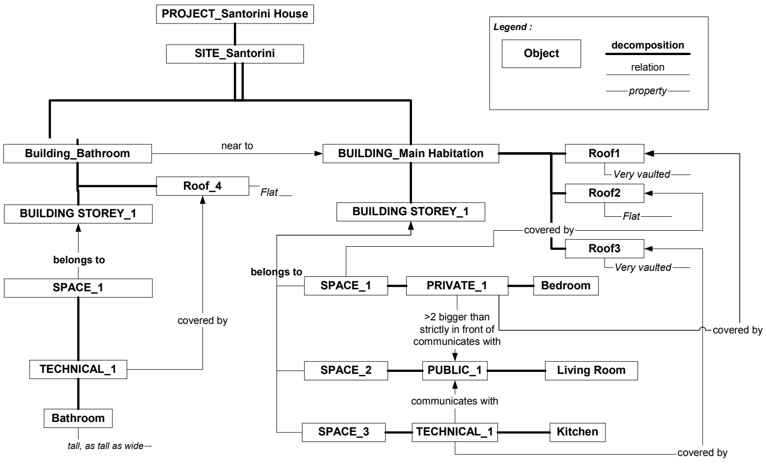
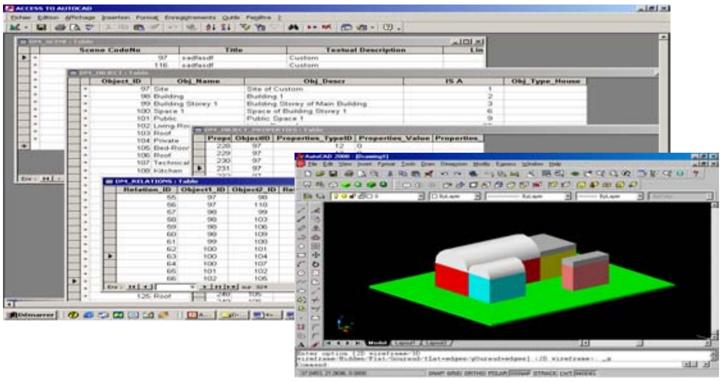


Fig. 10. An example of normalised description of Santorini's style house

The next phase concerns the generation of valid scene solutions. Solutions, (building scenes), are modelled geometrically by AutoCAD's solid modeller, and they are visualised through its design environment. Figure 11 illustrates a valid solution that satisfies the input description of a Santorini style building.


ACCESS TO AUTOCAD
Fichier Edition Affichage Insertion Format Enregistrements Outils Fenêtre ?
DPL_RELATIONS : Table
AutoCAD 2000 - [Drawing1]
File Edit View Insert Format Tools Draw Dimension Modify Express Window Help
Model Layout1 Layout2
SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

Scene CodeNo	Title	Textual Description	Lin
97	sadfasdf	Custom	
116	sadfasdf	Custom	

Object_ID	Obj_Name	Obj_Descr	IS A	Obj_Type_House
97	Site	Site of Custom	1	
98	Building	Building 1	2	
99	Building Storey 1	Building Storey of Main Building	3	
100	Space 1	Space of Building Storey 1	6	
101	Public	Public Space 1	9	
102	Living Ro			
103	Roof			
104	Private			
105	Bed-Roor			
106	Roof			
107	Technical			
108	Kitchen			
126	Roof			

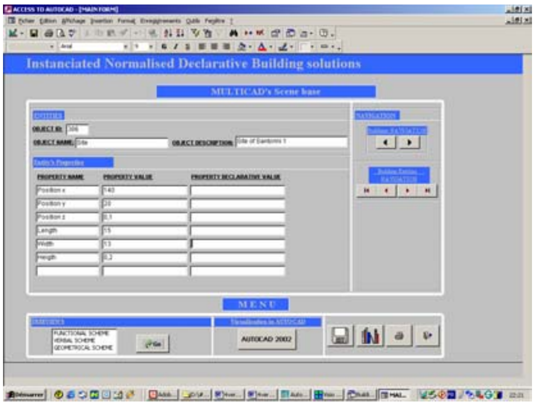
Props	ObjectID	Properties_TypeID	Properties_Value	Properties
228	97	12	0	
229	97			
230	97			
231	97			

Relation_ID	Object1_ID	Object2_ID
55	97	98
56	97	110
57	98	99
58	98	103
59	98	106
60	98	109
61	99	100
62	100	101
63	100	104
64	100	107
65	101	102
66	102	105

Fig. 11. A scene solution of Santorini house and the respective Scene Base records

During the visualisation phase, instantiated-normalised descriptions of the generated scene solutions are stored in the Scene Base of MultiCAD. Figure 11 also presents the respective records of the above Santorini building solution.

In order to manage the Scene Base of MultiCAD, we have developed a series of interfaces handling the respective records of the scene solutions and provide further functions of editing and manipulation of the later. The developed database management components provide the designer with various functionalities. For example, in figure 12, a main interface of data management is shown filled with the records of the entity Site of the above Santorini building solution.


ACCESS TO AUTOCAD - [MAIN FORM]
Fichier Edition Affichage Insertion Format Enregistrements Outils Fenêtre ?
Arial
Instanciated Normalised Declarative Building solutions
MULTICAD's Scene base
ENTITIES
NAVIGATION
Entity's Properties
MENU
FUNCTIONAL SCHEME
VERBAL SCHEME
GEOMETRICAL SCHEME
Go
AUTOCAD 2002
Démarrer

OBJECT ID: 306

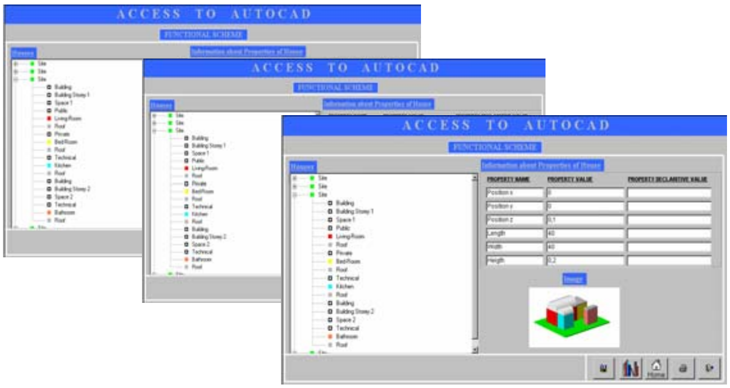
OBJECT NAME: Site

OBJECT DESCRIPTION: Site of Santorini 1

PROPERTY NAME	PROPERTY VALUE	PROPERTY DECLARATIVE VALUE
Position x	140	
Position y	20	
Position z	0,1	
Length	15	
Width	13	
Heigth	0,2	

Fig. 12. Interface for the data management of scenes

In order to facilitate the phase of scene understanding, the building solutions can be represented in the form of three different tree-views. In particular, they incorporate information concerning decomposing entities of the building focusing on the function of spaces, verbal description of the building's decomposing entities and geometric primitives of every space entity. For example, in figure 13, the solution of Santorini building (shown in fig. 11) is represented respectively in the form of the three tree-views.



Site
Site
Site
Building
Building Storey 1
Space 1
Public
Living Room
Roof
Private
Bed Room
Roof
Technical
Kitchen
Roof
Building
Building Storey 2
Space 2
Technical
Bathroom
Roof

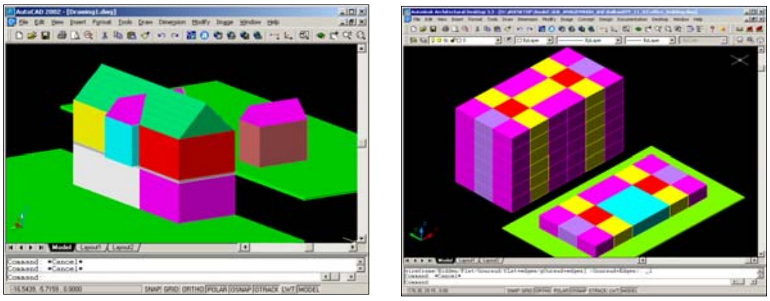
PROPERTY NAME	PROPERTY VALUE	PROPERTY DECLARATIVE VALUE
Position x	0	
Position y	0	
Position z	0,1	
Length	40	
Width	40	
Heigth	0,2	

Fig. 13. Interface of treeview representations of scene solutions

In order to take full advantage of the database management components and in specific the editing tools that they offer, records (building solutions) stored in the Scene Base may be fed back to AutoCAD in order to be modelled and visualised in its design environment. In this way, the designer has the ability without inputting any scene description (in fact, bypassing the description and the generation phase), to choose and edit any description of building solutions already stored in the Scene Base and visualise them in AutoCAD for eventual further manipulation.

Supplementary interfaces provide information about the building entities in a datasheet view. They offer the means to the user to either edit a parameter of building entities and then visualise the respective building to AutoCAD, or directly visualise building solutions without further changes.

We also input the description of a building of Metsovo style and that of an office building. Figure 14 illustrates an example of obtained valid scene solutions.


AutoCAD 2002 - [Drawing1.dwg]
File Edit View Insert Format Tools Draw Dimension Modify Image Window Help
Model
Layout1
Layout2
-16.5435, 5.7159, 0.0000
SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

```
Command: *Cancel*
Command: *Cancel*
Command:
```

Fig. 14. Scene solution of (a) house of Metsovo style and (b) Office building type floors

6 Conclusions

We developed a software application based on a commercial CAD system in order to support the declarative process of developing coherent building models based on architectural knowledge. A main contribution of this work is a new way to describe declarative buildings models through dialogs or Graphical User Interfaces, based on domain-specific knowledge (Architectural). Furthermore we propose a layered architecture-driven development of MultiCAD type software applications, based on a general-purpose commercial CAD and database systems. In specific, the implementation presented in this paper, which is based on AutoCAD 2002 and Access 2000, gives a test-bed for the above approach.

The resulted dialogue-based description of declarative models:

- Facilitates the capture of user requirements in a declarative way during the early phase of design process
- Permits the editing of design parameters and constraints in any phase of the declarative process in an interactive and user-friendly way
- Favours the completeness of normalised building scenes

- Facilitates the development of interface components in order to support this kind of description
- Provides coherency mechanisms according to the applied DKABM, which are enriched with specialised constraints concerning architectural style and building typology.

The proposed layered architecture-driven development of MultiCAD applications enables to take advantage of the existing components of commercial CAD and database systems. It also encourages the development of professional quality and easy extensible MultiCAD applications with rich capabilities of data integration and exchanges. The applications examples show, in three design cases, the easy build of user-friendly domain-specific environments for early phase of design process while proving the effectiveness of the proposed schema.

In our intentions, we include future research in the following domains:

- Further development of domain-specific environments based on declarative knowledge.
- Improvement of the architecture-driven development of MultiCAD applications.
- Improvement of the procedure of the dialogue or graphical user interfaces generation in a semi-automatic manner.

References

1. Lawson B., *Design in Mind*, Butterworth Architecture, London, UK (1994)
2. Rowe P., *Design Thinking*, MIT Press, Cambridge, MA, USA (1991)
3. Simon H.A., *The Science of the Artificial*, MIT Press, Cambridge, MA USA (1996)
4. Eastman C. M., *Building Product Models: Computer Environments Supporting Design and Construction*, CRC Press, Florida, (1999)
5. Flemming U, Woodbury R., Software Environment to Support Early Phases in Building Design: Overview, *Journal of Architectural Engineering*, 4(1):147-152 (1995)
6. Flemming U, Chien S.-F., Schematic Layout Design in SEED Environment, *Journal of Architectural Engineering*, 4(1):162-169 (1995)
7. Rivard H., Fenves S J., A Representation for Conceptual Design of Buildings, *Journal of Computing in Civil Engineering*, 14(3):151-159 (2000)
8. Woodbury R., Chang T.-W. Massing and Enclosure Design with SEED-Config, *Journal of Architectural Engineering*, ASCE 1(4):170-178 (1995)
9. Galle P., Towards Integrated, Intelligent, and Compliant Computer Modelling of Buildings, *Automation in Construction*, 4(3):189-211 (1995)
10. Carrara G., Kalay Y.E., Novembri G., Knowledge - Based Computational Support for Architectural Design, In: G. Carrara and Y.E. Kalay (eds.), *Knowledge-Based Computer-Aided Architectural Design*, (1994)
11. Pohl J., Myers L., A Distributed Cooperative Model for Architectural Design, In G. Carrara & Y.E. Kalay, (eds.), *Knowledge-Based Computer Aided Architectural Design*, (1994)
12. Chun H.-W., Lai E. M.-K., Intelligent Critic System for Architectural Design, *IEEE Transactions of Knowledge and Data Engineering*, 9(4):625-639 (1997)
13. Eastman C. M., Siabiris A., A Generic Building Product Model Incorporating Building Model Information, *Automation in Construction*, 3(1):283-304 (1995)
14. Marir F, Aouad G, Child T, Brandon P, "OSCONCAD: a Model Based CAD System Integrated with Computer-related Construction Applications", *ITcon The Electronic Journal*

(1998) 24-44

15. Kalay Y.E., P3: Computational Environment to Support Design Collaboration, *Automation in Construction*, 8(1):37-48 (1998)
16. Vassilas V., Miaoulis G., Chronopoulos D., Konstantinidis E., Ravani I., Makris D., Plemenos D., MultiCAD-GA: a System for the Design of 3D Forms Based on Genetic Algorithms and Human Evaluation, Springer LNAI Vol.2308, (2002) 203-214
17. Makris D., Ravani I., Miaoulis G., Skourlas C., Fribault P., Plemenos D., Towards a Domain-specific Knowledge Intelligent Information System for Computer-Aided Architectural Design, *Proceedings 3IA International Conference*, Limoges, France (2003)
18. Lucas M., Desmontils E., Declarative Modellers (in French), *Revue Internationale de CFAO et d'Infographie*, 10(6):559-585 (1995)
19. Miaoulis G., Plemenos D., Propositions pour un Système d'Information Multimédia Intelligent Dédié à la CAO – Le projet MultiCAD, Rapport de recherche MSI 96-03, Université de Limoges, France (1996)
20. Plemenos D., Declarative Modeling by Hierarchical Decomposition. The Actual State of the MultiFormes Project, Communication in International Conference GraphiCon'95, St Petersburg, Russia (1995)
21. Plemenos D., Tamine K., Increasing the Efficiency of Declarative Modeling. Constraint evaluation for the Hierarchical Decomposition Approach. *Proceedings International Conference WSCG'97*, Plzen, Czech Republic (1997)
22. Miaoulis G., Plemenos D., Skourlas C., MultiCAD Database: Toward a Unified Data and Knowledge Representation for Database Scene Modeling, *Proceedings 3IA International Conference*, Limoges, France (2000)
23. Miaoulis G., Contribution à l'étude des Systèmes d'Information Multimédia et Intelligent Dédiés à la Conception Déclarative Assistée par l'Ordinateur – Le projet MultiCAD (in French), Professorial dissertation, University of Limoges, France (2002)
24. Bonnefoi P.-F., Plemenos D., Constraint Satisfaction Techniques for Declarative Scene Modelling by Hierarchical Decomposition, *Proceedings 3IA'2000 International Conference*, Limoges, France (2002)
25. Plemenos D., Miaoulis G., Vassilas N., Machine Learning for a General Purpose Declarative Scene Modeler. *Proceedings International Conference GraphiCon'2002*, Nizhny Novgorod, Russia (2002)
26. Michelis P. *Greek Traditional House* (in Greek), EMP Press, Athens, Greece (1981) 41-60
27. Filippidis D. ed., *Greek Traditional Architecture*, (in Greek) Melissa publications Athens Greece (1988) 161-190
28. Laing A., Duffy F., Jaunzens D., Willis S., *New Environments for Working*, BRE and DEGW Publications, London, UK (1998) 26-33