Implementation of an Architectural Design Environment
The development of IDEA+

Stefan BOEYKENS[*], Herman NEUCKERMANS

Abstract: IDEA+ is an integrated design environment for architecture, which is currently being implemented at our department. It requires the development of an architecture-oriented data-structure and a prototype application.
In the research project, the focus lies on transitions between different design phases and scale levels, and at the same time integrating evaluation tools in the design environment.
The paper tries to give an overview of implementation issues and focuses on both programming and architectural topics:
"Representations" – How the project data will be displayed and at the same time, will serve as an interface to the design. How the different views on the data (2D, 3D, textual) will give other insights into the design.
"Transitions" – How the environment will try to follow the workflow of an architect/designer, by allowing a transition of the design into different scale levels and design phases
"Tests" – how the evaluation tests can be integrated in the design environment and at the same time be developed independently.

Keywords: CAAD, representation, transitions, design phase, scale level

1 Introduction
In the transition of the traditional to the digital architectural practice, a lot of attention is spent on the construction phase: generating construction documents (plans, sections) but also calculations/simulations or administrative documents. To generate these documents, we need detailed data and often complex tools. Typically, these "calculations" are done on a design that is more or less finalised. The major changes have been elaborated on a moment when only limited design info was available.
Early-design decisions could have a more profound impact on the output and also require a smaller amount of work, but there is a lack of design and evaluation tools that address the early-design phase. Our research efforts are targeted particularly at this phase. The development of IDEA+ is a part of this research.

1.1 What is IDEA+?
IDEA+ is an "Integrated Design Environment for Architecture". It handles both design and evaluation, is targeted at the architect/designer and proposes a computer

[*] Dept. of Architecture, K.U.Leuven, Belgium. stefan.boeykens@asro.kuleuven.ac.be

application for CAAD. It is not a general-purpose geometric modeller or drafting application. The “plus” stands for the added functionality of the design environment, by integrating evaluation tests into the design process.

1.2 Past research

1.2.1 The Conceptual Model [NEU 1992]

The conceptual foundation for this environment has been laid out in a scheme with different Design Phases (Sketch Design, Preliminary Design and Construction Design) and Scale Levels (Masterplan Level, Block Level and Space Level) together with appropriate tests at different levels.

1.2.2 The CORE Object Model [HEN 2000]

The Conceptual Model has been translated into a theoretical framework: the CORE Object Model. This set out the basic class structure and all internal relationships.
The development of the CORE Object Model made use of MERODE [DS 1994]. This software-development method helped defining the basic class structure and relationship-diagrams.

1.3 Current research project

The research project will finalise the CORE object model, define how transitions between design phases and scale levels can be handled [BOE et al 2002, BN 2003] and also elaborate one test case: daylighting [GN 2001, GEE 2003].

2 The data structure

Without going into detail about technical aspects of the data structure, it is important to remark the strict separation between CAAD entities (that contain architectural, conceptual meaning) and graphical entities. The design information is contained in the CAAD entities and is not a mere attribute of some graphical object.

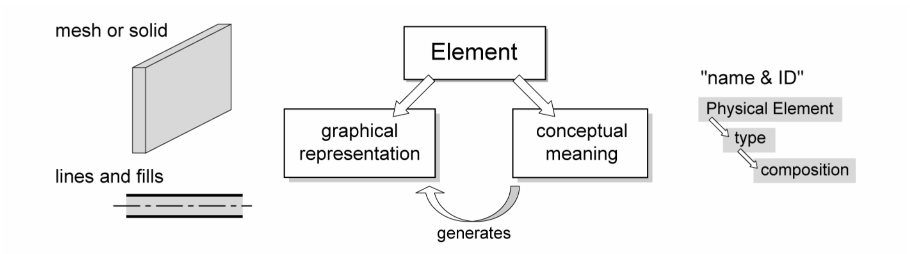

Figure 1. Separation between graphical and CAAD entities

The data structure is an object-oriented set of C++ classes and is meant to be platform-independent. The prototype application is a Windows-only application and uses the Microsoft Foundation Classes (MFC) and OpenGL.

3 Representations

Representations provide an interface to the project data. The only time a user sees the project data, is when a representation of that data is generated and, consequently, every change of the data will be executed from within a certain representation.
The CORE Object Model provides a representation-link, which connects a graphical entity to the CAAD Entity from which it depends. A representation is a collection of those links.
Current representations include a 2D-view (a plan-view), a 3D-view and a treelike hierarchical view. Other possible representations can be more text-oriented (e.g. a listing) or more conceptually oriented (a schematic view, where relations between elements are displayed).

4 Transitions between design phases & scale levels

An architectural project steps through *Design Phases*. The chosen phases are based on common architectural praxis (e.g. RIBA Plan of Work Stages[1] or the activity domain of the Belgian Board of Architects[2]) and are elaborated in [NEU 1992].

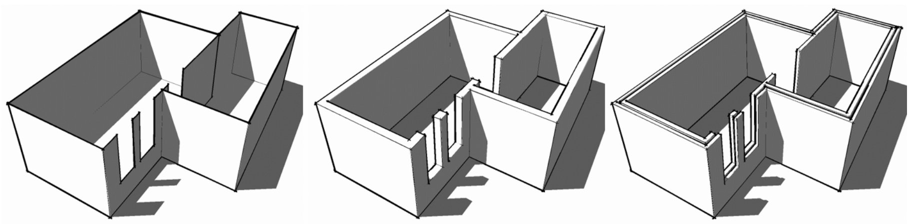

Figure 2. Design Phases: from Sketch to Preliminary to Construction Design

A Phase defines the amount of detail that is available. The Programming Phase (textual overview of project requirements) is not considered in the IDEA-project. The *Sketch Design* is when designer and client interact and the main design layout is decided upon. The *Preliminary Design* prepares the building permit (graphical and textual documents for the authorities). The *Construction Design* is where construction documents are prepared for the contractor (both graphical & textual). The design phases usually follow the same order (adding details to the Elements).
The *Scale Level* defines the scope on which the model is investigated: the *Masterplan Level* defines the main site, the main building envelope(s) and common information on structural concept and equipment. The *Block Level* defines the separation of the project in stories and/or rooms. The *Space Level* is where

---

[1] **e.g. http://www.riba.org/go/RIBA/Member/Practice_306.html**

[2] **e.g. http://www.ordevanarchitecten.be/nl/arch/activiteitsdomein.htm#top (in Dutch) or http://www.ordredesarchitectes.be/fr/arch/domaines_architecte.htm#top (in French)**

individual building elements are investigated (all physical elements, like walls, floors, roofs etc).

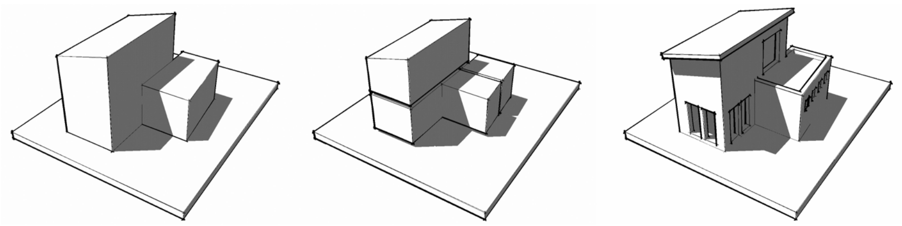

Figure 3. Scale Levels: from Masterplan to Block to Space Level

To allow the digital model to follow this workflow, we introduce *transitions*: going from one design phase to another or going from one scale level to another. We see transitions as a process of gradually enriching the project with additional information, by stepping back and forth between the different design phases and scale levels. It is certainly not a unidirectional workflow.

4.1 Transitions, Representations and CAAD Entities

Because the current scale level and design phase are properties of a certain representation (view), it is obvious that the transition has to update the current representation. This leads to certain elements not being displayed anymore and other (that were created during the transition) to be generated: graphical entities are created and linked to the CAAD Entity to which they relate.

When we switch between design phases, the composition of elements can change. When doing the reverse transition, however, no real changes can happen, since we do not want to delete any of the information we added with the previous transition. It is desirable to do the reverse transition from time to time, to evaluate lower level design decisions on a higher level. The difficulty lies in switching back and forth: when applying changes on a higher level, that affect the lower levels and vice versa. A strategy for this reverse transition was described in [BOE et al 2002].

5 Tests

To allow evaluation to be integrated in the design environment, we provide a generic structure for tests. They can be an integrated “module” or an external application, but the access to the project is defined in a pre-defined and abstract way in the data structure.

5.1 Available tests

The test for daylighting has been developed in our research group [GEE 2003] and some necessary connections have already been tried out based on collaborations with another research team. The daylighting test is the first one to be integrated into the IDEA+ application framework.

A prototype for a cost-estimating test, using the different design phases and scale levels has also been elaborated in a final year dissertation [WIT 2002]. The flow of cost calculations between different levels has been implemented as a Visual Basic module that operates inside Autodesk Architectural Desktop, but based on the conceptual model of IDEA+. We hope to integrate this testing mechanism into the main data structure and application, after finishing the object model and the generic structure for the testing applications.

5.2 Using tests in a "Design Environment"

The main reason for integrating tests in the Design Environment is getting feedback as early as possible. At the moment some basic layout is defined (e.g. a simple Masterplan sketch, where the site and a few building blocks are placed) we can already evaluate the design. E.g. when we think about a cost-estimation test, we can imagine having some library of mean prices per square meter for common building types. They can be derived from older projects but also by simulating one project in the design environment and calculating the average pricing for it.

5.3 The interaction of tests with the data-structure

Tests can query the project to generate output reports (cost, volume/area, heat loss, etc…); they can alter properties of elements (calculated area, calculated cost/m²) or store new data which is not yet available in the main data-structure.
Because we can't know at implementation time the data a test could possibly need, we allow the test to define a new dataset (a combination of parameters and values) and add it to CAAD Entities. This is a better solution than having a complete map of all objects with associated data stored by the test application. When an object is deleted, this data becomes obsolete and will be destroyed as well. When the scene is stored or loaded, the additional data is directly stored with the elements it belongs to.

6 Conclusions

The IDEA-application is partly functional: the user-interface and the main tools are decided upon and some of the main elements can be created and interacted with.

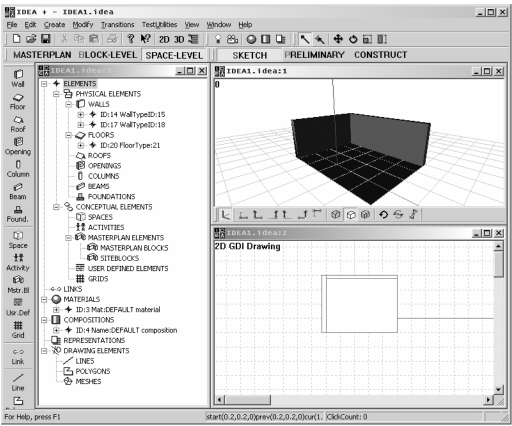

Figure 4. Screenshot of the Prototype, showing different representations.

A few representations serve as views to the project data (a 2D-plan view, a 3D-view and a treelike hierarchical view). The transitions and main testing interface still have to be elaborated. The daylighting test has been finalised but is not yet integrated into the design environment.
When IDEA+ is usable to do basic design studies and perform evaluation tests, we will evaluate it in design experiments (with students and/or chosen architects).
IDEA+ is a "work-in-progress" and still needs further research. It has never been the intention of developing a commercial-class software application. This is not possible in the current timeframe and with current staff limitations. The prototype application is a way to show the potential of such a design environment.

7 Acknowledgement

This research project has been funded by the Catholic University of Leuven (Belgium) with reference "Research Fund, K.U.Leuven (OT/00/18)".

8 Bibliography

[BN 2003] Boeykens, S., Neuckermans, H., 2003. Implementation Strategy for an Architectural Design Environment, Issues in the development of IDEA+, *Proceedings of the 3rd International Postgraduate Research Conference in the Built and Human Environment*, Lisbon (Portugal), pp761-771

[BOE et al 2002] Boeykens, S., Geebelen, B., Neuckermans, H., 2002. Design phase transitions in object-oriented modeling of architecture, *Proceedings of the 20th eCAADe Conference*, Warsaw (Poland), pp310-313

[DS 1994] Dedene, G., Snoeck, M., 1994. M.E.R.O.D.E.: A Model-Driven Entity-Relationship Object-oriented DEvelopment method, *ACM SIGSOFT Software Engineering Notes*, vol 19/3.

[GEE 2003] Geebelen, B., 2003. Daylight Availability Prediction in the Early Stages of the Building Design Process, *PhD thesis, K.U.Leuven*, Leuven (Belgium)

[GN 2001] Geebelen, B., Neuckermans, H., 2001. Natural-Lighting Design in Architecture, Filling in the Blanks, *Proceedings of the 7th International IBPSA Conference*, Rio de Janeiro (Brazil), 8 pages

[HEN 2000] Hendricx, A., 2000. A Core Object Model for Architectural Design, *PhD thesis, K.U.Leuven*, Leuven (Belgium)

[NEU 1992] Neuckermans, H., 1992. A conceptual model for CAAD, *Automation in construction*, 1(1), pp1-6

[WIT 2002] Withouck, P., 2002. Toetsen van een gebouwmodel: kostprijs-berekening in de verschillende ontwerpstadia, *Final year dissertation K.U.Leuven*, Leuven (Belgium) – only available in Dutch