An Integrated Software Environment for the Architectural Design Process

Mathias Schwenck & Sevil Sariyildiz
Delft University of Technology, The Netherlands

Faculty of Architecture
Department Building Technology
Chair "Technical Design & Computer Science"
Berlageweg 1
NL-2628 CR Delft, The Netherlands
E-mail: {i.s.sariyildiz / m.schwenck}@bk.tudelft.nl

Abstract

Many software systems are in common use in the field of architectural design. On the other hand, we consider a complete automation of architectural design as an unlikely proposition and undesirable for the architect. Therefore, the general objective is to support the designer during the whole process of architectural design in order to increase the efficiency and to improve the quality of the results. So far there are different tools providing such functionality. Nevertheless, there are no appropriate tools for many of the sub-processes. Furthermore, the current state of available design software is characterised by a lack of integration of different tools.

In this paper we will provide a survey on a project dealing with the solution of both problems. First we will give a general description of the support that software can provide to architects during the design process. We conclude that many different tools are needed which have to be integrated in an open, modular, distributed, user friendly and efficient environment. We will explain the necessity of integration and cover integration technologies.

Besides the aspect of integration we also deal with the development of tools which can operate in the integrated design environment. We suggest a strategy where the tool functions are specified on the basis of a transformation from hierarchical process descriptions of architectural design into a hierarchy of tool descriptions.

Key Words:
Computer Aided Architectural Design, Design Support Systems, Integrated Software Environments, Architectural Design, Process Modelling

Introduction

> *"Architecture is a science which is a mixture of an exact science and the art. The combination of these two important items makes architecture a difficult task. ... An architect has to combine these both primary elements in the design and in the same time while expressing the feeling of art, must take very good care of many other factors which play an important role in the building and design environment. The technical aspects on one hand, the social aspects on the other hand."* [Sari91]

Architectural design is a very complex process combining different types of requirements. Recent developments in fields like building technology or material science have increased the possibilities from which the architect may choose. Furthermore, current construction projects tend to become more and more complex. As a consequence the question arises how architects and others involved in the architectural design process can be supported in the handling of this complexity.

In this paper we will discuss the possibilities that can be provided by means of software systems. We are not considering other possibilities to support the process, e.g. advantages of the application of new design methodologies in the field of architecture are discussed in [ScVö96].

The Architectural Design Process

Generally, we consider architectural design as a process consisting of two main phases - conceptual design and materialisation. This classification is only a logical distinction. It does not imply a linear time sequence or a strong separation.

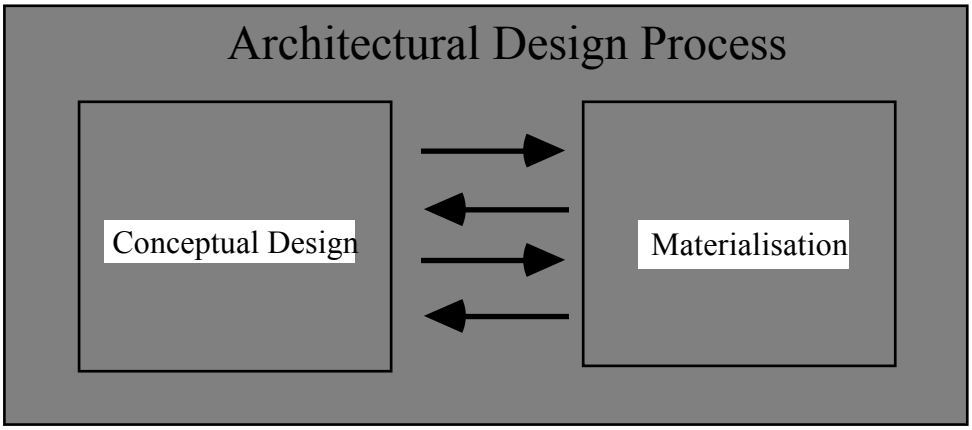

Figure 1: The Architectural Design Process

Conceptual design includes a global design of forms and relations between different parts of the object to be designed. In the materialisation step the forms are substituted by construction components and materials.
Once a concept has been determined, it must be materialised. That means that the geometrical model has to be transformed into components and materials. A large variety of tasks has to be executed during the materialisation phase. It includes dimensioning processes, determining the construction, materials and building details, considering building provisions, etc.

Software in the Field of Architectural Design

Many software systems are in common use in the field of architectural design. Nevertheless, there are no appropriate tools for many of the sub-processes. Furthermore, the current state of available design software is characterised by a lack of integration of different tools.

Both features are significant limitations to the level of support that software can provide. The lack of integration makes it impossible to implement efficient functions for design and data management. Furthermore, generic functions e.g. for permanent data storage have to be realised in each tool. Conversion programs are needed to exchange data. More generally, the designer has to deal more often with routine tasks like data management and data exchange. As a consequence he has significantly less time available for his creative design work.

Integrated design systems can be used to overcome this “bottleneck”. On the other hand, the implementation of these facilities is a complex process requiring significant resources. Furthermore, the introduction of added functionality may decrease the performance of an individual tool. [Wolf93] gives indications that these disadvantages can be compensated and that the overall efficiency will be significantly increased.

Requirements for a Support System for Architectural Design

We have already stated that the main characteristic of the system is to give support during the whole design process. For every sub-process of architectural design where the use of suited software can lead to improvements, these tools has to be available.

“Support” means, that the tools should provide functions to free the architect of routine tasks, to avoid faulty actions and to detect errors as early as possible, to support the architect by

increasing the amount of available information, to support the exchange of infomation between different partners participating in the building process, etc.
Based on these system characteristics we can specify the functional requirements for the support system. They includes features like:

- facilities for the persistent storing of design data
- support multiple users performing multiple design tasks concurrently
- guarantee consistency and integrity of design information
- access control
- support evolutionary design (version management)
- support view management
- design flow management
- facilitate tool integration.

All issues discussed very briefly in this section are described in greater detail in [SaSc96], whereas an example for the detailing phase of architectural design is given in [SaSc96a].

CAD frameworks as described for example in [Abel94] or [Wolf93] realise integrated software environments providing the functionality described above.

Integrated Software Environments

In [ScBr93] the term "integration" is described by distinguishing integration with respect to the following three dimensions:

- Data (Information)
- Control (Communication)
- User Interface (Presentation).

The *data integration* aspect of tools determines the degree to which data generated by one tool is made accessible and is understood by other tools.

The *control integration* aspect of a tool determines its communicational ability, i.e. the degree to which it communicates its findings and actions to other tools and the degree to which it provides means to other tools to communicate with it.

The *user interface integration* aspect is the degree to which different tools present a similar external look-and-feel and behave in a similar way in similar situations.

Integration has to be realised in all three dimensions. This avoids situations where limitations occur because of incompatible file formats, incompatible communication protocols or because of user interfaces that are not suited for the people working in the field of architectural design.

Instead of only developing design tools integrated system are also addressing the problem of the operating environment of these tools.

Figure 2 on the next page gives a schematic view of an integrated environment as used in [Wolf93].

The framework provides general services for the tools. It realises functions for data management in order to organises design descriptions and to provide access as well as design management functionality guiding the designer through the design process.

The interactions between tools and framework take place according to functions of the Tool-Framework Interface. The definition of the interface is a key issue in tool integration because the effectiveness of tool integration depends on it.

Within the integrated environment different tools can be used. Tools that are newly developed can be implemented according to the facilities of the framework, whereas existing software may be integrated using known tool coupling methods.

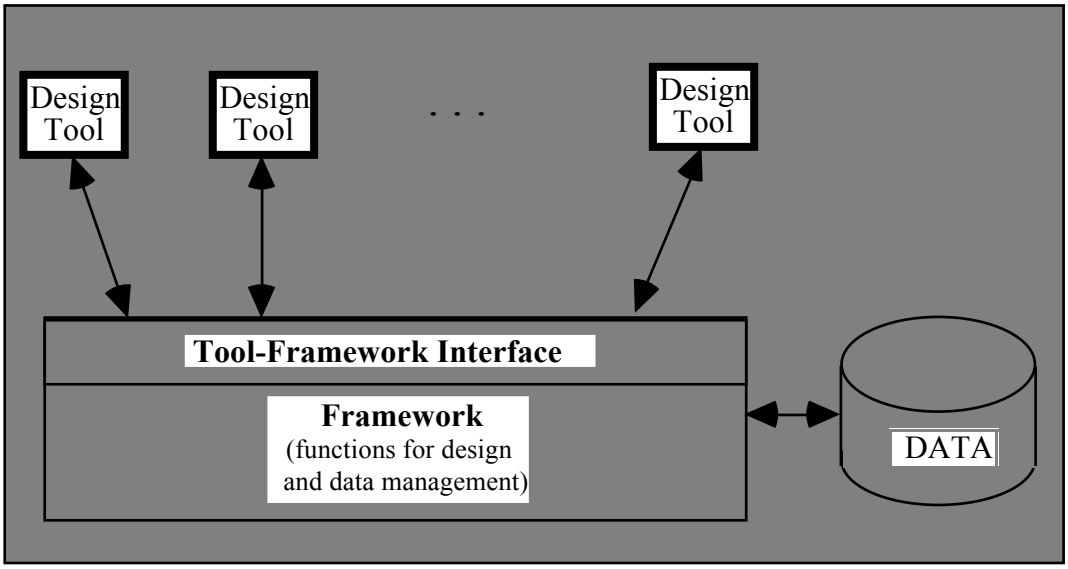

Figure 2: An integrated software environment

Generally, we can state that the implementation of a design framework is possible on the basis of known information technology methods and the experience from comparable developments in other design disciplines.

Process Descriptions and Tool Specifications

In the second part we will deal with the question how the functionality of design tools can be specified more precisely in order to optimise these tools corresponding to the needs of the user and to realise the demand to have appropriate tools for every sub-process of architectural design.

Why is this necessary? In our opinion the shortcomings of current software used in architectural design are partly caused by the fact that the demands of the process are not explicitly specified. Specifications are incomplete, inaccurate or not suited for software development processes. Furthermore, many demands are hidden, i.e. “secret wishes” of architects that are not represented by software specifications and consequently not realised in software tools.

To reach our goal to have tools available for all sub-processes of architectural design we need an explicit, comprehensive and correct specification, that represents all details needed. In order to get such a specification of tool functions we suggest the following strategy:

- First we need a model of architectural design describing all sub-processes and data involved. The generation of the process model is an iterative process that has to be executed step by step. It starts with a very general description of architectural design, where the process is decomposed into several sub-processes. In further refinement steps this decomposition process is continued step by step, i.e. the sub-processes are decomposed into new sub-proceses

representing a lower level. At the end, the architectural design process is described by a hierarchy of sub-processes.

- If the process description reaches a certain level of detail, it becomes possible to map this hierarchical process model into specifications of different design tools realising appropriate support for all sub-processes.

The fundamental advantage of this strategy is the inherent characteristic, that the tool specification corresponds to the process description.

If a suited level of detail is reached, the transformation will be relatively easy, because most processes will be directly mapped into a design tool. Nevertheless, there may be cases, where:

- A sub-process will be mapped to more than one design tool. Generally these tools have a very specific functionality. An alternative solution is the introducion of a further refinement level, but this decomposition should be avoided in cases where it is not related to the process and only requested to simplify the transformation.

- Several sub-processes may be mapped to the same tool, if their requirement are very similar. This tool will mostly provide a more general functionality.

A simple example of such a transformation process is shown in figure 3:

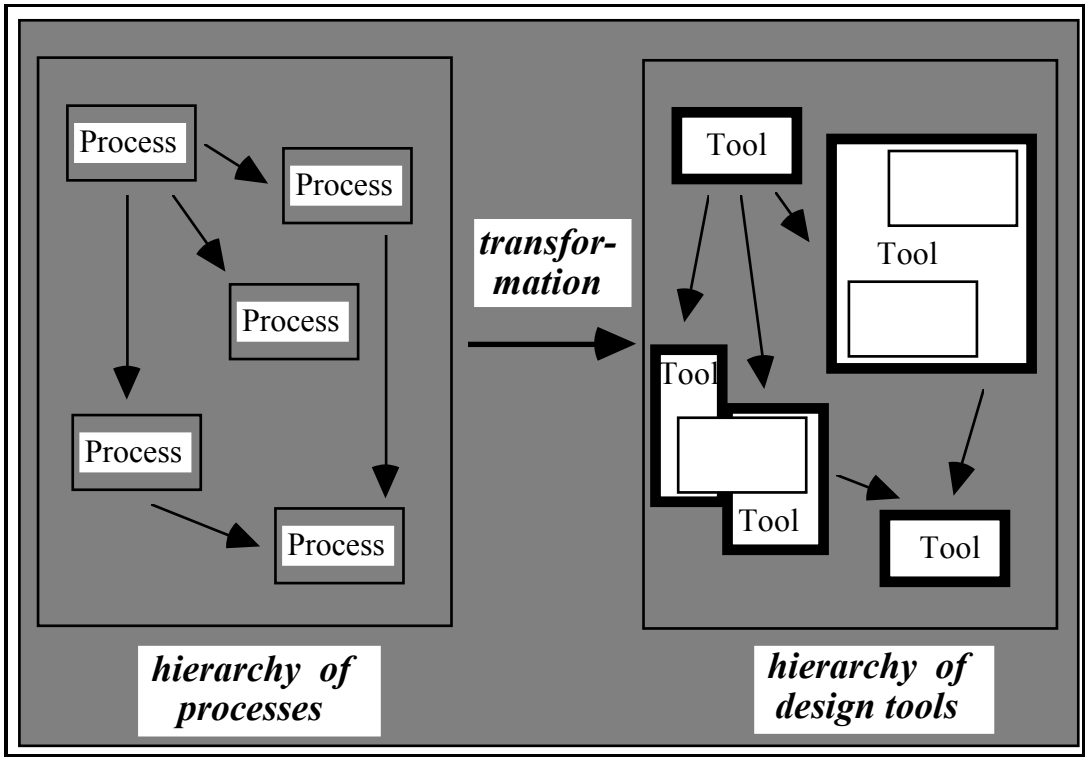

Figure 3: The transformation from process descriptions to tool specifications

Process models for architectural design, that are available in literature sources, could not be used as basis for this mapping because they do not meet the demands as described before. A new process decription has to be developed. In [ScVö96] first steps directed to generate such a general process model are presented. A lot of effort has to be put into the modelling, but using this suggestion will provide a description representing all demands including features that are hidden so far.

In our opinion the existence of such a process model can not be doubted. Although there may be no design processes that have been executed identically, there are many common features. The task is to extract these features and to combine them to a process model that is as abstract as necessary to be applicable in general and specific enough to represent a certain practical architectural design project.

Besides the use as basis for the specification of tools in an integrated design environment such a description is useful for other tasks. Within the context of this development it can also be used to implement design management function as a part of the framework.

Additionally, it can be used as basis for other research because the process model gives a better understanding of architectural design. Taking this into account it may be useful in many other areas including education purposes.

Summary, Conclusions and Future Work

The general motivation of this research is to develop software to support the handling of the increased complexity of architectural design. We have compiled a description of such a system providing general support during the whole process.

Instead of only developing design tools integrated system are also addressing the problem of the operating environment of these tools. They have to be integrated in an open, modular, distributed, user friendly and efficient environment.

Two major fields have to be addressed - the development of design tools and the realisation of an integrated system as their operation environment.

We have very briefly focused on the latter by discussing known technologies in the field of information technology and other design disciplines that can be used to realise such an environment. Regarding the first subject we have stated the need of a detailed specification of tool function. As a solution we have approached the following strategy:

- First we need a model of architectural design describing all sub-processes and data involved. This description has to be refined step by step.

- If the process description reaches a certain level of detail, we can start with a kind of a transformation, where this hierarchical process model can be mapped into specifications of different design tools realising appropriate support for all these sub-processes.

Using this strategy the main steps to develop such a support system are:

(1) implementation of a framework as basis for the integrated design system

(2) decision whether the tool specification are already implemented in available tools

(3) In this case these tools can be integrated using known methods for tool coupling.

(4) Otherwise new design tools have to be developed according to the framework.

References

[Abel94] Abel E., “Standards in CAD Frameworks”
Proceedings of the conference “Database Systems and Applications for the 90’s”, Delft, The Netherlands, 1994

[Sari91] Sariyildiz S., “Conceptual Design By Means Of Islamic-Geometric-Patterns Within A CAAD-Environment”
PhD-thesis, TU Delft, Faculty of Architecture, 1991

[SaSc96] Sariyildiz S., Schwenck M., “Integrated Support Systems for Architectural Design”
Proceedings of the 3rd Conference on “Design and Decision Support Systems in Architecture and Urban Planning”, Spa, Belgium, August 1996

[SaSc96a] Sariyildiz S., Schwenck M., “Design and its Materialisation till Detailing within Information Technology”
Proceedings of the International Conference on “Detail Design in Architecture”, Northampton, United Kingdom, September 1996

[ScBr93] Schefström D., Broek G. van den, “Tool Integration - Environments and Frameworks”
John Wiley & Sons, Chichester, 1993

[ScVö96] Schwenck M., Völker H., “Modelling the Architectural Design Process - Analysis of Traditional Methodology and Approaches to the Future”
Report, TU Delft, Faculty of Architecture, 1996

[Wolf93] Wolf P. van der, “Architecture of an Open and Efficient CAD Framework”
PhD-thesis, TU Delft, Faculty of Electrical Engineering, 1993