INTEGRATION OF BUILDING DESIGN TOOLS IN DUTCH PRACTICE

Wim Plokker & Luc L. Soethout
TNO Building and Construction Research
Department of Indoor Environment, Building Physics and Systems
P.O. Box 49
NL-2600 AA Delft
The Netherlands

ABSTRACT

Since two years, the Dutch building consultancy practice has been supported by an integrated design environment to base its advices on. This environment, called the Uniform Environment or UO in Dutch abbreviated form, has been developed by the Association for Computerisation in the Building and Installation Technology (VABI) and TNO Building and Construction Research.

The basic principle of the UO is that all data, associated with a building project, is stored in one database, irrespective of the design tools being used in the project. In this way, design tools can share information which is commonly needed by several tools. Each design tool has its own input forms, which have a strong resemblance with the input forms of the other tools. Known information from the input or the calculation of another tool is automatically supplied; the user only needs to input information, specific to the design tool.

Currently, several design tools have been incorporated into the UO, mainly in the area of HVAC, lighting and acoustics. Future improvements of the UO include, beside new design tools, a coupling to tools for cost estimation and specification generation. The integration of the tools in the UO will be improved by structuring the output of the UO as well as is currently the input. In the end, the user will be able to work problem-oriented instead of application-oriented.

INTRODUCTION

Since two years, the Dutch building consultancy practice has been supported by an integrated design environment to base its advices on. This environment, called the Uniform Environment or UO in Dutch abbreviated form, has been developed by the Association for Computerisation in the Building and Installation Technology (VABI) and TNO Building and Construction Research.

The need for the development of an integrated approach to building design arose from different sides. Designers, working with several technical calculation programmes, had to input for one project the same data over and over again, like the dimensions of a room. Incorporation of product data supplied by manufactures was difficult because of the large discrepancy in the format of the several product databases. Due to the amount of participants in the building process, communication is very important. Information technology can support this, by providing a consistent link between the software programmes of the participants.

The UO is the first serious attempt to deliver a useful integrated software platform for technical computation to the building industry. Started with emphasis on the building design, in particular HVAC design with relations to energy, indoor climate and comfort, the accent is now shifting to the building process as a whole, by incorporation of costing tools and tools for specification generation.

This paper intends to give a description of the current status of the UO, as well as the projects that are currently undertaken to develop the UO further. More applications will be added to the UO, as the UO is extended towards other faces in the building process. Besides a growth in breadth, a grow in depth is foreseen, where the UO is enriched by project management tools like a scenario manager. This tool will guide the user through the building design and beyond, thereby guarding the integrity of the project data and monitoring the overall process.

CURRENT STATUS OF THE UO

The UO offers the user an application framework for technical calculations related to the building industry. Several software applications are integrated into one design environment. Currently, the main emphasis is on the design of HVAC installations, the calculation of the energy performance of buildings and the prediction of the indoor climate in relation to comfort and energy use.

The key feature of the UO is the central database, in which all information associated with a building project is stored, irrespective of the design tools used in the project. All design tools inside the UO communicate with the central database in order to add project data to the database or to retrieve information from the database. In this way, design tools can share information, needed by more than one application. A good example of such information is topological and geometrical data. Most of the design tools need in one way or another the shape, structure and dimensions of a building or of the rooms inside a building. The user has to supply this information only once. Thereafter, all applications can make use of it, either directly, or after transformation. Another example of information sharing is when an application needs the outcome of another application as an input item. For instance, a design tool for dimensioning the radiators to be placed in a room can be supplied with the heat loss of the room, as supplied by a heat transmission programme.

The structure of the common database of the UO has been determined through a bottom-up approach. The information need of several existing design tools has been investigated, after which their common parts have been transformed into a product model. Additional information of the applications has been added to the core model in a consistent way. The advantage of this approach is that the central model of the UO stays close to the internal models of the applications that use the UO, and therefore guarantee the usefulness of the UO as a whole. A top-down approach, where a building core model is the starting point, probably results in a better structured model, but might not be well applicable in practical applications.

The central database is filled through the interface of the individual applications. Each design tool has its own input forms, which accommodate for the complete input required by the tool. Known information from the input or the calculation of another tool is automatically supplied; the user only needs to input information, specific to the design tool. Here, the advantage of a common project database is obvious: the user needs to enter information only once, even if more applications need this information. This has its impact on project management in terms of time (shorter production time) and of quality (less input errors and inconsistencies).

All VABI tools in the UO have a similar user interface, which can be constructed from a common library. This guarantees the same feel and touch for these applications. Such a consistent user interface lowers the learning threshold for users. This library might be available in the future for third-party developers as well.

Existing design tools with their own user interface are not required to adapt their interface to the UO. They only need to be hooked up to the UO on the data

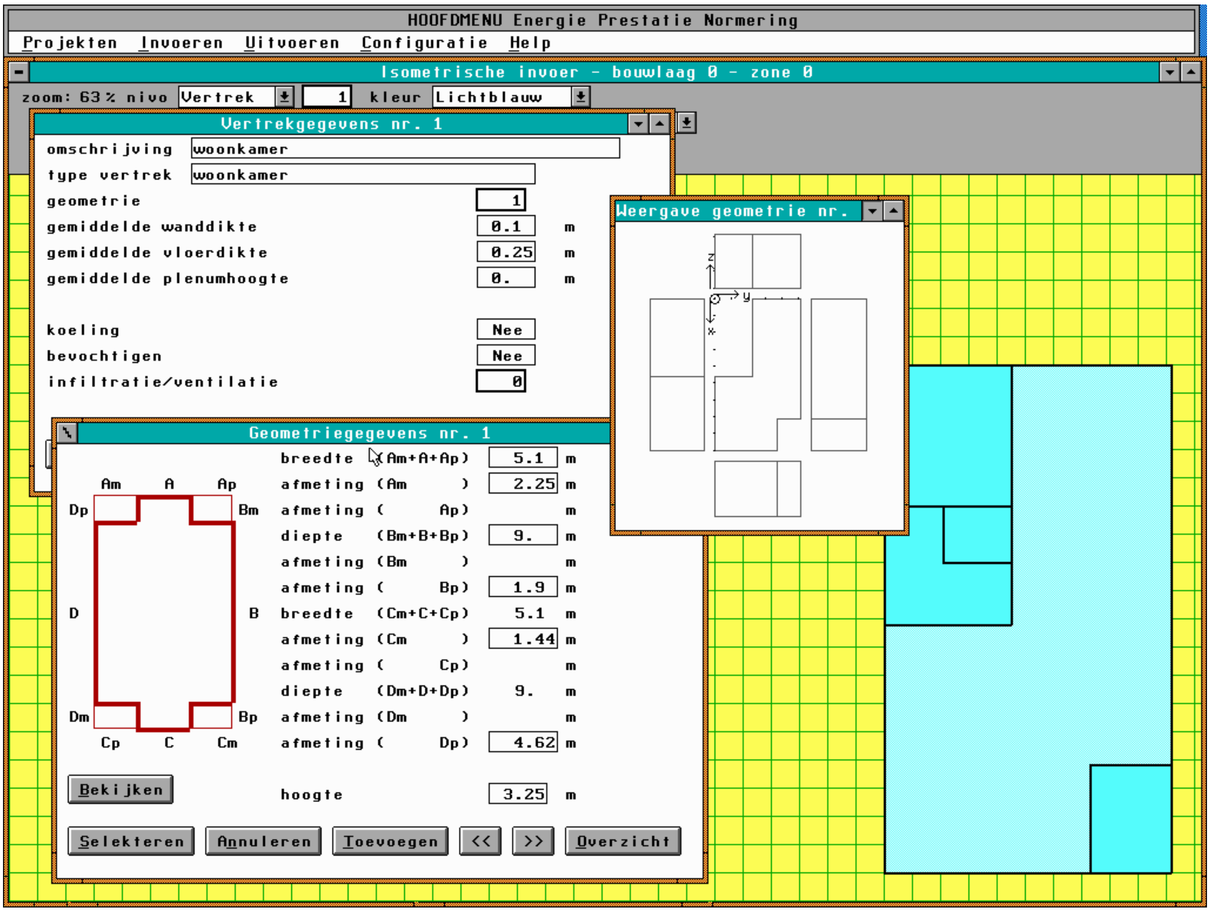

Figure 1   Graphical manipulation of rooms and walls

level. Therefore, a second library is available to supply code for the communication with the central database. This hides the low-level details of the actual database manipulations from the software developer in order to make a coupling on a rather conceptual level possible. The developer only needs to map the data model of the design tool onto the model of the UO. An extension of the core model of the UO is possible in consultation of VABI, when information required by a design tool is missing.

Using the common libraries does not imply that all applications in the UO depend on other programmes inside the UO. Each application stays self-contained and can be run as a standalone programme as well. They do not rely on common code like a central input facility.

The central database also provides the means for project management. Since building design is an interactive process, there will be several versions of a project. Version management ensures that calculations are always carried out on the same version of the project, independent of the tool used. The UO differentiates between two type of versions: major versions and minor versions or variants. A major version replaces the previous version, by storing the project under a new version number. All major versions are kept as an integral part and can be reused at any time. Variants are slight alterations of an original major version. Only the difference between the variant and the original major version is stored in the project database. The user can use variants for comparing different design options and testing which option meets the expectations most.

Another feature of the UO is that the input requirements can be adjusted to the level of detail of a particular design stage. Especially at the start of a project there is a need for global calculations. At a later point, when more information about the building is available, detailed calculations are required. All applications in the UO offer the user the possibility to adapt the level of input to fit best to the current building stage. This level can be adjusted per topic. Behind the scene, the simplified input by the user (on a low level of detail) is transformed into the detailed input required by the application. Unknown data are automatically added as default values. The applications themselves are the full tools, not approximated calculations. Four levels of detail are available:

1. building: the building is considered as a whole, without further subdivision in rooms; for example for the calculation of the global energy use.
2. zone: the building is divided in a few areas, where the climate installation is the same; for example for determination of the energy performance, according to a Dutch standard.
3. room: the building is divided in areas, separated by walls; for example for the calculation of heat loss or the dimensioning of radiators.
4. layer: the most detailed level, where walls are described in terms of layers, each with its own thermal properties; for example for calculation of overheating risk with a building simulation tool.

The user can make a first global calculation very quickly, by inputting the ground-plan of the building or a part of the building in a graphical way (see Figure 1). Properties of rooms (like temperature and heat loads) and walls (like mass and U-value) can be entered for all rooms and walls in the building at once. For example, the geometrical editor can recognise whether a wall is an outer or an inner wall, and generate default values for its properties. In a later stage, the user can do fine-tuning by adjusting the properties of individual walls. A graphical interface is also provided for the design of the duct layout (see Figure 2).

The success of the UO largely depends on the number and quality of the tools that are coupled to it. Many of the tools of VABI are meanwhile incorporated into the UO, mainly in the area of design of HVAC, lighting and acoustics:

- heat loss calculation;
- cooling load calculation;
- duct sizing;
- duct noise calculation;
- intensity of illumination calculation;
- energy performance calculation;
- thermal building simulation;
- radiator selection.

A schematic overview is shown in Figure 3. The labels (VA-101 etc.) refer to names of the corresponding VABI tools [1]. Many of these tools have been used for years by Dutch practitioners as standalone tools.

Other design tools are currently adapted to be part of the UO:

- calculation of window properties (Window Information System, WIS);
- computational fluid dynamics;
- interactive pipe design with CAD (Integrated Design System, IDeS).

Coupling to CAD-packages is also an important means of fast input. It is possible to enter data on

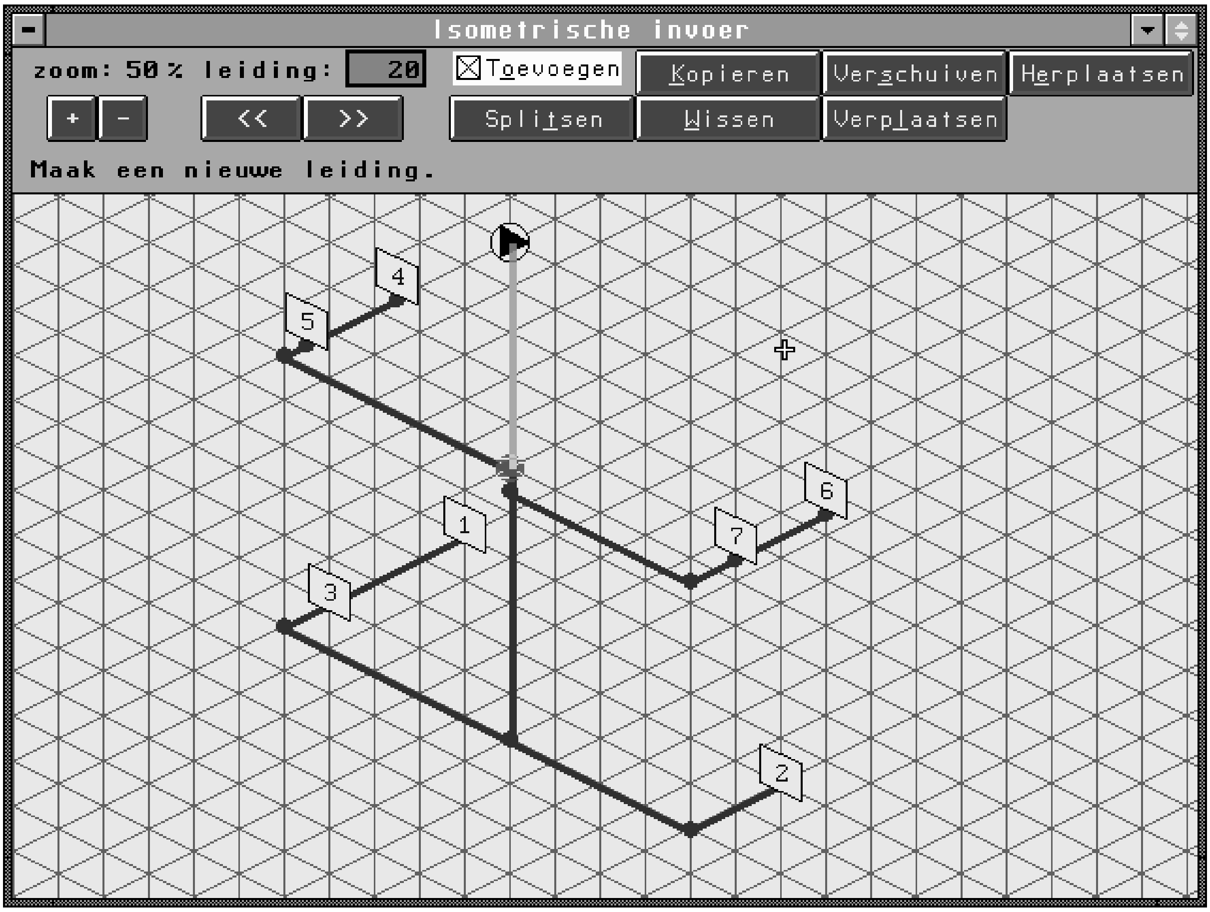

Figure 2 Graphical representation of duct layout

geometry from a CAD-package through DXF files or STEP physical files.

DEVELOPMENT OF THE UO

The UO, as it is on the market now, aims mainly at the integration of the design process in the HVAC area. Users start seeing the advantages of such an integrated system in terms of time profit and quality increase. They like to see the UO develop beyond the design stage and to incorporate other stages in the building process as well. This requires a coupling of the UO with other tools, like cost calculation, specification generation and administration software. Several developments support the growth of the UO in the breadth.

First, the UO has an open architecture. Third-party software developers can easily make their tools UO-compliant by using high-level library functions for the data exchange. The central database model can easily be extended to incorporate additional information for these new tools, after consultation of VABI, who manages the UO data model.

Second, data exchange between the UO and other applications can take place, according to the ISO-STEP standard. Much experience has been built up with these techniques in two related projects, DUS and COMBINE. DUS is an Dutch acronym for Data Exchange System, a project to investigate ways to exchange data between different disciplines involved in the building process (see [2], [3]). COMBINE was a EU project for research on integrated building design systems (see [4], [5]). Both project were dealing not only with the physical process of information exchange, but with the conceptual side as well.

Third, the UO is extended with a uniform coupling to product data, based on classification. As long as individual suppliers of components and product have their own format of supplying product data, it is difficult to development general tools for the generation of specifications and the cost calculation. A large Dutch project to come to an IT infrastructure for the HVAC line of business is working on a standard structure for product classification. This perfectly fits to the UO. In the design stage, calculations within the UO lead to a list of technical solutions, which satisfy the requirements. The actual products, which correspond to these technical solutions, can be easily found from standardised product databases. Eventually, a choice can be made, based on several aspects.

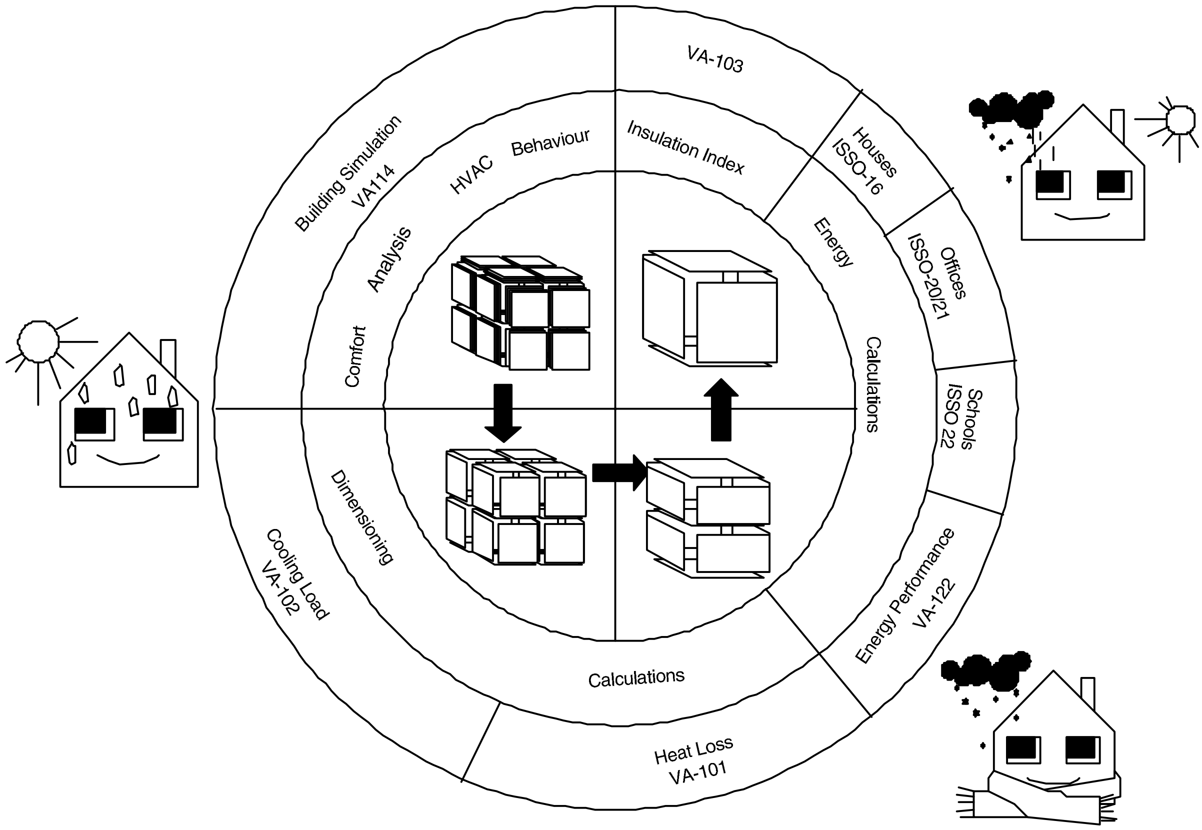

Figure 3 Overview of the UO, showing the kind of programmes involved and the different levels of detail

Besides a growth of the UO towards other stages of the design process, the number of tools for the design process itself can also be extended. For example tools evaluating the influence of the building on the comfort and health of inhabitants. Current research focuses on tools for indoor air quality and for the emission of building materials.

A final line of research involves the mechanisms of the UO itself, i.e. research in the depth. The current status is that the UO consists of a collection of tools, which store their data in a common database. The tools provide a consistent user interface for user input. The user is responsible for the value of the input and decides which tools to invoke and in which order. He also has to collect the output of the various tools and to interpret the results.

One aspect for improvement is an indication of the accuracy of the outcome of an evaluation with the UO. In many evaluations, the absolute outcome is not important, but trends in the output when the input is varied. To get information about trends and an indication what input changes have the biggest impact on the output, an automatic sensitivity analysis could be built in into the UO. This tool would automatically calculate a set of variants of the case as entered by the user, and plot the difference in output between them. In this way, the user does not only get one absolute output value, but also the variation of this output when certain input parameters are varied.

Another aspect for improvement is the output itself. Currently, each application generates its own output. Often, this output is very detailed, irrespective of the level of input. A great help to the user would be to structure the output of applications, in a way similar to the input. One way is to centre the output around the performance of the design with respect to the original design criteria (see Figure 4). The user is presented a simple overview of all criteria and a short message whether the design satisfies each criterion or not. More detailed information, behind each criterion is available on request.

This kind of presentation also requires that the tools are more integrated in the UO than currently is the case. The user is confronted with a system that is problem-oriented rather than programme-oriented. Evaluation of a design criterion might require the run of several tools before an answer can be obtained. These runs should be performed automatically by the system, after all input is present in the central database. This requires some management tool which controls the calculation process. The scenario, i.e. the order of invocation of the programmes and required

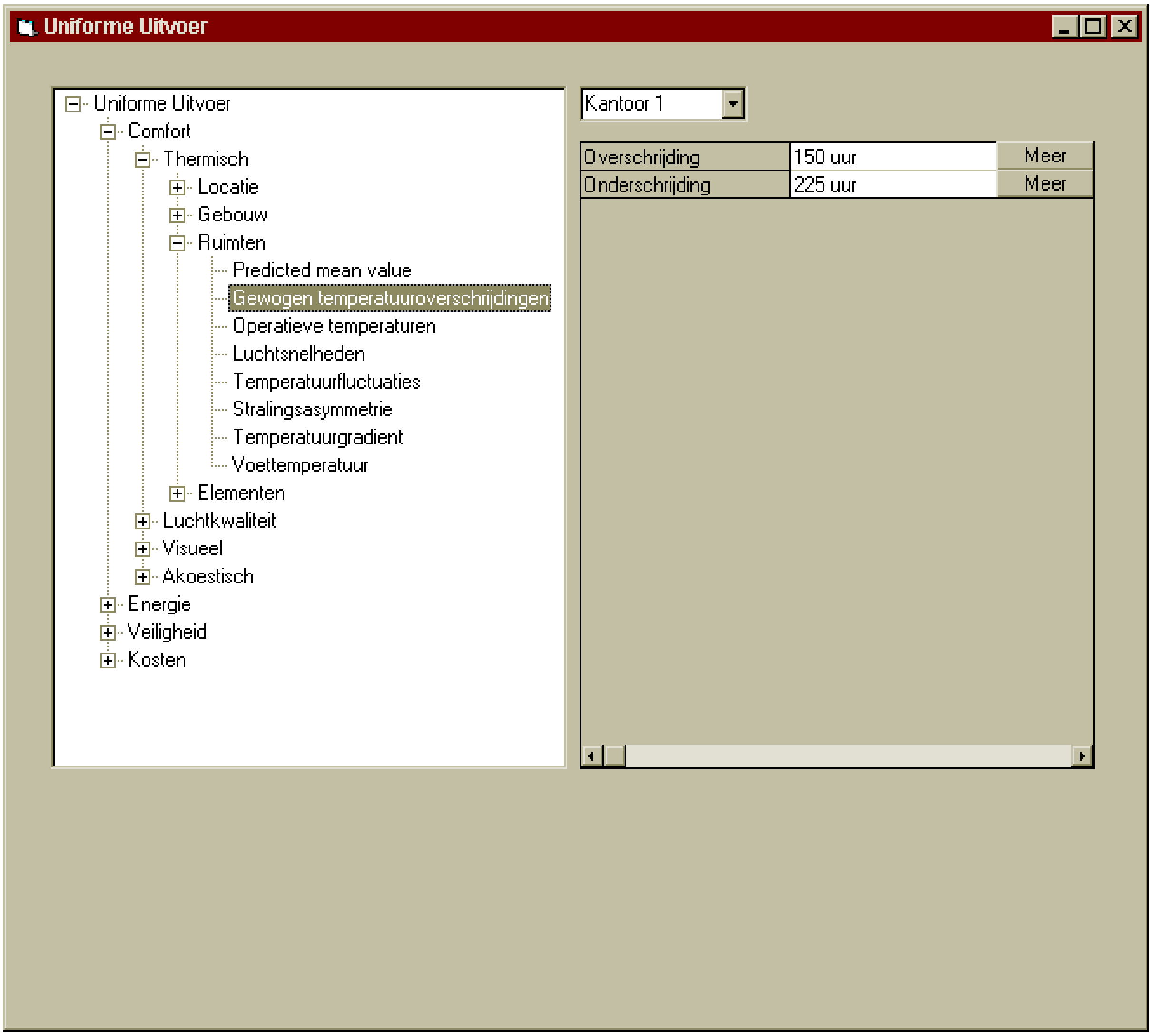

Figure 4 Output of the UO, structured around design criteria

interaction with the user, depends on the amount of intelligence built-in. The COMBINE project, mentioned before, has developed two prototypes of such a scenario manager, one with a rather shallow control, based on predefined scenarios, and one with a flexible strategy, where the system can evaluate the process after each evaluation and suggest the user which actions to undertake.

CONCLUSIONS

The UO is a first practical application of an integrated framework for HVAC development in the Netherlands. Currently, the focus is on the design stage.

New research investigates ways to extend the UO in several ways. First, by incorporating more design tools, not specifically HVAC-related. Second, by broadening the UO to incorporate other building stages as well, besides the design stage. Third, by offering exchange capabilities to other disciplines in the building process, like architecture and administration. Fourth, by tightening the integration between the tools in the UO, in order to get a better support for the user in making design decisions.

REFERENCES

[1] VABI Home Page (in Dutch), http://www.xs4all.nl/~vabi/

[2] van Dam, P.R., “DUS BOEK”, VABI, Delft, The Netherlands (1992); in Dutch.

[3] “DUS-Fase 5”, VABI, Delft, The Netherlands (1993); in Dutch.

[4] Augenbroe, G.L.M. (Ed.), “COMBINE 2 - Final Report”, TU Delft, Delft, The Netherlands (1995).

[5] COMBINE Home Page, http://erg.urd.ie/combine.html