

SIMGIS: GIS FOR THERMAL SIMULATIONS IN THE BUILT ENVIRONMENT

AUTHORS

Valeria Agnolotti, Christine Giger

ABSTRACT

Building simulation programs perform thermal simulations in a stand-alone scenario and, therefore, the resulting building automation is unable to handle a broad scenario of environmental changes. Due to the time needed for a building to react properly to short-term sudden weather changes, standard building automation is not enough to control energy consumption and thermal comfort efficiently: the sensors placed on the external envelop of a building provide information about weather changes too short in advance about weather changes. A meteorological station can inform in advance a group of buildings in an area about weather changes, but is unable, unlike a GIS, to identify the buildings which are really interested in the phenomenon and to send the weather changes information only to those buildings. The Swiss Federal Institute of Technology (ETH) has developed a new methodology to optimize building thermal performances by utilizing GIS, simulation, on line meteorological data and building automation.

INTRODUCTION

The development of new technologies and the improvement of existing building concepts play an important role in optimizing building behavior in order to improve the occupants' comfort and to decrease negative effects on the environment.

Building simulation estimates and models the influence of construction, installations and environment on a building and predicts its behavior. It is a quite common tool and often used to support design phase decisions, especially for low-energy or "passive" buildings.

In the pursuit of a more precise prediction of building behavior in the design phase of a project, first attempts at building simulation started in the 1960s and since then the programs have been constantly improved. Reasons for the development of such simulations are both economical and ecological aspects. At the same time, the needs of occupants for internal comfort have been increasing and therefore corresponding, but still cost-efficient, construction is required. Considering that buildings consume a third of the world's primary energy, the effort to reduce buildings' energy consumption is justified.

Nowadays, a large number of building characteristics and behaviors can be modeled and simulated. Many physical processes like heat, air and moisture flow, heating and cooling loads, artificial and day lighting, acoustics or smoke transport are implemented in simulation programs. Among many other tasks, simulation programs are used to design or improve heating, ventilation and air-conditioning systems or to estimate the total energy consumption of a building. Simulation programs also contribute to optimize the internal comfort (thermal comfort, acoustics, lighting, etc) for the occupants.

Building Automation include various interconnected systems. They work autonomously and are able to adapt to changing conditions in order to optimize the building's behavior.

Building simulation is applied at various stages in a building's life cycle (e.g. early sketch design, detailed design, construction or modification, maintenance, etc.). Building simulation software is used both in commercial projects and in research studies.

Building simulation models the physical processes influencing a building and predicts building behavior. It is a quite common tool and often used to support design phase decisions, especially for low-energy or "passive" buildings.

To further improve building energy balance and comfort, concepts are sought outside the classical areas of architecture and construction engineering. This paper presents a new approach consisting in connecting Geographical Information Systems (GIS) with building simulation and Building Automation (Intelligent Buildings – IBs).

The benefits and problems of a connection between GIS, IBs and building simulation software is subject of an ongoing research project at the Institute of Geodesy and Photogrammetry at the Swiss Federal Institute of Technology [Agnolotti, Giger, 2004].

Traditionally, building simulation software focuses on the basic building geometry, materials, usage schedules and climate data as input parameters and doesn't take into account many aspects of the environment in which the building is located. This assumption of a building on a blank plane oversimplifies reality and doesn't take

proper account of the fact that each building interacts with the surrounding built environment. Research and investigations report that building simulation can be significantly improved by taking more detailed environmental information into account. For this purpose instruments are needed to handle geographic information and carry out spatial analysis.

The results presented in this paper do not deal with all the aspects of this research field but are restricted to a few selected topics. The main part consists in the conceptual design of a data model for both building and environmental information. In preparation for this, an in-depth evaluation of available building simulation software was carried out. To show the possibilities of interaction between a GIS database and a simulation program, a simple prototype of an interface program was developed.

METHODOLOGY

Connecting building simulation, GIS and building control

Most building simulation programs consider the building as an isolated object standing on a blank plane and therefore all the physical processes are modeled for this special case. Many physical processes influencing a building could be described in more detail if simulation models would take into account the built environment of a building (e.g. the adjacent buildings). Not all parameters needed for the simulation would be known explicitly but would have to be derived from other data. Therefore, a tool is required which provides functionalities for spatial analysis. Spatial analysis is exactly the main usage of GIS and thus GIS is well suited for connection with building simulation.

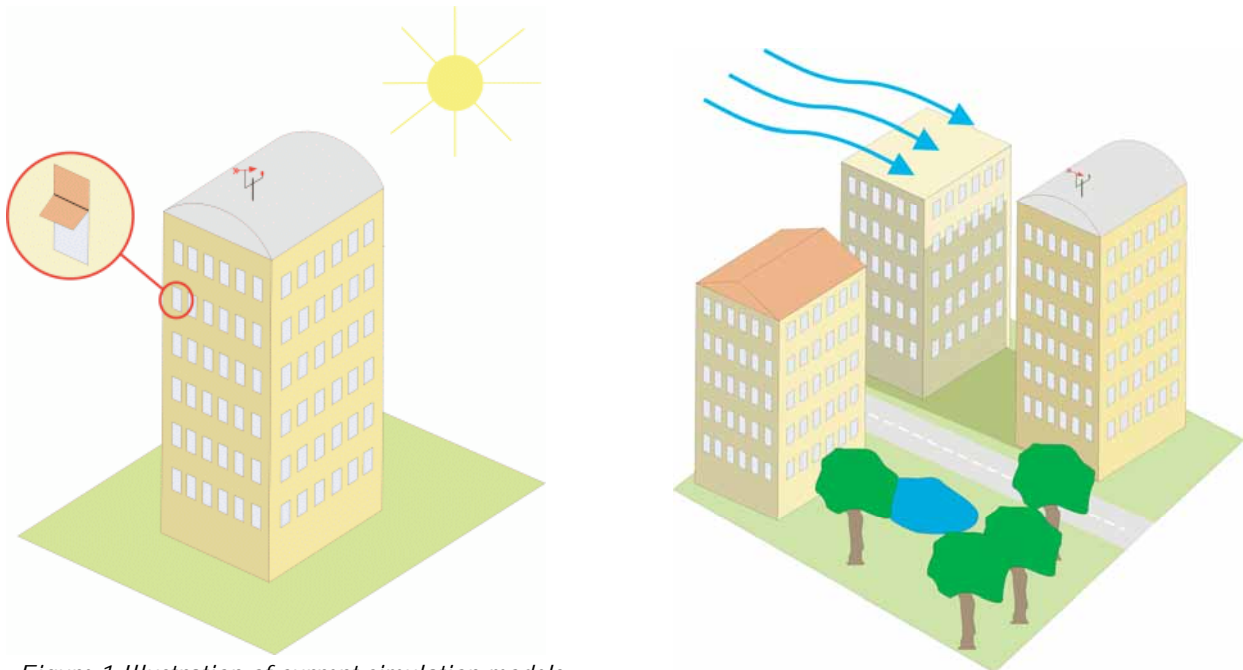


Figure 1 Illustration of current simulation models

Figure 2 Illustration of a simulation model including environmental parameters

An interconnection between building control and simulation could optimize the regulation of controllable building systems like windows, shading devices, heating systems or ventilation by simulating and comparing different versions of possible courses of action.

The core of the system architecture integrating building simulation, building control and GIS has to be a central unit consisting of a database and management software. The database contains all building information needed for simulation (e.g. geometry, materials, etc.) and environment information (e.g. ground surface, adjacent buildings, etc.). Additionally to static properties, the current status of the controllable building elements and the local weather conditions could be stored. Sensors installed inside the building and on its façade constantly acquire data and communicate it to the central unit. All systems and technical installations are connected to the central unit and operated by it. The role of a GIS in this context is to generate useful information by carrying out spatial analysis with existing and predicted data. Building simulation taking this additional data into account could provide information about the effects produced by changes in the environmental or building conditions.

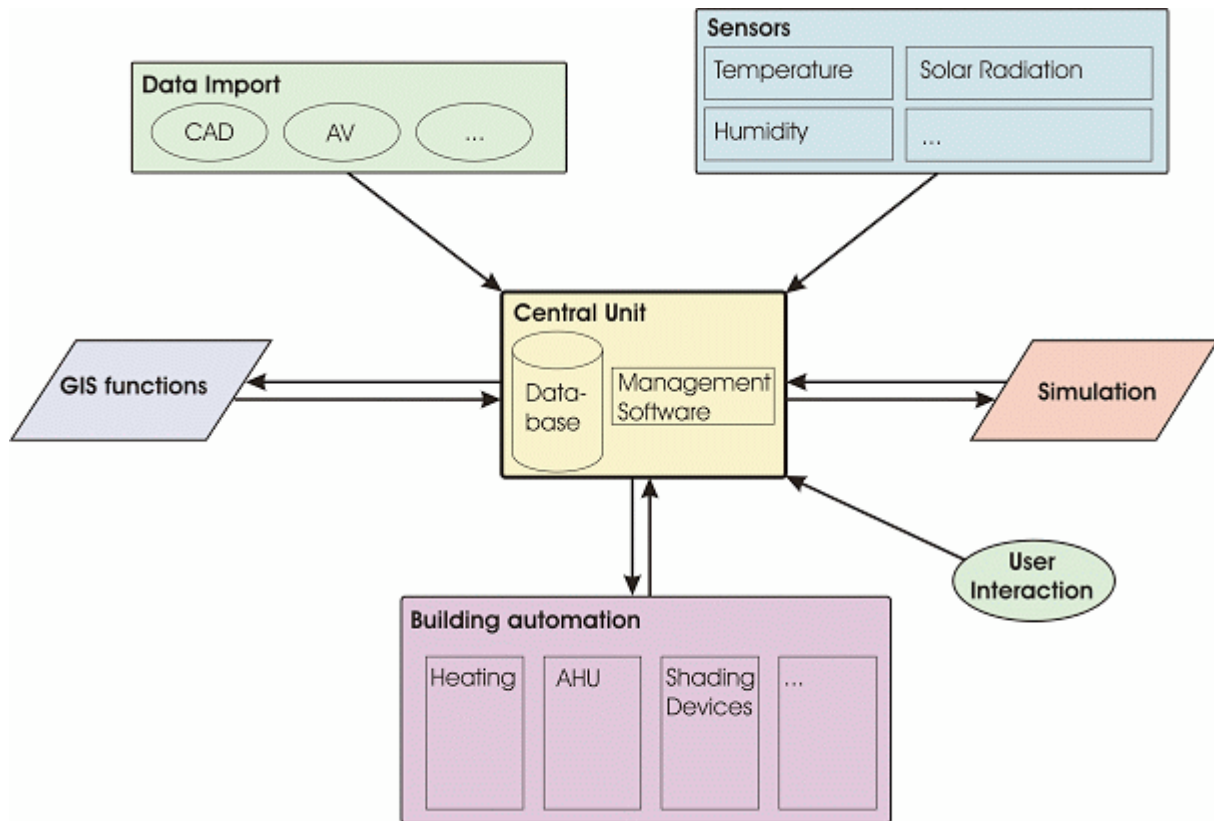


Figure 3 Concept for connecting GIS, IBs and building simulation

It is obvious that this concept needs a kind of “intelligence” able to evaluate the results of the simulations and to decide which actions are required to optimize the building behavior. In this concept the management software of the core unit represents the intelligence. As data is valuable only if it can be interpreted, “intelligence” is the most important part of such a concept but it is also the hardest one to put into practice.

The selected test thermal simulation software

DEROB-LTH is result of academic research at the University of Texas in Austin, USA, and has been constantly improved at the Department of Construction and Architecture at Lund University, Sweden.

It is a software package that simulates hourly energy performance of building structures and calculates the thermal comfort of single rooms. The name DEROB is an acronym for Dynamic Energy Response of Buildings. The input data for every thermal simulation software consists of building characteristics and a few environmental parameters. General attributes like the geographical location of the building, its time zone and climate data are one part of the information. The other part is the building geometry that has to be entered by specifying each of its elements like walls, windows and doors.

The geometry of an element is defined by its lower left corner (in a local coordinate system), its dimensions and its orientation in three-dimensional space. The properties of the building materials (concrete, brick, insulation, glass, wood, etc.) an element is made up of can be chosen from libraries that can be augmented by the user. DEROB can handle buildings with up to 8 volumes.

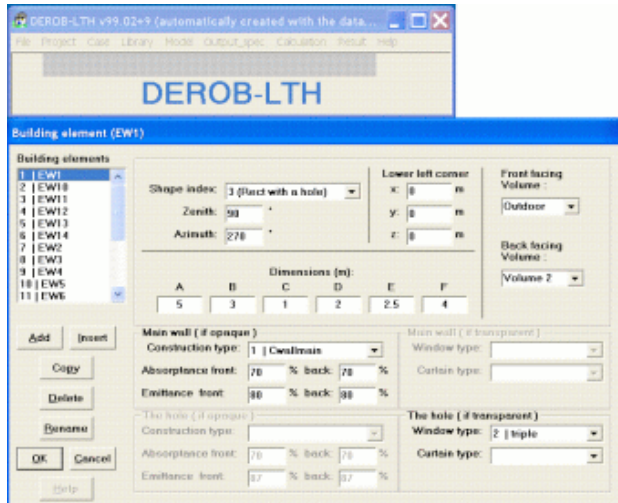


Figure 4 User interface of DEROB

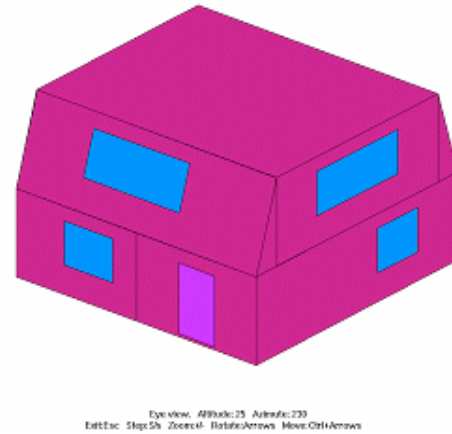


Figure 5 Screenshot from the DEROB viewer tool

BUILDING DATA MODEL

Goal

Based on the results of the simulation software analysis and especially on the parameters needed by these programs to calculate thermal balance for buildings, a data model describing all relevant elements of a building, their properties and their relationships has been developed. Since every object of the real world can be modeled in any degree of detail, generalizations had to be made in order for the model to stay simple and manageable.

In this case, only the core elements of a building – like walls, rooms, doors, windows and the like – have been modeled in detail. These are the elements absolutely necessary for any thermal balance simulation. Several other components (like heating/ventilation system, shading devices, balconies etc.) should be considered, but not worked out in all details.

The primary goal is to have a model that at least contains all information needed to start a simulation with the test simulation program. Other programs will need slightly different or additional information. The data model must therefore be easily extendable.

Modeling a building

The task of describing complex 3D objects in a formal way is not an easy one. The nature of such real-life objects – which are so easily and intuitively interpreted by humans – does not lend itself to abstract – and thus computer-readable – characterization.

There is no single, standard way to do it. Depending on the application, different modeling concepts prove convenient. Main driving force of 3D data technology has been and still is pure visualization applications and many of the popular techniques for describing 3D data are heavily influenced by this.

The first step in modeling a real world object – such as a building – is to describe it and its components as precisely as possible. One approach to do that would be to say that a building is made up of walls that enclose volumes (rooms). Both walls and rooms have their distinctive properties (attributes such as materials, thickness, air temperature etc.) and can themselves contain other objects – such as windows or radiators. It is essential to know which walls stand next to each other and which volumes they enclose. From these rough ideas it becomes quite clear that a good way to model a building is to use node-edge topology (boundary representation). It provides the features needed:

- Storing and addressing individual elements
- Storing attribute data for individual elements
- Connectivity and topology rules

These features become even more important if the ultimate goal is the calculation of a thermal balance simulation. There, the attributes of walls, windows and doors (i.e. the layers of materials they are made up of) play an important role and one needs to know which volumes are bounded by which walls and so on.

Creating the conceptual model

After having selected a method for modeling the building and having described the components, their properties and the relationships it should contain, the next step was to create a more formal conceptual model in a system-independent appropriate modeling language.

The Unified Modeling Language (UML) provides the possibilities needed to formalize the building model. Using a UML editor, a structural class diagram with all the classes, their attributes and methods as well as the relationships between these classes with their cardinalities was defined.

As the model grew, classes and attributes were added and classes were linked. More and more problems showed up which forced reiteration, discussion, checking against the parameters needed by DEROB and altering the model time after time, thus creating a more refined model with each iterative step.

Temporal Aspects

While the building itself is a static object, many properties like its usage (loads), heating, ventilation, shading devices status or the outside weather are highly dynamic and can change within minutes.

This leads to an area for which the "classical" building simulation programs are not designed. Even the so-called "dynamic" simulation programs that use one-hour or smaller time steps (like DEROB) are made for longer-term simulation and not for minute-by-minute indoor climate calculation with input from sensors, variance comparison and dynamic controlling of technical equipment.

The same applies to climate vs. weather. Whereas climate means a longtime average of the climatic conditions, the instantaneous weather is completely dynamic and can differ greatly from these average values. If totally dynamic simulation is the goal, a different kind of software has to be used or the existing programs have to be adapted.

A database can only be used to store static information (like geometry or material properties) and/or the system status for one moment in time (e.g. room temperature = 20°, heating system runs at 60 % power, window #4 is open, all blinds are lowered 30 cm etc.).

Results

The resulting building model offers the possibility to store most of the information needed to perform building simulation calculations in the provided class structures. It is meant to be generally usable and extendable.

So this model should be considered a skeletal structure from which a complete model that can handle almost all the data associated with a building could be built.

Some particularly important or interesting parts of the conceptual building model are described in the next paragraphs.

Main Geometry

The principal axis of the building model corresponds directly to the elements of a boundary representation model (see Figure 6):

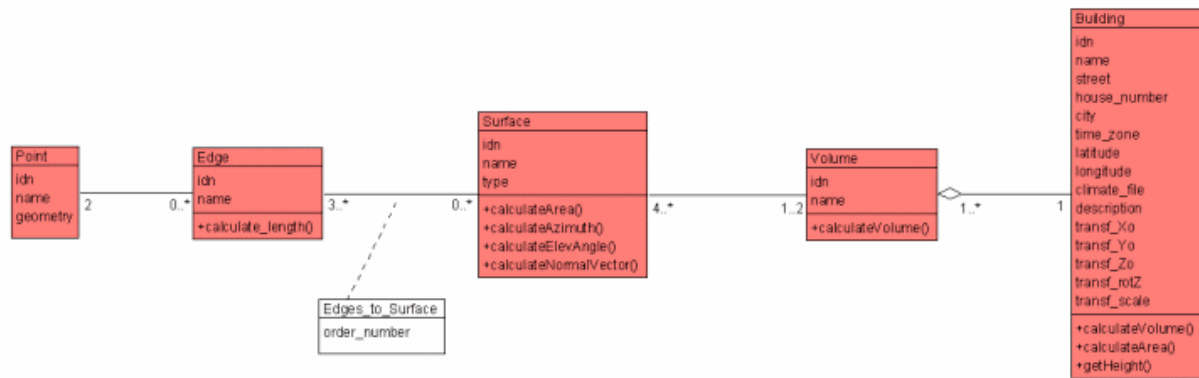


Figure 6 UML diagram of the main building geometry model

- The 'point' (or vertex) class contains the actual geometrical information as x-, y- and z-coordinates. It is assumed that these coordinates are entered in a local Cartesian coordinate system with a known relationship (in the form of a set of transformation parameters) to the global coordinate system of the environment.
- Two points make up an edge.
- Three or more edges bound a surface (wall, floor, roof). To make later geometrical calculations a lot easier, surfaces are considered to be plane. The order in which a surface's edges are referenced (represented by the "order number" attribute in the relationship class) determines its orientation.
- Four or more surfaces completely enclose a volume (room).
- A whole building can consist of one or more rooms. It is up to the user how detailed the interior of a building has to be modeled. For most thermal balance simulations, one volume (without interior walls) for all rooms heated to the same temperature is sufficient. This model allows for this as well as for precisely modeling the interior geometry of a building.

The "Surface" class and its equivalents

The "Surface" class is used to represent walls (exterior or interior), floors (interior or foundation) and roof elements (see Figure 7). The "type" attribute allows storing these distinctions.

The calculation of a surface's area, orientation (azimuth), elevation angle and normal vector are possible methods for this class. A surface is supposed to be defined without holes. That is if there are windows or doors in a wall (or a roof), these have to be defined separately in the corresponding "Window" or "Door" classes respectively and linked to the wall through the appropriate relationships. This is somewhat prone to errors and clearly one of the disadvantages of a simple boundary representation model.

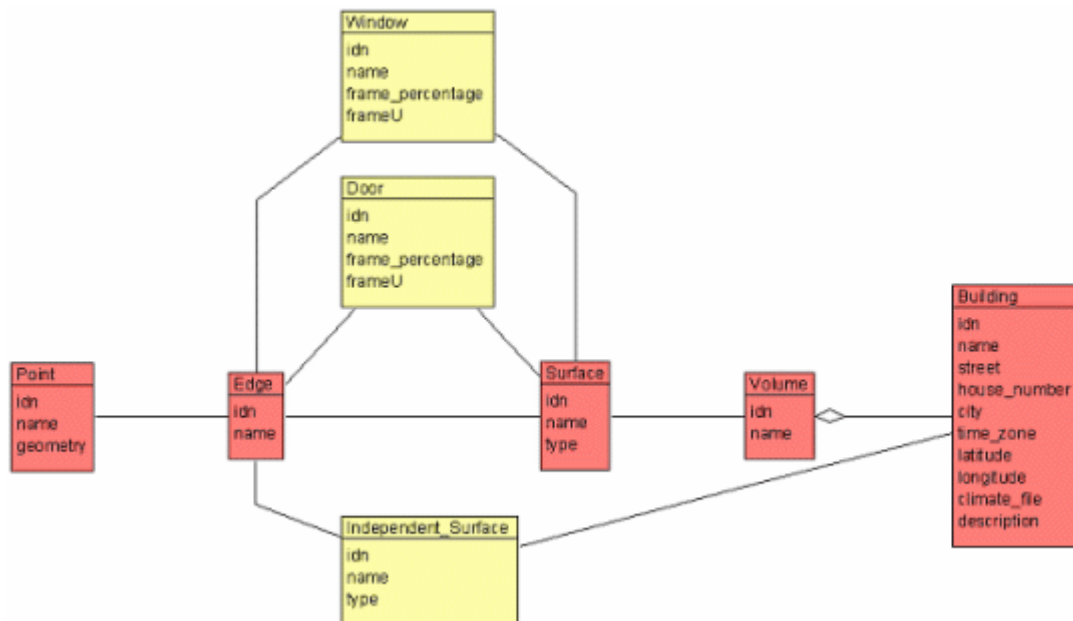


Figure 7 Simplified UML model including the surface classes

The class "Independent_Surface" is meant to hold surfaces such as balconies or protruding walls that are part of the building and might influence its energy consumption (e.g. by shading a window) but are not part of the envelope of a volume.

The "Building" class

The "Building" class holds general information about the building, some of which is of purely descriptive nature. Other parameters – such as longitude, latitude or the directory of a file containing climate data for its location – may be used in the simulation.

The "transf_" attributes describe a coordinate transformation from the local (building) coordinate system to a global coordinate system. A translation in x, y and z, a rotation around the z-axis (change of azimuth) and a general scale factor is assumed.

Possible methods are the calculation of the building's volume, area or height.

Materials

One of the factors that influence a building's energy efficiency the most is the composition of walls, windows and doors. The properties of their materials – such as conductance, transmittance etc. – and the combination of these materials in an effective way determine how much energy is lost through the envelope of a building or how much solar gains can be achieved. Thus this aspect needs to be modeled carefully.

As can be seen in Figure 8, each building element (walls, floors, roofs, windows, doors) is assigned an "element type". These types – for example "heavy brick wall" or "triple pane window" – are stored in the "Element_Type" class along with some attributes. These element types consist of material layers in a certain order and of a certain thickness. This information is stored in the "Layer" class.

As in the test simulation software, materials and their properties are kept in three separate material libraries, one each for glasses, gases and opaque materials. This system makes it possible to store the properties of each material in one central location where it can easily be changed if necessary. Each material can be referenced by any number of building elements. For example there could be "air" between the two panes of a window (which would thus have three layers: glass-air-glass) as well as inside a brick wall.

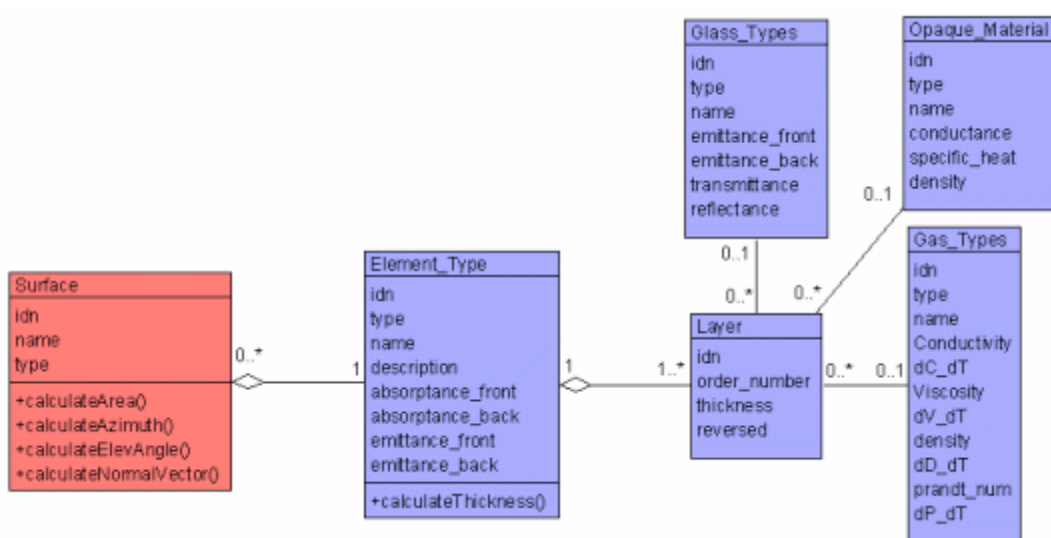


Figure 8 UML diagram of the material classes

Technical installations

Categories of objects not modeled in great detail (and not so prominently used in DEROB) are additional building components (like shading devices) and technical installations (like heating and cooling devices, air handling units and so on). These are very complex to model exactly, all the more so because of their dynamic behavior.

The specifications of the technical installations (such as maximum heating power etc.) could easily be stored as attribute data. However we didn't go deep in this field to decide what attributes have to be in the model. So all the current (though extendable) model provides for is a "type" attribute that could contain a link to a file describing the system and a "status" attribute stating whether the system is currently working or not (see Figure 9).

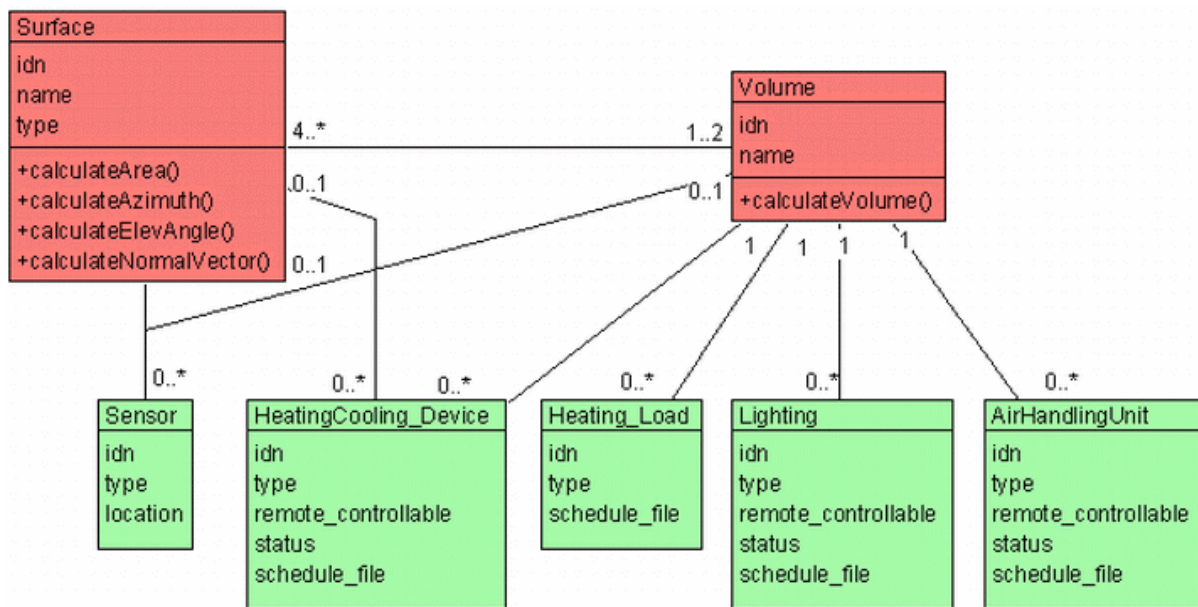


Figure 9 UML diagram of the technical installation classes

As described above, these systems are completely dynamic and can change their state constantly. This either according to a predefined schedule (which could be stored in a text file) or dynamically, regulated by actuators by means of input data from sensors.

Building loads

The same problem arises when trying to model volume loads, i.e. the heat generated by humans and appliances like computers or light bulbs inside a building. Here as well, average values for each day of the week could be provided in a schedule file.

Ensuring topological integrity

The boundary representation model relies heavily on topological information. Data entered into the model has to be carefully checked for topological correctness in order to maintain a consistent dataset.

The easiest part of this task can be described by the cardinalities of the relationships between the classes: e.g. a surface cannot be bounded by less than three edges – else it would have no area or not be closed. Integrity also has to be checked for classes without geometrical meaning, e.g. the material layers. The observance of the cardinalities can be checked without problems by counting entries in the relationship classes of a database.

Ensuring geometrical integrity is much more complicated. The following are a few examples of the conditions that must be satisfied and ought to be checked:

- The edges bounding a surface (including windows and doors) lie in a plane, form a closed shape and do not intersect each other. The order numbers (in the relationship class) must be in the right sequence in order to correctly determine the orientation of the surface (by calculating its normal vector).
- The surfaces bounding a volume form a closed space and do not intersect each other. This may not seem overly complicated for perpendicular rectangular surfaces (which are admittedly the most common in buildings) but can get really challenging for more complex shapes and angles.
- Since windows and doors can be made up of arbitrary edges, it has to be made sure that they lie in the plane of the surface they are related to.

Checking for topological integrity is a strong capability of ESRI ArcGIS. This opens up the possibility of using GIS functionalities to check the integrity of data entered into the database before running any export programs to the simulation software.

Therefore the system-independent conceptual model has to be extended to a Visio logical model, in order to be imported into ArcGIS to build a Geodatabase (see Figure 10). The following aspects have to be considered:

- define the geometry and attribute types of the feature classes
- organize objects and feature classes according to their attributes and functions
- implement the defined relationships and inheritances
- simultaneously build a graphical representation of the data model
- export data model as XMI-file
- check the validity of the model with a semantic checker.

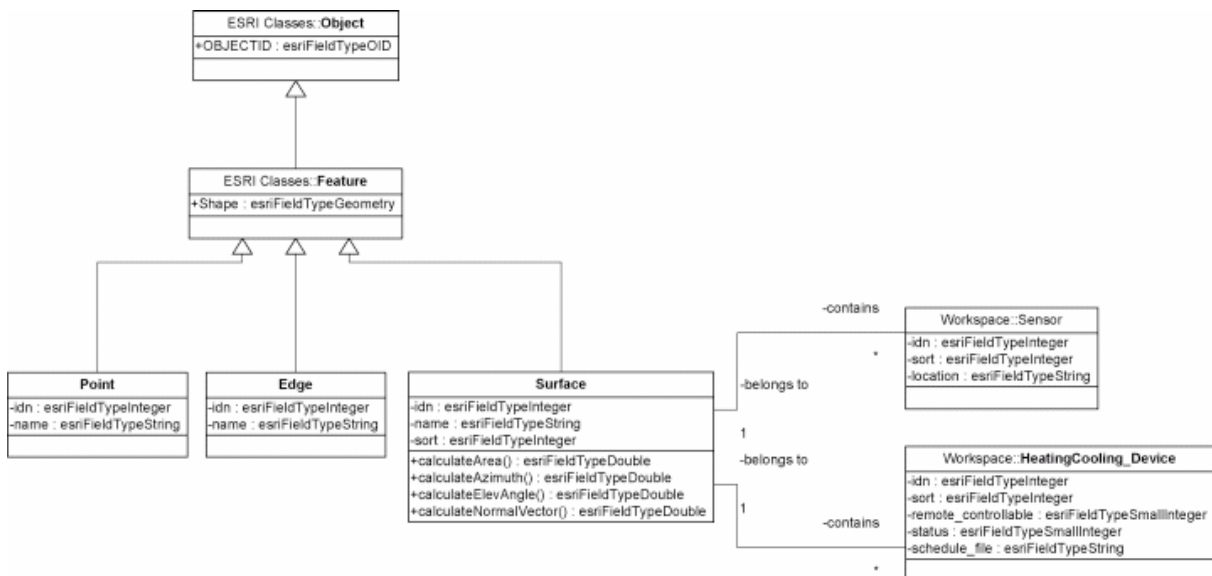


Figure 10 Visio logical data model example

BUILDING THE GEODATABASE

General Goals

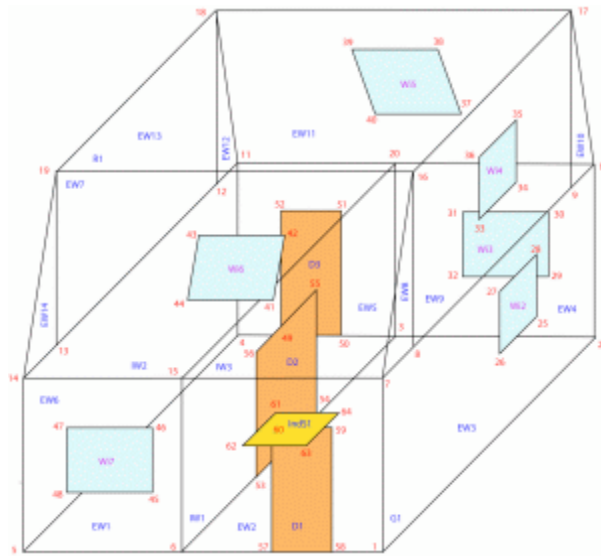
The goal of this next step is to create a database according to the conceptual model to prepare a structure to be later on filled with actual data. Preferably, all the elements described in the UML model (classes, attributes, attribute types, relationships, roles, cardinalities, value domains etc.) should be carried over to the database. The building model was not quite unproblematic since it has been designed to hold completely 3D geometry.

Defining a test building

In order to test the practicability of the database and to have simple data to test the planned extraction program, a small building with simple geometry was defined (see Figure 11). It contains most elements which DEROB is able to handle and covers some special cases like triangular surfaces, slanted surfaces or independent surfaces.

Having provided all geometrical elements like points, edges, surfaces, windows or doors with an unambiguous identification number, this data is ready to be entered in the geodatabase.

Figure 11 Sketch of the test building



Built environment 3D data

The Geodatabase should contain also all information about the built environment influencing the test building. Standard 3D urban data are generated from photogrammetric captured point clouds and can be imported into ArcGIS for 3D visualization as multipatch feature classes. Nevertheless, these are not complete 3D GIS geometries and can't be integrated in the Geodatabase according to the 3D model created for the building. Therefore a strategy has to be developed to extract both building and environmental geometries and attributes for the simulation program.

TEST INTERFACE/DATA EXTRACTION PROGRAM

Before implementing the logical model and the correspondent Geodatabase for the building, a test Access database has been built, to test the possibility of extracting geometry and attributes building data and to format them as input files for simulation.

General aspects

In principle an interface program between a database and a simulation software should be able to find all required information in the database and transform it into a form which is comprehensible by the simulation software. The complexity of this transformation depends mainly on the differences between the database data model and the characteristics of the input parameters that are required by a simulation program.

Rules must be formulated according to which geometry is converted so that the differences between the images of the building in the database and in the simulation program are minimal. The extraction of thematic data from the database is normally less complex than the extraction of geometrical information unless a simulation program requires a parameter that is not directly available in the database and must be calculated first.

If several simulation programs depend on information from the same database, the possibility of creating a data model specifically for one selected simulation program drops out and the interface program must be able to handle all the general transformation tasks.

Concept

The target of the interface program is the extraction of information from the building database, the transformation of this information and the generation of the input files that are required for a DEROB simulation. In this case the data model in the database corresponds to a point-edge-surface-model. As opposed to the capabilities of this database, DEROB only supports the construction of a building with a few clearly defined geometrical features.

Some assumptions were made for the development of the program:

- Geometry of the test building is limited to the geometry features provided by DEROB – namely rectangles, rectangles with a hole and triangles.
- Only one building is stored in one database even though the model provides for multiple buildings. This simplifies the required SQL statements for retrieving data.
- The number of windows and doors in a surface is also limited to one. There would be several not completely simple possibilities to process cases with multiple openings: the surface could be split into multiple surfaces or the windows or doors could be merged to one "hole". Some of the difficulties of processing these cases are to compute a fitting surface segmentation or to find reasonable geometric and material parameters for the combined hole.

The interface program for DEROB has been developed as a stand-alone program simple to use and creating all the required files for a simulation with DEROB. As programming language figures Visual Basic 6.0, which offers good functionality in database access and graphical user interface creation.

Results

Graphical user interface

The graphical user interface leads the user through all steps of the program and enables the specification of all the required data. First, the user has to specify the name of and the path to the database where the building is stored. For the placement of the generated files in the DEROB file structure, the user has to indicate the DEROB project directory and the names of the project and case to be created. It is also possible to write a case into an existing project or to overwrite projects and cases.

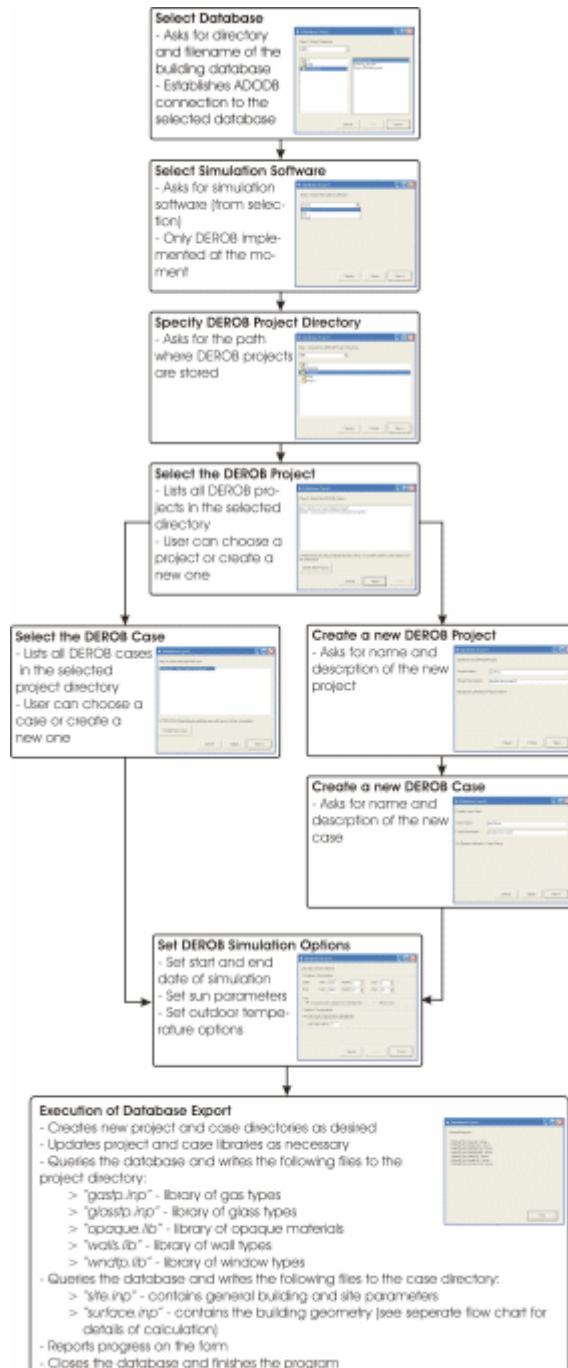


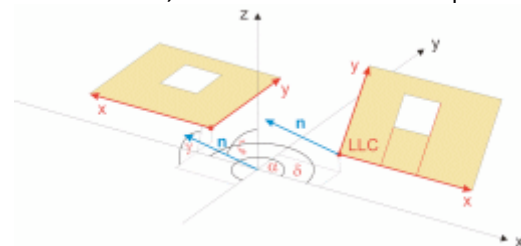
Figure 12 Flow chart of the interface program

For these file chooser dialogues, Visual Basic provides practical pre-defined elements like a drive-, directory- and file-selector box that can be placed on the form. The user selection is stored in a certain variable and can easily be read out.

In a last step, some parameters for the simulation are set: the start and end date of the simulation and whether sun and outside temperature data is available from a climate file. However, if the climate file provided (in the DEROB's .cli-format) specifies a different time period, the start and end date set in the options form of the database export program are overwritten by DEROB. After having specified all the required information the program executes the transformation, generate the required files and finish. Figure 12 shows the sequence of the GUI forms and the functions they contain.

Geometrical data transformation

DEROB models buildings with only a few different geometry features like rectangles, rectangles with a hole, triangles, four-sided polygons and five-sided polygons. As shown in Figure 13, DEROB geometry elements are described by their lower left corner (indicated in a local building coordinate system), the orientation of the surface (specified by azimuth α and zenith distance ζ of the plane normal vector) and a number of parameters



depending on the particular feature, like e.g. the dimensions of a rectangle. The azimuth has not actually been defined geodetically but as the angle of rotation in a normal mathematical sense: starting in the east and counting counterclockwise.

Figure 13 Illustration of building and surface coordinate system

The interface program executes the transformation procedure for each surface. In a first step the program retrieves all the points of a surface and arranges them according to the defined rotation direction. Because of the selected data model, it is possible that multiple points lie on the same straight line and so the program has to find all the points that are not corners of the surface. These points are left out, as they do not provide additional information about the surface. If the surface has a window or a door, the program searches for the corresponding edges and points. To calculate the lower left corner and the dimensions of the surface it is quite convenient for the surface to be parallel to the x-y-plane, so that the necessary operations can be carried out without considering the z-coordinates. This can be achieved by rotating the surface around the z-axis (δ) and the x-axis (γ). The angles of rotation can be deduced from the plane normal vector that is calculated using the cross product of two non-parallel vectors in the surface. The plane normal vector also serves to compute azimuth and zenith distance.

Further, the program has to acquire information about the materials of the surface and potentially of the door or the window and has to detect the volumes the surface is part of. After processing each surface the acquired data can be written in the file "surface.inp".

Generating the library files

All the thematic information required by DEROB has been stored in the database in quite a similar form to the DEROB file structure. This minimizes the complexity of the function generating the needed library files. DEROB uses an index structure in the library files that is analog to the primary key concept in relational databases. So the interface program must guarantee that this index structure is created well in the files and DEROB is able to correctly retrieve data.

Updating the program files

After having created all the files and folders in the right structure, an update of the program files "project.lib" and "cases.lib" is needed. The names of the generated projects and cases have to be written to these files in order that DEROB can find the added items, load the project and case and run the simulation.

CONCLUSIONS

Results

Some of the possibilities in the fields of building simulation and the potential of geographical information to optimize the performance of buildings have been explored. The extensible UML model of a building and the prototype of an interface program between database and simulation as well as the corresponding procedures are just basic results and call for further development.

A possible process workflow for defining a data model for a building and its environment, setting up a database according to that model and extracting data for simulation has been tested.

The conceptual and logical model

The definition of the building data model showed that UML is a very powerful tool for defining such conceptual models in a modern class-oriented style. On the other hand, it became evident that it is not as easy to transform such a model into a logical model, i.e. implement it in a database. Another issue with the node-edge modeling approach and relational databases is the need to implement extensive integrity and topology testing procedures. In this field, modelling for both building and built environment data have to be done.

The ArcGIS connection

ArcGIS is quite good at separating data and functionality. This concept seems promising and is even required for applications like the one described in this paper. However, a strategy has to be developed to support 3D geometry and functionality. Not only tools for visualizing 3D scenes are wanted but the execution of truly "spatial" analysis

The fact ArcGIS makes available libraries of GIS functions for languages like Visual Basic makes it easier to incorporate GIS functions into bigger program modules.

The simulation software

Most of the available building simulation software is clearly meant for static simulations in the design phase of a building construction project. Working with some of the programs – and DEROB in particular – has exposed some of their weaknesses and showed potential for improvement:

- Most programs can only handle a very limited range of geometric building elements.

- For totally dynamic applications, a simulation interval of one hour may still be too long. Only few programs can calculate shorter time steps.
- Only some programs offer programming interfaces and allow the user to adapt the actual simulation processes.
- Another area where good interfaces are lacking is the import of building data. No common format exists and not many programs are able to read CAD data. Manually entering all the data is very time consuming and prone to error.

While the results of this project include a prototype of an interface program to get data from the database to the simulation software, the opposite direction – transferring the results of the simulation back to the central unit and interpreting them – is just as important and has not yet been dealt with.

OUTLOOK

The concept for a coupling of GIS and simulation programs bears good potential for optimizing both energy efficiency and user comfort in buildings. The combination of existing systems can produce better results by using simulation to assess the effects of control decisions and by taking into better account environmental influences.

The big challenge will be to successfully network all the subsystems integrated in a building control system and to program central control software “intelligent” enough to make good decisions.

Another problem is the vast amount of data needed to feed such a comprehensive system. All details of a building, its geometry, material properties, specifications of technical installations and so on have to be entered into the central database. This information will come in many different formats and will have to be integrated into one consistent data structure. Environmental data is not available in great detail everywhere and often comes at a high price and in proprietary formats. In order for it to be useful for the task at hand, additional information may have to be gathered and a strategy has to be developed to magage 3D data together with building information for spatial analysis in ArcGIS.

ACKNOWLEDGMENTS

We would like to thank Stephan Landtwing, Michael Müller and Caspar Honegger for their help in the project and for the preparation of the graphics for this paper.

APPENDIXES

END NOTES

REFERENCES

- Agnolotti, V., Giger, C., Intelligent Building's Cell Behaviour in a Building Organism; Building Community Connection and Experience Exchange via Geographic Information System, Proceedings of 3rd International Congress Arquitectura 3000, The Architecture of In-Difference, Barcelona, Spain, June 30th - July 4th 2004
- Agnolotti, V., Giger, C., Enhancing Intelligent Buildings' performances through a GIS central control, Proceedings of 7th AGILE Conference on Geographic Information Science, Heraklion, Greece, April 29th - May 1st 2004
- Asimakopoulos, D. N., et al., Energy and climate in the urban built environment, James & James, London, 2001.
- Boed, V., et al, Networking and integration of facilities automation systems, Boca Rato CTC Press, 2000.
- Clark, G., Metha, P., Artificial intelligence and networking in integrated building management systems. - In: Automation in Construction, vol. 6, pp. 481-498, 1997.
- Clements-Croome, T., What do we mean by intelligent buildings? - In: Automation in Construction, vol. 6, pp. 395-400, 1997.
- Daniels, K, at al., Bauen für die Zukunft, HL Technik AG, München, 2002.
- Daniels, K, The Technology of Ecological Building, Basel ,1997.

- Dimson, B., Principles and challenges of sustainable design and construction. - In *Industry and Environment*, vol. 19(2), 1996.
- Flourentzou F.; Roulet C.-A., Multicriteria analysis of IEQ in sustainable buildings outline of a methodology EPIC 2002 AIVC Conference Proceeding, Lyon ; October (2002).
- Gallo, C.: *Architettura ecosistemica*, Gangemi, Roma, 1998.
- Gassmann, O., Meixner, H., *Sensors in Intelligent Buildings*, Wiley-VCH Verlag, 2001.
- Gay J.-B.; Rittmeyer P., Impact environnemental des constructions - Comparaison de variante d'implantation CISBAT'97 - Lausanne ; octobre (1997)
- Guillemin A., Morel N., Experimental results of a self-adaptive integrated control system in buildings : a pilot study *Solar Energy*, Elsevier Science Ltd // 75(5), pp. 397-403 ; May (2002)
- Guillemin A.; Molteni S.; Morel N., Application of Genetic Algorithms to adapt an energy efficient blinds controller to the user wishes CISBAT 2001 - EPFL // pp.331-336 ; 3 - 4 octobre (2001).
- Hartkopf, V. et al., An integrated approach to design and engineering of intelligent buildings – The Intelligent Workplace at Carnegie Mellon University. - In: *Automation in Construction*, vol. 6, pp. 401-415, 1997.
- Hartkopf, V., Loftness, V., Global relevance of total building performance. – In: *Automation in Construction*, vol. 8, pp. 377-393, 1999.
- Hendriks, L., Van der Linden, K., Building envelopes are part of a whole : reconsidering traditional approaches. - In: *Building and Environment*, vol. 38, pp. 308-318, 2003.
- Kua, H. W. et al., Demonstration intelligent building – a methodology for the promotion of total sustainability in the built environment. – In: *Building and Environment*, vol. 37, pp. 231-240, 2002.
- Mertens, E., *Das bioklima städtischen Baustrukturen : eine Analyse- und Bewertungsmethode für den gesundheitsorientierten Städtebau*, Köhler, Berlin, 1997.
- Oestreicher Y., Bauer M., Scartezzini J.-L., Accounting free gains in a non residential building by means of an optimal stochastic controller *Energy and Buildings* // Vol. 24(3) ; (1996).
- Pfund, M., Zanini, M., "3D-GIS", ETH Zürich, 2000.
- Roulet C.-A., Indoor environment quality in buildings and its impact on outdoor environment *Energy & Buildings*, Elsevier // 33, pp. 183-191 ; January (2001).
- Roulet C.-A., Solar Energy and Global Heat Balance of a City *Solar Energy*, Elsevier // 70/3, pp. 255-261 ; Mars (2001).
- Scartezzini J.-L.; Montavon M. (EPFL); Compagnon R. (HES-SO), Computer Evaluation of the Solar Energy Potential in an Urban Environment Proceeding of EuroSun 2002 Congress, Paper No 191, CD-ROM ; June (2002).
- Smith, S., The integration of communication networks in the intelligent building. – In: *Automation in Construction*, vol. 6, pp. 511-527, 1997.

AUTHORS INFORMATION

Valeria Agnolotti, Architect, Swiss Federal Institute of Technology ETH Zürich, Hönggerberg IGP HIL E 14.3, tel. +41 44 633 46 52, fax +41 44 633 11 01, e-mail agnolotti@geod.baug.ethz.ch, web http://www.geoit.ethz.ch/staff/valeria_en.html

Christine Giger, Prof. Dr., Swiss Federal Institute of Technology ETH Zürich, Hönggerberg IGP HIL D 45.2, tel. +41 44 633 30 51, fax +41 44 633 11 01, e-mail giger@geod.baug.ethz.ch, web http://www.geoit.ethz.ch/staff/christine_en.html