D3.3.1 - Use cases and specification of building monitoring system

WP3.3 - System monitoring for buildings

WP3 State Estimation and System Monitoring

MODRIO (11004)

Version 1.0

Date 14/02/2014

Authors

Per Sahlin EQUA Simulation AB.

Executive summary

WP3.3 in MODRIO aims at bridging the gap between the simulated model and the physical building. The end result of WP3.3 would be the derivation of a model that accurately estimates, as close as possible, the actual state of the system. To this end, WP3.3 calls for an approach which accounts for both parametric and structural inadequacies inherent in the original system mode in a generic fashion. This requires calibrating the initial system model through clustering several key parameters into a single normalized meta-parameter on a zone level. The Key Parameter Modifiers, being meta-parameters, attain normalized values of 1 and represent several different individual physical parameters that collectively contribute to the same physical process. Furthermore to ensure real-time visualization of the physical building through a dedicated GUI, we employ a direct State Controller to provide error correction and hence update the original system model to follow the physical behavior of the building. Moreover a thorough investigation of the Control Adjustment signals provided by the State controller facilitates the interpretation of the resulting optimized configuration of the meta-parameters by shedding light on the potential fault sources and disturbances causing the mismatch of the original system model. The Control adjustment signals should normally be near zero and the statistic of these signals is a key short term diagnostic mechanism that indicates a physical problem in the building (or in the improved-model related system). The purpose of the periodic learning events is to bring the Control adjustment signals to zero for the subsequent period. Hence the time evolution of the Key parameter modifiers is the main diagnostic tool for monitoring the long term health of the building and the relevance of the improved-model. If the improved-model perfectly reflects the real building, the value of all Normalized key parameters will be 1 after a tuning event. Moreover seldom re-calibration events are also negotiated, though offline, to account for major retrofitting and other functional changes that might occur to the building throughout its life cycle. Similarly retrofitting analysis is conducted in an offline fashion to allow the operator freedom of experimenting with new concepts and energy conservation methods without interrupting the ongoing calibration and control processes.

Table of Contents

1. Introduction

WP3.3 in MODRIO aims at bridging the gap between the simulated model and the physical building. The end result of WP3.3 will be the derivation of a model that accurately estimates the actual state of the system. To this end, WP3.3 calls for the development of a calibration algorithm that can handle an initial as-built simulation model of a building and fine tune it towards its current state.

The proposed approach relies on a simulation model of the building that has been developed in the design phase. Margins are small in this industry, and the cost of developing a tailored model for the operation phase would be prohibitive. In a general simulation context, such a design model is severely over-parameterized. Parameters correspond to physical measures of individual building-parts and the total number of parameters for an individual room counts in the hundreds. Meanwhile, the simplest possible simulation models of a room that are capable of adequately reproducing measured signals have an order of magnitude less parameters.

The proposed system targets an expert operator. It is envisioned that each such operator manages a large number of buildings from a remote location. Separate user interfaces may in subsequent work also be developed for the day-to-day building maintenance personnel, tenants and other stakeholders such as owners.

Furthermore, the present work is strictly aiming at the development and presentation of a model that is in real-time kept in tune with the physical building. Further use of this model for Fault Detection and Diagnosis or optimal control is not presently encompassed.

2. Vision and Architecture

At the heart of this approach is the notion of the meta-parameter, whereby a list of Key Parameter Modifiers is selected to describe the main physical actors of any single zone or system i.e. thermal mass, heating/cooling capacity, moisture storage, etc… The Key Parameter Modifiers, being meta-parameters, attain normalized values of 1 and represent several different individual physical parameters that collectively contribute to the same physical process. In this manner, by opting a compromise between a black box approach having no physical interpretation and taking head on the intractable optimization of numerous physical parameters any one building possesses, we obtain a generic approach of self-tuning a simulated building model through optimizing the configuration of the meat-parameters whilst attaining a translatable physical meaning of the end tuned result; therefore making way for fault detection, retrofitting analysis, and other meaningful interpretations of the final resulting system model.

Furthermore to ensure real-time visualization of the physical building through a dedicated GUI, we employ a direct State Controller to provide error correction and hence update the original system model to follow the physical behavior of the building. This provides us with two salient advantages; the first being the capability of manual fault detection through the direct visualization of an accurate current state of the physical building through the calibrated system model. Secondly a thorough investigation of the Control Adjustment signals provided by the State controller facilitates the interpretation of the resulting optimized configuration of the meta-parameters by shedding light on the potential fault sources and disturbances causing the mismatch of the original system model.

To better explain this approach in detail, we will examine the simulation phases that are encountered throughout a building life cycle and then introduce each component of the system model methodology to conclude this vision of achieving the WP3.3 goal of bridging the gap between simulation and reality.

2.1. Simulator Phases of the Building Life Cycle

In the WP3.3 context, we are working on already existing buildings that are not physically changed in conjunction with the exercise. However, another important scenario is when a building is significantly changed or new in conjunction with the application of the methodology. We will therefore include the building design phase in the description of the different stages of evolution of the system model.

Phase	*resulting model*	*Comment*
Design	A model that expresses Design Intent.	This design model is important as a baseline for contracted building performance. It should fulfil all requirements that are set to the building.
Commissioning	A model, which is adjusted to fit the As Built (and accepted) state of the building.	The process of going from design model to the as built model will involve changes to both the building itself. For already existing buildings that are not physically changed, i.e. the WP3.3 case, the as built system model is developed separately and is then calibrated against measured data to reach final tuned system model status.
Monitoring Launch	A real-time improved-model that has been prepared for use in the monitoring phase.	See list of changes wrt. As built model below.
Operation	1) States (variables) of the system model are continually adjusted to fit measured signals 2) Key Parameter Modifiers are at regular intervals, e.g. weekly, identified to fit measured signals.	The main results here are: 1. Diagnostic information about the health of the building. 2. A platform for exploring more optimal ways of running the building. 3. A platform to be used for more advanced controls that involve the system model itself.
Re-commissioning	A new version of the system model that reflects the new state of the	The re-commissioning can be triggered by physical (or control system) changes to the building, or a need to radically improve

	building.	system model fidelity.

2.2. Additions to the Improved-Model wrt the As Built-Model

Key Parameter Modifiers (KPMs) are introduced. A KPM is a meta-parameter specifically introduced for the purpose of model tuning. KPMs are normalized, i.e. if the model is in perfect tuning with measured data, the value of a KPM is 1. The concept of KPMs is further discussed below.

A mapping is created between control settings of the original system model and those of the building BMS. Some of the settings are parameters (fixed in time) and others are time series. If the operator changes a setting in the BMS, the corresponding set point change should automatically be done in the system model.

System model output signals are exported in order to match available physical sensor signals in the real building.

Real time input signals are created in the system model that allow a so called State controller (explained below) to intervene in local control loops of the building.

All impact from occupants in the original system model is replaced with inputs from an Occupancy model.

The automatic tuning is set at specific time intervals, where a snapshot of the current system model will be saved and stored for the utilization of automatic tuning through Key Parameter Modifier optimization. Hence this operation will not disrupt the real-time process of continuous state control of the system model on-line model. Moreover, once the optimization algorithm has concluded, a hot-start technique will be implemented to integrate the newly configured Key Parameter Modifiers into the real-time system model seamlessly and with minimal data loss. This approach will also be adopted to allow for off-line manual experimentation of the building controls and set points without the disruption of the real-time process. If the manually tuned system model scenario were to be adopted, then the integration into the current real-time model will also be through a hot-start technique for switching from the current model values to the newly tuned ones.

2.3. Improved-Model Related System

The improved-model (Figure 1) is simulated in real time along with the Real building. The system model receives signals from the Environment that should be equivalent to those affecting the Real building. Most obvious among these are data from a local weather station, with a complete set of weather data measures, including normal solar radiation, and wind speed and direction. Other possible external signals in this category could be real-time energy price information or utility load shedding signals.

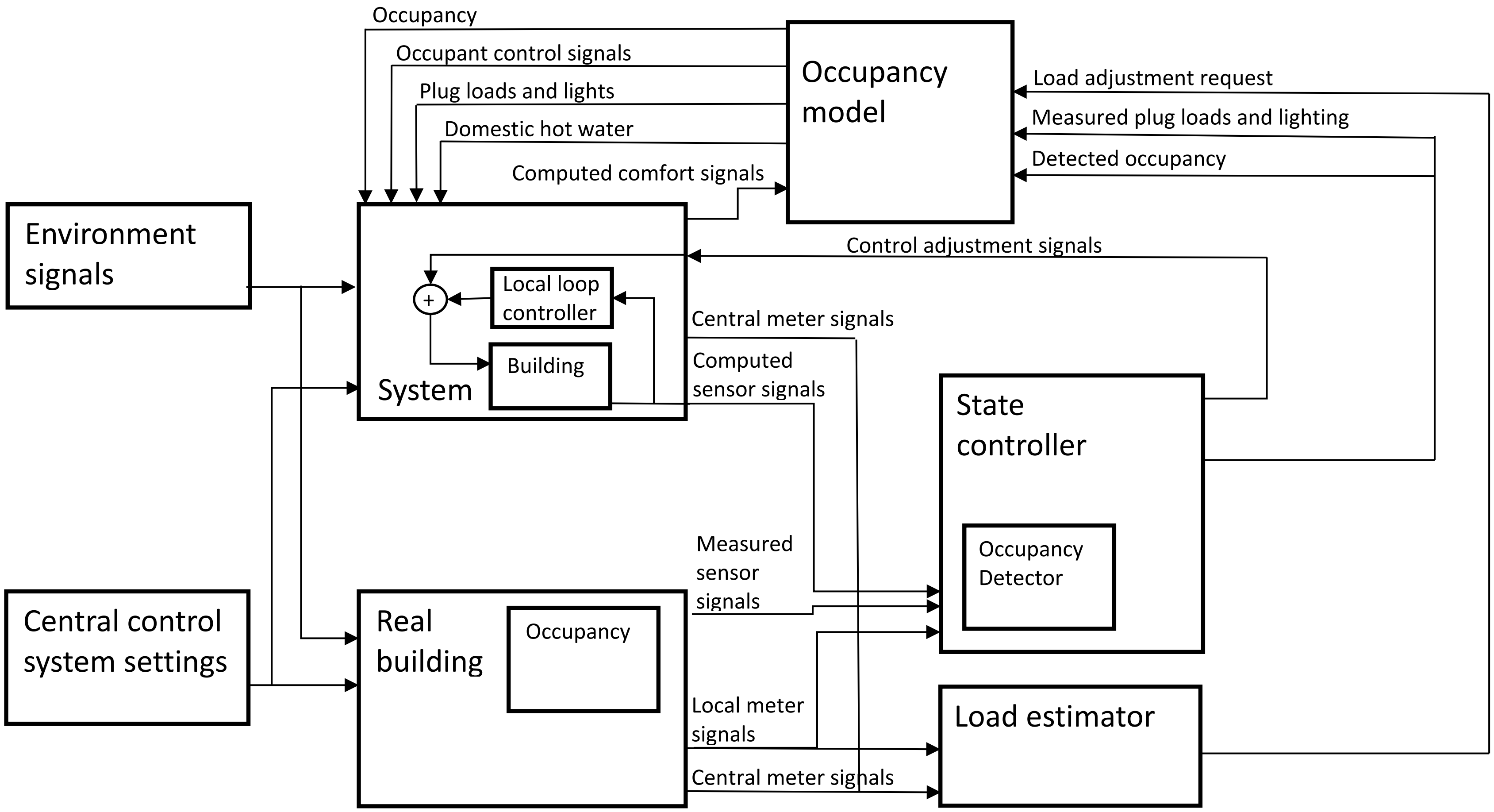

Figure 1. The improved-model related real-time system

2.4. Central Control System

The improved-model also receives settings and signals from the Central control system (BMS) of the building. All signals and settings that affect the operation of the building that are known to the BMS should be made available to the improved-model also settings that are under the authority of occupants. (Occupant settings that are unknown to the BMS are treated by the Occupancy model). The control setting signals sent to the improved-model must normally be mapped and transformed by a Central control signal transformer in order to match the signals that are available in the improved-model to cater to similar functions. This transformation module is not shown in Figure 1.

2.5. State Controller

A State controller receives all sensed signals from the Real building as well as equivalent computed signals from the improved-model. A Sensor transformation module will filter, map and transform real sensor signals in order to match the definitions of equivalent measures that are obtained from the improved-model. This module is not shown in Figure 1.

Special among sensed signals are Local meter signals, i.e. electricity use measures that reflect local usage for mains plugs and lights and that can be directly attributable to a given zone in the improved-model.

The purpose of the State controller is to compute required control action that, when allowed to act on the improved-model, will minimize the difference between sensed signals of the Real building and the improved-model.

A control action, called a Control adjustment signal, that will lessen the difference between a real measured and the corresponding computed signal is sent to the improved-model. This signal will interfere with the local control loop that controls the relevant signal and attempt to increase the output of the controlled device, e.g. mass flow through the radiator in a heating situation.

When direct measures of lighting and/or plug load power are available for a zone (Local meter signals), these are fed through the State controller to the Occupancy model (which in turn feeds them to the improved-model , overriding any occupant plug load and lighting operation models that may be active in the Occupancy model).

A special module of the State controller is the Occupancy detector. Its purpose is to detect possible occupancy. The method of doing this depends on available sensor signals. Lacking direct occupancy detection signals, the occupancy detector will be sensitive to other traces of occupancy, such as raised levels of temperature, CO_2 and humidity, or other events that would implicate an occupant being present such as a light or plug load switching event, a probable window opening etc.

2.6. Occupancy Model

The Occupancy model tries to mimic the behavior of real zone occupants, in terms of (1) actual presence (Occupancy), (2) any control action that the occupant is authorized to take but that is unknown by the BMS (e.g. change self-acting radiator valve settings, draw blinds, open window etc.), (3) signals to mimic real occupancy plug load and lighting action, and (4) signals for domestic hot water usage.

Occupant control action is modelled based on occupant presence and Computed comfort signals that are read directly from the improved-model.

When the Occupancy detector module of the State controller is unable to advice on the level of zone occupancy (e.g. if there are no sensors in place with relevance to occupancy), "-1" is sent in terms of Detected occupancy, in which case the internal stochastic presence prediction models of the Occupancy model will estimate zone occupancy.

Any stochastic signal generator in the Occupancy model must have the property of reproducing the exact same signal in two repeated evaluations with the same input, otherwise the optimization process for physical KMPs will be seriously impaired.

Various types of black and white box models can be tried for the Occupancy model. The training of these models is separated from the tuning process of the physical KMPs of the improved-model. However, having some parameters of the Occupancy model visible and given a "physical" interpretation could be a great advantage to assist the operation of the building by providing diagnostic information about occupant habits, preferences and possible wasteful behavior (such as regulating temperature by window opening rather than radiator settings in the heating season).

2.7. Load Estimator

In the common situation when Local meter signals are unavailable or incomplete, the Load estimator module attempts to reconcile improved-model plug loads and lighting power with the physical readings of central electricity meters. This is done by sending a load adjustment requests to the relevant Occupancy models. The occupancy models act on this request by increasing or decreasing the plug loads and lighting that is under its authority.

2.8. Regular Tuning of the Improved-Model

The Control adjustment signals should normally be near zero and the statistics of these signals is the key short term diagnostic mechanism that indicates a physical problem in the building (or in the improved-model related system). The purpose of the regular tuning events is to bring the Control adjustment signals to zero for the subsequent period.

This happens at regular intervals, e.g. weekly. The improved-model and the Occupancy model are tuned in order to learn changes in building physical parameters and in usage patterns. This is an off-line optimization exercise based on recorded signals, where the Key parameter modifiers are automatically identified in order to minimize the residual of needed Control adjustment signals since the previous tuning event.

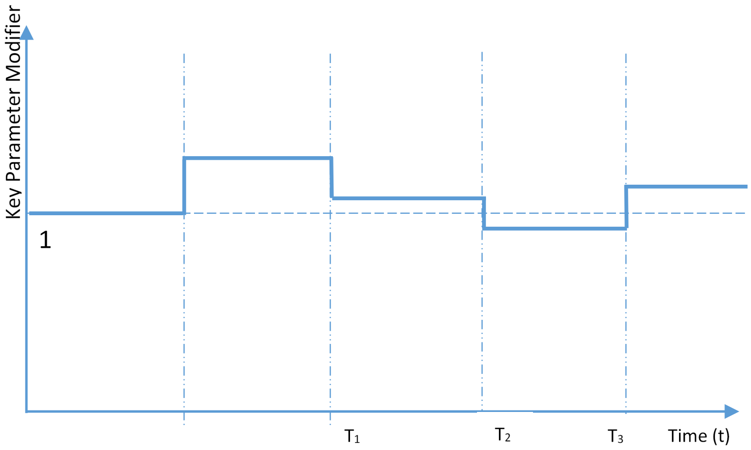

Figure 1 Tuning the Key Parameter Modifiers will occur at discrete time intervals, whereas they will start off normalized at 1 for a well calibrated model before drifting away.

The time evolution of the Key parameter modifiers is the main diagnostic tool for monitoring the long term health of the building and the relevance of the improved-model. If the improved-model perfectly reflects the real building, the value of all Normalized key parameters will be 1 after a tuning event.

2.9. Role and Selection of Key Parameter Modifiers

Among all the time-invariant, measurable properties of a building, that we call parameters, we can create classifications based on physical subsystems and physical processes. Subsystems are things like rooms (zones) or individual technical service systems such as air handling units. In each of these subsystems, a number of distinguishable physical processes are at play.

If we try to construct the simplest possible physical model of a subsystem and apply parameter identification techniques to compute parameter values based on measurement data, we will develop a sense for the minimum number of fundamental physical processes and the number of physical parameters required to represent them. For a zone, typical such models have less than a dozen individual parameters. These can be identified by less than half a dozen sensor signals.

Typical zone models that are used in the design phase, on the other hand, that have to perform without access to any measurements, normally have hundreds of parameters. The ambition in this phase is to make the best possible model with the information at hand and this leads to models that have a great deal of detail, e.g. describing the properties of each material layer of the walls, the precise areas of individual windows, radiators, wall parts etc. There is a gross mismatch in number of parameters between the a-priori design phase model and the model obtained by parameter identification from measurements.

With a reasonable number of sensors, it will be impossible to identify the hundreds of individual parameters of the design phase model. This is where the concept of Key Parameter Modifiers come in. The number of Key Parameter Modifiers of a zone model should be of the same order as the number of parameters of the simplest possible model for the zone, i.e. around a dozen. However, each Key Parameter Modifier may be bound to a large number of physical parameters. When the KMP is changed by some percentage, all the physical parameters that are bound to that KMP will change with a corresponding percentage. For example, when the KMP for 'envelope conductance' is changed, thermal bridges will be affected, as well as the conductivity of all wall materials, the U-value of each window etc. The KMPs are used to leverage our knowledge of how individual parameters will influence specific physical processes. For a zone in a WP3.3 building, the following KMPs representing fundamental physical processes are proposed:

- Envelope conduction
- Thermal mass
- Envelope leakage
- Solar aperture
- Moisture storage
- Heating capacity
- Cooling capacity
- Mechanical ventilation capacity
- Artificial lighting capacity

This process can be applied for different building systems, such as the air handling unit, boilers and chillers, and distribution networks, ending up with a comprehensive list of all Key Parameter Modifiers characterized by the whole building.

In summary, Key Parameter Modifiers describe and influence several other physical parameter in any single system. For instance Envelope Conduction, is one such Key Parameter Modifier that is normalized to 1 and influence the U*A values of the building zone. Similarly the KPM Solar aperture influences the individual shading coefficients of all windows and this single normalized meta-parameter summarizes the overall combined effect of all the sub-layered physical parameters.

3. Use Case

A building is not a physics project, however that does not mean we can't conduct experiments on it every now and then, especially before it has been commissioned. Requiring a strict thermal comfort level often shies the building operator and owner from testing the set points and limitations of their buildings. Furthermore the time response of the building, often in the order of hours, limits us from varying the physical behavior of a building in the case of short or intermittent occupant absence. However, to better understand an existing system and properly uncover its underlying structure and principles, some form of system identification through excitation is required. This follows that observing the normal behavior of a building in response to well behaved inputs and excitations is in adequate to create an accurate model. For instance to properly model the structure and form of occupants' behavior, excitation of the building systems when the occupant is absent is essential. Otherwise an occupant's daily activities will undoubtedly mask the smaller frequencies exhibited by the building's own systems; this can be severe in the case of smaller buildings where occupant behavior plays a major role in a building's behavior.

Hence whenever the occupancy presence model permits, and without venturing much outside the thermal comfort zone, scheduled excitations such as system shutdowns and set point variation can be conducted to fine tune the simulation model for accurate future prediction values. An accurate simulation model in turn yields more reliable Fault Detection and

Diagnosis signals, which can be manually inspected through the real-time visualization of the building's performance and behavior. Having an accurate baseline to base your upper and lower limits of performance for the building's systems, allows for easy identification of outliers and irregular patterns. Moreover a representative simulation model not only can be used for FDD purposes, but also for identifying energy saving measures.

An example of such an excitation exercise in the commissioning phase of the building can be described as follows:

The building is prepared for the measurement sequence in the following ways:

1. All radiator thermostats (or valves) are set in the fully open position.

2. Any user-controllable outdoor air vents are set in their design positions.

3. All shading devices are set in their fully open position.

The test procedure is the following:

1. Switch on fans and turn off any heat supply, allowing building to free-float as long as is practically possible with respect to available time and freezing risk. Depending on the building time constant, free floating for up to two weeks may be necessary.

2. Switch fans off and turn on heating supply to full power. Allow the building to heat up completely.

3. Switch fans back on while leaving full heating power, allowing the building to stabilize at the new state.

4. Return radiator thermostats to design set points.

At regular intervals, e.g. monthly, the model is re-calibrated in the occupied building. This is done by temporarily shutting off heat supply and, in all combinations, ventilation fans for periods that are sufficiently long to obtain reliable signals, yet short enough not to impose unacceptable discomfort. These calibrations are preferably done during the night, and can be part of a regular night-setback scheme.

Access to such a calibrated real-time model opens several possibilities, among them are:

1. Objective and accurate assessment of the thermal performance of a delivered, unoccupied building in the commissioning phase. Faulty building envelopes or HVAC service systems can immediately be exposed.

2. On-line monitoring of actual heat-flows for tenant billing systems. Present systems disregard major heat contributions from neighboring apartments. This makes present direct supply heat measurement systems fundamentally unfair and this, in turn, is a major preventive factor for broad introduction of tenant billing.

3. Computation of optimal set-points with respect to forecasted weather and building usage.

4. Control of operative temperature rather than air temperature. Human thermal comfort is not primarily related to air temperature but to a mixture of air and radiation temperatures, often quantified as operative temperature. In spite of this, present control systems maintain air rather than operative temperature. This is because operative temperature is difficult and expensive to measure directly.

5. Detection of faults in a building and aberrations from contracted building usage, e.g. windows left permanently open, tampering with HVAC systems, too many occupants.

Moreover through running several handpicked Energy Conservation Measures via a What-If-Analysis the user will be able to reflect upon the effects of different scenarios on a building's energy performance and thermal comfort variation, in order to evaluate properly each ECM in accordance to cost and other selection criteria. Nonetheless, not all ECMs require manual What-If-Analysis. Depicting occupancy presence, occupancy behavior, and weather patterns, allow for the development of a strategic operation scheme for the building systems, optimizing energy performance whilst maintaining comfort integrity and minimizing operational costs. This optimization approach which can be achieved through a Model based Predictive Controller, allows for automated energy saving measures to take place with minimum input from the user, requiring only the definition of the cost function or minimization target, as well as requiring the outlining of the constraint functions or prohibitive limitations of the program to be set by the user.

4. Graphical User Interface

The targeted user is the building expert operator, and he/she requires not only a comprehensive outlook of the building performance but also a more detailed and clear insight to each separate building system component. For the expert operator to be able to properly assess the condition of the building, and be able to identify opportunities for improvement and increasing operation efficiency, a realistic and precise virtual representation of the physical building through component localization mapping must be sought for. This translates to the operator being able to pin point local building equipment, piping and duct installations, structural and mechanical components through the virtual environment presented by the GUI. This alleviates the inherent difficulty presented in locating and identifying physical building components in real life situations.

The ambition has been to develop a system where visualization of the geometric model as well as visualization of simulation results and online measurements can be performed in a single, fully interactive, 3D environment. This 3D environment is totally integrated with IDA Indoor Climate and Energy (IDA ICE). The tool we have chosen is The Visualization Toolkit (VTK), which is a popular tool for interactive 3D computer graphics and visualization. VTK is very well suited for visualization of computational results, but not so well suited for geometric modeling. However, VTK is written in the programming language C++, and the source code is freely available, so the geometric modeling capability has with some effort been implemented.

4.1. Building

The 3D visualization environment is integrated with IDA ICE as a new tab of the system window, the 3D plan tab, Figure 3.1.

The facades of the building are shown as semi-transparent polygons, Figure 3.1. An object in the 3D plan is selected by clicking on it. It is then shown in red. When a building section containing a number of facades is selected, its properties are shown in the sidebar. Some properties can be edited from there.

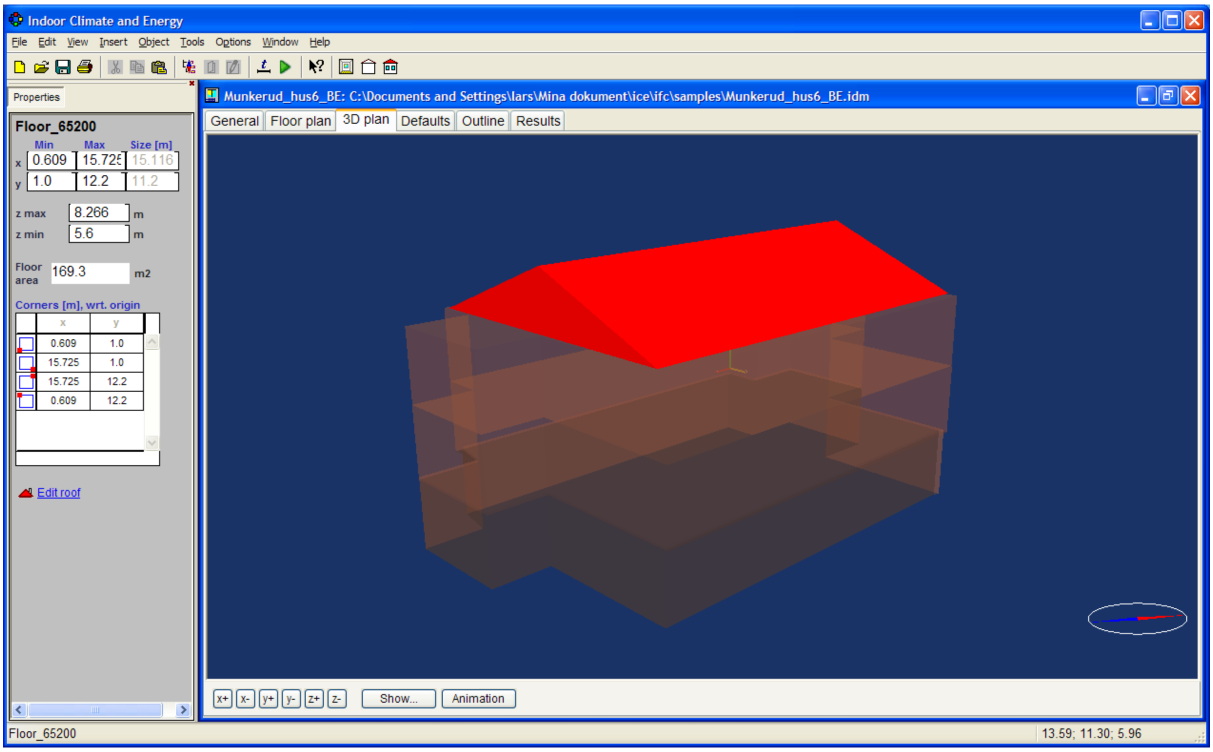

Figure 3.1 Building visualized with semi-transparent polygons

By selecting Edit roof from the sidebar a 2D roof editor is opened, Figure 3.2. From there very complex roof shapes can be created.

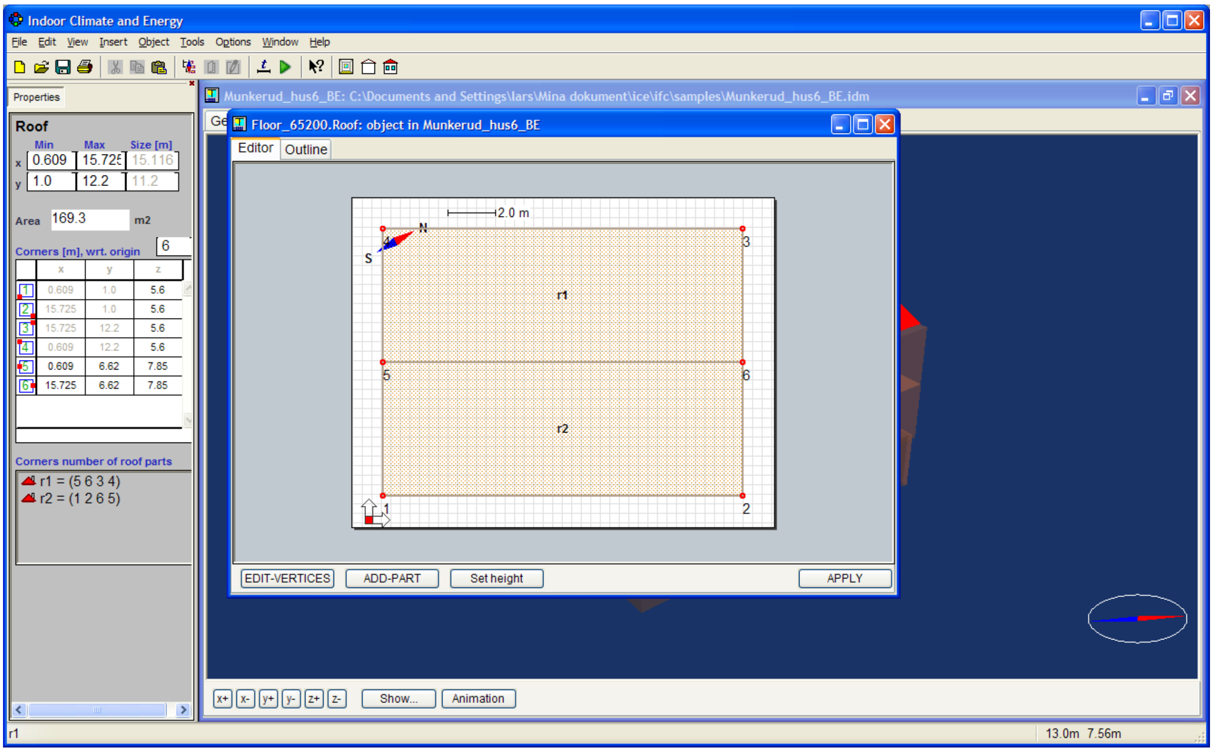

Figure 3.2 2D roof editor

4.2. Coordinate system and compass

The coordinate system of the building is shown in the 3D plan window and a compass, with its red arrow always pointing to north, is displayed in the bottom right corner of the 3D plan window, Figure 3.3.

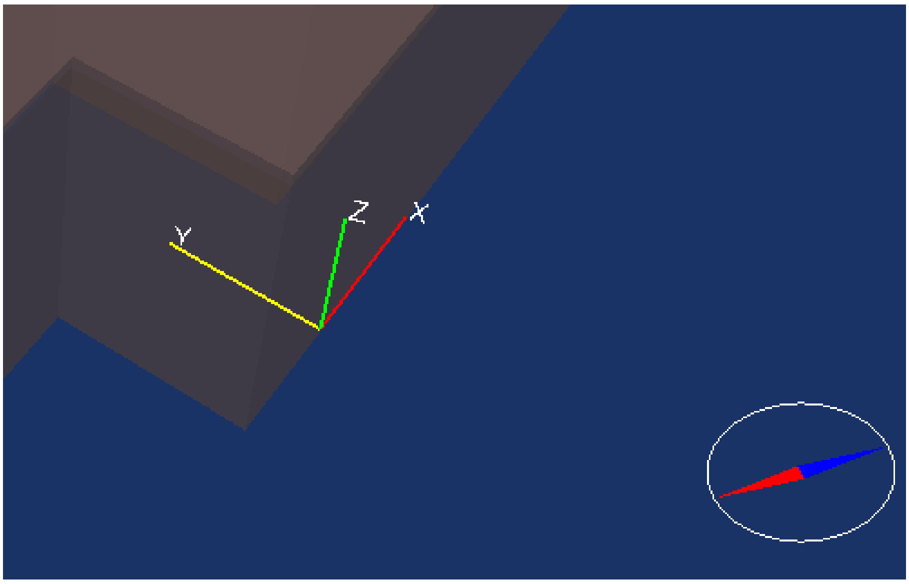

Figure 3.3 Coordinate system and compass

4.3. Navigation in the 3D plan window

The 3D model can be interacted with in real-time. The focal point is a point in the center of the 3D window, towards which the view is always oriented. The left mouse button rotates the model around the focal point while the model always is oriented so that the positive z-axis is up. The middle mouse button (or both the left and the right mouse buttons on a two button mouse) moves the model up, down, right or left while the focal point is set to a point in the center of the 3D window. The right mouse button zooms in and out on the focal point.

When Set focus is chosen from the right mouse button menu, or the hot key 'F' is pressed, the focal point is set to the point under the cursor.

To see the entire model, Zoom extents is chosen from the right mouse button menu, or the hot key 'R' is pressed.

When the system is saved, the last viewpoint (view position and focal point) is saved, so that the 3D model is viewed from the same position when the 3D window is opened the next time.

4.4. Zones

The floor, ceiling and walls of a zone are visualized as layered polygons where each layer corresponds to a specific material in the wall construction, Figure 3.4. Windows and openings cut out holes in the polygons. When a wall of a zone is selected, the geometric properties of the zone are shown in the sidebar.

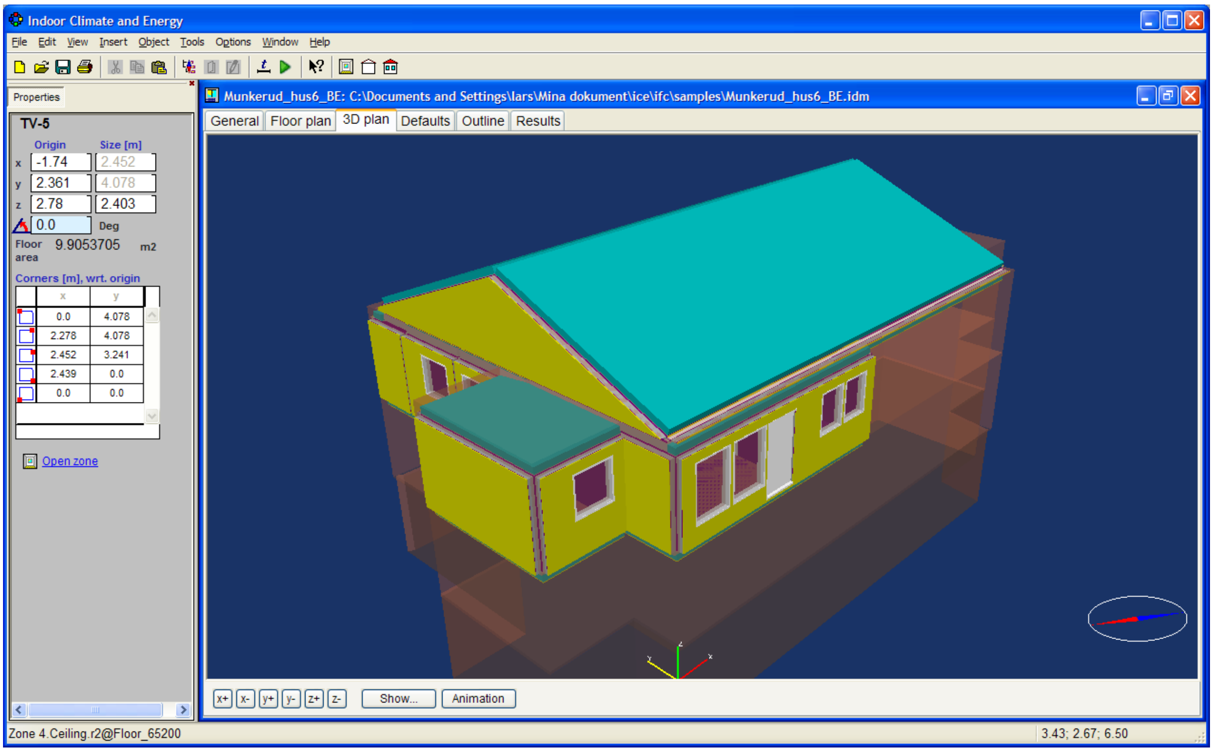

Figure 3.4 Zone walls visualized with layered polygons

4.5. Cutting the model

The 3D model can be cut in all directions with the buttons x+, x-, y+, y-, z+ and z-, Figure 3.5 and Figure 3.6. Pressing the Ctrl-button makes it possible to move the cutting plane.

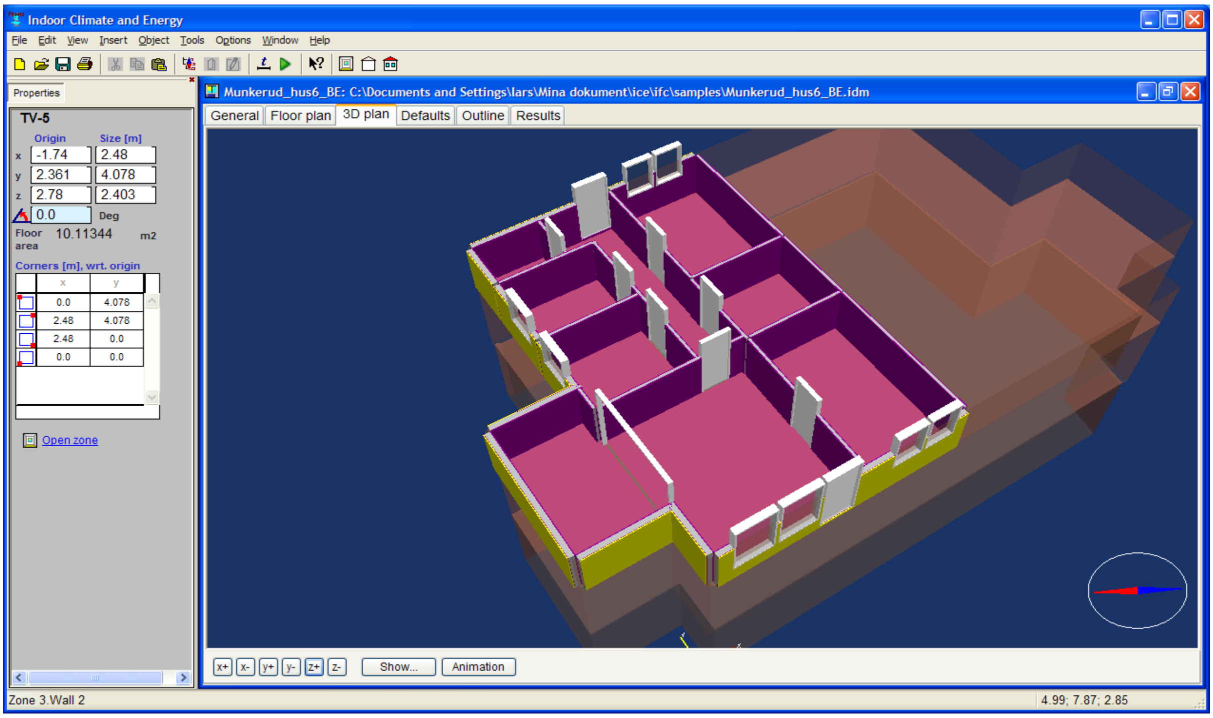

Figure 3.5 z+ cut of the 3D model

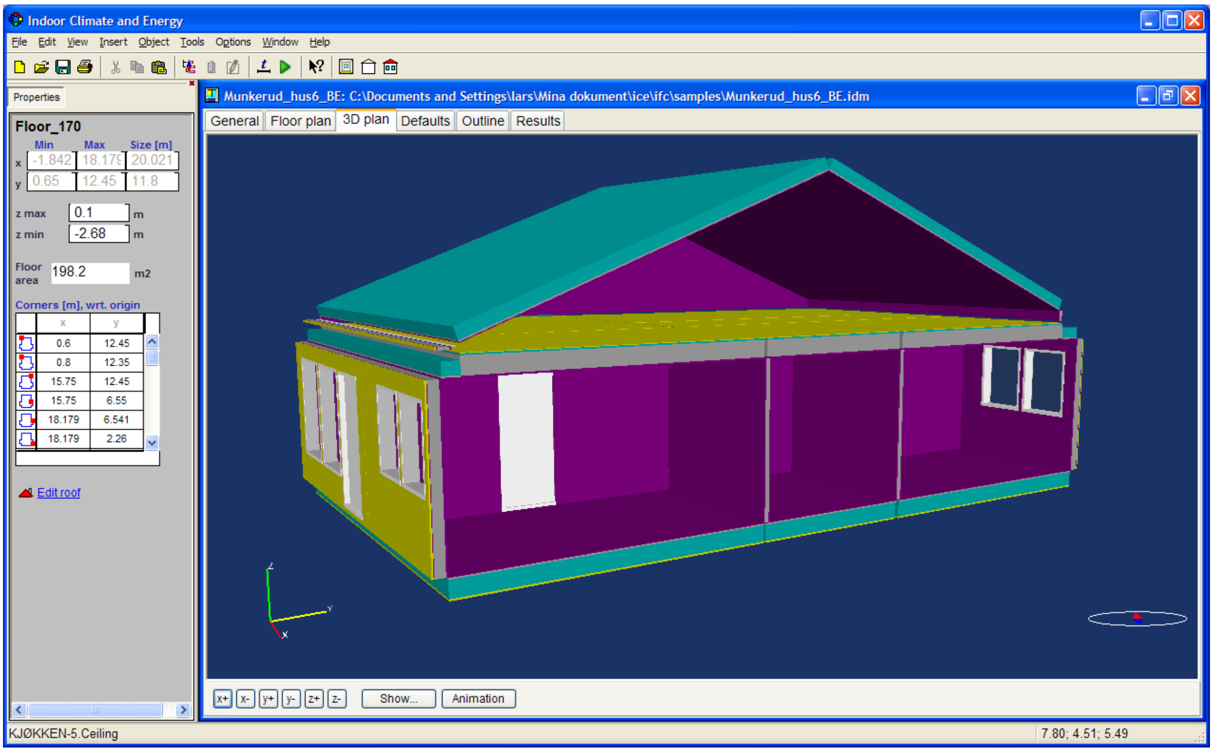

Figure 3.6 x+ cut of the 3D model

4.6. Controlling visibility

Visualization of objects in the 3D plan window can be turned on/off from the Show objects of type dialog, Figure 3.7. The Show objects of type dialog is opened from the right mouse button menu.

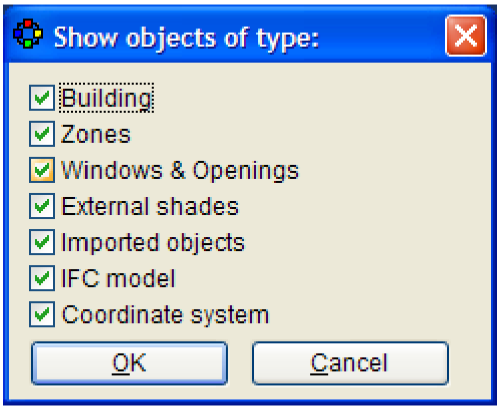

Figure 3.7 Show objects of type dialog

The model can be viewed in wire frame mode by selecting Wire frame from the right mouse button menu, Figure 3.8.

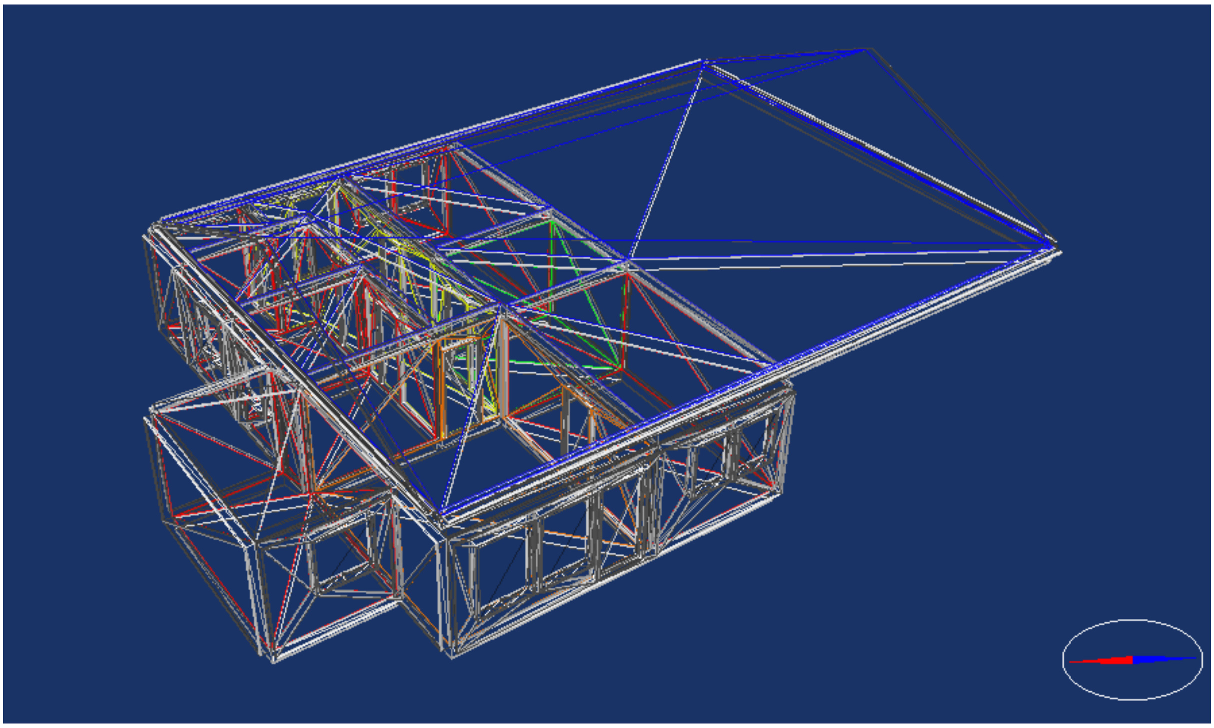

Figure 3.8 3D model in wire frame mode

The background color of the 3D plan window can be changed by choosing Background color from the right mouse button menu.

4.7. Windows and openings

Double-clicking on a wall will open a 2D wall editor. There objects like openings and windows can be added and edited, Figure 3.9.

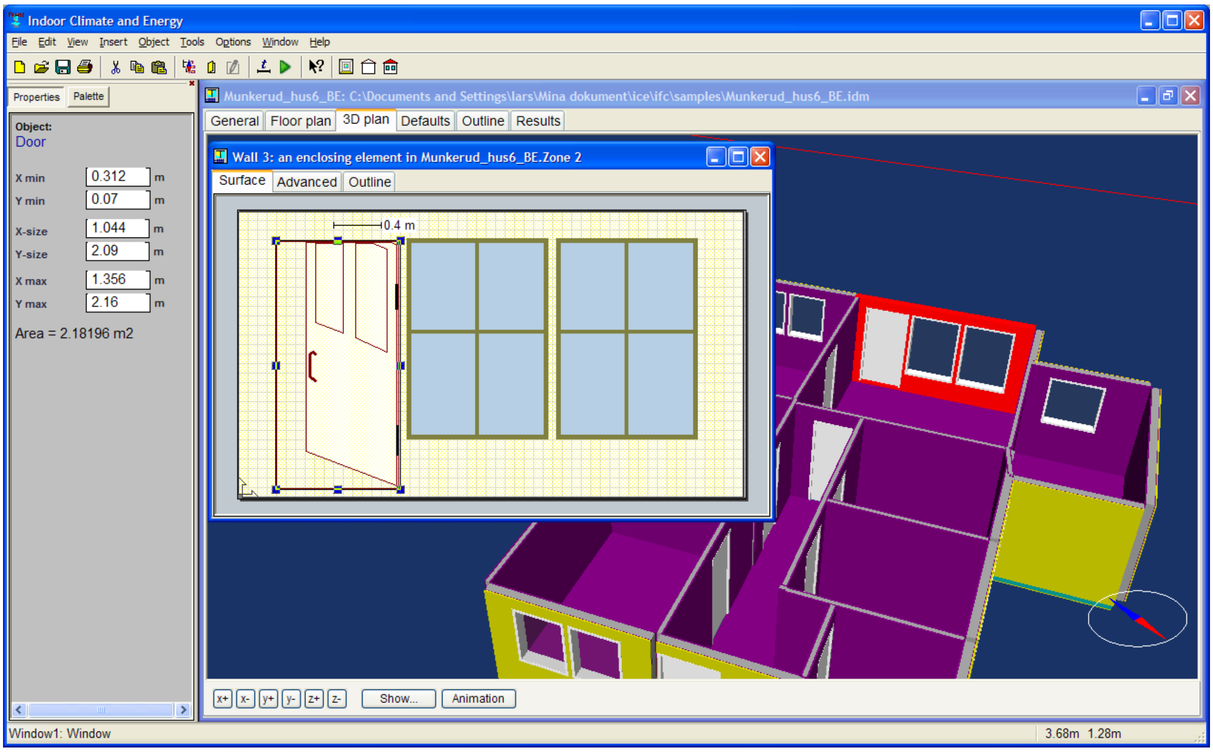

Figure 3.9 2D wall editor

The 3D model of an opening is different if it is sometimes open, Figure 3.10, always closed Figure 3.11, or is an opening without door, Figure 3.12.

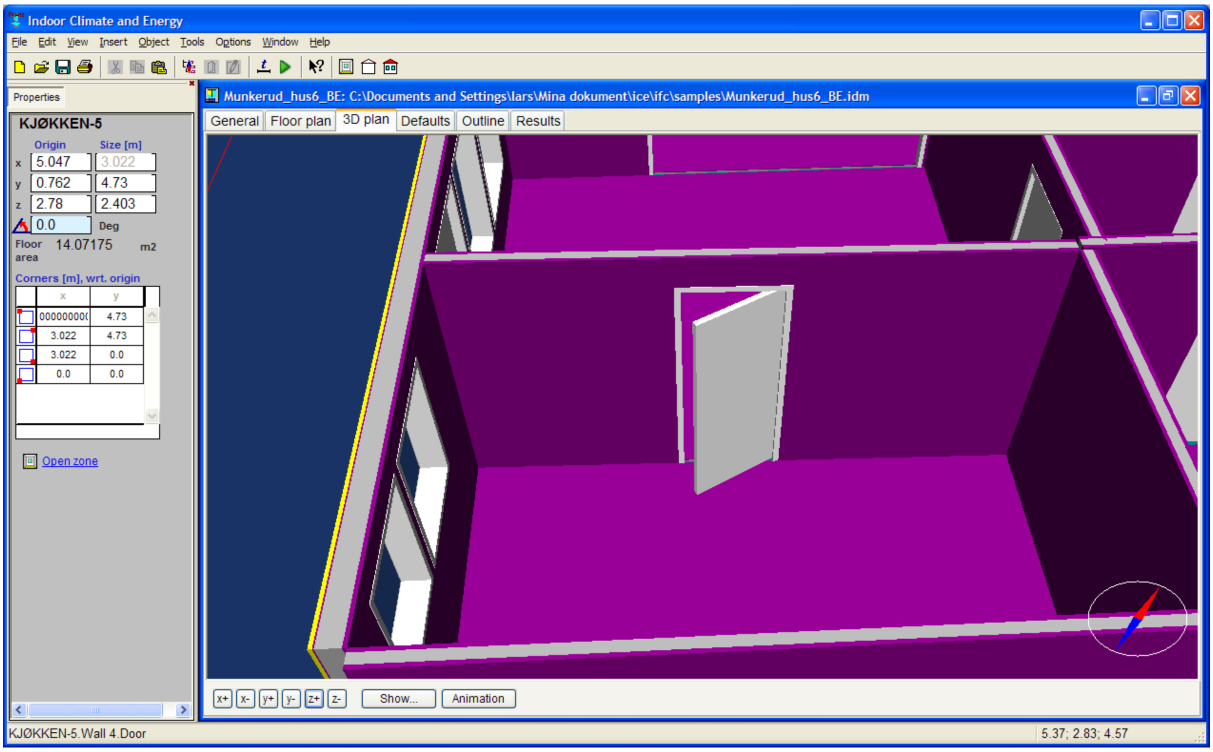

Figure 3.10 3D visualization of door that is sometimes open

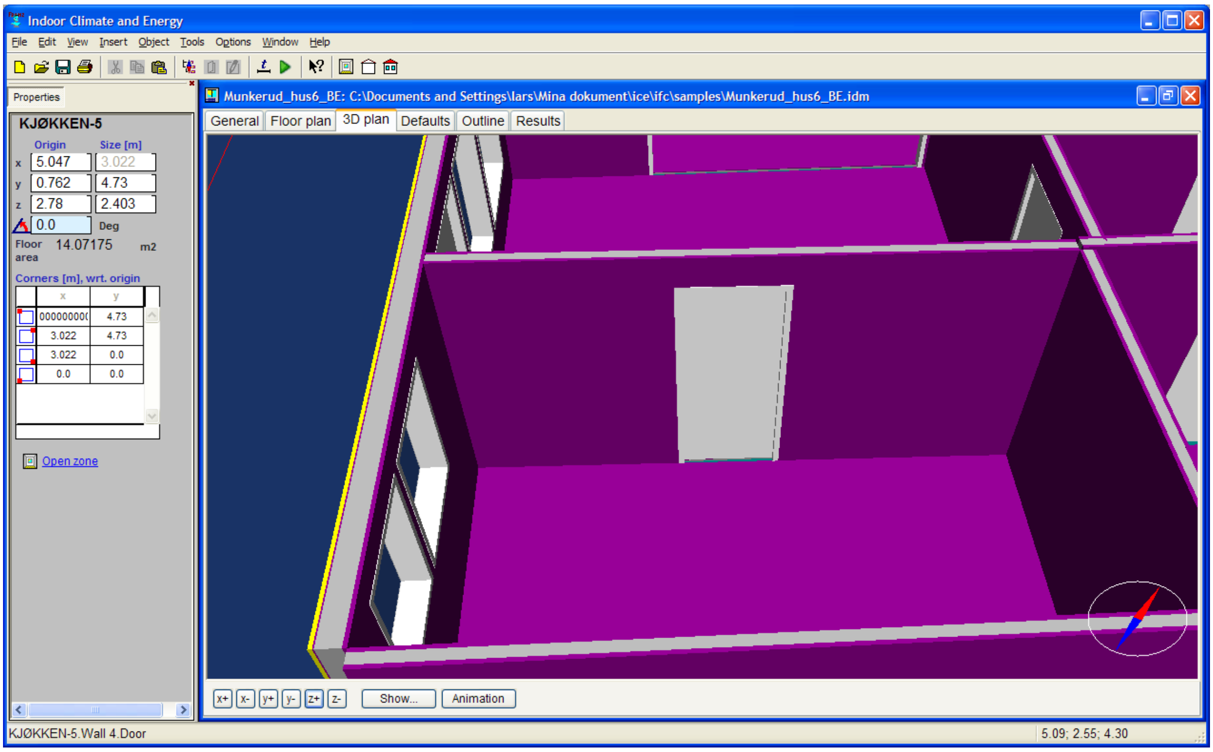

Figure 3.11 3D visualization of door that is always closed

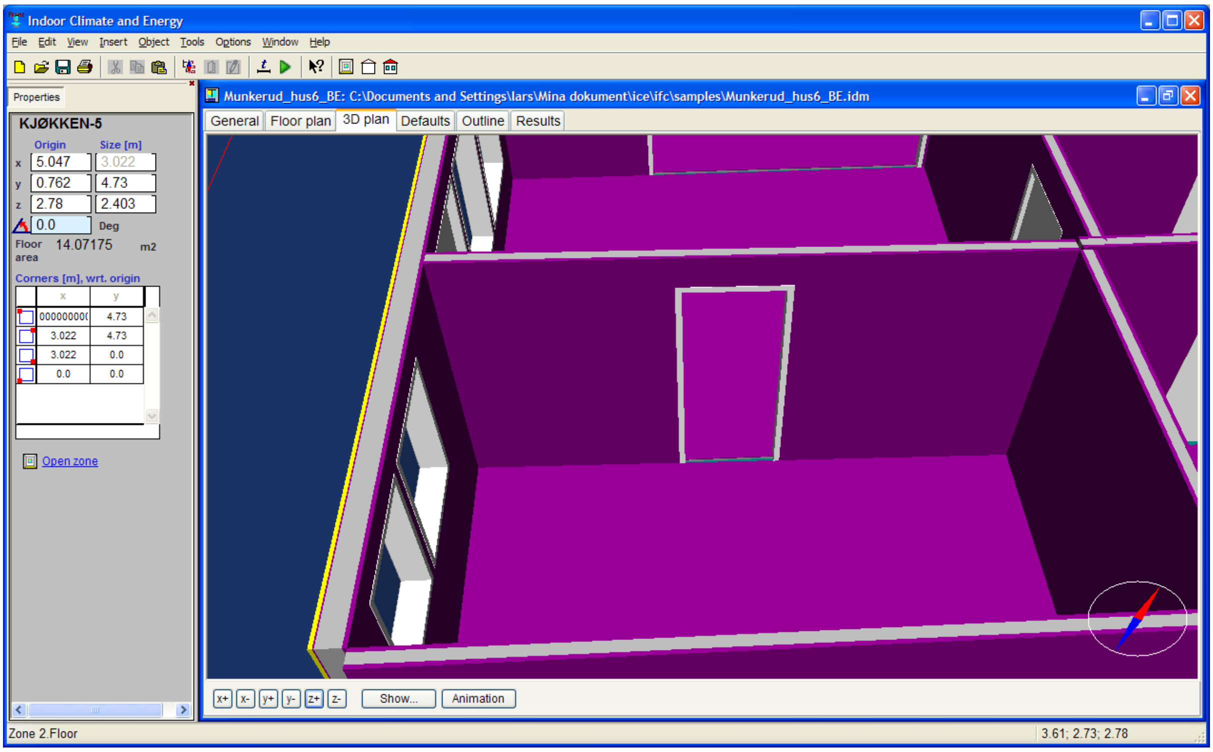

Figure 3.12 3D visualization of opening without door

A window is visualized as a frame and a transparent surface. The transparent surface is placed at depth from the outside corresponding to the window's recess depth, Figure 3.13.

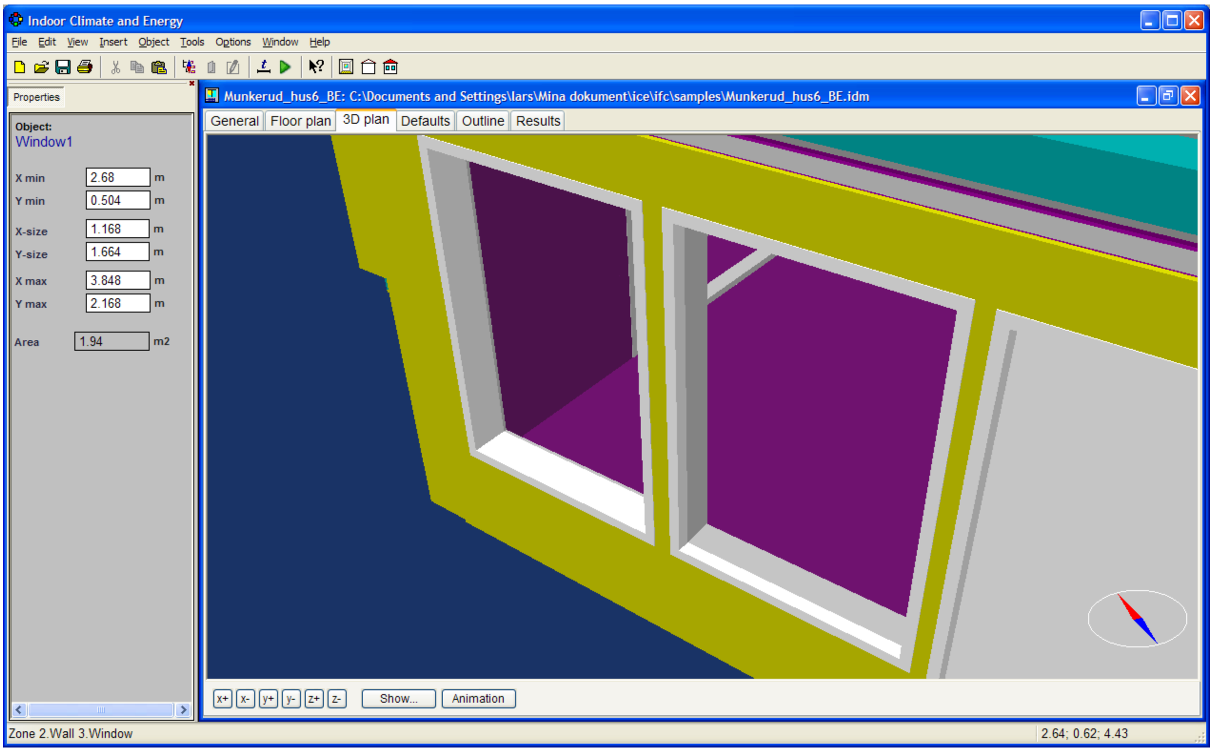

Figure 3.13 3D visualization of two windows with different recess depth

4.8. External shades

Double-clicking on a window will open the window dialog, Figure 3.14. In this dialog external window shading can be added.

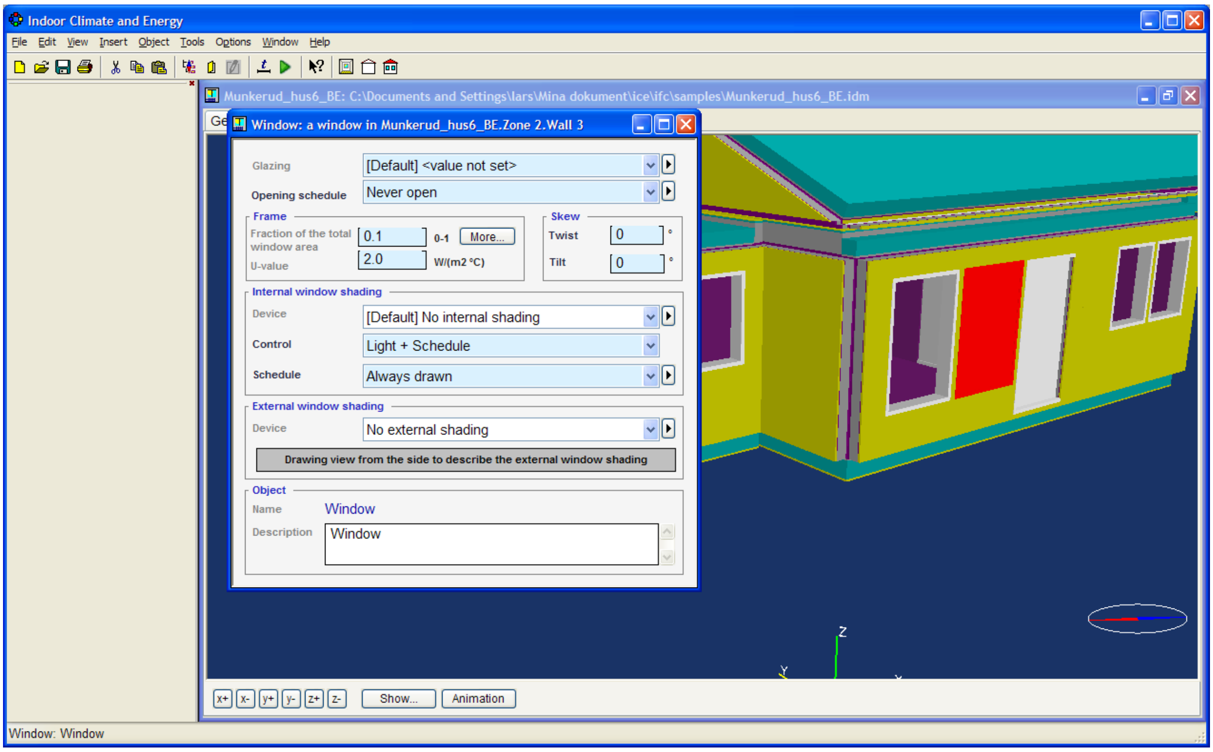

Figure 3.14 Window dialog

Double-clicking on the 'Drawing view from the side to describe the external window shading'-button will open the shading editor, Figure 3.15. There shading elements can be added and edited.

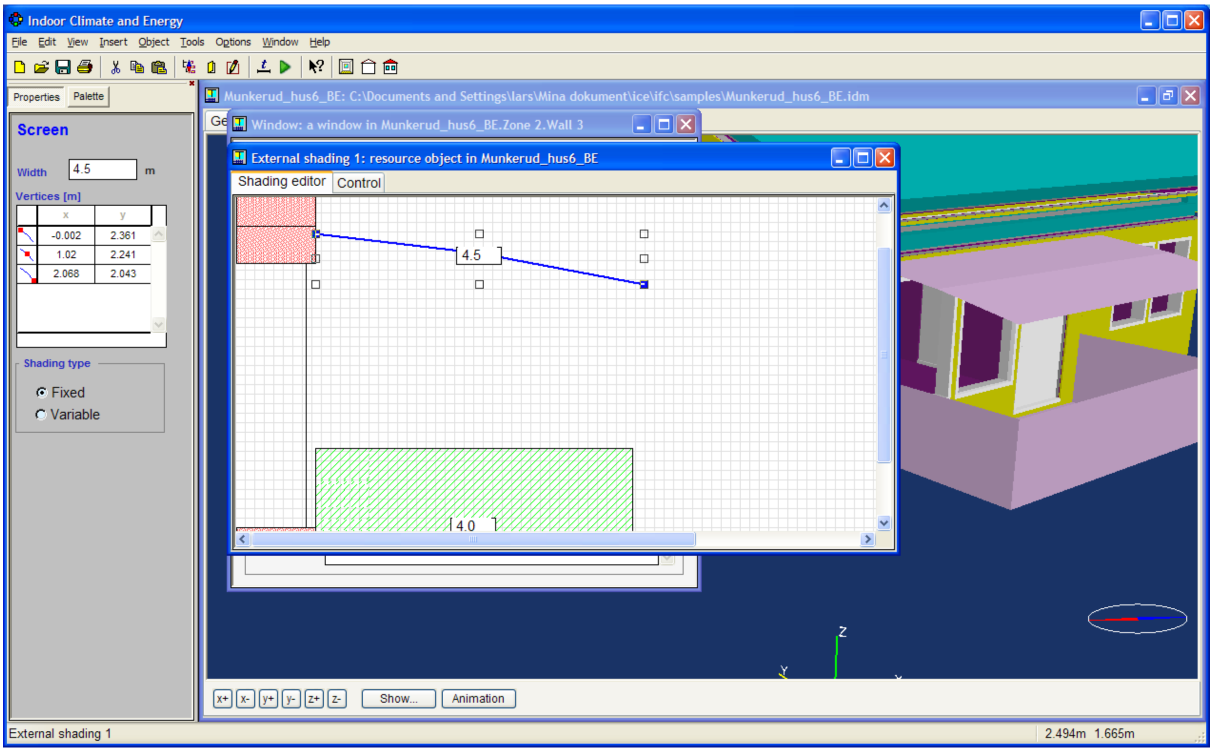

Figure 3.15 2D shading editor

The shading surfaces are visualized in the 3D plan window. These include balcony with sides, simple screen, see Figure 3.16, and marquee with sides, side fins, Figure 3.17.

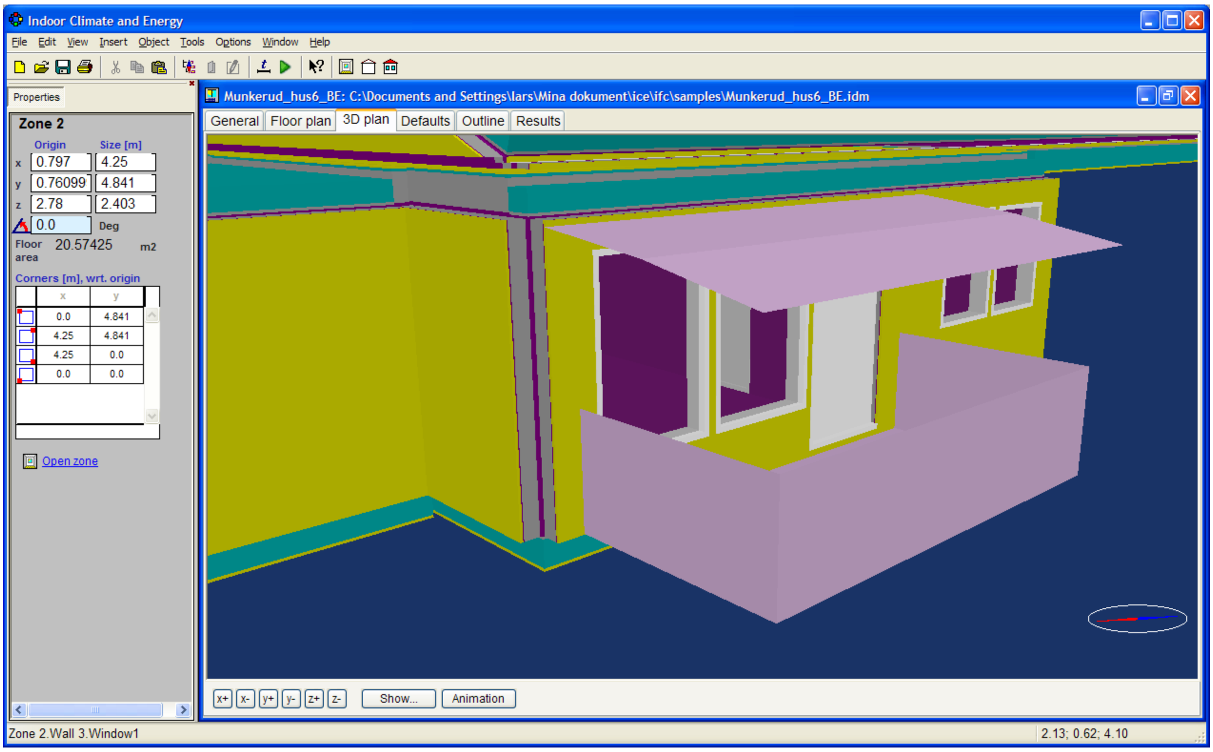

Figure 3.16 3D visualization of a balcony and a simple screen

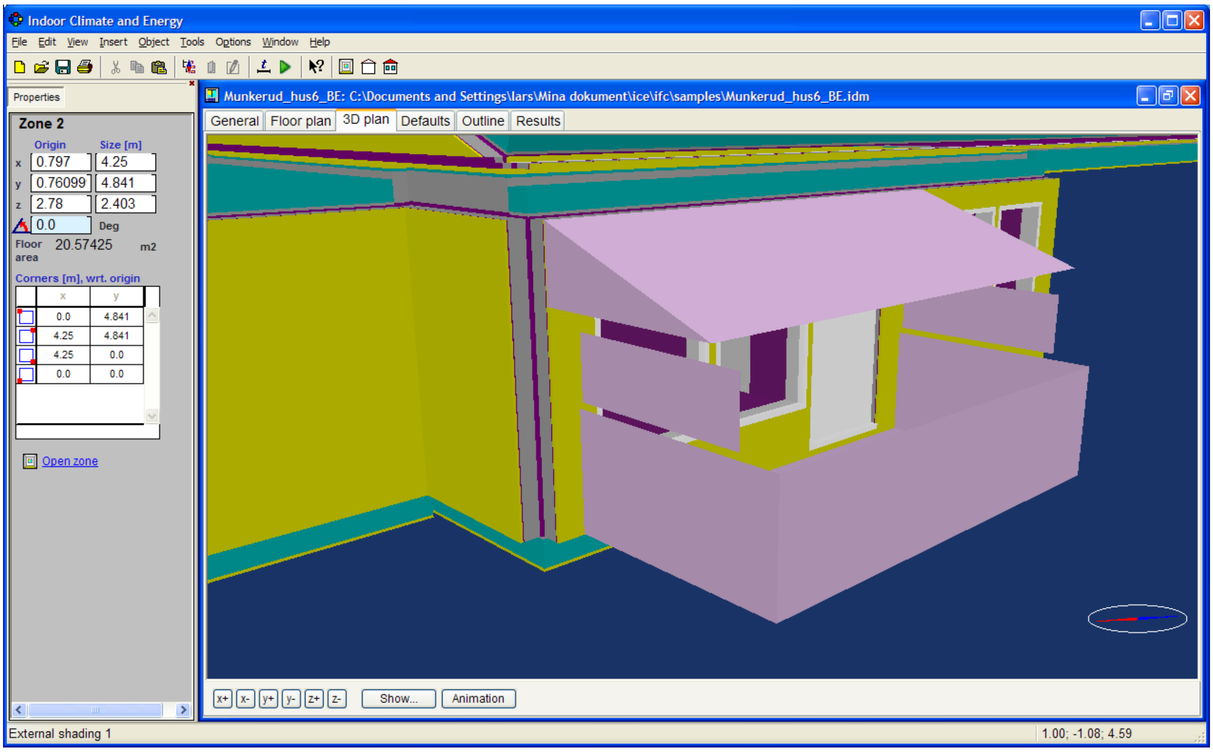

Figure 3.17 3D visualization of a balcony, a marquee and side fins

Shading buildings can also be defined to shade the windows. These are added and edited from the Site plan, Figure 3.18.

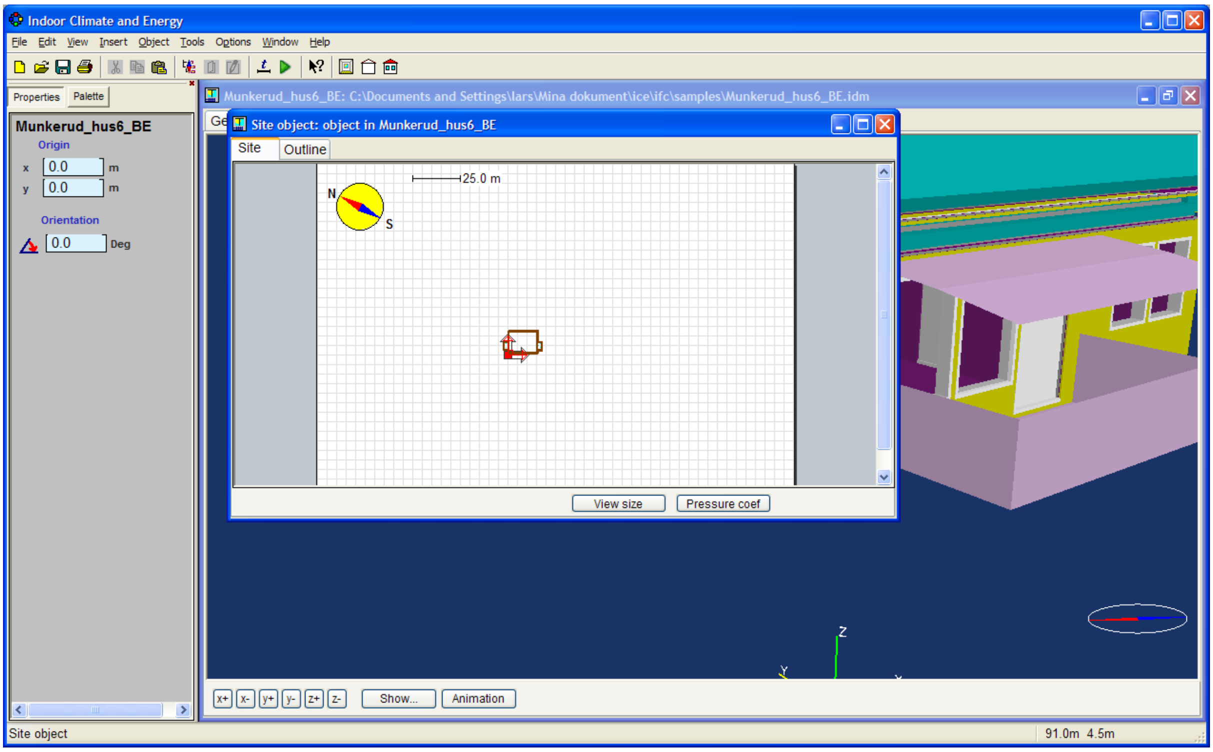

Figure 3.18 Site plan

The shading buildings are visualized in the 3D plan window, Figure 3.19.

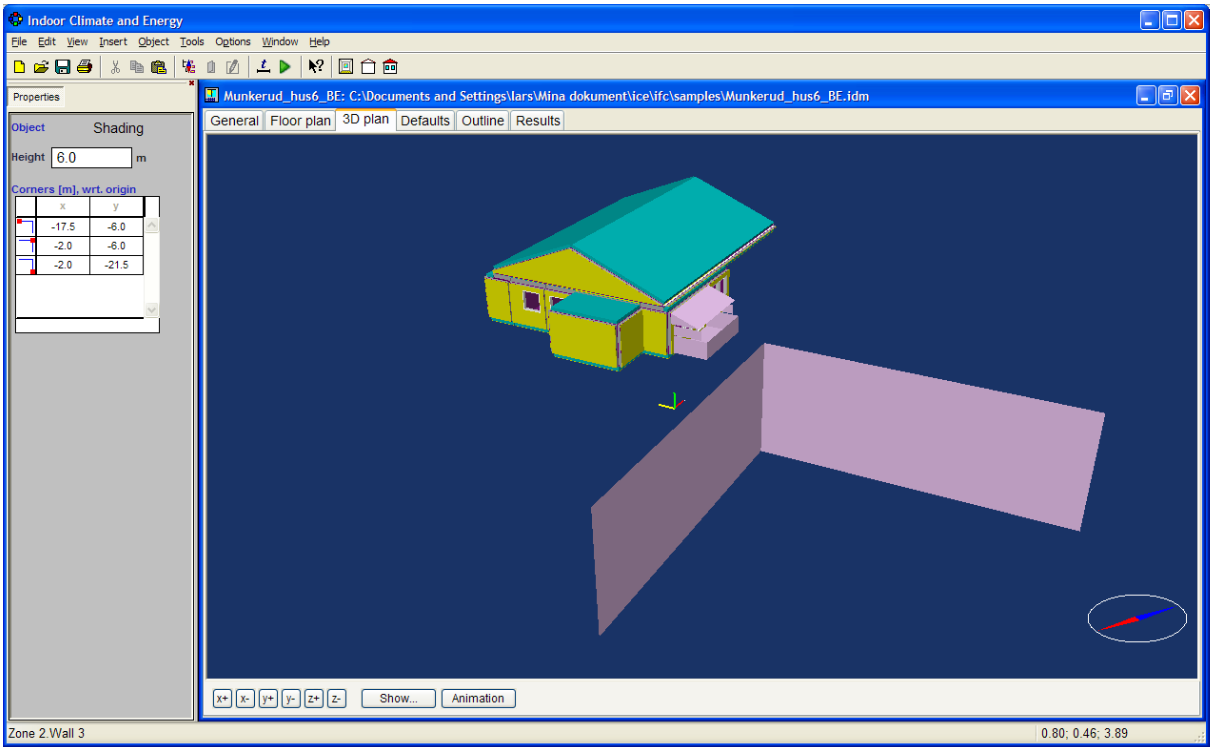

Figure 3.19 3D visualization of a shading building

4.9. IFC import

IFC models can be imported into IDA ICE from the Floor plan tab. When an IFC model has been imported, its geometric properties are visualized in the 3D plan window, Figure 3.20. The IFC import facility will in the framework of I3CON be extended to support IFC 2x3.

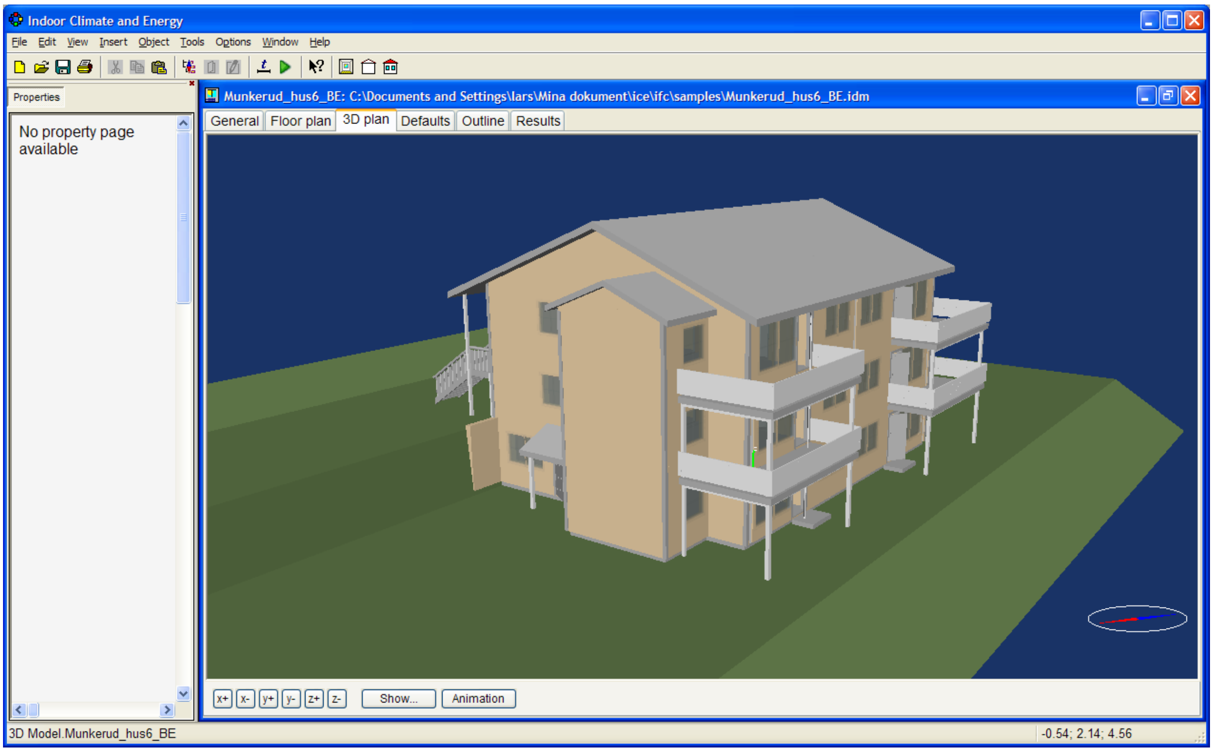

Figure 3.20 3D visualization of an imported IFC model

The IFC model can be cut just as the rest of the model, Figure 3.21.

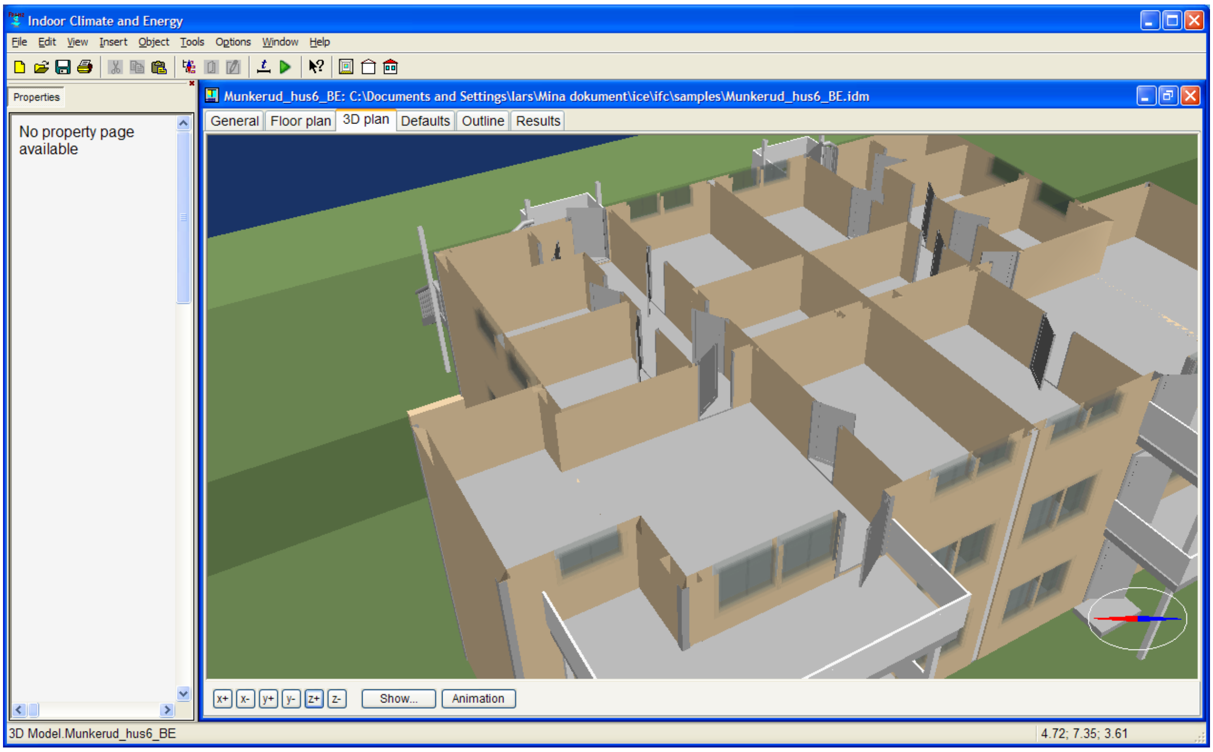

Figure 3.21 z+ cut of IFC model

The corresponding cut is shown in 2D in the Floor plan tab, Figure 3.22. There, zones that automatically take the shape of IFC spaces can easily be added.

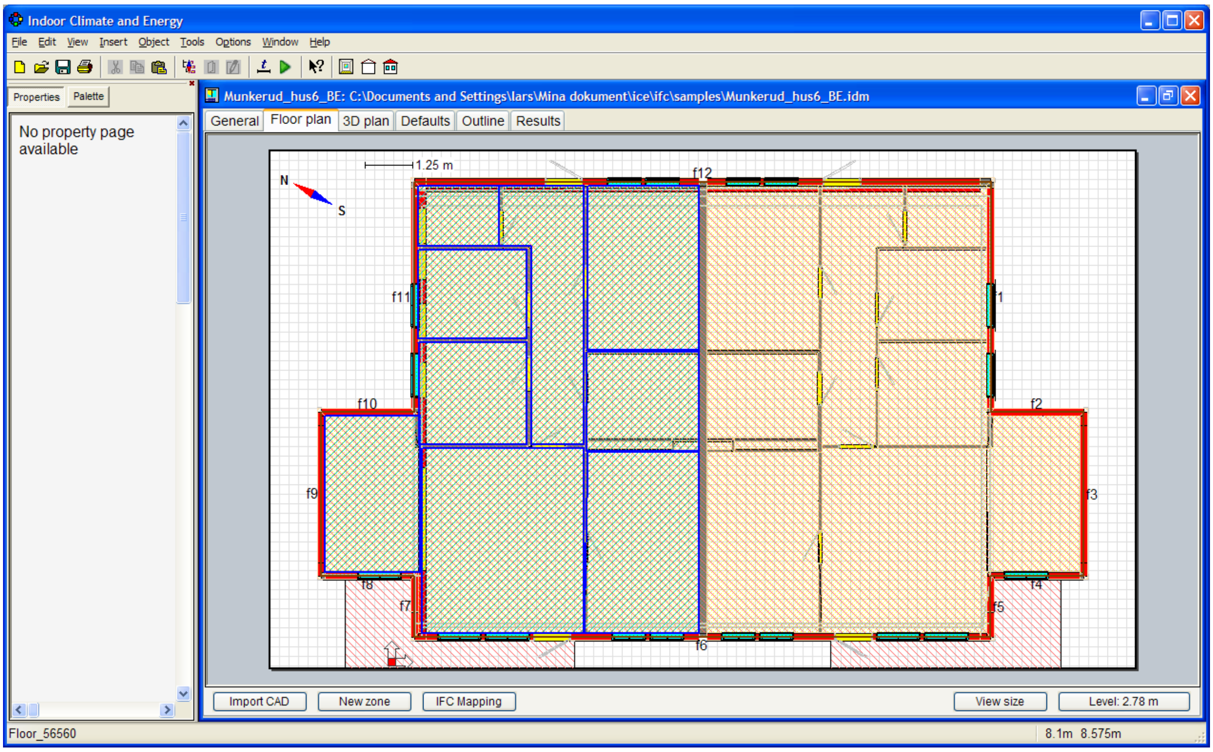

Figure 3.22 Floor plan with IFC model and added zones

The new zones are shown in the 3D plan window, Figure 3.23, 3.24 and 3.25.

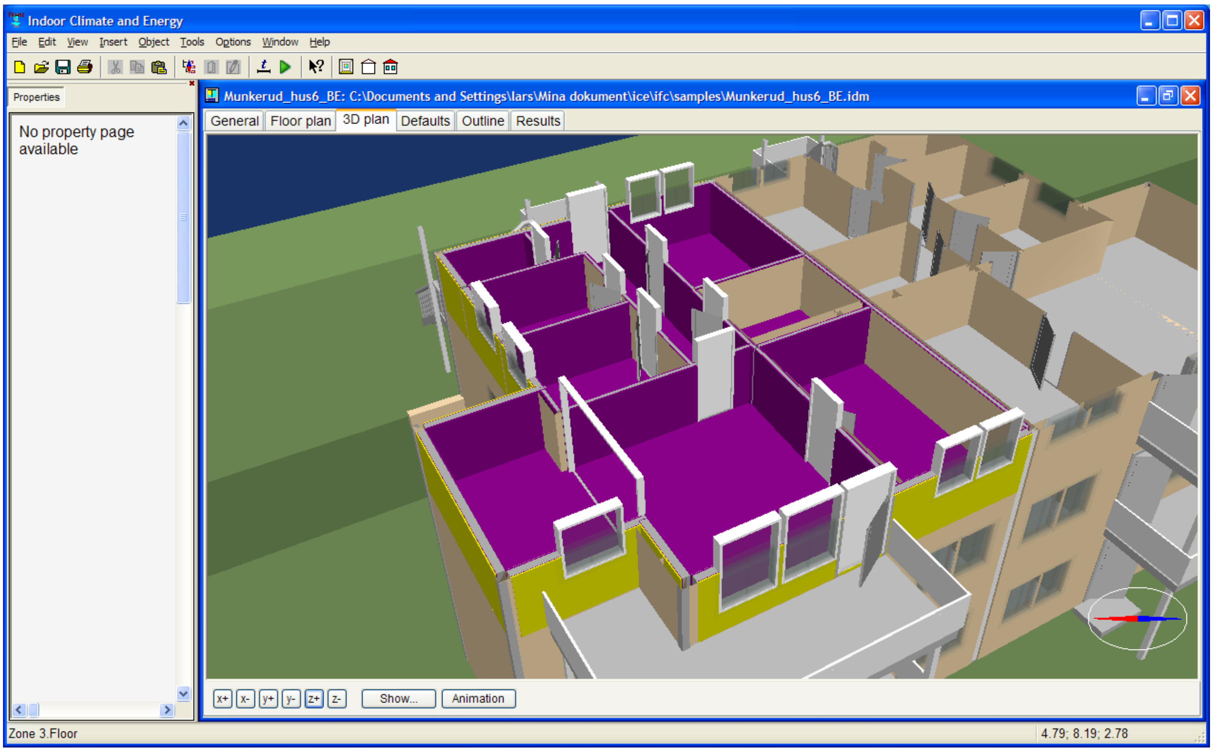

Figure 3.23 3D visualization of IFC model and added zones

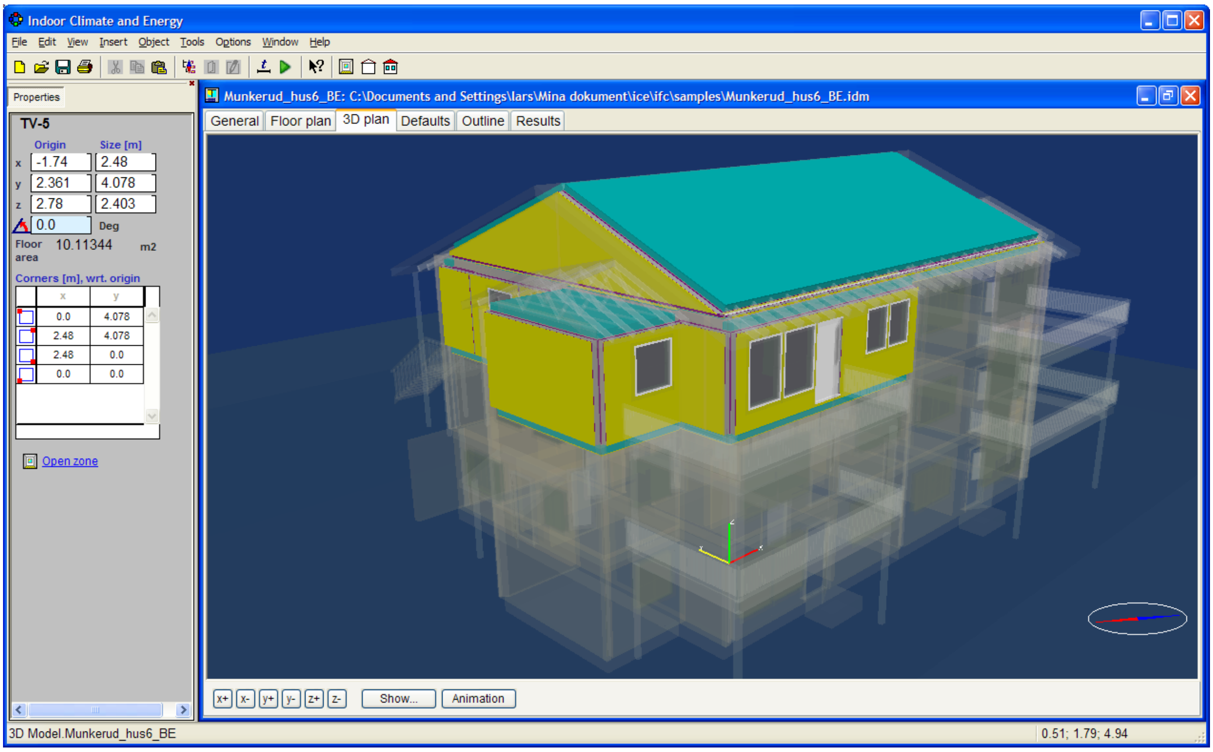

Figure 3.24 3D visualization of IFC model and added zones

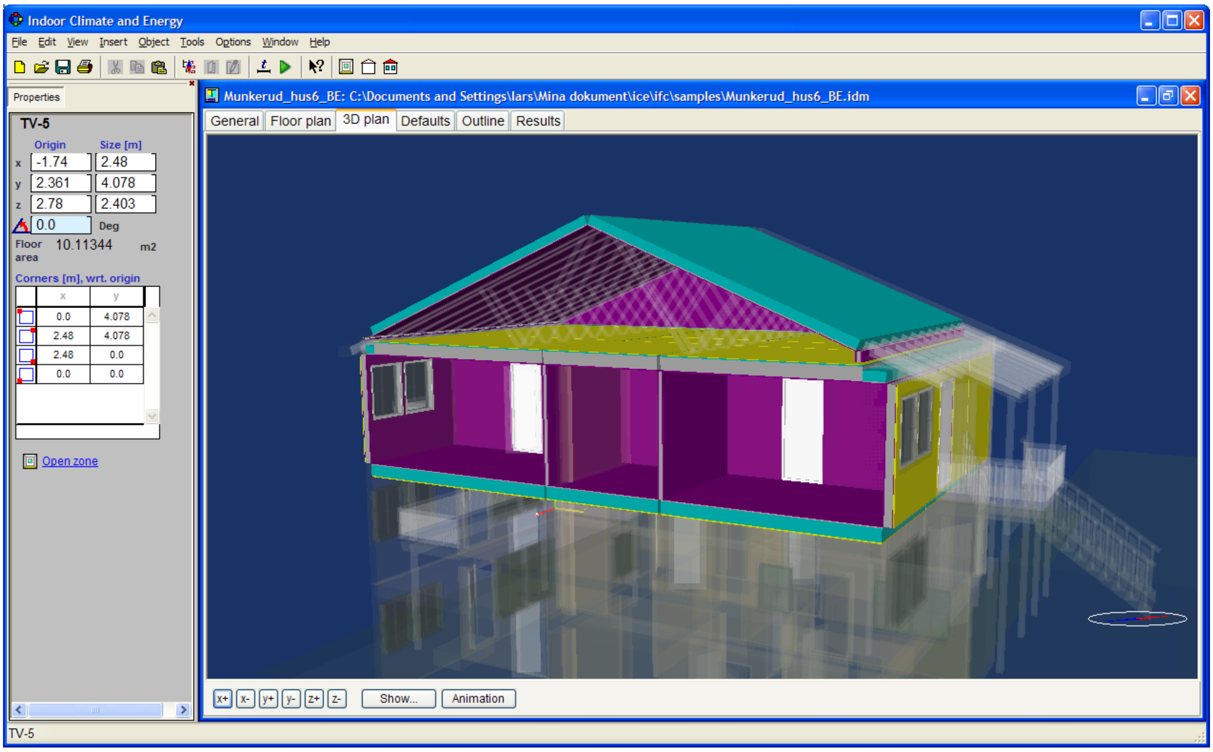

Figure 3.25 3D visualization of IFC model and added zones

4.10. Import of other CAD formats

Additional objects can be imported into the 3D model in a number of CAD formats, Figure 3.26. These objects are taken into account in the shadow calculations as well as the IFC model with all its details.

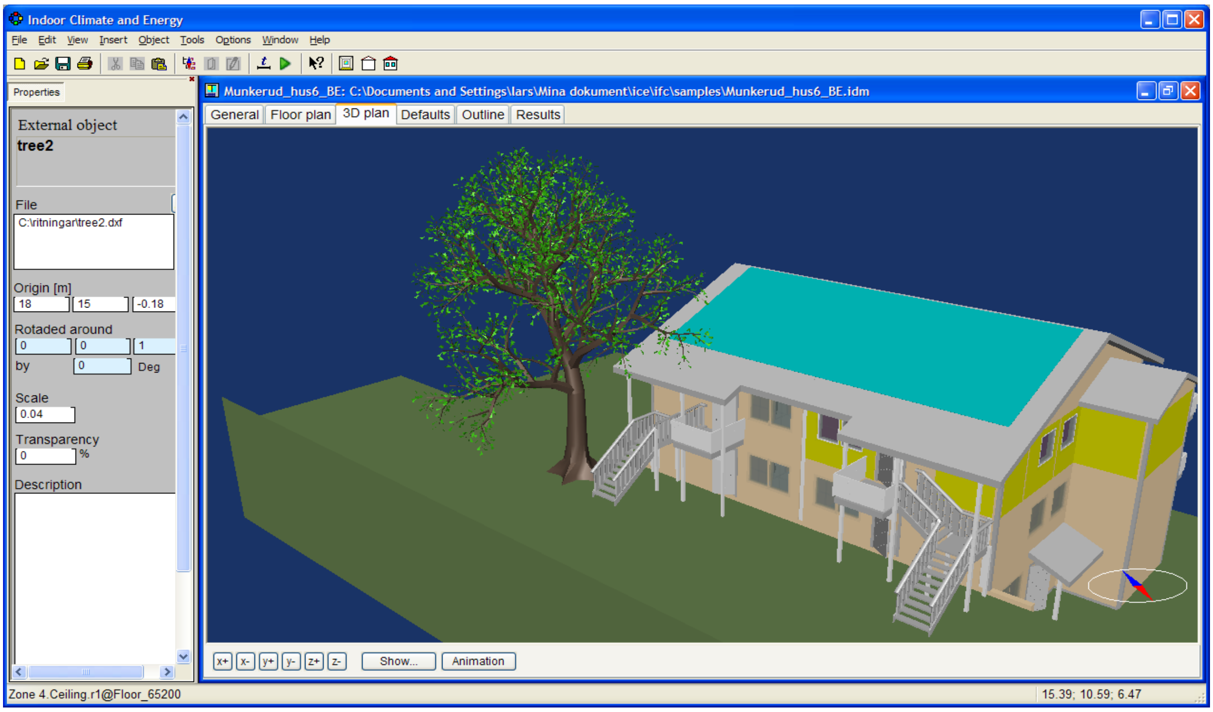

Figure 3.26 CAD model of a tree imported into the 3D model

4.11. Visualization of parameters and variables

In the General tab a summary of the model is shown in a list, Figure 3.27. When one of the box icons in the header of the lists is clicked, a graphical overview of the selected parameter is shown in the 3D plan tab.

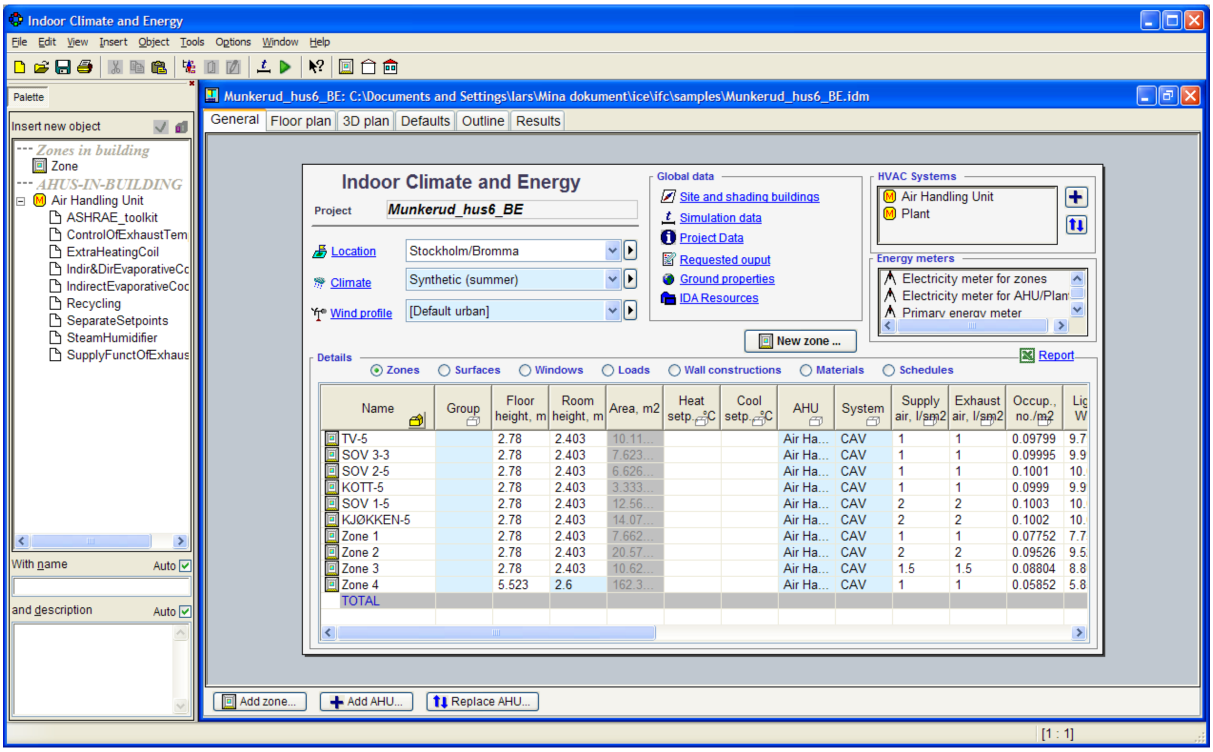

Figure 3.27 General tab with list of parameters

Figure 3.28 shows the names of the zones in the 3D plan window.

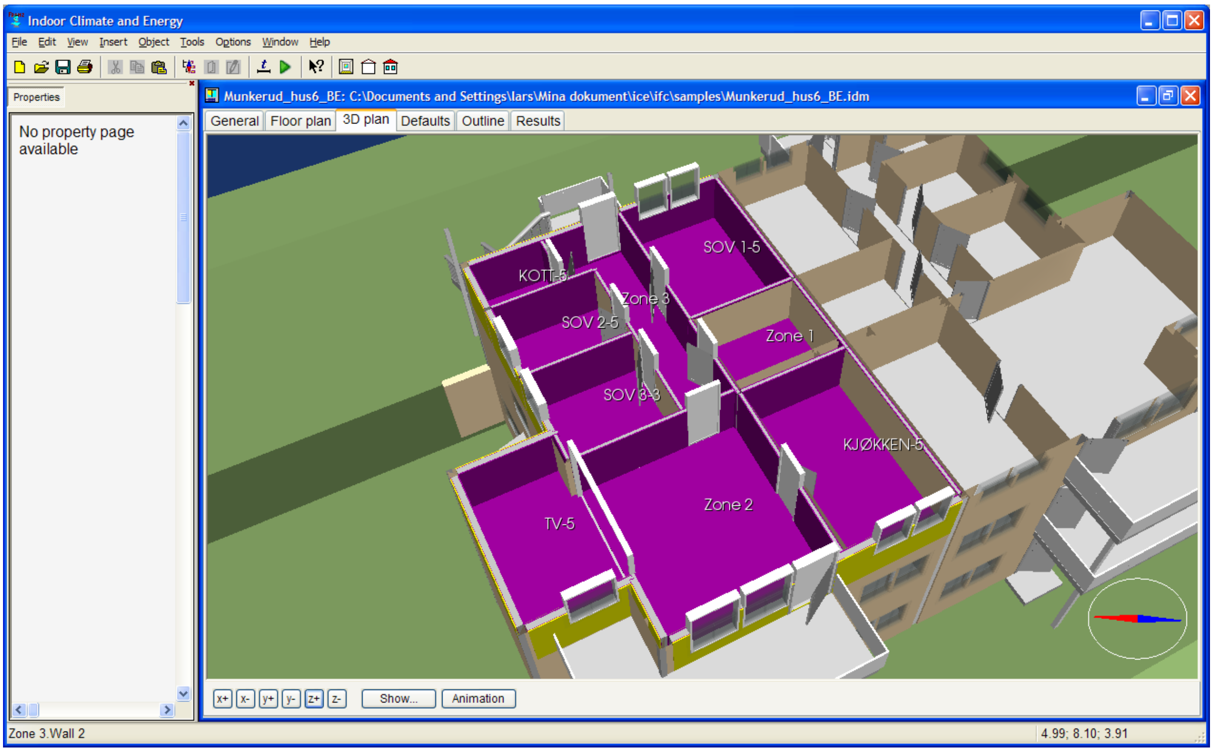

Figure 3.28 Zone name parameter visualized in 3D model

Figure 3.29 shows the number of occupants per square meter of each zone with colors on the walls.

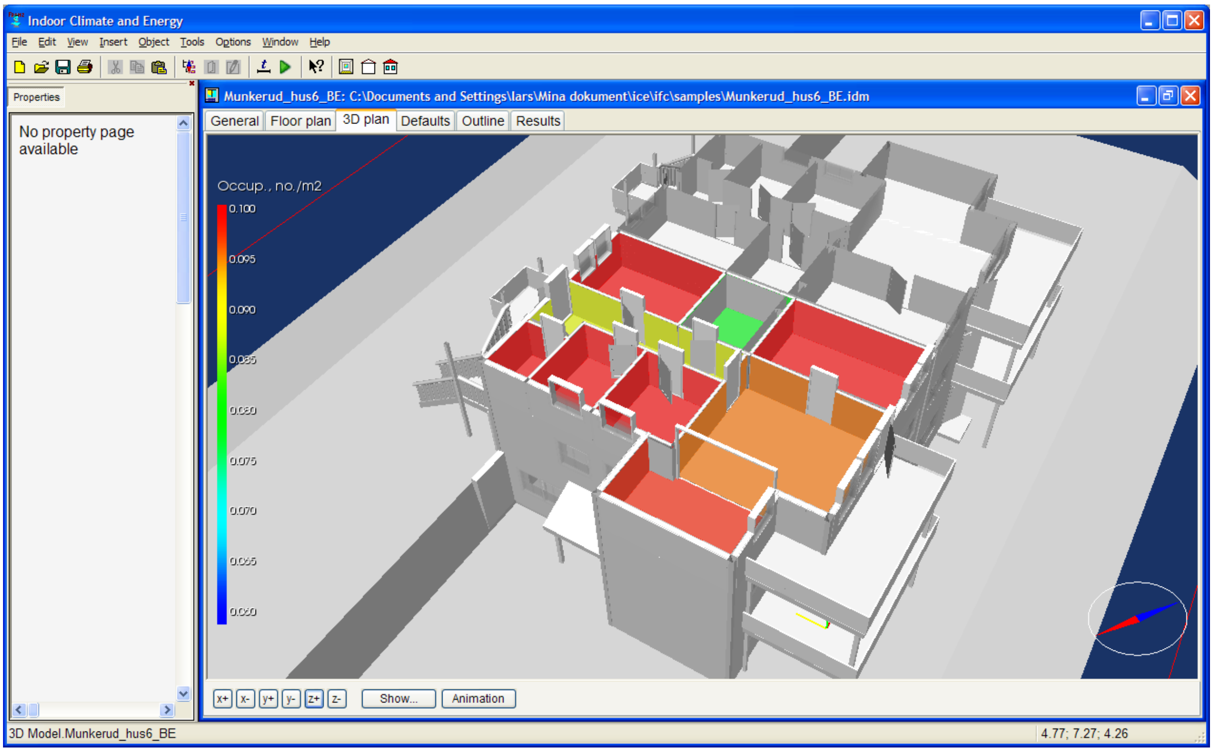

Figure 3.29 Occupant no/m2 parameter visualized in 3D model

Figure 3.30 shows the amount of supply air in liters per square meter for each zone with colored arrows.

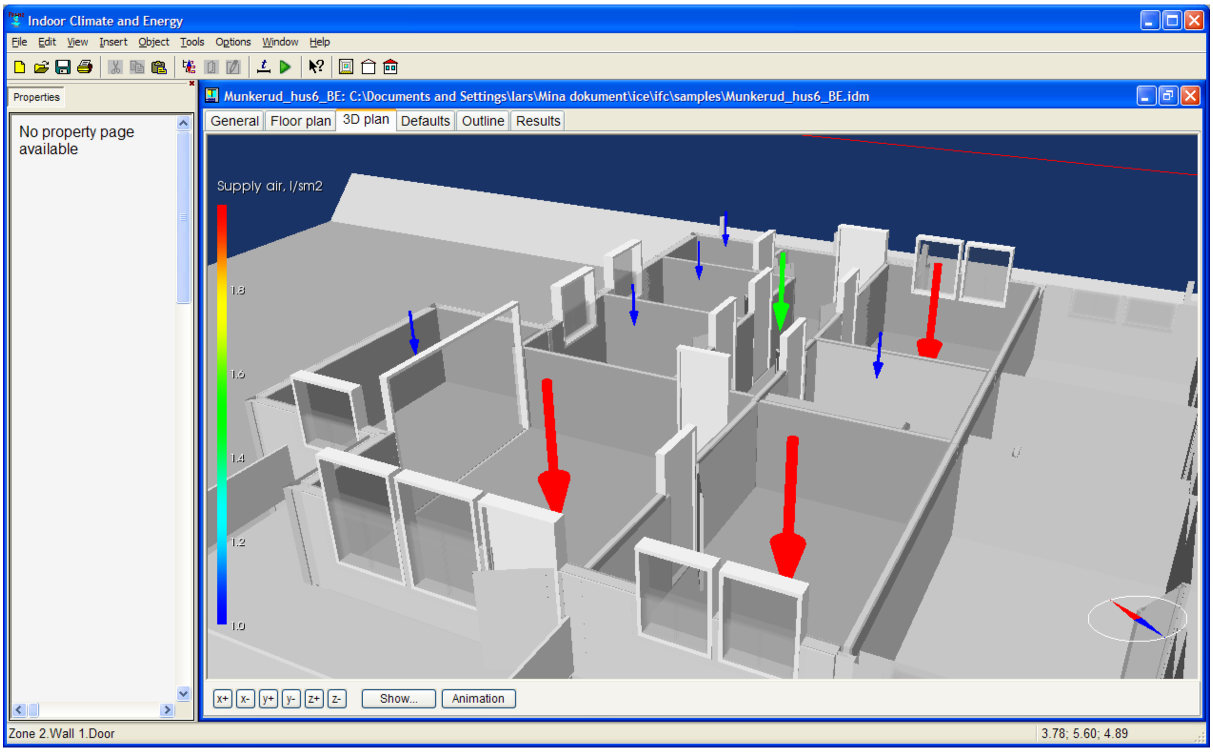

Figure 3.30 Supply air parameter visualized in 3D model

Figure 3.31 shows occupant schedules for each zone with colors on the walls.

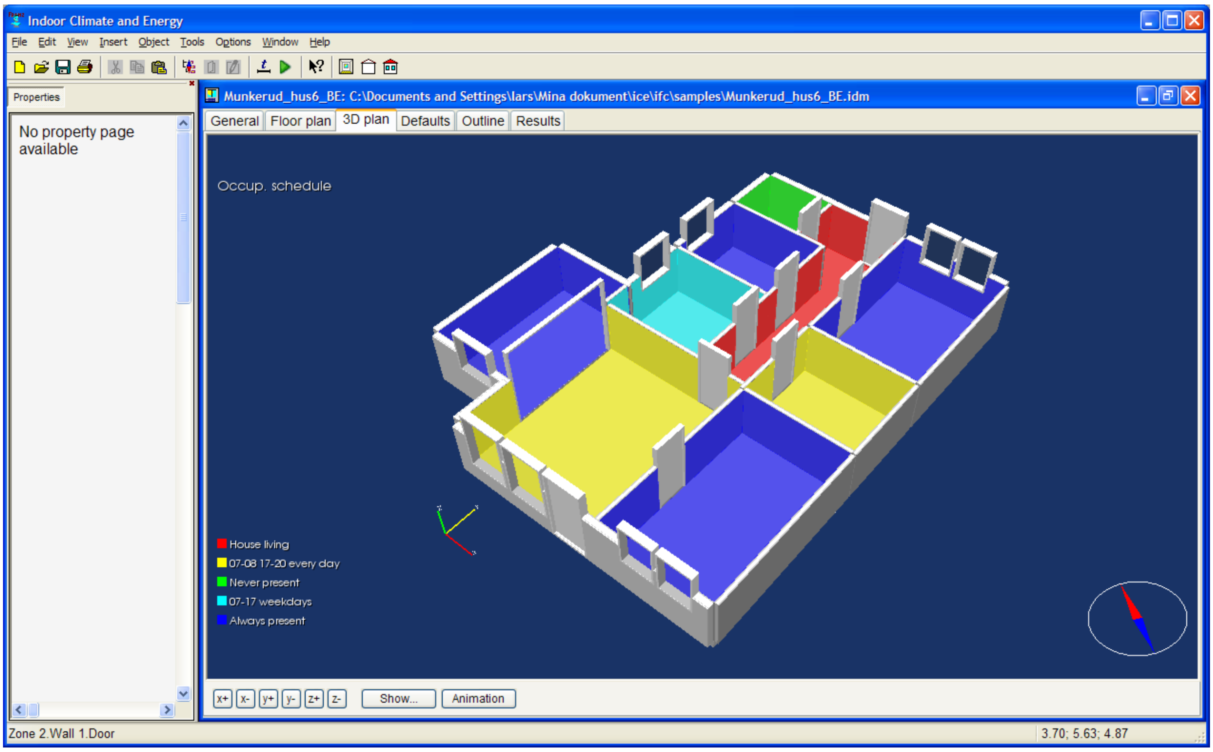

Figure 3.31 Occupant schedule parameter visualized in 3D model

Variables are visualized in the same way as parameters, only that their time dependence is animated. Animation of a variable is initiated by opening the 'Show animated results' dialog with the Animation button, Figure 3.32. Select variable to animate and click Show.

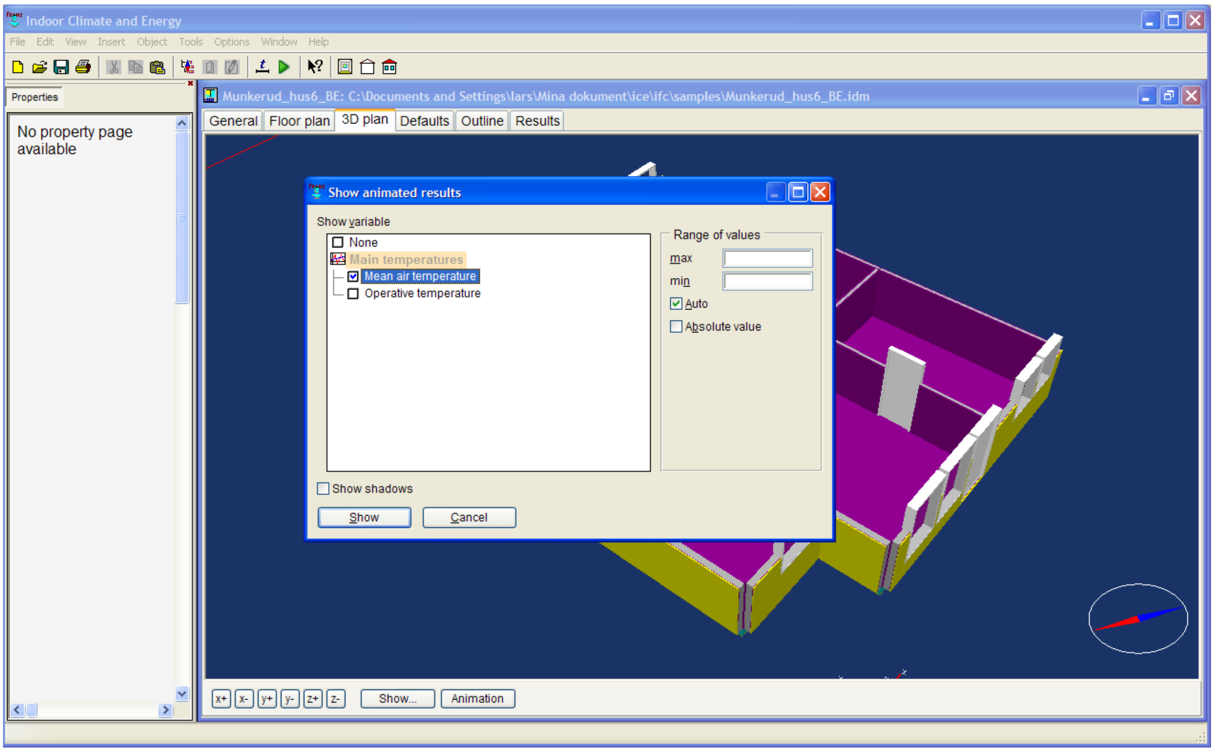

Figure 3.32 Show animated results dialog

4.12. Shadow visualization

The sunlight and shading of the model are visualized in the 3D plan tab with complex real-time shadows. Visualization of a shadows is initiated by opening the 'Show animated results' dialog with the Animation button, Figure 3.33. Check 'Show shadows' and click Show.

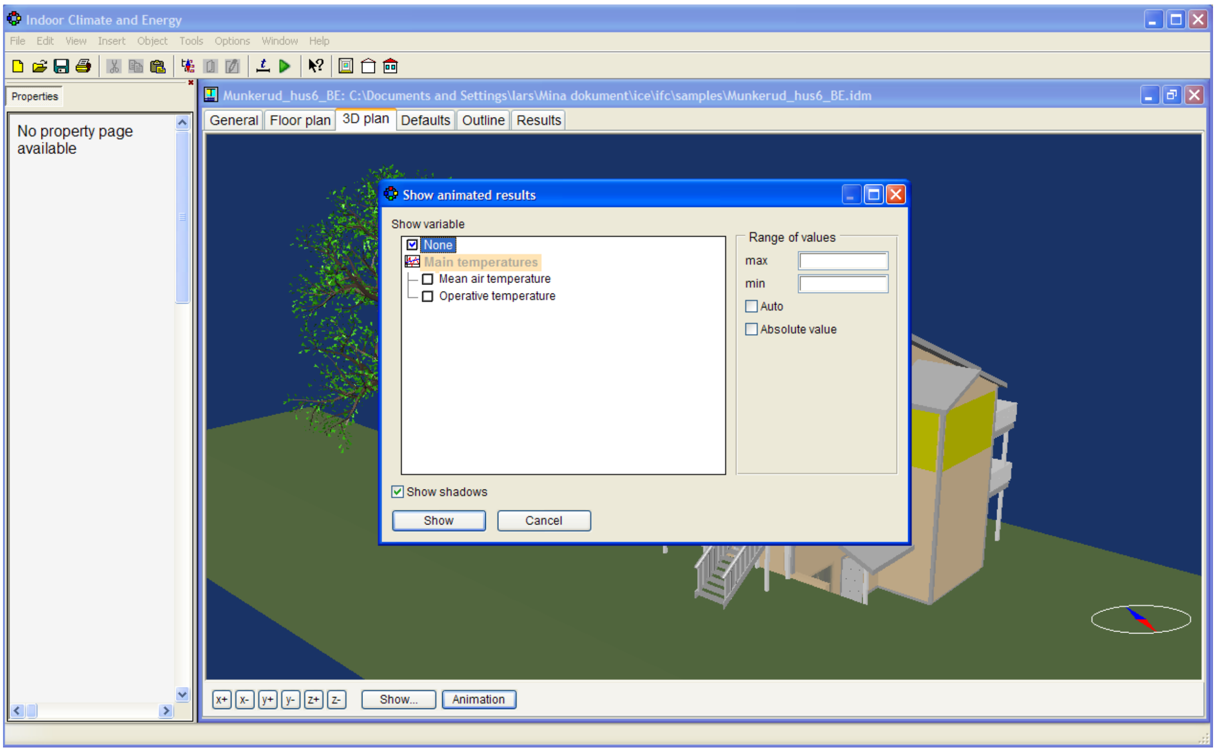

Figure 3.33 Show animated results dialog

The shadows are quite detailed and the model can be interacted with in real-time while the shadow are animated to follow the sun path of the day, Figure 3.34.

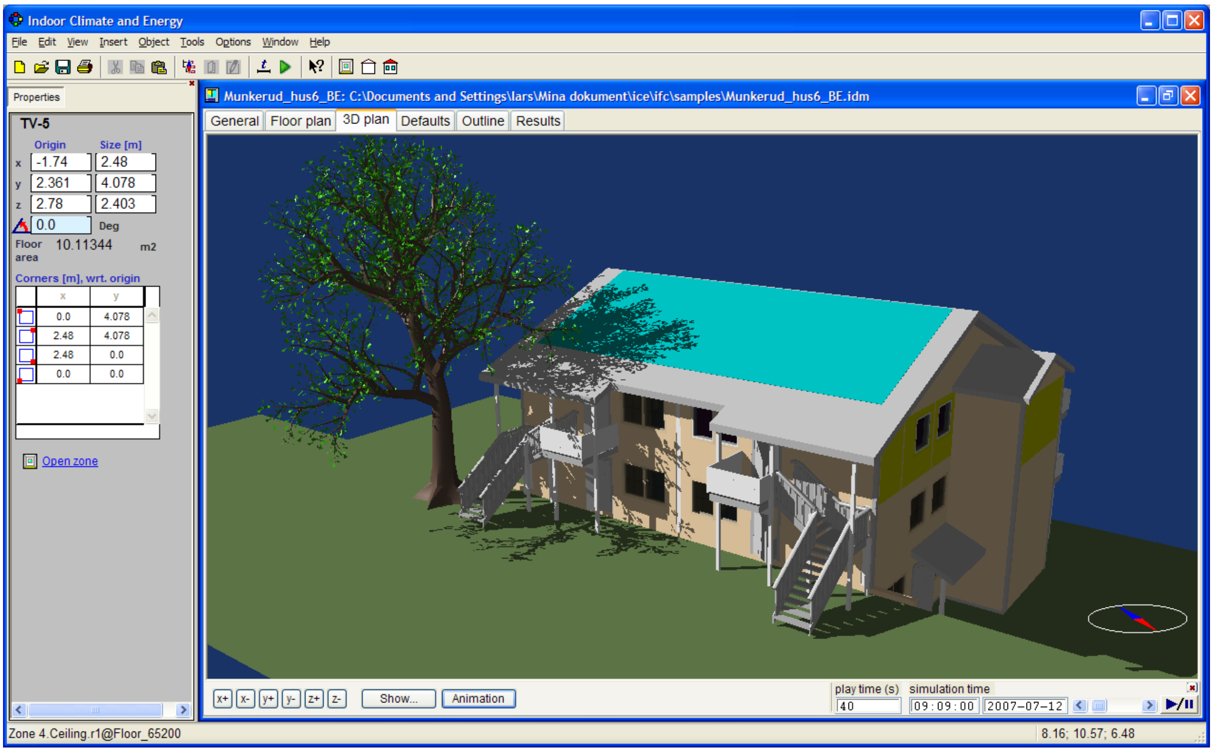

Figure 3.34 Real-time shadow visualization in the 3D model

Even as the 3D model is cut in different directions, the shadows stay the same and are animated and fully interactive, Figure 3.35.

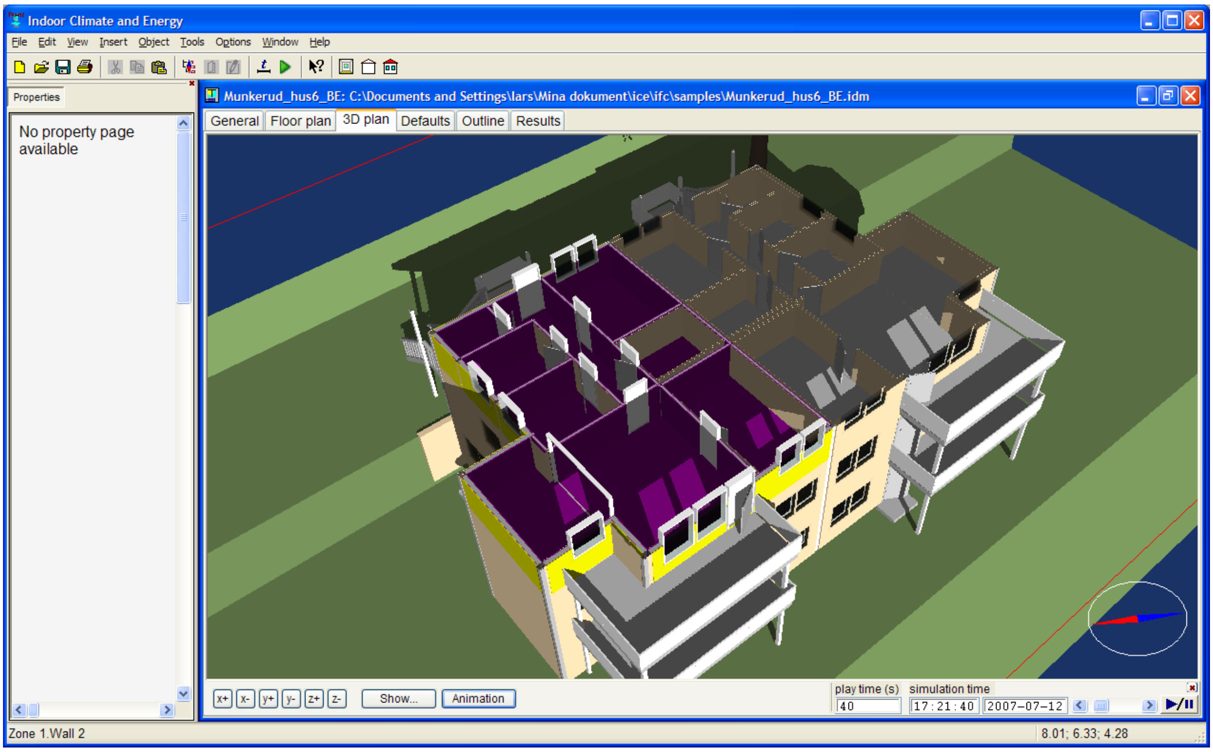

Figure 3.35 Real-time shadow visualization in a z+ cut of the 3D model

4.13. Printing the 3D window

A high resolution image can be rendered from the 3D plan window by choosing File > Print from the menu bar.

5. Development environment

Even if the role of the simulator as a field device is conceptually simple, the adaptation of a variable-time step differential-algebraic solver to the required real-time asynchronous communication is not trivial. The simulator in its basic configuration may run considerably faster than real-time over a longer period of time, yet simulated time will slow down considerably as the simulator is negotiating event situations, e.g. when an on-off controller switches state, or when numerically difficult passages must be resolved. A continuous monitoring of the time lag of the simulator is needed and alarms must be issued when the time lag grows unacceptably long or the simulator fails for some other reason.

The following basic functionality is needed in the real-time simulator:

1. Input signals are read together with relevant timestamp (current time).

2. Interpolation and possibly extrapolation functions are fitted to input signals to the previous few time steps in order to allow input evaluation at any point in time up to the current time.

3. The time step control of the simulator attempts to take a step that reaches the current time. Possibly failing this, the simulator has to approach the current time by several smaller steps.

4. As soon as output signals exist for any time points up to the current time, they are reported.

5. The time lag between current time and the time of the output signal is reported.

In order to achieve this synergy between the simulation model and the real building, co-simulation has to take place in a common test bed. The test bed should be able to connect and synchronize the simulation model with different network sensors, Building Management Systems, and actuators. Moreover the test bed has to have the capability of extending the features of the simulation program and thus is future robust, being able to be easily upgraded and extended to add features to the existing platform. To achieve this as a proof-of-concept and at a prototype level, Building Controls Virtual Test Bed developed by Lawrence Berkeley National Laboratory, based on an open source Ptolemy II from the University of California at Berkeley, can be utilized. BCVTB relies on BACnet stack, which itself is an open source implementation, allowing the exchange of data with BACnet compatible building automation systems. Furthermore BCVTB allows the co-simulation between MATLAB/Simulink and IDA-ICE, thus extending the features of IDA-ICE simulations. Therefore BCVTB would be an adequate solution tool for realizing real-time visualization and control of the physical building through the simulation model.

For commercial deployment, OPC (Open Platform Communication) is adopted. OPC (OLE for process control) is an open standard for inter process communication with widespread industrial usage. It is currently based on Windows technology, but extension to a Web services architecture is well underway.

We plan to base the real-time simulator on the OPC standard and to implement the following basic operations:

1. Connect to OPC server (with given server name)

2. Disconnect from OPC server

3. Open channel (OPC group) for sending or receiving signals; signal names are given here

4. Close channel

5. Read data from channel (called asynchronically by OPC server)
6. Write data to channel (OPC group)
7. Get server time

For off-line simulations that are carried out by a Model Predictive Control algorithm, a facility to exercise and modify any model remotely must be developed. These simulations may require special hardware for sufficient efficiency and may therefore be carried out on an external system. A short description of the Client-Server operations is as follows:

Client login

Send:

Client contract id (encrypted)

Simulation request {IDA Application; (Add-ins); 'Custom code; 'Estimated max time <<seconds CPU>>}

Receive:

Session {'Granted <<id>> | 'Denied <<reason>>}

Rights {Supported IDA Application; (Supported Add-ins); 'Custom code; Storage quota <<MB>>}

Account status {Free simulation <<seconds CPU>>| Balance <<seconds CPU>>}

Simulation or model modification request

Send:

Model {Actual model | Stored model id <<modification script>> [<<Custom code>>] | Solver input file [<<Custom code>>]

Return specification {Actual model | (individual variables ['End-state only])

Receive:

Data according to Return specification | Error messages