Fourth National Conference of IBPSA-USA
New York City, New York
August 11 – 13, 2010

# AGENT-BASED MODELING AND SIMULATION OF INDIVIDUAL BUILDING OCCUPANTS

Gerhard Zimmermann
Informatik, University of Kaiserslautern, Germany
e-mail: zimmermann@informatik.uni-kl.de

## ABSTRACT

Occupant behavior determines the performance of buildings much more than many building details. This is especially true in organizations with flexible work hours, if occupancy is taken into account by the control systems and users interact with the building. Therefore, a model is presented that expresses individual user requirements, roles, and activities at office work places. A multi-agent approach with automatic model-based code generation is used to model users and user groups and to create prototype simulators for research. In order to be able to test user models realistically, the user model is extended by multi-agent models of the building performance, equipment, and control system.

## INTRODUCTION

Buildings - commercial, domestic, and public - are major consumers of energy and producers of $CO_2$. Therefore, it is of utmost importance to provide building designers with reliable tools to analyze and improve buildings in respect to energy consumption, $CO_2$ footprint, and user comfort during their lifetime.

Today, building performance simulation environments exist that provide good results for the physical and technical aspects of buildings in relation to thermal, air quality, light, and acoustic properties. These simulators are based on efficient numerical solutions of mathematical systems representing the building physics. Many systems take user behavior and automatic control into account, but with very strict assumptions.

Lately, it has become obvious that especially the energy consumption of buildings and the user comfort are very sensitive to real user and control system behavior, especially if users perform part of the control manually. Significant energy savings are possible if user occupancy and manual control are considered properly, without a reduction of user comfort. Building simulation environments are just beginning to reflect this influence by combining building performance simulators with generators for occupancy patterns and user interactions.

The presented research has focused on the problem of integrating building users, usage, control, equipment, and the physical structure into one system, using a homogeneous modeling and programming paradigm. A wide range of paradigms is available, ranging from Fortran to AI concepts. The primary focus is modeling, software production, and run-time efficiency. It was found experimentally that multi-agent systems with model-based automatic software generation provided a good compromise for modeling and simulation of buildings as systems. More details will be given in the paper.

No claim is made that this approach is superior to the existing building simulation solutions. The goal rather is to show an approach that has advantages for research, development, and analysis of user models and building control systems.

This paper focusses on the dynamic user behavior and comfort requirements and their integration with existing solutions for the dynamic behavior of buildings and intelligent control systems to predict and analyze the properties of building systems and user comfort as realistic as possible. Because humans are very complicated and unpredictable, the model of individuals is restricted to user properties that are relevant to the analyzed building properties.

## RELATED WORK

Manual interaction with light switches, dimmers, and daylight control by shades has been studied and energy savings of greater 50% have been realized. Therefore, results of appropriate field studies have been used to create stochastic models of user behavior in regard to natural and artificial lighting. For example (Hunt 1979) predicted manual switch-on probabilities. These first stochastic models were based on cumulated averages and did not take changing environments into account. As a considerable improvement, a stochastic and dynamic model was created by (Reinhart 2004) called Lightswitch 2002 with 5 minute time steps and individual user decisions. Four types of users are considered. Lightswitch was integrated with esp-r by a system called SHOCC (Bourgeois et al. 2006) and shows how the light load effects the total building energy consumption significantly. Lightswitch 2002 still assumes relatively regular work hours with 3 breaks and plus/minus 15 minutes random occupancy change times. In contrast, field studies in several offices show much more irregular occupancy patterns with up to 50% unoccupied work places during work hours (Mahdavi and Pröglhöf 2009).

In a related field study, window openings in relation to air temperatures and user behavior have been studied in offices over seven years (Haldi and Robinson 2009). As a result, sophisticated stochastic models of the relations have been derived for the application in performance simulations.

Another field study has been conducted to characterize activity schedules, occupancy patterns and movements of individuals in offices (Tabak 2008). An algorithmic approach USSU has been developed to predict and simulate detailed individual schedules of job related and personal activities in relation to activity locations with random properties based on parameters derived from field studies. Results from USSU have been used as daily occupancy data for building performance simulations (Zimmermann 2008a). Tabak's and the author's research have been performed in parallel with a mutual exchange of ideas. (Hoes et al. 2009) have extended the SHOCC - esp-r coupling by USSU to introduce refined occupancy patterns.

Personal comfort has been studied in many field-studies in different climate zones. A thorough evaluation of many of the data (de Dear et al. 1997) led to new ASHRAE standards for heating and cooling, taking outdoor temperatures into account. Still, the standards assume that all people have the same comfort requirements under equal conditions. We interpret the data differently and believe that 100% satisfaction can only be achieved if individuals can enforce individual comfort settings as explained in (Zimmermann 2008a). This conforms with the findings of (Arens et al. 2010) that perfect control of effective temperatures in offices does not lead to higher user satisfaction, but results in higher energy consumption.

## MODEL BASICS

The previous chapter has shown that there is a clear trend to highly dynamic models of individual users in changing environments. Significant performance prediction errors can occur if such user behavior and appropriate control strategies are not considered in building simulations. The presented modeling and simulation approach tries to advance the capabilities for representing a wide range of different user types and individuals as they are found in organizations with flexible irregular work hours, instant dynamic responses to changes, and individual preferences with reasonable modeling and programming effort and run-time efficiency.

Universities share many of the features of office buildings, although with a large variability of activities and schedules. Therefore, a university office has been chosen as an ideal example to test user models.

The computational model coming close to individual behavior is the *multi-agent system*. Agents in the meaning of computer science are entities that have specific features: autonomous, reactive to other agents and the environment, proactive, having internal states, and containing at least one thread of control. Additional features such as artificial intelligence are not considered here. Agents share many features of objects such as information hiding, but go beyond in their capabilities.

Here agents are interpreted and realized as *autonomous processes* that *communicate asynchronously*. Agents can be composed hierarchically and are not limited to model humans as the name suggests. Technical systems, physical components, or even abstract entities can be modeled as agents as long as they perform autonomous processes and communicate with and influence other agents. For example, a heating coil transforms the air temperature and communicates with the control system, the hot water supply, the fan, and other agents. This stretches the interpretation of agents very far, but it serves the goal of a homogeneous model.

### User behavior basics

The goal of models of individual behavior and the interface to performance simulation and building control systems are primarily *activity schedules* for each individual that include the activity locations or more precisely the temporal assignment of activities to *functional units* as for example work places. Activities depend on the *role* of the individual in the organization or household, on the activities of others, the *task* to be performed, resource and environment requirements, and personal *preferences* and characteristics. Activities also depend on static and dynamic building and functional unit properties. Therefore, a feedback-loop also exists from the building back to the individuals and user activities. The feedback can be short term such as a manual setpoint change as reaction to a temperature change or long term learning effects and user adaptations.

### Building system model overview

To structure this complex interdependence network, the components of a building system are assigned to five *domains*, each divided into *material (matter)* and *abstract* entities:

- *user activity domain: individual* and *activity*
- *functional unit domain: inventory* and *place*
- *control system domain: controller* and *sphere*
- *service system domain: equipment* and *zone*
- *building domain (*including the outdoor): *structure* (e.g. walls) and *space*.

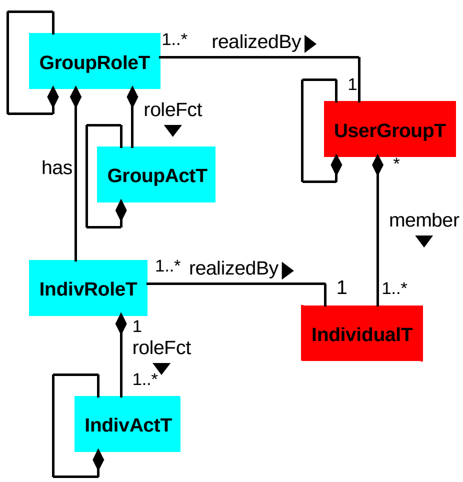


*Figure 1 User activity domain model. Abstract types are blue, material types are red*

Figure 1 shows the simple agent type model of the *user activity domain*. A user group can be composed of user groups and consists of individuals as members. A group role can be composed of other group roles, can have its own group activities, and can also be composed of individual roles with individual activities. Group roles and individual roles are assigned to user groups and individuals. This assignment can change over time, but at every point in time it is a one-to-one relation.

The trailing "T" at the object type names is a generalization and means that for a specific application domain, new types with different names and special features can be derived as a refinement and to simplify modeling. For example, in the university office domain, role names such as StudentRole, ProfessorRole, TechnicianRole can be introduced to define an ontology for universities.

The *functional unit domain* model provides the possibility to model part of the organization as abstract agent types. Only the *place* needs a material realization called inventory, mostly furniture that defines a work place or a meeting place.

One reason to introduce places is to give activities a location. From the location of different places that are used by consecutive activities of a role, the times to move between places and the density of individuals in circulation places can be deducted and simulated.

Another reason is that a *place* is also an abstract geometric entity that has to be geometrically related to *zones* of the service domain and to *spheres* that control zones. Different zones may provide light, ventilation, air, or radiation temperatures. The places define the requirements for the control spheres of the zones. The requirements depend on the type of the place, the conducted activity, and the preference of the user. If zones overlap several places, requirements from different places may be contradictory. The place dynamically bundles and negotiates the requirements and calculates resulting individual comfort levels using feedbacks from the building.

In the *control domain*, agents perfectly match distributed control processes and the communication between processes, sensors, and actuators. A control process can be a simple PID-loop or a sophisticated neural network. The control process can be proactively controlled by an internal clock or reactively by changes in the environment.

Agents can be used to model the *service domain*, e.g. by decomposing HVAC systems into units such as coils, fans, or ducts as agents, each agent evaluates the physical equations of the unit. The exchange of state variable data is performed by asynchronous communication. Agent clocks with fixed periods are possible, but not necessarily. Asynchronous event driven execution can be used to minimize computational requirements. Due to the hierarchical decomposition of agents, evaluations at higher levels of the tree are also possible, e.g. at AHUs or entire building zones.

The same decomposition can be applied in the *building domain*, as has been shown in several case studies by comparing equation solver based performance simulators with multi-agent based simulations (Zimmermann 2001). It has to be pointed out that the building domain was not completely covered as for example in esp-r, but only the effects significant for the specific building and the specifically asked questions, for example relative energy consumption and user comfort in relation to different control strategies.

## DYNAMIC USER BEHAVIOR MODEL

### Activity scheduling methods

In offices, the individuals are responsible for most of their activity scheduling. This is a decision process that takes many factors including personal preferences into account. The more we understand this process, the better we can model and simulate it.

In the area of transportation, decision theory is relatively far advanced (Arentze and Timmermans 2006). From this theory three decision making levels are adopted here: strategic, tactical, and operational. At the strategic level major goals and milestones are ordered by priorities and sets of tasks defined to fulfil the goals in time. At the tactical level, activity schedules for the near future are defined to fulfill the tasks. At the operational level, the activities are executed with the goal not to exceed the scheduled time frames.

In the area of office work, we are not yet at this level of sophistication. The strategic level is not part of the simulation, but is executed manually by providing parameter sets for individual behavior as input to simulation. The parameter sets are extracted from field studies. From organizational information about roles and workflows, activities are derived, parameterized with frequency, duration, and other observed features. Stochastic models are used for the tactical level based on these parameters to generate schedules according to simple rules. By randomizing some of the parameters with given probability functions, different schedules for individuals with identical roles are derived for every day and with the necessary time resolution, simulating typical behavior patterns at the operational level.

### Classification

Several classifications can be applied to roles, tasks, and activities. First of all, the type of organization determines the existing roles. Here, a university department is used as example.

An individual takes on different roles during a day, for example housewife, professor, and recreation role. A role consists of several task that become activities when scheduled. Tasks and activities that can be classified as

- Job or work related: attend lecture, write paper, attend meeting, programming, homework, paperwork, make copies, travel, ...
- social: chat, call home, go to lunch, take a break, ...
- physiological: go to bathroom, smoke, get a drink, ...

The latter two will be merged into personal tasks and activities because overlaps exist anyway. Tasks and activities can be further classified as

- planned: meetings, travel, lectures, lunch, ...
- continuous: check e-mail, write report, prepare lecture ...
- spontaneous: visitors, external interrupts, personal activities,...

Planned tasks are entered into the schedule first, often for a long time in advance, continuous fill the free time in between planned activities, others occur spontaneously at run-time and are not scheduled in advance. A third important distinction is between

- single individual: desk work, go to bathroom, ...
- group: meeting, lecture, consultation, ...

A group task requires more than one person to share the activity at the same time. Members do not have to be in the same location, as for example in a tele-conference.

Another distinction is between primary and secondary activities: Primary activities are scheduled. During the execution of a primary activity, secondary activities such as going to the printer or asking somebody for help can occur spontaneously without entering the schedule.

It has to be stressed that the classification is used for scheduling purposes only, not to model a workflow. Activities can be interrupted if necessary or split into smaller ones if exceeding one day, as for example writing a thesis.

### Activity refinement level

When activities are defined and modeled it is necessary to consider the influence of activity types on the expected results of the building system simulation. For example, if only the energy consumption for HVAC is analyzed, it may not matter which work is performed at a desk. What matters is if and when the workplace is occupied, because a user adds to the heat load as a person, with the used equipment, and needs air conditioning when present. Short breaks may not be of any consequence and may be inhibited. If detailed lighting control is to be tested, different kinds of desk work may require different light levels and even short breaks can trigger light switching. If the traffic in circulation spaces is a concern, every movement and its route has to be considered.

### Role scheduling

Roles are composed of role specific activities. The first activity is started when an individual takes the role (e.g. enters the office). The last activity ends when the same individual quits the role. Typically, the times for take and quit will be morning and afternoon, but some students or office workers may start in the evening and quit in the morning to use the quiet times.

Role start time and quit time are decisions of the individual and the scheduling will be the responsibility of corresponding *individual* agent. Nevertheless, it has to be coordinated with pre-scheduled role activities. It is assumed that the individual knows the schedule of the planned activities when deciding on the start times.

The start time is based on the assumption that it varies randomly every day in a certain time range. Another assumption can be that the probability density function of the start time resembles a bell shape. For the purpose of building performance simulation, an equal distribution within the range is sufficient. The start time is defined by three parameters: average start time, start time range, and probability. The probability models the possibility that an individual does not go to the office, for example because of travel or other reasons. Such events are assumed to happen randomly.

For the quit time, average duration and range are sufficient. It is assumed, that the predicted quit time can be overruled by activity scheduling when necessary, for example because of a meeting.

Especially in university offices, every day of the week requires a different role schedule for most individuals because of regular lectures. This pattern is repeated after a week. Therefore, a different set of parameters is defined for seven consecutive days. This set may change throughout the year. In other organizations different regularity patterns may apply.

**Job related task scheduling**

Primary job related activities are scheduled in the *role* agent. Scheduling starts out with planned tasks. Planned tasks can be regular such as lectures or irregular such as meetings. *Regular planned tasks* are defined by start time, duration, location, probability, and priority. The probability can be used to model for example that a lecture is sometimes skipped.

*Irregular planned group tasks* such as meetings or cooperative work are scheduled next. One planning role is responsible for the other roles. In principle, this requires negotiations between the participating roles to find free slots in all schedules. If priorities are to be considered, this can become as complicated a process as in real life. To simplify the process, irregular group tasks are scheduled in the group role agent that aggregates the task members. All individual role agents report their planned activity schedule for the day to the group role. The group role searches for gaps in the schedules of all participants. The group role then decides randomly based on the frequency, duration, duration range which group tasks are to be scheduled that day, checks for the best fitting gaps and possible locations, schedules the activities and notifies all member roles. Best fitting means that the planning role has priority, the rest of the members will have to change their schedule or will not participate. A *gravity* function is applied to avoid small time gaps.

*Irregular planned single individual tasks* are scheduled next according to the same parameters as the group tasks, but in the individual role agent.

*Unplanned activities* may not be the least important in the workflow, but for scheduling purposes they get the lowest priority, are scheduled last, and just fill the remaining free time between planned activities. If more than one such activity exists, this time has to be split fairly. Therefore, unplanned activities are defined by a time share and a preferred duration. All time shares together must add up to 100%. The scheduler selects the activity which needs a share next and stops it when the preferred duration is reached and it has filled up its share.

The scheduling is performed at midnight for the time interval from the next take role to quit role. The latter time can be extended if planned activities overlap the quit time. Midnight is necessary to synchronize all roles for group task scheduling. Rescheduling during the active time of a role can occur through external requests. Such requests can be modeled by an external agent or by timed events in an input file. Other spontaneous and also secondary activities are handled in the activities themselves.

Moving between places is an activity by itself, but is not scheduled. It is inserted automatically when the scheduled activities are activated and a move is necessary.

**Personal activities**

Personal activities are more specific for the individual and less for the role. How often a person needs a coffee is not scheduled. The frequencies of different personal activities are determined by individual parameters, together with average duration, duration range, location, and priority. For energy consumption and comfort measurements, a simple stochastic model based on equal distribution is sufficient. With the defined frequency and while the role is taken, the individual sends requests to the role stating the requested activity, duration, position, and priority. The role knows which activity is active and sends the request to it if the priority is high enough. The activity then triggers a move to the location, waits for the return and continues.

## MULTI-AGENT MODELING

**The SDL modeling environment**

The used modeling language SDL stands for Specification and Description Language (Olsen et al. 1994). It is based on a hierarchical structure of concurrent processes that are connected by asynchronous communication channels. The processes are modeled as extended finite state machines. State transitions are triggered by asynchronously arriving messages (signals). The messages are queued. Timers can be started and also generate signals to model the simulation time. State transition can result in task executions and in sending new messages. This short description may show that SDL models can be interpreted as multi-agent models. The tasks model all calculations and are expressed in a programming language similar to C.

One major advantage of SDL is that models can be automatically translated into executable computer programs, using a commercial tool environment (Rational). This model-based code generation is the reason why simulators can be created from scratch or by reuse of model components at an abstract model level.

**Example agent structure**

As a case study an open-plan office at the Carnegie Mellon University with the name Robert L. Preger Intelligent Workplace (IW) has been chosen. The South wing IWS has 10 spaces for faculty, students, and visitor work places and a circulation place. Figure 2 shows the agent structure in all five domains.

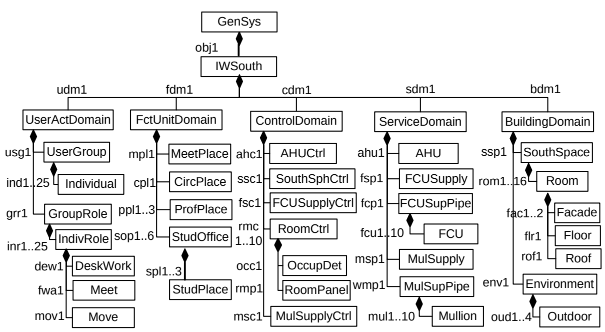


*Figure 2 Agent structure of IWS office simulator. Rectangles denote agent types, the names to the left of the types denote the agent instances*

In this paper we are only interested in the user activity and the functional unit domains. The user group usg1 is composed of 25 individuals ind1..25. The group role grr1 is composed of 25 individual roles and each of them of three activities modeled as agent types *DeskWork*, *Meet*, and *Move*.

Correspondingly, the IWS provides 1 meeting place, 3 places for professors, 6 student office places with 3 work places each, and a common circulation space. This list of 21 work places (desks) means that some individuals time-share a desk or just come to meetings, e.g. undergraduate students that come for coaching.

For the purpose of simulating user behavior, the geometric relations are of little concern. In the service domain, fan coil units provide heating and cooling. In the building domain, adjacencies and facades have to be modeled for the calculation of heat transfer and the necessary heating and cooling properties. Solar radiation loads through windows are considered.

Since the IWS only offers a small meeting place for maximally 8 individuals, most of the larger meetings and lecture are external and are modeled by nominating the exit as destination and origin of moves to and from external activities. In the same way personal activities are modeled as external. Smaller meetings such as student coaching are typically conducted at work places in the IWS. This short description shows that the example features enough different properties to demonstrate a large variety of user behaviors.

**User comfort**

Users are not only characterized by their activities, but also by comfort requirements. Environment comfort requirements are air and radiation temperature, air quality, light level, noise level, and other parameters. The requirements are determined by the individuals' preferences, by the activity performed, and its place. The control system, the service system, and the building interact to provide the comfort at the place. The user can interact directly with the systems, e.g. opening a window, or indirectly by interacting with the control system, e.g. changing set-points.

In the model all requests and interactions are channeled through the users place of activity and at the place the requirements are measured against the actual state variables to determine the comfort level and provide feedback. It cannot be assumed that all requirements can be met, e.g. in open-plan offices the ability to provide different climate values at adjacent places is limited, as has been shown by simulation (Zimmermann 2008a) and by measurements.

User comfort can be expressed as the percentage of dissatisfied persons (PPD) on the basis of Fanger's work. For the purpose of model evaluations with the existing uncertainties, much simpler models have been used (Zimmermann 2008b).

**Message-Transition Charts (MTC)**

Although SDL models have a graphical representation and a formal syntax and semantic, large complex concurrent systems are difficult to design and debug using SDL models only. Therefore, a new, more abstract model has been introduced to visualize the dynamic interaction of agents. Similar to Sequence Diagrams in UML, MTCs model the message interaction between agents in selected situations. Since messages trigger state transitions, the agents are abstracted to just the state transitions participating in the interaction. Figure 3 shows an example with two communicating agents to

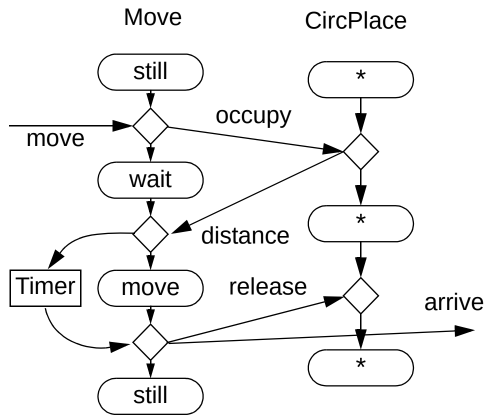


*Figure 3 MTC showing the Move-CircPlace agent interaction*

explain the simple syntax and semantics of MTCs. Ovals are states, diamonds trigger points, arrows denote state transitions and messages. Trigger points can also have conditions.

The *Move* agent is in state *still*, when the message *move* arrives and starts the state transition to *wait*. During the transition it sends a message *occupy* to agent *CircPlace* with the starting and destination place names as parameters. *CircPlace* is in one of two states, *empty* or *occupied*, represented by the symbol *. During the state transition *CircPlace* calculates the Manhattan distance *dist* between the two transmitted places and sends the *distance* back. *CircPlace* also counts the number of simultaneously moving occupants *nP* to decide on the next state. The distance is transformed into a duration by *Move* to set the *Timer* for the state transition to *still* and to send *arrive* to the destination place. This shows a simple situation for the purpose of design and documentation. Such MTCs are manually translated into SDL state transition diagrams and refined with tasks for the necessary calculations. Figure 4 shows the corresponding SDL graph page that formally describes the major state transitions of *CircPlace*. This is one of nine pages of the complete SDL description of this agent type.

## PROTOTYPE EXPERIMENTS

As test-bed the IWS was used to experiment with different user temperature preferences in open-plan offices and with space partitions of different heights between sections. Heat exchange between offices considers air flows through the connecting openings and heat flow through the partition elements. Long wave radiation exchange could be neglected compared to air exchange heat flows. The temperature control was based on occupancy due to individual schedules. Figure 5 shows a result of a full year simulation of the IWS with a Pittsburgh weather file with 6 min. time resolution. All thermal processes have been triggered with the same 6 min. clock period. All activity related events have been processed instantaneously with a time resolution of less than one second.

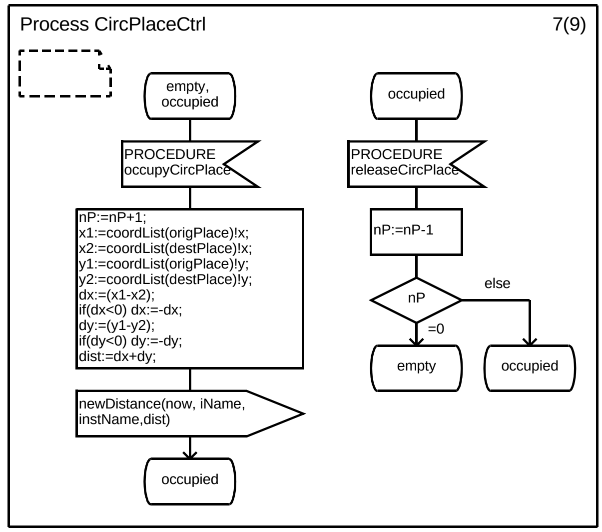


*Figure 4 SDL process diagram of the two state transition of agent type CircPlace in Figure 3*

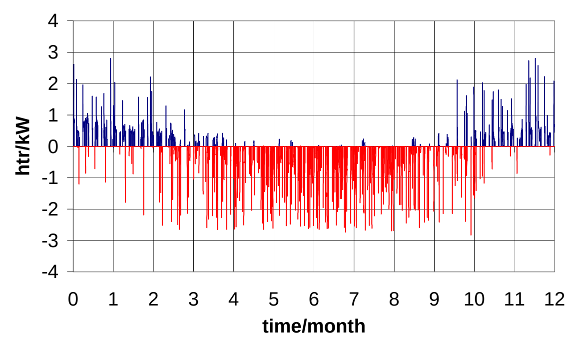


*Figure 5 Heat transfer rate (blue for heating, red for cooling) of one office in the IWS*

*Table 1: Yearly energy consumption of the IWS*

| Exp | occ | setT °C | setBack °C | totalE MWh | heatE MWh | coolE MWh |
|---|---|---|---|---|---|---|
| 1 | no | 20 - 25 | 5 | 5.8 | 0.4 | 5.4 |
| 2 | 24 hours | 20 - 25 | 5 | 23.1 | 7.8 | 15.3 |
| 3 | user | 20 - 25 | 5 | 17.2 | 4.9 | 12.3 |
| 4 | user | 22 - 23 | 7 | 27.4 | 9.9 | 17.5 |

Some results in regard to energy consumption are shown in Table 1 as an example. Exp 1 shows a lower limit for no occupancy, Exp 2 the case for no occupancy control and full time comfort setpoints because of the irregular occupancy at all times. Exp 3 shows the possible energy reduction with occupancy control, despite very irregular user schedules. Exp 4 demonstrates that a narrow setpoint interval increases energy consumption considerably.

The presented activity scheduling for individual users is demonstrated using the SDL modeling and code generation environment (Rational). To give an idea of the effort and model complexity: 40 agent types had to be modeled, generating 287 instances (agents) at run-time. These agents executed (acted) concurrently and communicated asynchronously. Only by applying very strict software engineering principles and tools (Metzger and Queins 2002) was it possible to create and debug the SDL model. Once the model was created, automatic code generation, compilation, and run-time set-up of the prototype took less than one minute. Therefore, debugging the running simulator took place in the abstract SDL model with very short turn-around times.

It has been argued that the agent-based solution is too slow for performance simulations. The measured runtime for simulating the example building for 1 full year with a real weather file with 0.1 hour time resolution took 45 minutes on a Windows laptop with a 1.2 GHz CPU.

The created simulator can be adapted to other geometries as long as the number of spaces and occupants is within the same limits and fan coil units are used for heating and cooling. All geometries, adjacencies, material properties, role assignments etc. are parameters that are entered via files. The extension to larger structures or other service systems or control algorithms is easy, but requires changes to the model.

## CONCLUSION

The project has shown that it is possible to model and simulate individual building users with complex roles, activities, schedules, and personal requirements as independent agents in a homogeneous agent-based performance simulation environment. Combining abstract models and tools from the repertoire of computer science and software engineering with results from the fields of building performance simulation and user behavior field studies, experimental simulation prototypes could be developed with very reasonable effort and execution times. This environment has been used to develop and analyze different user activity and preference models and their influence on energy consumption and user comfort. A few characteristic results have been shown. The environment will be used in future to refine the building functions, e.g. including lighting, and to refine the user model, e.g. user adaptation to the building system. The author expresses the hope that we will better understand user behavior through field studies in buildings with individual user control potentials.

## REFERENCES

Arens, E. et al. 2010, Are ’class A’ temperature requirements realistic or desirable?, Building and Environment 45 (2010), 4-10

Arentze, T.A. and Timmermans, H.J.P. 2006, New theory of dynamic activity generation, TRB 85th Annual Meeting, Washington DC, USA

Bourgeois, D. et al. 2006, Adding Advanced Behavioral Models in Whole Building Energy Simulation, Energy & Buildings Vol.38, 814-823

de Dear, R. et al. 1997, Developing a Thermal Model of Thermal Comfort and Preferences, final report, ASHREA-rp884

Haldi, F. and Robinson, D. 2009, Interactions with window openings by office occupants, Building and Environment 44 (2009), 2378-2395

Hoes, P. et al. 2009, User behavior in whole building simulation, Energy & Buildings 41, 295-302

Hunt, D.R.G. 1979, The use of artificial lighting in relation to daylight levels and occupancy, Building and Environment 14, 21-33

Mahdavi, A. and Pröglhöf, C. 2009, Toward empirically-based models of people’s presence and actions in buildings, Proceedings BS2009, Glasgow, Great Britain

Metzger, A. and Queins, S. 2002, Specifying Building Automation Systems withPROBAnD - a Method Based on Prototyping, Reuse, and Object-orientation, Lecture Notes in Informatics (LNI), P-5

Olsen, A. et al. 1994, Systems Engineering Using SDL-92, Elsevier 1994

Rational, IBM Rational SDL and TTCN Suite 6.3, http://www-01.ibm.com/software/de/rational/

Reinhart, C.F. 2004, Lightswitch 2002: a model for manual control of electric lighting and blinds, Solar Energy 77(1), 15-28

Tabak, V. 2008, User Simulation of Space Utilization - System for Office Building Usage Simulation. Ph.D. thesis, Department of Architecture, Building and Planning, Technical University of Eindhoven, the Netherlands

Zimmermann, G. 2001, A New Approach to Building Simulation Based on Communicating Objects, Proceedings Building Simulation Conference (IBPSA) 2001, Rio de Janeiro, Brazil

Zimmermann, G. 2008a, Individual Comfort in Open-Plan Offices - A Case Study, DDSS2008, Leende, the Netherlands

Zimmermann, G. 2008b, Modelling and Simulation of Individually Controlled Zones in Open-Plan Offices, Proceedings ECPPM’2008, Sophia Antipolis, France