Towards a Context Control Model for Simulation and Optimization of Energy Performance in Buildings

Marco Jahn & Markus Eisenhauer
Department of User-Centered Ubiquitous Computing
Fraunhofer Institute for Applied Information Technology, Sankt Augustin, Germany

Radu Serban, Alfons Salden, & Andries Stam
Almende B.V., Rotterdam, The Netherlands

ABSTRACT: The process of optimizing building energy performance requires complex measurement, estimation and analysis. To address the complexity of designing for energy efficiency, models need to describe a building’s different subsystems: the structure, processes, occupants, control equipment, environmental conditions, etc. For example, a Building Information Model (BIM) describes a building’s structure, while Business Process Models (BPM) can formalize processes going on in a building, and a Context Control Model (CCM) describes heterogeneous devices in relation to a building’s context.

To express the interdependencies between these subsystems an Enterprise Energy Performance Model (E-EPMM) is needed, integrating and extending the subsystems’ models.

In this paper we focus on the physical subsystem (the building and devices) presenting a Context Control Model (CCM) for devices and appliances as an integral part of the E-EPMM. Further, we describe a middleware approach for instantiating and using the CCM and its application in a multi-agent based energy performance simulation framework.

1 INTRODUCTION

Global warming is one of the major problems mankind will face during this century. Modern societies and governments invest huge efforts in reducing CO2 emissions by e.g. reducing the amount of energy produced by fossil fuels. Another strategy is to reduce the amount of consumed energy by optimizing energy efficiency.

For example, due to its large share of total energy consumption, significant savings potential lies in the residential and commercial buildings sector. According to the Smart 2020 Report (The Climate Group, 2008), global building emissions make up 8% of total emissions in 2002 (i.e. 3.36 GtCO2e (Giga-Tonnes of CO2 emissions), excluding the energy used to run the buildings). The report forecasts an increase of this number up to 11.7 GtCO2e in 2020 and at the same time identifies a savings potential of 15% that can be achieved with the help of ICT: “Globally, smart buildings technology could potentially reduce emissions by 1.68 GtCO2e and be worth €187 billion ($295 billion) of energy savings and €29 billion ($45.7 billion) in carbon costs. This value can be captured by ICT and other high-tech companies.” (The Climate Group, 2008).

To increase efficiency and minimize waste of energy, different strategies can be applied. For example, exchanging old equipment with more energy efficient devices, implementing building energy management systems, or motivating people to consume less energy. Of course, only the combined implementation of these strategies will unleash the full potential of energy savings. Further, it is important to consider the whole lifecycle of a building from the design phase to the operational phase.

To allow ICT-based measurement, analysis, and optimization of a building’s energy performance in its different phases we need to identify and be able to modify the different aspects that have an impact on this performance. For example, if a building would be retrofitted with a new wireless Building Management System (BMS), this change would have a huge impact on the energy performance. Or, if business processes should be optimized towards increased energy efficiency, this could affect other processes or the usage of equipment.

We call these different aspects that affect a building’s energy performance the subsystems of a building: The physical subsystem (buildings, devices, environmental conditions in the building), the human subsystem (occupants, with their occupancy and usage behavior), and the enterprise subsystem (enterprise business processes and business goals).

For translating these subsystems into the ICT world, we need models to describe them: For example, a Building Information Model (BIM) describes a building’s structure, while Business Process Models (BPM) can formalize processes going on in a building, and a Context Control Model (CCM) describes heterogeneous devices in relation to a building’s context.

When looking at the different models and the subsystems in reality, we recognize a gap between the real interdependencies and the models. For example, a Building Management System might be completely unaware of the structured data that is available in
a BIM. Vice versa, a BIM might not include information about devices for measuring and managing a building’s energy performance.

To bridge this gap of interoperability, we propose an Enterprise Energy Performance Management Model (E-EPMM) expressing the interdependencies between the subsystems, integrating and extending the subsystems’ models. Such E-EPMM serves as the basis for managing a building’s energy performance throughout its lifecycle. It can be used for design and simulation as well as for energy management in the operational phase.

In this paper we describe the E-EPMM as an integrated model of the different subsystems, focusing on the Context Control Model. In contrast to BIM and BPM no commonly accepted standard or methodology exists for designing a CCM. We describe our approach to modeling and instantiating a CCM by using a middleware for managing heterogeneous devices. We further present a multi-agent based simulation framework, using the E-EPMM for optimizing processes towards energy performance.

2 ENTERPRISE ENERGY PERFORMANCE MANAGEMENT MODEL (E-EPMM)

The goal of the Enterprise Energy Performance Model (E-EPMM) is to overcome issues of interoperability within the ICT-based energy performance management during the whole lifecycle of a building. Katranuschkov et al. have identified three gaps in building design and management practice, namely (1) the lack of a common data repository, (2) the lack of software interoperability, and (3) the insufficient use of simulation and monitoring during the whole lifecycle (Katranuschkov et al., 2011). This leads to the problem that ICT support in the different lifecycle phases is mostly restricted to one phase. The same goes for energy performance management, if even possible. Furthermore, eventual installations in the retrofitting/refurbishment phase also need to be reflected by the models for energy performance management.

The E-EPMM extends current Energy Performance Models by incorporating and integrating multiple dimensions: the physical sub-system, the human sub-system, the enterprise sub-system, and the general surrounding environment. By explicitly incorporating the enterprise as actor in this ecosystem, the energy performance model is expected to better adjust to the characteristics of the business domains.

The goal is to integrate several aspects of an industrial sector: (1) The processes and governance of business and ICT infrastructures, particularly the aspects concerning optimization of cost-efficiency and other business-related Key Performance Indicators (KPIs). (2) The operations and management of the Building and ICT Infrastructure, particularly the optimization of energy-efficiency and its related KPIs (such as, GHG emissions reduction). (3) The agent-based modeling of actors, both stakeholders and supporting systems, with their needs and business-related issues; this includes devising of agent-based negotiation schemes which enable robust and efficient enterprise energy management in terms of the identified KPIs under dynamic and unpredictable situations.

Figure 1 shows the different parts of the E-EPMM and its role in the context of ICT-based energy performance management.

The Building Information Model (BIM) is an abstract representation of the physical and environmental aspects of the building ecosystem, incorporating architectural metadata and environmental parameters. Several standards for BIMs exist, e.g. IFC core ISO/PAS16739, and formats supporting this standard - gbXML, CityGML, landXML, or CIMSteel. The BIM is needed in the E-EPMM to model the physical subsystem of a building.

The Business Process Model (BPM) is an abstract representation of the functions, processes which the building supports in its daily use by its occupants. It models the enterprise subsystem. From the existing standards for representing BPM, e.g. BPMN/BPEL, BPDM, SBVR, AQPCF, BMM, PRR, we focus on open standards such as BPMN, supported by most commercial tools used in CAD/architecture design practice.

The Context Control Model (CCM) describes the heterogeneous devices in relation to a building’s context and is part of the physical subsystem. Such devices can be monitoring equipment like sensors and smart meters but also appliances used in business processes. Further, the E-EPMM needs to include the effect of additional factors, such as environment and human occupancy, on the building energy performance.
2.1 Interoperability Issues

In order to support interoperability of our proposed device models and energy performance simulation tools with existing CAD and energy analysis tools and data standards, we use gbXML as a starting building block for describing CCM and the relation between devices in the control system and occupant activities. We extend the Green Building Energy Performance information with:

1. Location/spatial information (specific for BIM);
2. Process/activities and performing roles information (specific for BPM);
3. Equipment/device information (specific for CCM);
4. Measurement/performance indicators information (specific for KPI);
5. Energy cost/impact information (specific for ANM).

Table 1 shows a possible mapping between the elements of gbXML and the elements of E-EPMM.

<table>
<thead>
<tr>
<th>gbXML Element</th>
<th>Possible Mapping to E-EPMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>gbXML/Campus</td>
<td>BIM/BuildingPosition</td>
</tr>
<tr>
<td>gbXML/Construction</td>
<td>BIM/BuildingStructure</td>
</tr>
<tr>
<td>gbXML/Schedule</td>
<td>BPM/TenantActivities/Activity/Schedule</td>
</tr>
<tr>
<td>gbXML/WeekSchedule</td>
<td>CCM/Device/Schedule</td>
</tr>
<tr>
<td>gbXML/DaySchedule</td>
<td></td>
</tr>
<tr>
<td>gbXML/IntEquip</td>
<td>CCM/Equipment/Devices/Sensors</td>
</tr>
<tr>
<td>gbXML/ExtEquip</td>
<td>CCM/Equipment/Devices/Actuators</td>
</tr>
<tr>
<td>gbXML/Meter</td>
<td>ANM/EnvironmentalSpecifications</td>
</tr>
<tr>
<td>gbXML/Weather</td>
<td>BIM/EnvironmentalConditions</td>
</tr>
<tr>
<td>gbXML/AirLoop</td>
<td>KPI/InternalAirQuality</td>
</tr>
<tr>
<td>gbXML/Meter</td>
<td>CCM/Equipment/Devices/Sensors</td>
</tr>
<tr>
<td></td>
<td>ANM/EnvironmentalSpecifications</td>
</tr>
<tr>
<td>gbXML/Zone</td>
<td>BIM/OccupancyInformation</td>
</tr>
<tr>
<td></td>
<td>KPI/EnvironmentQuality</td>
</tr>
</tbody>
</table>

3 CONTEXT CONTROL MODEL

The Context Control Model (CCM) is part of a building’s physical subsystem, describing heterogeneous devices and their relation to the building (e.g. location in the building). Such model is particularly important to bridge the gap between the early design phase and the operational phase as it provides a common information model for devices (e.g. employed by a BMS) and a relation to the Building Information Model.

During the lifecycle of a building both, the structure and the installed devices can undergo significant changes, requiring an adaptable and extensible model that is able to reflect these changes.

3.1 Basic design of the CCM

The most basic dependencies the CCM describes are between devices and locations. Both, devices and locations themselves are described in more detail by their respective taxonomies. For example, devices can be sensors, actuators, computers, smart meters, and so on. Locations are buildings, rooms, etc. where the location information needs to be inline with existing standards like gbXML. Another information that needs to be modeled are the device capabilities. We need to know e.g. if a device can measure or influence environmental conditions, if it can count people or simply be switched on and off. Figure 2 shows a simplified example of a sensor instantiation in the CCM. Sensor TempXY is an instantiation of class TemperatureSensor, which means that it has the ability to measure the EnvironmentalCondition Temperature. This relation is implicitly inherited from the TemperatureSensor class. In Chapter 4 we will describe how such a model will be instantiated so it can be used by applications.

3.2 Application of the CCM

The CCM (as a part of the E-EPMM) provides important information about the physical subsystem during a building’s lifecycle. In the early design and simulation phases of a building it can be used to model devices that are known a priori or to simulate potential devices or Building Management Systems that could be installed. In the operational phase the CCM contains the building’s state and provides important information to energy management and performance optimization.

Now that we have a defined a model for describing a building’s physical subsystem (especially devices), we need to think about how to apply and use such model in real-world applications. This is a serious issue, because – as described above – we deal with many different kinds of technologies and communication protocols. As we want to be able to deal with these different technologies and be open to ever-changing environments, we need to abstract from...
specific device and subsystem technologies and create a common information and access layer to work with.

Our approach to achieve a high degree of interoperability is to use middleware to manage and instantiate the CCM. Our goal is to use existing standards where possible and to develop reusable and extensible software components and models. In the next section we describe our middleware approach to achieve this goal and how the LinkSmart middleware helps to fulfill these requirements.

4 LINKSMART MIDDLEWARE FOR MANAGING HETEROGENEOUS DEVICES

LinkSmart middleware is used to connect the CCM with the real-world devices. It provides software components for device management and messaging infrastructure to access devices. This is a major advantage because we can integrate existing devices/subsystems into our applications quite easily.

The LinkSmart Middleware is a generic middleware for developing Ambient Intelligence (AmI) applications (Eisenhauer et al., 2011). It is the result of the FP6 European Project HYDRA (HYDRA Project).

LinkSmart provides a framework and software development tools for integrating heterogeneous networked devices into AmI applications. Further, LinkSmart comes with software components that provide functionality typically for AmI applications (e.g. message encryption, event management, or device discovery). In the following we will provide a short overview of the LinkSmart software architecture and concepts we apply and develop further in the domain of energy efficient buildings.

4.1 LinkSmart Architecture

LinkSmart implements a service-oriented architecture providing to software developers a set of components (called managers) they can select from, depending on their specific requirements (see Figure 3). This architecture also adheres to the principles of loose coupling and separation of concerns.

Each manager encapsulates a set of operations and data that realize a well-defined functionality. Some of these managers are essential (e.g. Network Manager) while others provide optional functionality (e.g. Context Manager or Storage Manager). Each manager has a clearly defined role, offering a set of services to be used by other managers or application level components. Further, as LinkSmart aims at supporting the development of distributed AmI applications, managers can be deployed on different hosts, communicating via Web Services. In consequence LinkSmart supports the development of scalable applications, from simply connecting two computers to full-fledged pervasive environments supporting e.g. security, distributed storage and context awareness.

4.2 Core Managers for eeBuildings Applications

LinkSmart has been and currently is applied and further developed for developing smart home (Jahn et al., 2010) and energy efficient building environments (Jahn et al., 2011; SEEMPubS Project; Adapt4EE Project; SEAM4US Project). Therefore, we currently strive to define a subset of LinkSmart optimized for energy efficient building environments. Such subset also includes the core managers implementing the basic functionality of LinkSmart:

A basic concept of LinkSmart is to abstract from heterogeneous devices and network protocols and to provide common Web Service interfaces for devices. This means, every LinkSmart device exposes its interface as Web Service (and managers do so as well). A device service can be identified by its HID (HYDRA-ID), which is unique inside a LinkSmart network. A LinkSmart Network is formed by distributed Network Managers that take care of the communication among devices and managers. Every service can register itself at a Network Manager and thus take advantage of communicating inside the LinkSmart network.

The Network Manager enables network communication by creating an overlay P2P network that implements SOAP Tunneling as transport mechanism for Web Service calls (Milagro et al., 2008). This concept allows direct communication among all

\(^1\) http://sourceforge.net/projects/linksmart/
devices inside a LinkSmart network, no matter if they appear behind a firewall or NAT (Network Address Translator). Further, the HID addressing scheme allows devices to transparently publish and use services anytime anywhere regardless of network boundaries or fixed service endpoints. If a device wants to consume a service of another device, the Network Managers of both devices take care of routing the Web Service calls, using the services’ HIDs.

Another core component of LinkSmart is the Event Manager. For smart home and eeBuilding applications it is essential to be modular, extensible and provide low coupling of components, as set-ups can change when devices are removed or new devices are added to the environment. The Event Manager addresses these requirements, implementing a publish/subscribe mechanism for LinkSmart services. Thus, we are able to develop loosely coupled applications, which are flexible enough to face the requirements of dynamic AmI environments. The Event Manager handles all subscriptions and is responsible for publishing events via a Network Manager, compliant to the LinkSmart communication model.

4.3 Devices, Subsystems and Proxies

A core requirement of AmI applications is to support a wide variety of heterogeneous devices and communication protocols. This is also important in the domains of smart and energy efficient buildings. Especially in the refurbishment phase of a building it is necessary for an ICT system to be open to the removal, introduction or exchange of devices.

LinkSmart comes with software components and tools to foster the seamless integration of devices into new or existing LinkSmart applications. The proxy concept allows developers to hide the complexity of the underlying device technology and expose the functionality as LinkSmart Web Services. In AmI application design for larger spaces like commercial or public buildings it is important to take into account existing ICT installations like BMS, or security systems. In LinkSmart the integration of such subsystems can also be realized by proxies. Of course, the complexity of a subsystem proxy depends on many factors, e.g. availability of an open API, communication protocol, etc.

As tool support LinkSmart provides a model driven approach for device integration. The device ontology supports semantic interoperability between the different types of devices. The device ontology is based on the FIPA device ontology specification (FIPA, 2002) and the AMIGO project vocabularies for device descriptions (AMIGO, 2006). It contains basic information about devices e.g. device description and manufacturer (Sarnovsky, 2007). The LinkSmart Device Development Kit utilizes the device ontology to help developers creating LinkSmart devices (proxies). Further, it is used during the runtime device discovery process for semantic device discovery (Kostelnik et al., 2008).

4.4 LinkSmart and the CCM

The original LinkSmart device ontology defines a general taxonomy of device classes which can be used and extended for specific application domains. An ontology for smart energy efficient buildings is currently being developed in various research projects, e.g. SEEMpubS, Adapt4EE. This ontology will contain information about common devices and capabilities in the energy efficient buildings domain. For example, a device can have the ability to measure temperature or luminance or other environmental condition. Devices can also have the capabilities to control certain environmental conditions like the temperature while other devices may just have an energy consumption profile.

There is a clear need for having an explicit representation of the role of each device in the energy performance, e.g. through its direct consumption or other building systems and occupant activities that depend on it. For this purpose, we use the mappings indicated in Table 1. This information is used by an agent-based Context Manager, which is able to recognize the impact of devices and occupant activities on the building energy performance, e.g. whether a device is wasting energy without servicing any ongoing occupant activity.

4.5 LinkSmart Implementation

The LinkSmart reference implementation is built for OSGi environments. OSGi adheres to the principles of service- and component-oriented programming, providing a Java-based modular service platform. Components (called bundles) can be installed, started and stopped at runtime, not requiring a reboot of the whole environment. Each bundle publishes services that can be looked up and consumed by other bundles. Consequently, LinkSmart managers are available as OSGi bundles that can be plugged together on demand. Further, each manager publishes a SOAP Web Service interface to facilitate remote communication among components.

Components for semantic device discovery are implemented in .NET. As all communication is routed through the Network Managers, the underlying implementation details of a manager or device proxy are not important.

LinkSmart has been designed to meet the requirements of different target users: (1) Middleware developers extend the basic functionalities of the core middleware components or develop branches for specific domains. (2) Device developers are responsible for creating device proxies and keeping
track of the device ontology. They program the translation between devices or subsystems and LinkSmart Web Services. (3) Application developers build applications by selecting (and if necessary extending) existing LinkSmart managers and device proxies.

5 AGENT-BASED SIMULATION AND OPTIMIZATION FRAMEWORK

The added value of using a Multi Agent-based approach (Kuehne et al., 2005; Ito et al., 2008; Cioffi-Revilla, 2010) to model the building and its elements, including the control equipment, the occupants and their activities, and the environmental factors as active components engaged in interaction as part of the Building Ecosystem, has been proven (Zimmermann, G. 2006b; Zhou et al, 2011). By applying it, we are able to better understand and adjust the interactions of building elements, to meet their intended goals in an energy-efficient manner. The primary objectives of an agent-based approach to coordination are to study the interactions between relatively autonomous entities which lead to emergent properties, such as the ability to coordinate actors to achieve a global conflict-free and feasible schedule: Can the different actors of the building ecosystem, self-interested but unable to achieve their goals without collaboration, achieve near-optimal local coordination, in the form of a global schedule which is both efficient (e.g., Pareto optimal) and robust? How can these agents make and coordinate their decisions in order to achieve a globally efficient and robust schedule in a partially observable and non-deterministic environment?

5.1 Typical CCM Usage Scenario

The typical scenario employed for Energy Efficiency Analysis using agent-based simulation is described below.

1. The Architect defines the Building Design, and selects one architectural change, e.g. conversion of a room in the building for a new purpose or for improved energy performance tuning.
2. The maximum allowed and the actual measured/estimated Occupancy Load Factor for the building segments estimated to be impacted by the change (e.g., how many occupants are expected per room, per m² of space, per process, or during specific time intervals) are defined as part of the Occupancy Model (OM). This is based on international, national and local property-specific industry standards, defining maximum and/or standard values.
3. Using inputs from the building’s Monitoring and Measurement Framework, the measurements from existing control equipment supporting the Building Energy Performance Management are integrated into the current Building Model.
4. Based on the Current Business Process Model with associated services (BPM), enhanced with occupancy information from the Occupancy Model, the BPM/Activity Designer provides the set of activities impacted by the room conversion. From the set of building activities and occupant activities supported, the so-called Future Business Process Model is produced, which includes the new to be supported processes.
5. The Architect initiates a What-If analysis to study the impact on Energy consumption with different occupancy load factors and different equipment installed in the room. This analysis is done using an agent performance simulation platform, which takes as inputs the Building Information Model extracted from the current CAD design, as well as the Business Process Model and Occupancy Model. The simulation platform has a configurable Energy Performance Model, which allows one to either estimate, based on earlier measurements, or to manually specify the impact of activities associated with devices and occupants, on the building Key Performance Indicators.
6. Based on the results of the What-If analysis, a set of alternative designs is selected, each containing the changes and ensuing combinations of effects associated with each type of change.
7. Once a short list of designs has been selected for further refinement, a simulation of the integrated model of the new room as part of the larger building ecosystem is done. Here, populating data with measured and learned occupancy patterns in the neighboring building segments is done.
8. Subsequently, an end-to-end stress test is done, for the measured or learned occupancy model, with an estimation of energy consumption range in worst case scenario. The result of this phase is a set of differences of estimated financial costs, power consumption, CO² emissions and waste produced by the change of the building ecosystem generated by the room being transformed.
9. The Architect selects the elements of the design which meet the purpose of reconversion and present energy-efficiency, having withstood the stress test.

5.2 CHAP Agent-based Model

For the energy performance simulation, we employ our existing multi-agent modeling and simulation tool (Munroe et al, 2005) which supports constraint solving for scheduling, collaborative decision mak-
ing, distributed coordination and optimization through learning and negotiation. This tool, called CHAP, for Common Hybrid Agent Platform, provides support for adaptation and evolution of application-specific data models and logic, support for integration with existing applications and deployment on different types of enabling infrastructures, such as wireless sensor mesh or mobile ad-hoc networks. CHAP uses a general-purpose, evolvable associative memory, a set of reusable AI modules implementing building blocks of intelligent behavior, a configurable agent component deployment engine allowing agents to run on mobile devices and to interact with sensors, and a toolset for data visualization. These components can be tuned and adapted to a particular application or business domain using diverse (enabling) ICT infrastructures.

Each device part of Building’s Context Control Model is represented as an autonomous CHAP agent. Its activity is the result of composition of multiple aspects of computation – data acquisition and processing, knowledge extraction and manipulation, and resource planning, task and energy management. This corresponds to a goal-aware, utility-aware, and adaptive rational agent, whose lifecycle is SENSE-REFLECT-PLAN-ACT, i.e. simple task selection, composition and chaining rules. The computing cycle of a CHAP agent is formalized as a transformation of the internal state based on environmental conditions observed:

\[ A' := \text{next\_state}(A,\text{Env}) = \text{Sense}(A,\text{Env}) \circ \text{Reflect}(A,\text{Env}) \circ \text{Plan}(A,\text{Env}) \circ \text{Act}(A,\text{Env}) \]

These lifecycle activities are supported by the main specialized components of the CHAP platform: LINKS component is responsible for Interaction Management with external world, i.e. Acquiring knowledge through Sensing and Applying plans in practice through Acting; NETS – Task Management, Planning and Scheduling; MEMO – Knowledge Management; MOTOR – Energy and Lifecycle Management. MEMO can be viewed as a distributed, extensible and adaptable tuple space, which stores all observations of the agent about its environment (acquired from LINKS), all knowledge of how to use its existing capabilities (NETS), and all plans and detailed actions for the agent to sustain itself and to change its environment, i.e. actions (including structural transformation and adaptations of CHAP components themselves) to be performed through its actuators (MOTOR).

Example: The Enterprise Energy Performance Management Model (E-EPMM) is defined as a multi-objective function of the Occupancy Model (OM), which is optimized by means of negotiation between self-interested agents with different objectives.

\[ \text{EEPMM}(\text{OM}) = \text{MultiObjectiveOptimisation-ByNegotiation}(\text{Agents, Environment}) \]

Environment is a dynamic set of time-dependent constraints, representing the static aspect of the domain:

\[ \text{Environment} = \text{Env}(t, \text{Resources, ResourceConstraints, NegotiationSchemes}) \]

\[ \text{ResourceConstraints} = \{\text{Resource}(i,t) \mid i=1..n, t=1..m\}, \text{ with } \text{Resource}(i,t) = \langle \text{BIM}(\text{OM},t), \text{BPM}(\text{OM},t), \text{CCM}(\text{OM, BIM, KPI, t}), \text{KPI}(\text{BIM}) \rangle \]

Each agent is seen as a composition of aspects, including prioritized objectives Objectives(t), which is a set of available activities with an indication of the resources required, and their utility as dynamic valuation. The agents try to rationally apply the available activities in order to apply the given objectives, using their resource constraints on available resources:

\[ \text{Agents} = \{\text{Agent}(i) \mid i=1..n \text{ agent index}\}, \text{ with } \text{Agent}(i) = \langle \text{Agent}(i), \text{Objectives}(t), \text{OM}(t), \text{ResourceConstraints}(t,i) \rangle \]

\[ \text{Objectives}(t,i) = \langle \text{Activity}(i,j), \text{Resources}(i,j), \text{Utility}(i,j,t) \rangle \mid j=1..m \text{ is index for Activities} \]

The impact of Occupancy Model (OM) on the current Building Information Model (BIM) is given as:

\[ \text{BIM}(\text{OM},t) = \langle \text{BuildingStructure}(\text{BIM}), \text{Environment}(\text{BIM}(\text{OM},t)), \text{Equipment}(\text{BIM}(\text{OM},t)) \rangle \]

The main BIM-related parameters influencing the building energy performance are: building’s structure position and topology (e.g., large volume, high rise wrt its external environment, large windows, many doors, materials, isolation), external conditions, occupants with their activities, and equipment.

\[ \text{BuildingPerformance}(\text{BIM}) = \text{BIM}(\text{BuildingStructure, ExtEnvironmentalConditions, Occupants, Equipment}) \]

The only truly dynamic parameters that can be influenced are equipment and occupant activities.

\[ \text{Equipment} = \text{CCM}(\text{EquipmentPlacement} \circ \text{EquipmentActivationStatus} \circ \text{EquipmentPerformance}) (\text{BIM,OM,t}) \]

Occupants are expressed in terms of their Occupant density per BIM element and BPM element:

\[ \text{Occupants} = \text{OM}(\text{BPM,BIM,t}) \]
BPM is influenced by occupant activities, i.e. Services, Activities and Roles changing over time. In an adaptive Energy Performance Management Model, CCM must adjust itself to the Occupants’ activity patterns.

The resource dependency and impact of device/agent activities on the building energy performance are expressed as evaluations:

\[ \text{CCMEnergyKPIImpact} = \text{KPI}(\text{CCM}, \text{BPM}, \text{BIM}, t) \]

The objectives of agents which are part of the CCM sensor-actuator device cloud are expressed in terms of this energy performance impact. These evaluations are influenced by the building occupants’ activity patterns, building performance metrics, and available coordination/negotiation schemes.

### 5.3 Implementation of CCM using the CHAP Agent-based Modelling and Simulation Framework

The problem of coordinating the CCM sensor-actuator device cloud for energy efficiency is viewed as a constraint satisfaction problem, more precisely a Distributed Resource-Constrained Multi-Project Scheduling Problem with uncertainty and partial knowledge (dRCMPSU), in which the resource constraints are provided by device performance specifications and occupants’ activities impact on building’s energy performance KPIs. For solving RCMPSU, we implemented agent simulations based on CHAP agent platform, able to accommodate autonomous agent negotiations (Ter Mors et al, 2008; Mao et al, 2008; Mao et al, 2009) or collaborative decision support problems through voting (Ferro et al, 2009).

An RCPSP problem involves the construction of a “project” schedule specifying for a list of activities the start and/or end-time in such a way that a set of resource constraints (time and other resource usage, such as energy, computing time, etc) are satisfied, and a set of objective functions (describing objectives such as minimization of impact on building energy performance) is optimized.

Each “project” in CCM is a list \( A = \{a_0, a_1, ..., a_n\} \) of activities available for the CCM sensor-actuator device cloud to support the building services (e.g., movement detected, turn on lights, start heating, increase heating, emergency exit door opened, start sprinklers, etc). Each activity \( a_i \in A \) has an estimated processing time, or duration, which is subject to uncertainty factors, such as occupants’ movement, availability of computing time, etc. It also has a start time (or release), end time (or deadline), and a set of dependencies on other activities and on resources \( R = \{r_0, r_1, ..., r_m\} \). CCM is an activity network emerging as result of interaction of multiple resource-constrained and inter-dependent “projects” (for instance, an actuator device “project” depends on a sensor device “project”). All devices and controls are represented as “project” agents, while the critical resources such as available energy quatum, time, computing time, but also device services (such as sensing, notification), are represented as “resource” agents. Some agents can be “resource” agents as well as “project” agents.

The internal computation and evolution cycle of each agent \( a_i \in \text{Ag} \) can be seen as an activity network (represented as task compositions \( T(a_i) = < t_0 \circ t_1 \circ \ldots \circ t_i \circ \ldots \circ t_n > \), with \( t_i \in A \)) undergoing continuous transformations \( T(a_i) = < t_0 \circ t_1 \circ \ldots \circ t_i \circ \ldots \circ t_n > \rightarrow T'(a_i) = < t_0 \circ t_1 \circ \ldots \circ t_i \circ \ldots \circ t_n > \). The transformations are triggered by selection of alternate activities based on highest utility, as provided by the objective function valuation. As mentioned, agent’s valuations of task utilities depend on the energy performance impact of each task/activity, utility which is time-dependent.

We define set \( E : \text{PowSet}(\text{Ag}) \times \text{PowSet}(T) \times \text{PowSet}(\text{Time}) \rightarrow R \) as the set of objective evaluation functions defined over task compositions, which evaluates societal performance aspects (e.g. total execution time, total resource cost, etc). This allows selection of the best candidate from the alternative candidate task compositions or to adapt them from a structural, functional or organizational perspective. Each agent can have its own objective function, used for negotiating resource exchange with other agents during operation that can be altered on-the-fly, based on new acquired knowledge.

Each event \( e \) occurring in the CCM sensor-actuator device cloud is represented as a tuple \( (a_i, \tau, c, t) \) where \( a_i \in \text{Ag} \) is the set of agents in charge of the task, \( \tau \in T \) is the task, part of a set of activities \( A \) plus all uncertainty-producing events (incidents) \( i \in I, c \in C \) is a power set of context elements and \( t \in \text{Time} \) is the set of relevant time points.

The goal of the scheduling problem is to find a suitable set of time points \( t_i \in \text{Time} \), such that the impact of an incident \( i \in I \) on an agent or group of agents \( a_i \) working on a task \( \tau \) is minimal, while taking into account the context \( c \) and the times \( t_i \) at which all tasks need to take place.

For estimating the performance of potential solutions, and as such constructing recommendation schemes that fit new incidents occurring, the Windmill approach allows comparing new events to past ones. This is done by an estimation function for \( v \) which weighs past solutions by their relevance using a relevance metric \( \delta \) and aggregates the associated performances according to its weights. The performance function associated with each event \( e \) is described by \( v: A \times G \times T \times C \times \text{Time} \rightarrow [0,1] \); higher values correspond to desirable outcomes and lower values to less desirable ones. The performance function comprises measurement of, for instance, the op-
erational performance (e.g., response times) individual judgments (i.e., ratings) or the workload.

This estimation of performance and concurrent running of all optimizations implements a distributed constraint satisfaction algorithm, which is able to select the most preferred solution, i.e. the activation schedule of a set of activities, which maximizes a specific set of objective functions.

6 CONCLUSION

In this paper we present an innovative agent-based approach for energy performance modeling and simulation in buildings, based on an advanced Context Control Model. We describe our middleware approach to model and manage heterogeneous devices (as part of the CCM) in energy efficient buildings, and to simulate their activity.

This approach takes into account the need to adjust building monitoring and control equipment based on occupants’ activities, and views the ecosystem formed by occupants and the monitoring and control devices as the main factors influencing the internal environmental conditions of a building. Representing explicitly the occupants and the effects of their activities on the building ecosystem makes it possible for building designers to incorporate energy efficiency analysis in the early phases of building lifecycle and to produce better performing buildings. It also allows one to provide a more granular management and control of building equipment (sensors and actuators), and to use this information to achieve a higher building energy performance.

7 ACKNOWLEDGEMENTS

The work presented in this paper has been carried out in the Adapt4EE project, co-funded by the European Commission within the 7th Framework Programme (FP7/2007-2013) under grant agreement n° 288150.

8 REFERENCES