SEVENTH FRAMEWORK PROGRAMME

ICT systems for Energy Efficiency

Project Title:

Occupant Aware, Intelligent and Adaptive Enterprises

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 288150

Deliverable

Training and Model Optimization

Deliverable No.		**D7.3**	
Workpackage No.	**WP7**	Workpackage Title and task type	Pilot Implementation, Simulation Model Training & Evaluation
Task No.	**T7.4**	Task Title	Agent Based EMS Training / Learning and Adapt4EE model Optimization
Lead beneficiary		**ALMENDE**	
Dissemination level		**PU**	
Nature of Deliverable		**R**	
Delivery date		**September 2014**	
Status (F: final; D: draft; RD: revised draft):		**F**	
File Name:		**Adapt4EE-Deliverable-D7.3.docx**	
Project start date and duration		**01 November 2011, 36 Months**	

Authors List

Leading Author (Editor)			
Surname	Initials	Beneficiary Name	Contact email
van Krevelen	D.W.F.	ALMENDE	rick@almende.org

Co-authors (in alphabetic order)				
#	Surname	Initials	Beneficiary Name	Contact email
1.	van Beusekom	S.	ALMENDE	suki@almende.org
2.	Stam	A.W.	ALMENDE	andries@almende.org
3.	Stellingwerff	L.	ALMENDE	ludo@almende.org

Reviewers List

List of Reviewers (in alphabetic order)				
#	Surname	Initials	Beneficiary Name	Contact email
1.	Elmasllari	E.	FIT	erion.elmasllari@fit.fraunhofer.de
2.	Krinidis	S.	CERTH	Krinidis@iti.gr
3.	Simoes	P.	ISA	paulo.a.f.simoes@gmail.com

Document history			
Version	Date	Status	Modifications made by
0.5	June 2014	Draft version of the table of contents.	ALMENDE
0.8	September 2014	Second Draft, integrating received comments from plenary meeting presentation	ALMENDE
0.9	October 2014	Released Deliverable for peer review	ALMENDE
1.0	October 2014	Final version refined based on peer review	ALMENDE

List of definitions & abbreviations

Abbreviation	Definition
ACADIA	Association for Computer-Aided *Design* in Architecture
AEC	Architects and Engineering Community
BIM	Building Information Modelling
BPM	Business Process Model
BPMN	Business Process Modelling Notation
BPS	Building Performance Simulation
CAD	Computer-aided Design
CERTH	Centre for Research and Technology Hellas
CIB	Conseil International du Bâtiment (Int. Council for Building)
CIMA	Centre for Applied Medical Research (UNAV)
COBIT	Control Objectives for Information and Related Technologies for IT governance
CUN	Clínica Universitaria de Navarra
D&E	Design & Engineering
eCAADe	Education and research in Computer Aided Architectural Design in Europe
E2B EI	Energy Efficient Building European Initiative
EC	European Commission
ECM	Enterprise Content Management
E&D	Engineers & Designers
EE	Energy Efficiency
EORBMR	Enterprise Open Reference Business Model Repository
eTOM	enhanced Telecom Operations Map
EU	European Union
FhG	Fraunhofer Gesellschaft zur Förderung der angewandten Forschung e.V.
FP7	7th Framework Programme (EU)

GSRT	General Secretariat for Research and Technology (CERTH)
IBPSA	International Building Performance Simulation Association
ICT	Information and Communications Technologies
IESSA	Installations and Energy Section of the School of Architecture
IF	Impact Factor (ISI / Thomson Reuters)
IoT	Internet of Things
ISI	Institute for Scientific Information (Thomson Reuters)
ITI	Information Technologies Institute (CERTH)
ITIL	Information Technology Infrastructure Library
IPR	Intellectual Property Rights
MFB	Method and Functional Building Block
NCEUB	Network for Comfort and Energy Use in Buildings
OMI	Open Models Initiative
SaaS	Software-as-a-Service
SoA	State of the Art
SME	Small Medium Enterprise
STREP	Specific Targeted Research Project
SWOT	Strengths, Weaknesses, Opportunities and Threats (Analysis)
ToC	Table of Contents
VA	Visual Analytics
VPN	Virtual Private Network

Executive Summary

The efforts reported in this deliverable concern Adapt4EE Task T7.4 ("Agent Based EMS Training/Learning and Adapt4EE Model Optimization"), particularly the definition, execution and reporting on Adapt4EE system training programmes for enterprise network operators from the considered pilot domains. These training programmes are delivered in the form of interactive training or learning algorithms used to optimize the enterprise models for intelligent agent-based simulation. This optimization relates to both the adaptation of enterprise models to new domains or designs, and also the calibration of enterprise business process models based on actual measurements obtained from pilot sites.

This deliverable defines *optimization algorithms* for (automated) adaptation the enterprise simulation models to specific domains. These algorithms are to a large extent based on principles of self-organization, thus yielding behaviours related to process creation, process management and equipment allocation that inherently accommodate variations in the scenery as introduced by a designer.

Furthermore, *training algorithms* have been specified for calibrating the enterprise simulation models to more closely match realistic behaviours as observed in the pilot sites. The calibration problem has been formalized mathematically as an optimization problem and two optimization approaches have been provided, one is BPM driven and explores and scores all activity sequences possible with the given BPM, the other takes a bottom-up evolutionary approach and tries to reconcile increasingly more observations with fewer and fewer business process instances. Each of these algorithms has been *validated* showing they adhere to the formalized optimization rules.

As part of the Adapt4EE system, integration aspects are discussed concerning how to *prepare the raw measurement data* obtained from pilot sites for successful application of the training algorithms, as well as how to *apply the training algorithms* in order to obtain calibrated, domain specific enterprise simulation models.

Lastly, an application user manual is provided for key actors (e.g. enterprise network or building management operators) to provide their feedback and adapt the enterprise network models as needed.

Table of Contents

List of figures

List of tables

1 Introduction

1.1 Scope of the Deliverable

This deliverable reports on the implementation and validation of training/learning algorithms used by the intelligent agent-based enterprise model simulation (EMS) of Adapt4EE, corresponding to the results of the activities that took place in Adapt4EE Task T7.4 ("Agent Based EMS Training/Learning and Adapt4EE Model Optimization").

The efforts in Task T7.4 reported here concern the definition, execution and reporting on Adapt4EE system training programmes for enterprise network operators from the considered pilot domains. In the context of intelligent agent based simulation, these training programmes for enterprise network operators are delivered in the form of interactive training or learning algorithms used to optimize the enterprise models for intelligent agent-based simulation. The aforementioned programmes for training the Adapt4EE system, referred to here as training algorithms, are wrapped as such within user interfaces for key actors (including enterprise network operators and building management operators) to provide their feedback and adapt the enterprise network models as needed.

The main goals of Task T7.4 are:

- To define *optimization algorithms* for (automated) adaptation the enterprise simulation models to specific domains;
- To define *training algorithms* for calibrating the enterprise simulation models to more closely match realistic behaviours as observed in the pilot sites;
- To *prepare the raw measurement data* obtained from pilot sites for successful application of the training algorithms;
- To *apply the training algorithms* in order to obtain calibrated, domain specific enterprise simulation models;
- To *validate the algorithms* by examining the calibrated and optimized domain specific simulation results; and
- To provide a specific methodology or user manual for key actors (e.g. enterprise network or building management operators) for collecting the necessary feedback to be fed into the pilot requirements as well as for the adaptation of the enterprise models, if needed;

1.2 Adapt4EE project concept[1]

Energy Efficiency is considered to be a key component of the European energy policy underlying the fundamental objectives of the European Union's (EU) 2020 strategy. Recent and past surveys indicated that buildings are a major constituent of the urban ecosystem, accounting for almost 40% of the overall energy demand in Europe [1]-[2]. Construction products (and especially those of commercial use) constitute energy intensive systems that comprise energy demanding assets & facility operations and, most importantly, occupants who are the main driving operational force, performing everyday business processes and directly affecting overall business performance and energy consumption.

Extensive industrial practice throughout the years and respective market surveys demonstrated that most crucial decisions concerning construction products happen in the early phases of the design process. Specifically, the findings of recent research studies indicated that appropriate design improvements, tailored with the support of building performance simulation software, could reduce energy use in both existing and in new building envelopes [3]-[4]. In this context, early design products comprise features that determine to a large extent energy performance and thus can provide critical evidence to simulation and analysis tools for thorough evaluation of design alternatives. To cope with this, modelling and simulating the energy efficiency of buildings and various facilities semantics has now been established as an integral part of the design process and many simulation tools are commercially available to designers & engineers.

Adapt4EE aims to address several shortcomings of existing and rather complex building tools, such as the lack of a holistic and systems-based view of buildings, the efficient separation of algorithms and simulation that will stimulate key players (D&E community) to easily assess the energy use on specific attributed domains. The main purpose of the project is to develop a building simulation framework focusing on the early design phases of a construction product, which will be able to provide the key stakeholders with the necessary simulation results that fully take into account both (i) the descriptive data of a building (material, components, equipment, space layout, etc.) and (ii) the information related to the dynamic behaviour of the building due to its occupancy fully taking into account the organization that is going to be "housed" within the building.

---

[1] The content of this Section has already been provided in other Adapt4EE deliverables, and is included here only to contextualize the readers not familiarized with the project.

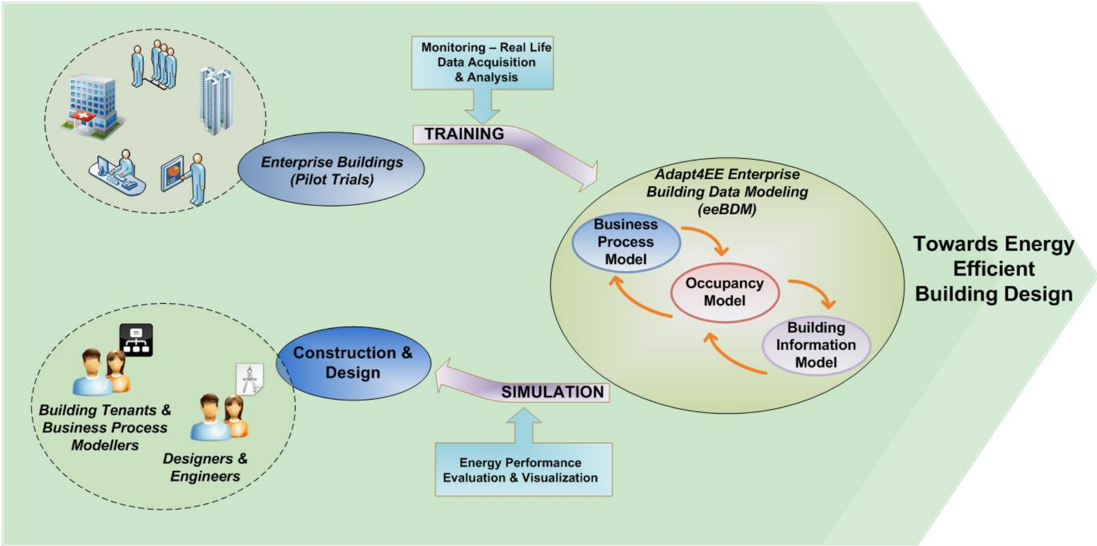

Figure 1: Adapt4EE Project Concept Overview

As may be seen in Figure 1, Adapt4EE aims to deliver and validate a *holistic* building simulation framework that takes into account the fusion of two different but complementary worlds: i) the Building Information Modelling (BIM), and ii) the Business Process Modelling (BPM), having as main catalyst the human factor (presence and movement). The incorporation of information about the dynamic behaviour of a building (e.g. organization that will be "housed" in the building) at the early design stages of a building will further improve the ability of designers, engineers and respective stakeholders (business modellers and/or building owners and tenants) to analyse the energy performance of the construction products as well as to allow them for optimisation of its energy consumption based on multi-dimensional or multi-criteria constraints. Moreover, design decisions on energy performance optimization should be based on sound and realistic estimates of the actual future energy consumption of constructions during operation, taking also into account potential consequences on business operations affected by early design decisions and vice versa.

In short, Adapt4EE aims at further augmenting the contemporary architectural envelope by incorporating business and occupancy related information into the building design phase so as to eventually provide the necessary shareable knowledge for effectively analysing the energy consumption of enterprise buildings as well as for further reconciling the differences between the energy performance of "real" and "simulated" construction products.

1.3 Structure of this Document

The remainder of this document is structured as follows:

- The next section discusses how the enterprise simulation models being developed in the Adapt4EE project are able to adapt to new domains.
- Section 3 explains the realized training algorithms for calibrating the domain specific enterprise simulation models for more realistic occupancy behaviours based on measurement data obtained from pilot sites.
- Section 4 briefly summarizes how the training algorithms are integrated within the Adapt4EE system, particularly how the raw measurement data is prepared for the training algorithms, and how the trained calibration results are persisted for use in other modules.
- Section 5 describes several implementation aspects of the training algorithms, including the environment, maintenance, and performance issues.
- Section 6 provides a short user manual for the targeted users on how to apply the algorithms on their own sets of (new) measurement data.
- Finally, Section 7 concludes the Deliverable.

2 Enterprise Model Adaptation

One important advantage of the Adapt4EE system is its ability use enterprise domain knowledge obtained in one site and reuse it for the evaluation of another. The ability to reuse the improved occupancy modelling mechanisms and structures across multiple construction product designs of similar enterprise domains offers a valuable tool for construction product designers and engineers, by extending the effectiveness of augmented performance assessments as offered by the Adapt4EE system to completely new construction products within the domain.

Realizing this ability to reuse occupancy modelling mechanisms and structures however poses some special reusability requirements to the enterprise simulation model. This section describes how these are tackled by principles of self-organization, as discussed firstly below. The following sections cover three mechanisms that allow the enterprise simulation model to adapt to new construction product designs.

2.1 Self-Organization

Self-organization refers to the ability of organizations to perform its functions without depending on a strict hierarchy or leadership [1]. The individual components show properties like autonomy, interactivity, and following simple rules that collectively produce desired group or macro behaviours that "emerge" from the micro scale interactions. Typical examples from nature are birds flocking, fish schooling, or termite armies cooperating according to simple rules yet being able to confuse predators or build enormous complex structures. Self-organization properties of complex adaptive systems have been studied extensively using agent based modelling and simulation.

As explained in an earlier Deliverable D4.2 ("*Integrated Enterprise Model & Intelligent Agent Constituents*"), one of the main reason for applying an agent-based modelling and simulation approach is the support for heterogeneity across individuals. Whereas traditional occupancy flow models used in many construction design evaluation simulators typically assume homogeneity among (groups or types of) occupants, agent based modelling enables behavioural deviations and preferences per individual. Besides providing more fidelity in the simulation results, this characteristic also has another advantage: *adaptation*.

In contrast to modelling flows of crowds based only on their current positions (e.g. assuming the Markov property, where behaviours are very much short-term oriented as the next state depends only on the current), in the intelligent agent based enterprise simulation models occupancy is a result of individual (autonomous) decision making with respect to participation in complex business processes as specified for the relevant enterprise domain. These individual decisions in turn may even be based on individually varying preferences. This means that each decision on whether or not some individual occupant or resource such as building elements (rooms, corridors, etc.) or equipment artefacts (computers, beamers, etc.) should participate in some business activity, all relevant agent constituents (representing the construction design, equipment artefacts and occupants therein) are polled for their current and expected availability.

Although much more computationally demanding, this ad-hoc approach to occupancy modelling is inherently self-organizing, enabling the model to change its structure, even during a simulation run, without affecting the (realistic) occupancy behaviours of individual model components, the agent constituents. As indicated in Deliverable D4.2, this allows for performance comparison not only across construction product design alternatives, but also across usage scenarios including disruptive events such as temporary blockades.

The ad-hoc characteristic which enables this inherent model flexibility is due to several decision making algorithms, three of which are discussed below. Others, like the employed shortest path algorithm by Dijkstra [2] are considered well known to the reader and will not be elaborated on in this section. The input for these algorithms are parameters specifying mathematical distributions which may be calibrated using the training algorithms described in further sections.

2.2 Process Creation Behaviour

Every intelligent agent based enterprise model simulation case (i.e. replication or simulation run) starts with a Scenario Management agent, which in turn creates all business process instances as specified for the enterprise simulation at hand at the appropriate simulated time instants.

2.2.1 Safeguarding Simulated Business Performance

Rather than simply extrapolating the relevant frequency for each type of business process (as specified in the business process model, or trained from observations), the Scenario Management agent keeps track whether the created business process instances are actually able to complete given the current building design, before issuing the next instance of some business process to be managed. This safeguards against counting many business process instances reportedly being performed in the simulation case, whereas in actual fact this could never realistically occur given the limited resource (availability of occupants, building elements, or equipment artefacts).

For instance, a designer might have started a new enterprise simulation based on his new building design which adds several corridors or an accidental blockade between rooms relevant for some business process in the enterprise domain. While the business process was observed to occur once every hour in alternative designs, the new design is only able to accommodate the complete business process once a day due to the amount of travelling involved in traversing the new corridors circumventing the blockade. In this case the business performance indicator would drop significantly for the new design. Simply extrapolating the originally observed frequency would have missed this change, and simply added the number of process instances as expected whether they can realistically be completed or not.

2.2.2 Business Process Independency

In such cases where the enterprise model agents are unable to accommodate the business processes in the ratios originally specified (or trained from observations) due to limitations in the new building design, the model constituent agent must choose how to proceed based on its assumption of business process in- or interdependency. The options explored here include:

- *No dependency*: simply continue generating business processes according to each type's frequency, assuming no dependency among the business processes; or
- *Full dependency:* Reduce the amount of all business processes to maintain the originally specified or observed ratios, assuming complete dependency among all business processes.

By default the model constituent agents assume no interdependency, a design choice made for two reasons. Firstly, the assumption that all business processes are fully interdependent and should always occur in the originally specified or observed ratios seems quite strong when regarding an entire enterprise domain with typically multiple processes occurring in parallel and independently (to illustrate, consider hospitals, offices, restaurants, etc.) Secondly, it appears to be more informative to know which business processes a new building design is unable to accommodate such that the design may be altered accordingly. Such information would be lost if the agents are trying their utmost to accommodate the original process ratios thus reducing the overall business performance for all business process types and hiding the "culprit" business processes.

2.3 Process Management Behaviour

The main purpose of the Process Management agent, once created, is to ensure its respective business process performed by the most eligible (e.g. nearest or otherwise best) occupants and resources available.

The pseudo-code for this algorithm is given in Figure 2:

Randomly draw task path A from (calibrated) transitions of process P

Negotiate allocation of best available resources R* for all those required in A

For each activity a in task path A

For each resource r in those required for this task (R_a sub R*)

Await readiness of other allocated resources $R_a \setminus r$

Confirm "usage" interval (t_{now}, t_{now} + duration of task a) to r

Figure 2: Process management algorithm pseudo-code

2.3.1 Reducing Resource Allocation Overhead

This algorithm introduces a lot of extra messaging between the enterprise model's constituent agents, due to the constant checking of their availability for allocation to currently active business processes. In order to reduce the resource allocation negotiation overhead, failed attempts for allocation are retried in static intervals of for instance 1 (simulated) hour. Although these negotiations do not take any simulated time within the resulting simulation case event trace, the model constituent agents do require CPU time to compute and interact before the simulation time can continue.

To illustrate, if for some process instance (e.g. activities related to a particular patient's surgery) the required resources could not be allocated since they're unavailable (currently out of the building or otherwise occupied) then allocation of the process instance will be reattempted later when enough resources of the required types may have become available again. Although simulation time has paused, many messages have been send to and fro to determine which resources of the required type were currently available.

Again, a trade-off is involved in this behaviour or policy. Choosing the interval for reattempts too short (high frequencies) will result in a lot of negotiation overhead, whereas long intervals (low frequencies) result in more resources being left "unused".

2.3.2 Separate Equipment Allocation

Process Management agents assume that equipment artefacts required for some process are available when a building element has been allocated, without specifically allocating timeslots yet for the equipment usage. Resource allocation negotiations thus mainly cover the availability of occupants per type and building elements per type. Instead, any required equipment artefacts, particularly of types that have limited service capacity (e.g. a computer screen or printer) will schedule their precise usage events lazily, that is, only once the next activity actually commences.

To illustrate, in the surgery process example mentioned earlier a number of displays may be required for some of the activities involved. These displays will allocate their actual times of usage once the respective activities commence or vice versa, allocating free time slots in their individual schedules accordingly. More details are provided below.

2.4 Equipment Allocation Behaviour

As indicated above, the allocation of equipment artefact usage intervals for equipment types with limited usage capacity is handled lazily, postponing commencement of the next activity until all allocated equipment artefacts have become available.

2.4.1 Lazy Equipment Allocation

The motivation for having a lazy equipment allocation approach is twofold.

The first reason is that the equipment usage durations (as specified by the business process modellers, as well as estimated based on real measurement data obtained at the pilot sites) are typically much shorter than the (estimated) activity durations they are required for. As such, allocating the equipment artefact for the full duration of some activity will over-represent its usage in the resulting simulation case and reduce the performance indicator fidelity.

The second reason or rather a positive side-effect is of a more pragmatic nature, in that it also improves simulation performance since the negotiation overhead is further reduced to building element types and occupant types, excluding all known equipment artefacts.

2.4.2 Synchronous Equipment Usage

When allocating the equipment artefacts, another assumption must be made regarding the synchronicity of their usage, particularly when more than one equipment artefacts are required for performing some business activity. The assumptions explored are:

- *Full synchronicity*: This assumption is the strongest and implies that the activity can't commence until all equipment artefacts are simultaneously available; or
- *No synchronicity*: This weaker assumption allows the activity to start regardless of equipment availability, but does not guarantee consistency.

The current behaviour models make the first assumption, delaying activity start until all equipment artefacts are simultaneously available, thus replicating the specified or calibrated task duration and maintaining consistency in resulting business activity and equipment usage.

With the second option, there is a risk that no time slot is found available within the business activity's allocated interval. Instead of the activity stopping before all required equipment usage is logged causing inconsistency, the activity will only complete after all allocated equipment artefacts have finished their time slot, thus delaying to completion of the entire business process if needed and in effect slightly reducing simulation performance indicator fidelity.

The trade-off made here is to either have occupants and resources wait for simultaneous availability and reduce their average business performance until such time, or to simulate business activity until all required equipment artefacts became available and were used and increase the average process cycle time.

3 Calibrating Enterprise Models to Specific Domains

The enterprise occupancy models rely mainly on accurate business process models. The accuracy of the business process model parameters thus becomes very important in producing valid simulation results to serve as a basis for the performance analysis of the construction product designs. This section explains how the accuracy of the business process models specified by the enterprise modeller can be calibrated using measurements obtained from prototypical pilot sites, accompanied by a mathematical formulation of the calibration task as an optimization problem.

The first part of this section gives a formal definition of the optimization stage followed by a description of the calibration stage. Based on these definitions some comments are made on preparing the raw measurement data are prepared for the optimization algorithms that try to solve the optimization problem. Finally, the remainder of this section presents the design of two optimization algorithms, one based on heuristic search, another on evolutionary computation.

3.1 Challenge

The main challenge in calibrating the business process models using occupancy sensor data is discovering the context in the training data. Basically the training or calibration algorithm has to guess what activity the occupants were likely engaged in based solely on their position in some room, or rather, on the occupants' combined trajectories through space over time. Once this context of likely activity engagement given the trajectories has been estimated, a calibration may be generated that fine-tunes the modelled business processes to reflect the activity engagement in future simulation runs by the Agent Simulation Module (ASM).

To illustrate the challenge of lacking occupancy context, consider two occupants who were observed to be in a meeting room for some time may have been performing one of several meeting activities defined as part of various processes. To select the most likely process context, any extra contextual information becomes relevant, such as the position of these occupants before and after this observation, as well as the duration of the overlap in their individual visits to the meeting room. Perhaps in one matched candidate process, the respective process describes these occupants performing some activity at their desks, whereas another matched candidate process specifies the occupants would be arriving from another (remote) office location. Let's assume in this example that the observed occupants were at their desk prior to visiting the meeting room next door, hence the likelihood of the former candidate process should then be much higher than the latter and the business process model calibration should reflect this as such, that is, more sitting and less travelling.

3.2 Calibration in Stages

As hinted above, the process of calibrating business process model parameters is divided into two stages (at least in the implementation reported in this deliverable):

1. Finding the best (or optimal) mapping of observations to business processes; and
2. Calibrating the business process parameters to the observations.

3.2.1 Mapping Observations to the Business Process Model

Figure 3 presents the mathematical or formal definition of the first stage. This stage is concerned with finding a mapping that correlates observations obtained from an enterprise

Given:

- A set of building elements:

  `space( be_id, space_type ), …`

- A set of equipment artefacts per building element:

  `equipment( ea_id, ea_type, be_id ), …`

- A set of occupant groups or roles:

  `role( r_id ), …`

- A set of business processes:

  `process( p_id, <p_activities> ), …`
  …where `<p_activities>` contains zero or more activity specifications:
  `[ activity(a_id, a_duration*, <a_requirements>, <a_transitions>), …]`
  `<a_requirements>` contains zero or more of:
  `space_req( ea_type )`, and/or
  `role_req( [ (r_id, [occ_ref, …]), …] )`, and/or
  `equipment_req( [ (ea_type, ea_duration*, [ea_ref, …]), … ] )`; and
  `<a_transitions>` is a *probability mass function* over business activities:

  `transition( [ (a_id, t_likelihood*), …] ) )`

- A set of building element occupancy and equipment artefact usage observations:

  `observation( obs_id, t_enter, t_leave, occupancy( be_id, r_id ) ), …`
  `observation( obs_id, t_start, t_stop, usage( ea_id ) ), …`

…find (optimize) a mapping of observations to activity types in instances of (likely) business process types (`p_instance_id`):

`map( obs_id, p_instance_id, a_id ), …`

…such that mappings are preferred that*:*

1. maximize the amount of observations |`obs_id`| mapped to some `p_instance_id`;
2. maximize the number of activity requirements matched by observations;
3. maximize the proportion of activity requirements matched by observations;
4. minimize the amount of intermediate inactivity (delay or idle time); and
5. minimize the number of valid business process candidates, with a minimum of 1.

Figure 3: Formal definition of the optimization stage in business process calibration

(pilot) building with the business process model provided for the respective enterprise domain. The observations regard building element occupancy and equipment usage.

The optimization problem assumes that several items are given, including the set of building elements, the setoff equipment artefacts per building element, the set of occupant groups or roles, the set of business processes (some directed graph of business activities each requiring zero or more resources), and finally a set of observations related to building element occupancy and equipment artefact usage.

3.2.2 Calibrating the Business Processes Using the Mapped Observations

The final stage is then to (re)calibrate all the *-marked attributes in the activity specifications (<p_activities>) by defining mathematical distributions that reflect the actually observed values. These attributes include the activity durations (a_duration), equipment usage durations (ea_duration), and activity transition likelihoods (t_likelihood).

Other attributes relevant for simulation concern the start timing of processes (e.g. only on Mondays) as well as the intervals between process instances (e.g. daily, weekly, or

P ← processes (Tasks x Resources (Roles, Equipment Types, Space Types, Time) x Transitions)

S ← scenario (Spaces per Type, Equipment per Type, Occupants per Type)

O ← measured Space Occupancy intervals (in scenario S)

U ← measured Equipment Usage intervals (in scenario S)

L ← maximum candidate task path length

C ← set of all activity sequences [a_1, a_2, …, a_n] in activity graph (incl. cycles) for all processes p in P where n =< L

$s_{threshold}$ ← threshold match score in [0, 1] below which observations are considered non-matching

For each activity sequence c in C

a. For each activity a_c in sequence c
   i. For each Space Occupancy interval o in O
      1. Skip o if space type or role of o do not occur in (some task of) sequence c or activity a_c
      2. O' ← subset of O where time and space overlap/match o
      3. v ← largest interval of overlapping o' for required roles R_t
      4. s_1 ← required (a) vs. observed (O') role occupancy match
      5. s_2 ← specified (a) vs. observed (v) role occupancy duration match (quadratic distance function, $0 < s_2 < 1$ for $t/2 < v < t$; $s_2 = 1$ for $v > t$, since multiple activities may have been performed by the occupant in o')
      6. U' ← subset of U where time and space overlap/match o
      7. W ← intervals of equipment usage u' for U'
      8. s_3 ← required (a) vs. observed (U') equipment type usage match
      9. s_4 ← specified (a) vs. observed (W) equipment type usage duration match (quadratic distance function, $0 < s_4 < 1$ for $t/2 < w < 2t$; $s_4 = 0$ otherwise)
      10. s_{ao} ← activity-observation match score = average(s_1, s_2, s_3, s_4)
      11. if $s_{ao} \geq s_{threshold}$
          a. add match(o, a_c)
b. For each activity a_c in sequence c
   i. O" ← subset of O where for all o in O" a match(o, a_p) exists such that a_p matches any previous action in sequence c
   ii. s_a ← average $_{o \text{ in } O''}$(s_{oa})
c. s_c = average $_{a \text{ in } c}$(s_a)

Figure 4: Pseudo-code for the BPM driven observation mapping approach.

monthly). Although these attributes are not specified in the BPM, providing mathematical distributions for them based on observations will increase the realism of simulation runs.

A more detailed description of the available types of mathematical distributions is given in paragraph 4.2.

3.3 BPM Driven Observation Mapping Approach

One approach to mapping observations of occupancy and equipment usage to business activities specified in a BPM is to start from the BPM and try to match all observations in a heuristic manner. In this case, all candidate sequences of activities up to some sequence length are generated for all processes. Then each activity or step in these candidate sequences are matched with observations that are “nearby” (in space and time) resulting in a match score per activity-observation pair (s_{ao}), as shown in Figure 4. This activity-observation match score is an average of:

- the roles or occupant group match (s_1, either 1 or 0);
- the quadratic distance of matching occupancy observations’ duration (s_2, in the range [0, 1]);
- the proportion of matching equipment usage observations during the occupancy (s_3 in the range [0, 1]); and
- the quadratic distance of the matching equipment usage observations’ durations (s_4 in the range [0, 1]).

3.3.1 Validating the BPM Driven Approach

Considering only activity-observation matches with scores larger than some threshold value ($s_{threshold}$ in the range [0, 1]), each candidate activity sequence c can be scored by averaging the scores of all matched observations (o_a) for sequence step up to and including activity a (obtaining sequence step scores s_a) and averaging again for all activities a_c in sequence c (obtaining sequence scores s_c).

Considering the optimization rules stated in the formal problem definition of Figure 3, the first rule of maximizing the number observations matched is handled by the repeated iterations over building element occupancy as well as the equipment artefact usage observations (sets O and U respectively, as applied in steps a.i, a.i.2, a.i.6 and b.i of Figure 4).

Similarly addressed are optimization rules 2 and 3, regarding the maximization of respectively the number and proportion of requirements matched by observations, since all possible sequences of activities possible in the BPM (the set C of generated candidate activity sequences) are considered. The scores of sequences containing more activities are likely to be higher since more requirements regarding equipment usage may be matched.

Scoring the activity-observation matches in this manner implies a bias towards matching observations that occurred in close vicinity (in both time and space) of each other, thus addressing optimization rule 4 regarding minimization of the total idle time.

Finally, the last optimization rule, regarding minimization of the number of validly matching business processes, is inherently optimal in this business process driven approach, since the final scores concern a valid sequence of activities for precisely one business process.

3.3.2 Performance

The BPM driven observation mapping approach was implemented in the Java programming language, and executed on an Apple MacBook Pro with a 2GHz Intel Core-i7 running OSX 10.9.4 and Oracle’s Java Virtual Machine version 7. The following measurements were obtained using this setup:

Table 1: Performance results for the BPM Driven Optimization Approach

Case	Number of processes	Number of space types	Number of candidate sequences	Number of candidate sequence activities	Number of observations	Stage 1 duration (sec)
1	2	2	22	34	6	19.9
2	2	2	22	34	18	20.8
3	19	9	494	1321	101	50.4
4	19	9	494	1321	171	55.6
5	19	9	494	1321	241	61.4

3.4 Evolutionary Observation Mapping Approach

The evolutionary approach is implemented as a standard Genetic Algorithm (GA). A GA typically initializes a population of individuals with their genome representing candidate solutions (their phenotypes). After evaluating their individual phenotypical fitness, recombination and mutation operators are applied to these genomes in order to generate new generations of offspring representing new and hopefully fitter candidate solutions. These are again evaluated and, based on their fitness, (de)selected for further generations of offspring. The following describes the particular details of each GA component.

3.4.1 Population

A population consists of several individuals, typically around 100. Each individual genome encodes a phenotype, in this case a candidate solution to the observation-activity mapping problem as formulated in Figure 3.

Each gene or position in the genome corresponds to an observation of either building element occupancy (enter and leave room event pairs) or equipment artefact usage (start and stop equipment use event pairs). This means that the genome length is determined by the number of observations. Note that observations which are relevant to no business process activity in the BPM are ignored from the encoding sequence. For instance, occupancy of corridors is not explained specifically by any business activity, and hence ignored in the optimization process.

Each gene encodes two references, one concerning an activity instance of some business process and the other concerning a process instance. The idea is that through evolutionary operations the observations are grouped together into the same context or business process instance, as they increasingly merge into the same activity and/or process by sharing their instance reference.

Each genome thus represents a complete mapping of observations to business process instances. In order to ensure that the initial population of genomes encodes for valid mappings, each gene is initialized with a unique references for both activity and process instances within that genome and population. This means that, initially, all observations are considered to be unrelated and occurring on their own within some business process activity.

3.4.2 Fitness Evaluation

After each iteration or generation, the population is evaluated in order to select the fittest individuals for breeding new offspring. Fitness evaluation is an important part of any optimization algorithm, since this is what characterizes the search landscape and algorithm performance. Fitness in this case is determined by a score or rather penalty that should be minimized, which aggregates a total of three fitness criteria or types of error.

First to minimize is the number of inconsistencies in the phenotype or mapping. Inconsistencies mean that a particular grouping of observations can never occur in any path or sequence of activities for any of the business processes specified in the BPM. These inconsistencies may occur regardless of any safeguards present in the recombination and mutation operators.

Second to minimize is the number of unique activity and process instances in the phenotype or mapping. This entails that phenotypes having smaller numbers of unique instances are considered fitter. In other words, the less process instances are needed to explain all the observations, the better. This principle echoes Occam’s razor, that is, to prefer the shortest explanation possible.

Finally, another aspect to minimize is the total idle time between consecutive activities per process instance. It is assumed that activities occur nearly consecutively, with some room for intermediate passage through corridors, etc. Intuitively one would prefer mappings with observations grouped together temporally over ones that assume immense gaps (of unobserved activity or inactivity) between the grouped observations.

3.4.3 Breeding Pipeline

The breeding pipeline refers to the method in which new offspring is generated for future generations and determines how the search space is traversed. The breeding pipeline typically combines both the fitness information with some form of randomness.

In this case the fitness helps to determine which individuals will become “parents” for recombination of their genomes. In fact, the recombination operator requires two parent genomes, one selected uniform randomly, the other selected with likelihoods proportional to their fitness. This ensures that local optima are further explored via fitter parents, but also that less-fit parents may help escape local optima in favour of finding global optima.

Recombination between genomes occurs by swapping gene sequences between multiple cross-over points such that resulting offspring produce valid phenotypes or mappings (i.e. the time/resource allocation constraints still hold, including process cycles, resource recurrence, parallel activities, multiple activities per occupant, etc.)

Finally, the offspring pass through a mutation operator which occasionally mutates a pair of randomly selected genes' values with the effect of joining some unlinked observations, i.e., observations previously assumed to belong to different activity or process instances are now joined to the same group of observations. As with the recombination operator, this operator retries mutations across gene pairs until it generates phenotypes or mappings that are again valid given the available BPM. The gene pairs are more likely to be selected for joining when they are "nearby" in terms of the respective observations' times.

3.4.4 Validating the Evolutionary Approach

The pseudocode in Figure 3 specifies five optimization rules by which one may validate an optimization algorithm implementation such as this evolutionary approach. Each is discussed briefly below.

By design, the evolutionary approach generates solutions that always map all observations to some process instance, hence it inherently meets the first optimization rule, that is, to maximize the amount of observations mapped to some process instance.

The second and third optimization rules concerning respectively the number and proportion of activity requirements matched by observations is approached via the mutation operator which continues to increase the number of observations matching the requirements of a single process or even activity instance with each generation.

The fourth optimization rule is also addressed by the mutation operator, that is, mutation gene pairs are selected with a preference for "nearby" combinations in terms of observation time, hence minimizing the overall idle time per process instance regarding the distance between mapped observations.

Finally, the last optimization rule to provide a minimal set of business processes explaining the data is also addressed. Again this is mainly due to the mutation operator, since the number of valid business process candidates decreases as more observations are mapped to the same process instances, leaving less candidate processes that may explain them. Both the recombination and mutation operators have built-in safeguards that produced offspring remain valid, meaning that at least one business process exists that can produce a sequence of activities that could explain the mapped observations.

4 Integration

As part of the Adapt4EE system which is centred around the Common Information Model Interaction Module (CIMIM) described in Deliverable Dx.x, the Agent Training Module (ATM) described in Deliverable D3.4 ("*Adapt4EE Multi Agent Management System*") must import measurement data from the CIMIM and export its calibration results back to the CIMIM for use in other modules, mainly the Agent-based Simulation Module (ASM) described in Deliverable D4.2 ("*Integrated Enterprise Model & Intelligent Agent Constituents*").

4.1 Importing Measurement Data from the CIMIM

Although initially the Semantic Reasoning Module was aimed to derive the semantic events describing the raw sensor data, same challenges were encountered that further would reduce the calibration algorithm efficacy. Here we briefly explain two of these challenges.

First, aggregating individual room occupancy data into complete trajectories of occupants became an issue, since the tracking devices across the site do not tell us right away whether a person that was seen leaving room A and entering room B sometime after is the same person or not. Sensor algorithms reduce all visual features to some abstract and temporary occupant id that remains consistent while the occupant is in a single room. However, especially since not all corridors were completely watched and important visual features are not reused, hybrid semantic reasoning was unable to cope with the information gaps. Therefore the algorithms must assume only sensor data provided by RFID data is reliable, containing information on which groups were present in what number and during which time intervals.

The second challenge regards the relation between equipment usage and particular occupants or groups. It was not yet possible to formulate a semantic query in the IAM that would produce matches of which occupants were using which equipment. Although the position of occupants and equipment is known, not all equipment have the same kind of minimum proximity indicating which occupant is actually using the equipment. Furthermore, the usage events are typically "on/off" events occurring only a few times a day (turn on your computer in the morning, turn it off possibly the next day). In order not to make too many assumptions that may impose new bias on the calibration results, the solution was to leave it to the calibration algorithm to decide which occupant was likely using the equipment. In cases where equipment usage is exclusively assigned to the context one business activity and not another, this may have a slight impact on the calibration fidelity of the excluded business activities. The equipment usage events have been converted simply into time ranges or intervals during which equipment was active, and as such serve as constraints for the observation mapping optimization algorithm to help reduce the search space somewhat and guide the algorithms.

4.2 Exporting Trained Calibrations to the CIMIM

After calibration stages have completed, the calibration results should be stored in the enterprise domain's Common Information Model (CIM). In order to accommodate the calibration results, the CIM schema's `<trainingParameters/>` element has been extended with `<processTemplate/>` elements that enable the definition of mathematical distributions for any BPM attribute. In fact, a whole range of mathematical distributions is supported, enumerated by the type `distributionTypeEnum`, including:

- Constant values;
- Beta distribution;
- Binomial distribution;
- Cauchy distribution;
- Chi-squared distribution;
- Enumerated distribution (continuous or discrete values);
- Exponential distribution;
- F distribution;
- Gamma distribution;
- Geometric distribution;
- Hypergeometric distribution;
- Levy distribution;
- Log-Normal distribution;
- Normal distribution;
- Pareto distribution;
- Pascal distribution;
- Poisson distribution;
- T distribution;
- Triangular distribution;
- Uniform distribution (continuous or discrete values);
- Weibull distribution;
- Zipf distribution; and
- other distributions not explicitly named.

To specify exactly which parameter value is described by a distribution, several reference attributes are included in the respective schema type "`tDistribution`":

```
<complexType name="tDistribution">
    <sequence>
       <element name="property" type="a4eeuni:tProperty" maxOccurs="unbounded"
    minOccurs="0"/>
    </sequence>
    <attribute name="type" type="a4eeuni:distributionTypeEnum" />
    <attribute name="processRef" type="string" />
    <attribute name="valueRef"
       type="a4eeuni:processValueRefEnum" />
    <attribute name="otherType" type="string" use="optional" />
    <attribute name="activityRef" type="string" use="optional" />
    <attribute name="usedEquipmentRef" type="string" use="optional" />
    <attribute name="usedSpaceTypeRef" type="string" use="optional" />
    <attribute name="otherRef" type="string" use="optional" />
</complexType>
```

As shown in this schema type definition, a distribution must refer to a specific process ("processRef") and value ("valueRef"), the latter of which may take one of "`nextActivity`", "`waitingTime`", "`executionTime`", "`transportTime`", "`timeOfUse`", "`concurrency`" or "`other`". These have the following meaning:

- `nextActivity`: Distribution over an activity's likelihood per `nextActivityRef`. Requires the `activityRef` attribute to be set.
- `waitingTime`: Distribution over an activity's waiting time. Requires the `activityRef` attribute to be set.
- `executionTime`: Distribution over an activity's execution time. Requires the `activityRef` attribute to be set.
- `transportTime`: Distribution over an activity's transport time. Requires the `activityRef` attribute to be set.
- `timeOfUse`: Distribution over an activity's used equipment time of use. Requires the `activityRef` and `usedEquipmentRef` attributes to be set.
- `concurrency`: Distribution over an activity's or process time of exclusive usage where 0 percent means non-exclusive-utilization and 100 percent means utilization-dedicated to the given refs.
- `other`: Distribution over a process value. Requires the `otherRef` attribute to be set.

Consider the following example:

```
<processTemplate type="EnumeratedDiscrete" processRef="BP ISA Ordered Stock
Analysis (SKELETON) (BPMN 2.0)CIM" activityRef="196109"
valueRef="nextActivity">
   <property name="196112" type="double" ns0:type="xsd:double">1.0</property>
</processTemplate>
```

In this case the calibration result specifies the type of mathematical distribution as a discrete enumeration (“EnumeratedDiscrete”) for process “BP ISA Ordered Stock Analysis (SKELETON) (BPMN 2.0)CIM” and activity “196109” concerning its value for the “nextActivity” attribute. The probability mass function of this enumerated distribution captured by nested <property/> elements provides precisely one option for the nextActivity value, that is, activity 196112 with probability 1.0. In other words, the optimized activity-observation mapping contains only one follow-up activity for 196109 which is 196112.

5 Implementation

The calibration algorithm and other tooling required for training the business processes has been developed as an open-source project to foster future development and reuse. The technologies are platform independent and focused on web-based application so as to maximize the accessibility to Adapt4EE system users.

5.1 Runtime Environment

In order to support wide adoption of the tooling, system requirements have been reduced to a minimum for the (server-side) operating system or platform as well as the (client-side) browser application.

5.1.1 Operating System Requirements

The runtime environment of the calibration algorithms is part of a web service that may be deployed on any application server. Demonstrations and screenshots shown in this deliverable are based on a version of the web service that was running on a virtual Linux host that served a simple dedicated Jetty application server. Since this setup is Java-based, almost all important operating systems available via Platform-as-a-Service (PAAS) solutions are supported.

5.1.2 Browser Requirements

Browser requirements are minimal thus enabling broad accessibility to the system once online. In fact all HTML5-compatible browsers are suited to run the web-based application of the training algorithms, which basically covers most of the browsers currently used on mobile and desktop devices.

5.2 Code Maintenance

The calibration algorithms and enclosing training application code has been organized with the aim to support a large community and foster future usage and adoption.

5.2.1 Programming Languages

Platform-independent Java and JavaScript programming languages have been applied which are ideally suited for prototyping efforts in heterogeneous environments typically encountered in international research projects such as Adapt4EE.

5.2.2 Code Structure

Software project management has been implemented with the well-known Maven system, which provides standards and conventions for structuring the code and managing third-party software libraries and dependencies.

5.2.3 Open-source

Main collaboration on the code base occurs via a web-based versioning system Subversion (SVN), although some libraries that are common to Almende’s efforts in multiple projects are maintained separately on the openly accessible Github repository.

.

6 Application

Finally, and perhaps most importantly, the training algorithms integrated within the Adapt4EE system must be made available to the system's users for application to their domain specific enterprise simulation models.

6.1 User Interface

The user interface for calibrating business process models shown in Figure 5 involves two steps: to select an enterprise model or CIM; and to start a calibration algorithm.

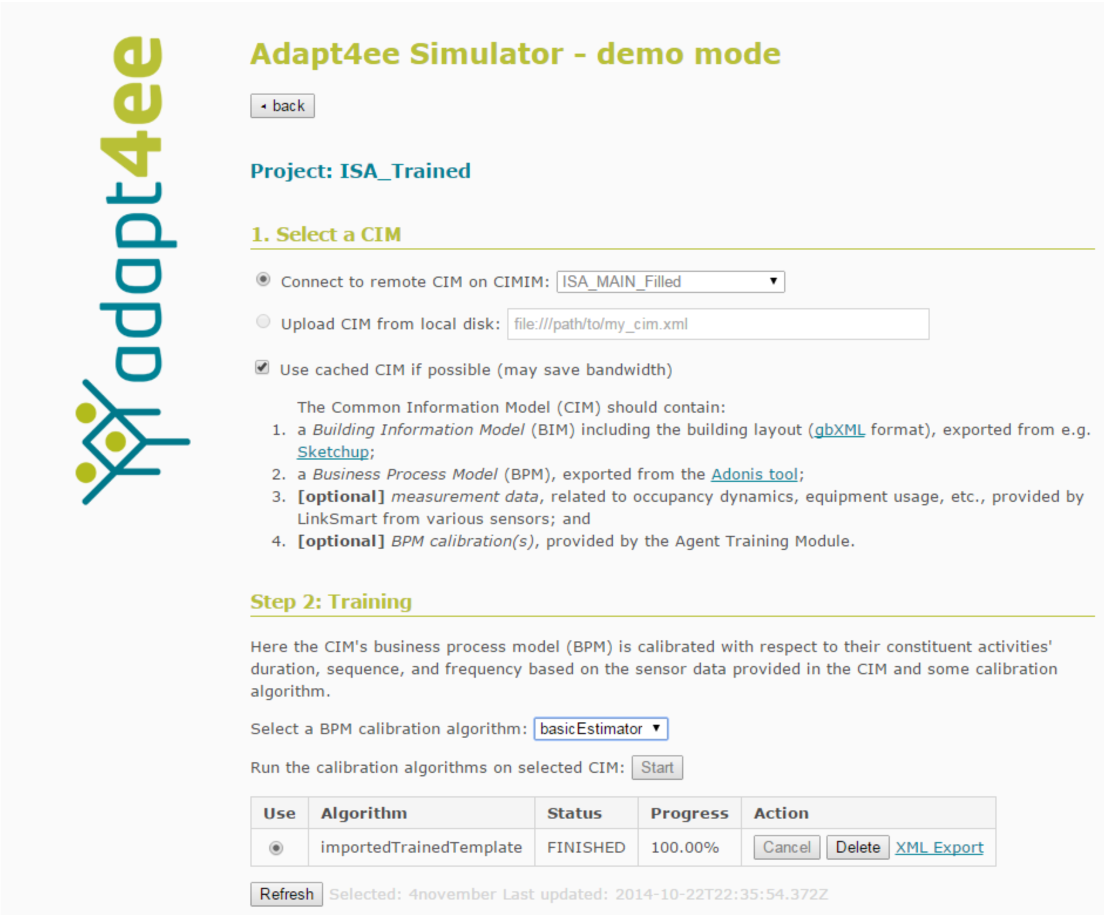

Figure 5: BPM Calibration User Interface

6.1.1 Selecting an Enterprise Model for Calibration

The first step in calibrating a business process model is to select the respective enterprise model or CIM for which a new calibration will be generated. The CIM must contain at least (1) a building information model or BIM including the building layout in gbXML format (exported from third party software such as Google Sketch-up) and (2) a business process model exported from the Adonis tool that has been customized for the Adapt4EE system, but also (3) some measurement data containing the observations to guide the calibration, and possibly (4) some earlier calibration results.

6.1.2 Training the Business Processes

The next step is to select a training approach, either the BPM driven or the evolutionary approach, click "Run" and wait for the optimization algorithm to finish. Once finished the resulting trained process templates may be "exported" separately, which will provide the XML contents as for instance shown in Figure 6.

Figure 6: Trained Business Process Template Export

6.1.3 Applying the Business Process Calibration

Finally, the user selects which of the calibration results should be applied in the current simulation run by the Agent Simulation Module. To this end, a radio-style checkbox is shown to the left of each available (finished) calibration result, as shown in Figure 5. Once selected, any new simulation run (performed in another part of the GUI) will apply the calibrated business processes to generate the building element occupancy and equipment artefact usage behaviours.

7 Conclusion

Corresponding to Adapt4EE Task T7.4 ("Agent Based EMS Training/Learning and Adapt4EE Model Optimization"), the efforts reported in this deliverable concern the definition, execution and reporting on Adapt4EE system training programmes for enterprise network operators from the considered pilot domains. These training programmes have been delivered in the form of interactive training or learning algorithms used to optimize the enterprise models for intelligent agent-based simulation. Finally they have been wrapped within user interfaces for key actors (including enterprise network operators and building management operators) to provide their feedback and adapt the enterprise network models as needed.

This deliverable defined optimization algorithms for (automated) adaptation the enterprise simulation models to specific domains, in particular self-organizing behaviours related to process creation, process management and equipment allocation.

Furthermore, training algorithms have been specified for calibrating the enterprise simulation models to more closely match realistic behaviours as observed in the pilot sites, in particular by formalising the calibration problem mathematically and providing two optimization approaches (BPM driven as well as evolutionary). Each of these algorithms has been validated showing they adhere to the formalized optimization rules.

Integration aspects have been discussed concerning how to prepare the raw measurement data obtained from pilot sites for successful application of the training algorithms, as well as how to apply the training algorithms in order to obtain calibrated, domain specific enterprise simulation models.

Lastly, an application user manual has been provided for key actors (e.g. enterprise network or building management operators) for collecting the necessary feedback to be fed into the pilot requirements as well as for the adaptation of the enterprise models, if needed.

8 References

[1] European Commission, Energy Efficiency Plan 2011, available at http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:52011DC0109:EN:HTML:NOT

[2] European Parliament, Directive 2010/31/EU of the European Parliament and of the Council of 19 May 2010 on Energy Performance in Building, available at http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32010L0031:EN:NOT

[3] Clarke, J. A. (2001). Energy simulation in building design. Ed. Chiel Boonstra, Ronald Rovers, and Susanne Pauwels. International Conference on Sustainable Building 2000. Vol. 2003. Butterworth-Heinemann.

[4] Kim, H., Stumpf, A. & Kim, A. (2011). Analysis of an energy efficient building design through data mining approach. Automation in Construction 20(1): 37-43.

[5] Kauffman, S. A. (1993). The Origins of Order: Self-organization and Selection in Evolution. Oxford University Press.

[6] Knuth, D. (1977). A Generalization of Dijkstra's Algorithm. Information Processing Letters 6(1): 1–5.