Semi-automated deployment of Simulation-aided Building Controls

Sergio Leal, Florian Dubisch, Florian Stift, Gerhard Zucker, Peter Palensky
Energy Department
AIT - Austrian Institute of Technology
Vienna, Austria
Email: [sergio.leal, florian.dubisch, florian.stift, gerhard.zucker, peter.palensky]@ait.ac.at

***Abstract*—The deployment of Simulation-aided Building Controls is a complex process due to the uniqueness of each building along with an increasing complexity of building systems. Typically the deployment tasks are performed manually by highly specialized personnel which results in a poorly documented and extremely scattered deployment process. This paper introduces a workflow for the deployment of a Simulation-aided Building Control service suitable for supporting the operation phase of a building. Some tasks in the deployment process may benefit from machine support, especially data intensive, repetitive and error prone tasks. These may be fully or semi-automated. The proposed approach reduces the complexity of the setup procedure, decreases problems related to the uniqueness of the infrastructure and supports the documentation of the deployment process. Specially large facility management service providers may profit from this deployment process.**

I. Introduction

The increasing energy prices and energy demand pose new challenges for modern society. Overall, according to the European Commission Directorate General for Energy and Transport [1], buildings are one of the fastest growing energy consuming sectors. At the same time the increased complexity in building systems provides new challenges for building operation processes. Different communication protocols, bus systems and sensing/actuating systems are used to control the building environment.

Among the objectives of building automation are assurance of thermal comfort, visual comfort and indoor air quality. In this context efforts are currently focused on a solution that satisfies energy requirements, assures the operational needs and increases energy efficiency. To accomplish these goals, it is important to have continuous energy management complemented by the implementation of energy and cost savings strategies.

This can be provided by a Simulation-aided Building Control service (SaBC). A SaBC is a dynamic system which main purpose is to support simulation-aided indoor environmental control operations in buildings. In an SaBC, future control steps are projected from available model data through simulation. Then computed values of pertinent performance indicators, are predicted, compared, and ranked. Due to the amount of information gathered by the model, it can also support other building tasks such as those concerned with building logistics and management.

SaBC deployment is particularly challenging, when compared to conventional automation systems, because there is a huge fragmentation among system manufacturers with regard to system architecture and data communication protocol.

The majority of the solutions presented by previous studies were custom made and presented a low portability. Which means that, even if a building control solution worked perfectly on a given building, there were no guarantees that it would be possible to port it to another building, and initiate operations immediately. Also, most of the building control solutions presented in the past were domain-centered and offered low communication between building systems. Therefore, architectures [2] [3], tools and strategies [4] have been proposed to automate the deployment of standalone software entities in complex, large-scale environments. Although few contributions have approached the deployment of SaBC applications.

The aim of this work is to present an improved, partially automatic deployment process workflow for SaBC where human supervision can be reduced along with the deployment costs. Specially large facility management service providers may profit from this deployment process.

The SaBC deployment process is divided into several tasks. These tasks are setup building model, setup physical test environment, setup building model server, setup application interface, setup building automation interface, setup controller and system testing.

In order to define the deployment process tasks, business process modeling (BPM) diagrams are used. BPM enables a multi-layered representation of processes with different levels of detail [5]. This paper focuses on the visual representation of the model [6].

A proof of concept has been made in a passive house office located in Vienna.

II. Related work

Building automation systems (BAS) provide automatic control of the indoor environments conditions. Usually dedicated field buses are used to interconnect BAS devices. These must be more flexible, manageable and interoperable than industrial field buses [7].

To address the diversity of systems and integration issues, a building automation task definition and communication infrastructure is necessary [8]. Kastner et al. analyses requirements, application models and services for this building automation task definition. It also summarizes the most important open systems, such as BACnet, Lon-Works and EIB/KNX.

Real-time simulation-based building control in the building performance domain (thermal and visual indoor environmental controls) has been extensively discussed in the past [9]. Following research has supported this approach and provided further proof of concept mostly in terms of validating implementations in lighting and shading domains [10].

Brunner [11] presented a building model service (BMS) where an object tree continuously updated from sensor data reproduced the current building state. The building model was concurrently accessible to multiple clients and supported by persistent storage. Shared Object Model (SOM) [12] were used to describe the building model. SOM is a fairly straightforward object hierarchy and served well as simple model representation of the building. Brunner argues that, in comparison to the complex Industry Foundation Classes (IFC) [13], the advantages are significant, since a simpler model was easier to create, maintain and update.

The definition of deployment is still nowadays subject of discussion [14]. On the SaBC deployment context the following definitions are considered: Carnagiza et al. [3] define deployment as the delivery, assembly and maintenance of a particular version of a software system at a site. Chazalet et al. [14] extends the Carnagiza et al. process by introducing a management activity.

III. Methodology

The deployment of a SaBC service is a complex task, involving different stakeholders, each of them with different roles in the SaBC deployment process. These roles can have a predominant hardware or software component. Among the roles that have a stronger hardware component are those from building hardware designers, such as Building Planners and Building Integrators. The software roles are those from BAS designers, such as Control Engineers and Software Engineers.

The SaBC deployment process is performed by a Master Integrator [15]. This is an inter-disciplinary role that executes all tasks of the deployment process, managing the coordination between the hardware and software roles, along with the communication and feedback to building owners, facility managers and users.

A work-flow model is proposed to support SaBC deployment. The SaBC deployment process is divided into several tasks. These tasks are setup building model, setup physical test environment, setup building model server, setup application interface, setup building automation interface, setup controller and system testing (figure 1).

During the building model setup task, information concerning the building and its systems is acquired. The building model server, the application interface, the building automation interface and the controller may be configured and tested in parallel, that is, independent of each other. Finally a series of tests are performed on the whole system. To maintain simplicity the deployment process is represented as sequential, but in reality it is iterative, where for example negative test result may implies modifications on previous stages.

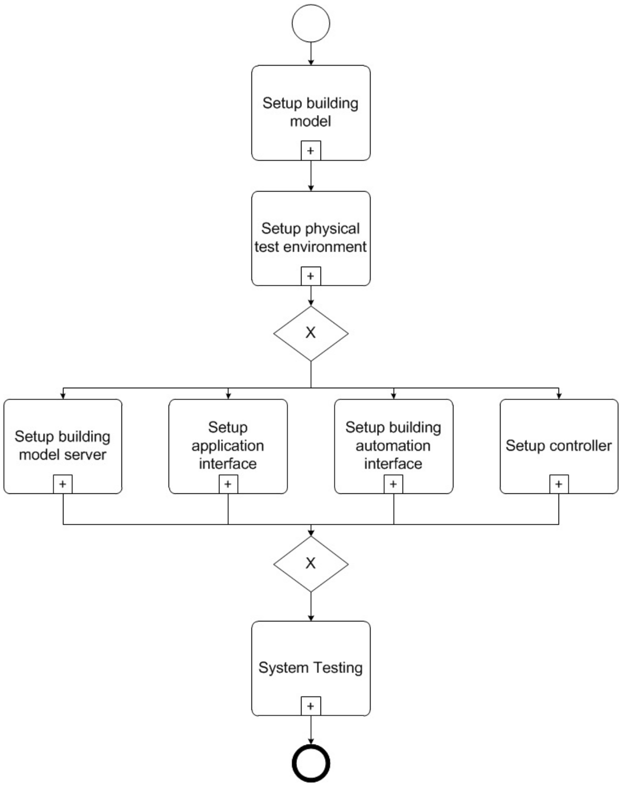
Setup building model
+
Setup physical test environment
+
X
Setup building model server
+
Setup application interface
+
Setup building automation interface
+
Setup controller
+
X
System Testing
+

Fig. 1. Top level view of the deployment process

A. Setup building model

In order to understand the building model setup task, a further level of detail is necessary, as shown in figure 2.

The first task in the building model setup is the definition of a naming scheme. This scheme is used throughout the entire project, unifying nomenclature and simplifying communication. It contains information concerning relevant objects in the model, from technical objects (such as sensors, actuators, luminaries, valves, etc.) to building elements (such as walls, windows and furniture). The naming scheme has to be adapted to the deployment case.

Subsequently, two tasks are initiated in parallel. On the one hand, space and space enclosure data is collected; on the other hand, technical equipment data is gathered. This data may be collected from a Building Information System, if available. On a closer look at the space and space enclosure data collection procedure, the following tasks are performed. At first the space and space enclosure data are requested; this information is typically generated by an architect. Later, when the building model is available, the space and space enclosure geometry data are extracted, along with material data, such as surface properties, color and reflection. Spatial building model instances are generated from these data. Then, the Spatial building model entities are validated. In parallel with the space and space enclosure collection procedure, the technical

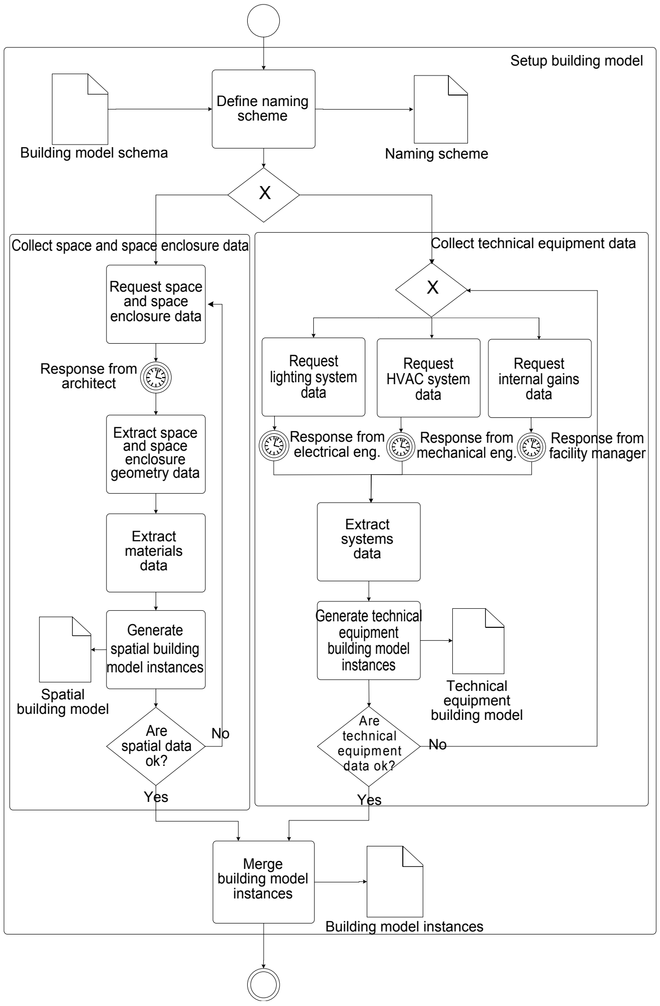
Setup building model
Building model schema
Define naming scheme
Naming scheme
X
Collect space and space enclosure data
Collect technical equipment data
Request space and space enclosure data
Response from architect
Extract space and space enclosure geometry data
Extract materials data
Generate spatial building model instances
Spatial building model
Are spatial data ok?
No
Yes
X
Request lighting system data
Request HVAC system data
Request internal gains data
Response from electrical eng.
Response from mechanical eng.
Response from facility manager
Extract systems data
Generate technical equipment building model instances
Technical equipment building model
Are technical equipment data ok?
No
Yes
Merge building model instances
Building model instances

Fig. 2. Building model setup process

equipment data collection is performed. For each building systems a data request is sent to each specific professional (e.g. lighting system data from Electrical Engineer, Heating, Ventilating and Air Conditioning (HVAC) system data from Mechanical Engineer and internal gains from the Facility Manager).

When all data are available, they are combined in one technical equipment document. This process results in technical equipment building model instances. After, technical equipment building instances validation is performed. Once both space and space enclosure and the technical equipment data are accessible, they are merged into building model instances representing the building.

B. Setup physical test environment

The physical test environment setup process is shown in figure 3. First a typical room in the building environment is selected for tests. The selected room will be used during the deployment process as testbed. So it must be representative of the deployment environment and conform with the following criteria:

- have the predominant room architecture (for example, if the building is an office building, a standard office should be selected.)
- be equipped with the standard technical equipment (luminaires, HVAC, etc.)
- have the standard materials (walls, windows and furniture material have to be representative of those predominant in the building)
- be on the principal room orientation (North, South, East, West façade)

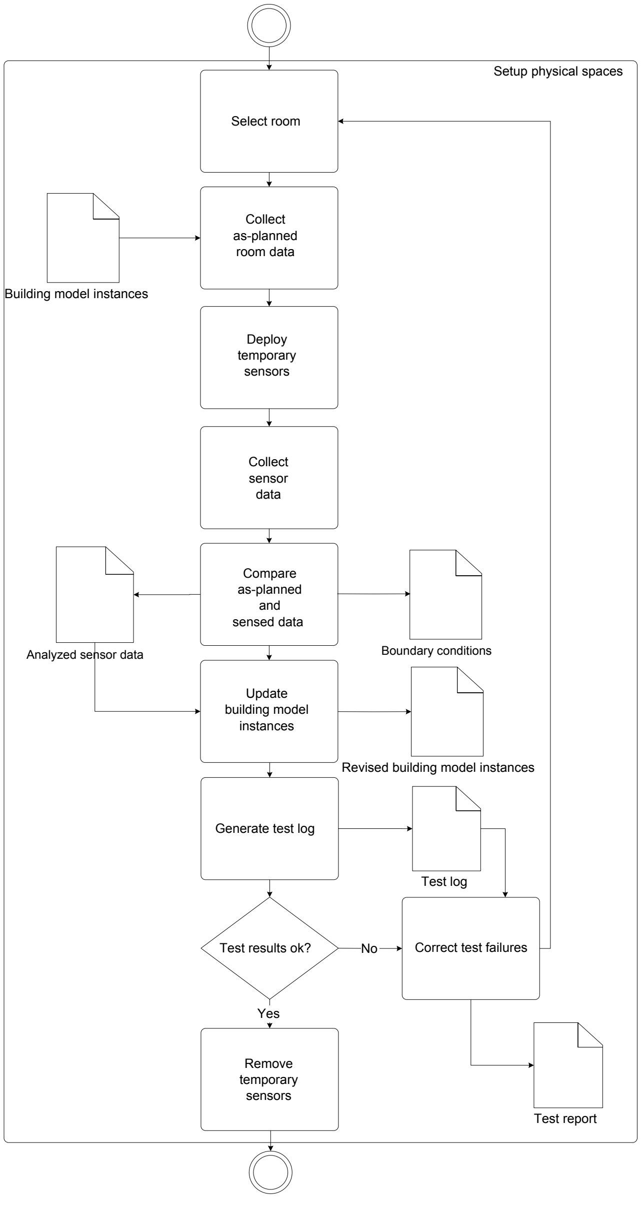

Fig. 3. Physical test environment setup process

Accordingly, information concerning the selected room is gathered. This information includes Light Reflectance Values from different materials of objects in the room, including room boundaries. Other technical information may include data for thermal analysis, air flow analysis and air quality analysis.

In the lighting system example, information may include, luminaires technical specification, light distribution in space analysis, light intensity degradation over time analysis and so on. An array of sensors is distributed in the environment and for a period of time data is acquired from the test room. Sensor calibration is assumed to be performed by the manufacturer.

When all data are collected, these are analyzed and compared with the as-planned data, generating boundary conditions and sensor data analysis documents. This is used to update the building model instances. If the test results are correct this task is completed. Otherwise, errors are corrected and the process is restarted. At that moment, a test report is generated, containing the calibration use case and its results.

C. Setup building model server

The building model server setup process is shown in figure 4.

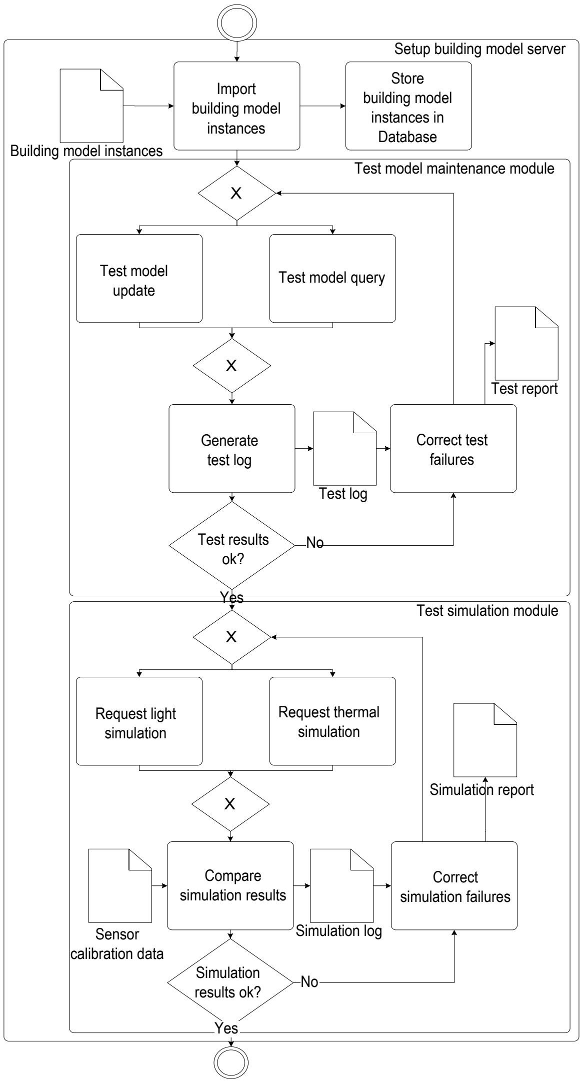

Fig. 4. Building model server setup process

First the building model is imported into the SaBC database. This task has to be performed after any change on the building model.

Afterwards the model maintenance module is tested. This module is responsible for model update and model queries. Model updates include off-line model update and on-line sensor updates. Model queries include model geometry queries and sensor state queries. Consequently, further documentation is generated. This documentation includes a log file reporting relevant tasks taken place during these tests, and a test report, describing each test case and its results. In case of failure, corrections are performed and test re-runs are done.

Furthermore the simulation modules are tested. Simulations are requested for each simulation tool. Then a comparison between simulation results and calibration values is done. The calibration results are generated during the physical test environment setup task (III-B). As result a log file reporting relevant tasks performed during simulation is generated, and a simulation report describing simulation use cases and results is created.

D. Setup application interface

An application interface is provided by the SaBC system to enable client applications interaction with the building model. The application interface setup process is shown in figure 5.

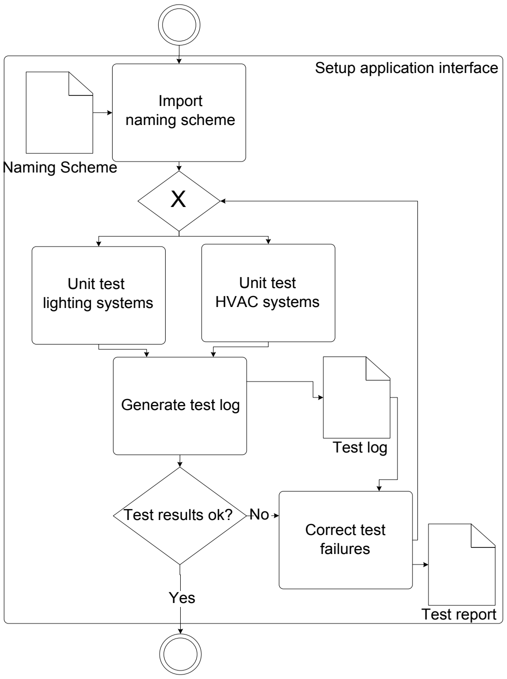

Fig. 5. Application interface setup process

Using the previously generated naming scheme, a mapping between the building model and the application interface is made. This involves the definition of a correspondence between building model entities and interface entities. As an examples, for lighting control, each luminaire or group of luminaires is given an ID (LUM01, LUM02), which may be referred to by client applications.

A unit test is performed for each building system independently. In a typical lighting system unit test, client application would establish communication with each lighting device present in the building model and would try to change its state.

A log is generated, compiling all tests results. Then failure identification is done, resulting in a test report document. If the test results are satisfactory, then the setup of the application interfaced is completed. Otherwise, failures are corrected and the setup process is re-initiated.

E. Setup building automation interface

The building automation interface setup process is shown in figure 6.

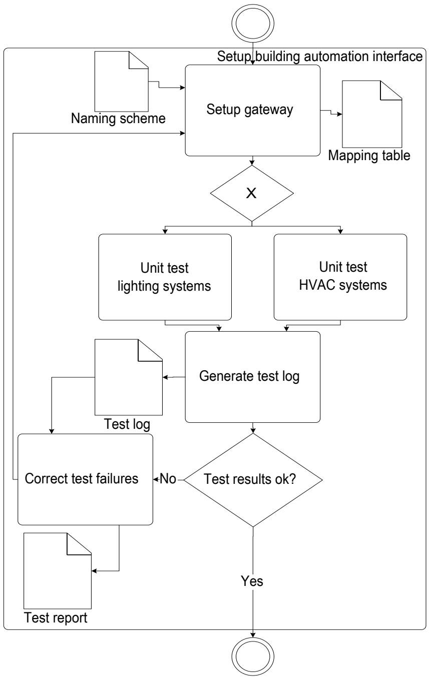

Fig. 6. Building automation interface setup process

The interface between the SaBC system and the physical building is performed through a gateway. The gateway setup is done using the formerly established naming scheme. A direct association between the real world and the virtual world sensors and actuators is made, resulting in a document containing a mapping table.

In parallel, for each available system, a unit test is executed similar to the unit test system described earlier (section III-D). A test log is produced. If the tests are satisfactory, this setup phase is completed; otherwise, test logs are analyzed to identify possible failures. A test report is made, including test use cases description and results.

Once the situations that created the test failures are identified and corrected, the setup phase of the building automation interface is re-initiated.

F. Setup controller

The controller setup process is shown in figure 7.

First, relevant sensors, actuators and other building technical elements are selected from the previously generated building model instances. For each building system, comfort and energy utility functions are loaded and set. Existing libraries of comfort and energy utility functions are used.

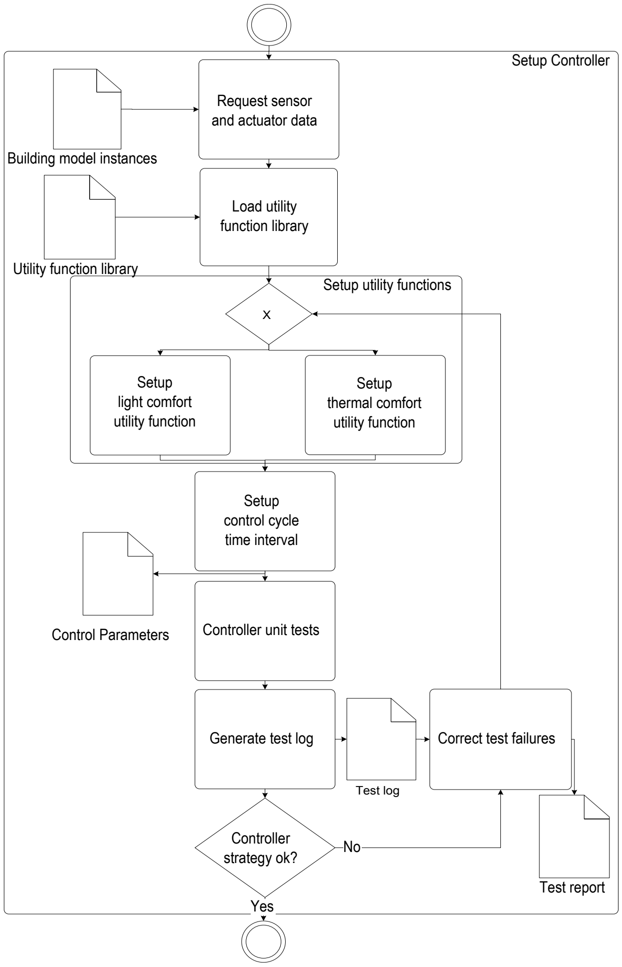

Fig. 7. Controller setup process

Afterward, time interval for the control cycles are defined and stored in a control parameter document.

A controller unit test is performed, producing a test log. If the tests are satisfactory, this setup phase is completed; if not, failure situations are corrected and the controller setup phase is re-initiated.

G. System testing

When all SaBC system modules are set up and unit tests are completed, full system tests are performed (figure 8).

In the context of SaBC deployment process the most significant tests are performance, security, scalability and recovery tests. Other tests, such as stress tests and load tests, are also important, but the former are considered the most important ones from the SaBC deployment process point of view.

Performance tests determine system performance in different conditions, such as night periods when users are absent, and relevant daily periods when users are present (for example, test with 3 hours periods). Performance testing is designed to

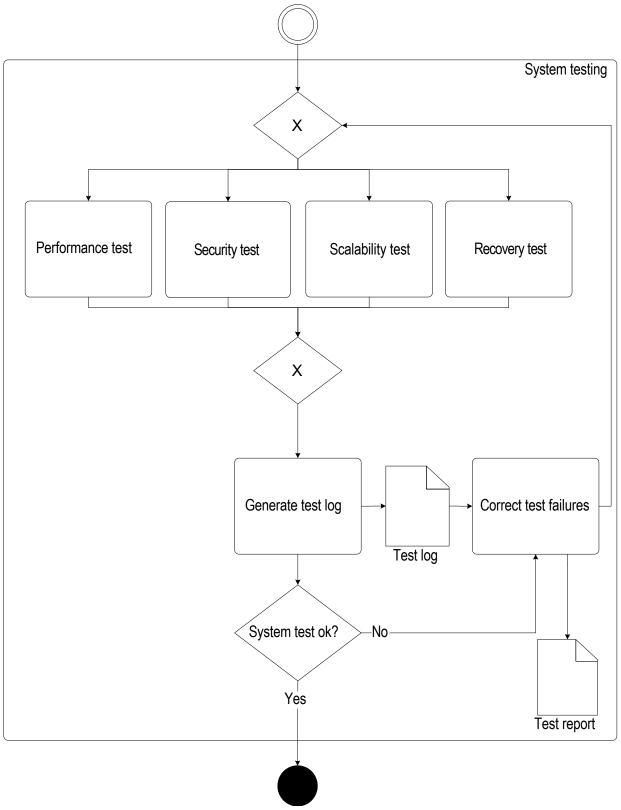

Fig. 8. System testing process

test the run-time performance of software within the context of an integrated system [16].

Security tests are concerned with security issues, such as devices take-over, middle-man attacks, flood-attacks, Denial Of Service (DOS) attacks, and so on. It attempts to verify that protection mechanisms built into a system will, in fact, protect it from improper use [16]. This test is performed by simulating such attacks. On the SaBC context it is particularly important to test flood-attacks and DOS attacks. On a SaBC, large quantities of data (sensor data, simulation data, model data, etc.) are continually communicated. If the SaBC cannot ensure secure recovery from flood-attacks and DOS attacks, substantial quantities of relevant data can be lost, affecting for example the building model accuracy, simulation assistance (in case simulation results would be lost during an attack) or even the entire SaBC system availability.

Scalability tests are an extension of performance tests. The purpose of scalability testing is to identify major workloads and mitigate bottlenecks that can impede the scalability of the SaBC and establish the growth limit of the system [17]. In the SaBC case, it is relevant to, for example, determine simulation workloads and mitigate bottlenecks on sensor data acquisition and model data update.

Recovery tests establish recovery procedures upon failure situations. It forces the software to fail in a variety of ways and verifies that recovery is well done [16]. Among the test performed are sudden device failure, network failure, communication failure, or even full system failure. On the SaBC system it is important to test sudden device failure, especially on the context of building model update and consequent simulation accuracy. Also communication failure is of major importance, due to the major negative impact of miscommunication on the overall performance and accuracy of the SaBC.

Naturally, after all tests are done, a test log is generated, describing relevant tasks. If the test results are satisfactory, the deployment is completed; otherwise, failure identification is performed, creating a test report, describing test cases and results. After the identified failure situations are corrected, the system testing is restarted. Failure correction may in some cases imply changes in previous steps. This means that the deployment process may have to be restarted.

IV. Proof of Concept

Initially information concerning the building, building systems and SaBC were collected. The test building is a passive house office building built in 2008. It is located in Vienna's 21st district and features a gross building area of 11,700 m^2 and a net building area of 7,500 m^2 [18].

The test building has a Siemens control and monitoring system. It is a building automation product which has an easy-to-understand graphic display providing simple, intuitive control of building systems. It uses BACNet [19] to communicate with field devices.

Regarding the scope, it was determined that the main objective for this building is to allow load shifting regarding technical equipment electrical energy consumption (excluding building lights due to safety reasons). This means to use load management technique that aims to move electrical energy demand from the peak hours to off-peak hours of the day.

Load management techniques are still not possible in the current Austrian energy market. Currently there is no difference between peak and off-peak hours electricity prices. Still, it was decided to investigate the possibility of shifting peak load in order to be prepared for future applications.

A prediction of the near future technical equipment electrical energy consumption is made for a certain time period. As input a building model, weather prediction and internal gains standard load profiles are used. As output, a prediction of the building energy consumption and ranking of the main technical equipment electrical energy consumers is shown. This allows for example, the facility manager to reduce consumption by turning off a pump for 30 minutes or reducing a fan to 20% of its capacity for 1 hour.

Figure 9 shows how system modules for test building relate to the modules in the SaBC system architecture.

Dymola Modelica [20] was selected as building model server and application interface and Siemens Desigo as building interface. The facility manager uses Siemens Desigo Insight V3.4 [21] to interface with the building automation system. TRNSYS [22] was selected as thermal simulation service. The controller is the building facility manager. This occurs often in real world application of SaBC systems, where

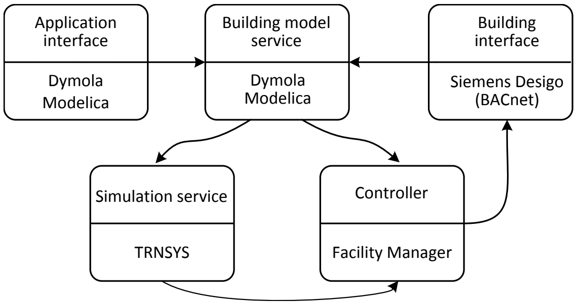

Fig. 9. Test building - SaBC system architecture.

the controller is not a software application, but a person who has to make decisions related to the building operation. This is mainly due to security and safety reasons. Unfortunately it cannot be guaranteed that control suggestions sent to the controller will be actuated.

The facility manager is experienced with building automation systems and has been operating this building since the beginning of the operation phase. According to the facility manager's experience, load shifting was only possible during summer time. Only the ground water pump and auxiliary pump for the concrete core activation could be actuated by reducing the operational frequency for a period of time, since the building thermal storage ensured comfort maintenance. All other system devices, including lighting devices, are critical to the building operation and could not be deactivated or its operation reduced.

Constraints regarding the well pump's operation range emerged during the controller setup task. The well pump cannot operate below 30 Hz according to the manufacturer (as shown in Figure 10). This means that the pump power consumption is always at least about 2 kW. This is, even though the cooling demand of the sub-system would already be satisfied at lower frequencies (15Hz), which would mean a power consumption of less than 1 kW.

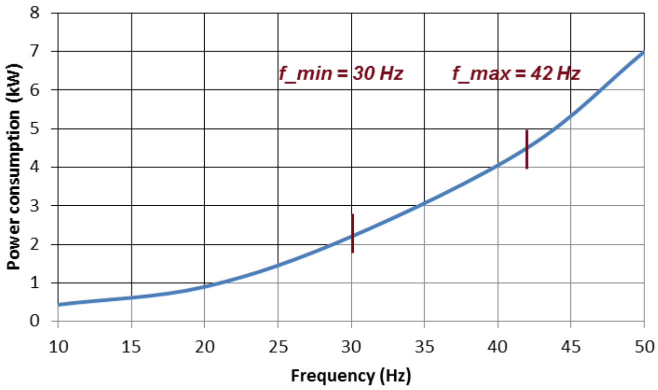

Fig. 10. Well pump power consumption and operative constraints.

Also, the building facility manager does not allow the pump to operate above 42 Hz with the argument that there is no gain in efficiency and that the pump longevity is reduced when operated over this threshold.

The control cycle was defined as follows. Every hour a scheduled system task is initiated (for example a cron job in Linux or a scheduled task in Windows). This task starts the Dymola Modelica application, which gathers building model and weather predictions data for the simulation service.

Three scenarios were defined, namely:

- well pump runs at 30 Hz for the next hour.
- well pump runs at 35 Hz for the next hour.
- well pump runs at 40 Hz for the next hour.

Simulations are requested for each scenario, using standard internal gains according to ASHRAE Standard and Norms [23]. Figure 11 shows the geometric thermal model visualization using Google Sketchup [24].

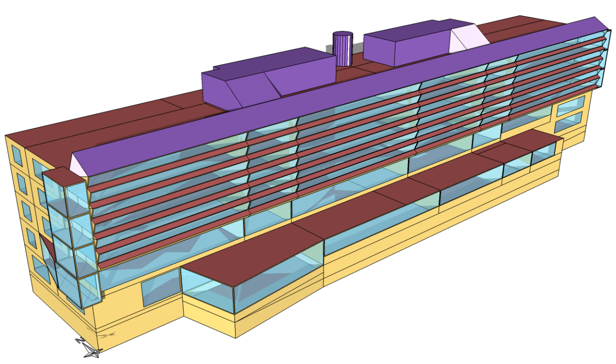

Fig. 11. Test building geometric thermal model [25].

When simulation results are available these are evaluated, ranked and compiled in a report email which is sent to the facility manager. The facility manager can then decide whether to actuate or not. The ranking is based on a the electrical consumption of the well pump and the indoor temperature. Which means that the most efficient scenario is running the well pump at 30 Hz, except on very hot summer days. This results from the combination of the scenarios and goals defined for the SaBC deployment.

In order to have realistic performance tests, these have to be performed at least for a period of 1 year. This ensures system performance is tested for at least one heating season, one cooling season and two intermediate seasons. This makes system testing a long and resource intensive process. Due to time limitation it was not possible to test system performance for this period of time.

The SaBC performance test, in this context, means to verify if a control cycle is performed within the selected time interval. Since the selected control cycle time interval was set to 1 hour by the facility manager, system performance is not an issue. The control cycle is performed in 4 minutes.

The SaBC security test, in this context, inspects if it is possible to influence the simulation output by breaching security within the SaBC. Since the SaBC system is not public (there is no connection to the internet) nor publicly accessible (computer is located in a restricted area) system security requirements are ensured.

The SaBC scalability test, in this context, assesses if by adding rooms or zones to the thermal model, the system still performs as desired. Since it is not foreseen that either the building or the building system will be scaled, system scalability is not an issue in this SaBC deployment test case.

The SaBC recovery test, in this context, confirms if the SaBC would still perform after the SaBC recovers from an unexpected power-down test. Since the SaBC is started periodically by an scheduled system task (for example a cron job in Linux or a scheduled task in the Windows Task Scheduler), system recovery is not an issue in this deployment test case.

V. Conclusion

This paper contributes to the definition of a workflow for the deployment of SaBC services. This minimizes the considerable complexity of the process, which requires careful coordination of tasks and integration of heterogeneous data and processes from different sources. The process model is intended to be sufficiently generic such that it may be used for projects independent of their size, SaBC control domains or SaBC capabilities. It allows an assessment of the deployment tasks. The SaBC deployment process is divided into several tasks. These tasks are setup building model, setup physical test environment, setup building model server, setup application interface, setup building automation interface, setup controller and system testing.

During deployment process test technical and time limitation were revealed. Load shifting potential is low due in the test building to technical limitations, consequently the SaBC advantages are overcast. It would be desirable to test the deployment process in other buildings and building automation systems. Due to time limitation it was not possible to test system performance for the period of time of one year, as desirable. The SaBC performance is highly depended of the simulation tools performance. The collaboration of many specialists may limit the implementation of the such a solution.

The deployment process described here has the potential to improve the management of the deployment data, simplify the deployment process and avoid errors by supporting tasks which are data intensive, repetitive and prone to error. Another benefit of the described deployment process is that the deployment can be performed by non-specialized personnel, with fewer training necessary. Specially large facility management service providers may profit from this deployment process.

Acknowledgment

This research was partially funded by the Austrian Research Promotion Agency (FFG) with the Klima- und Energyfonds program, project number 834677.

References

[1] European Commission Directorate General for Energy and Transport, *European energy and transport trends to 2030.*, Report, Brussels, 2003.

[2] A. van der Hoek and R. S. Hall and A. Carzaniga and D. Heimbigner and A. L. Wolf, *Software Deployment: Extending Configuration Management Support into the Field*, CrossTalk The Journal of Defense Software Engineering, VOL. 11, NO. 2, pp. 9-13, FEB 1998.

[3] A. Carzaniga and A. Fuggetta and R. S. Hall and D. Heimbigner and A. van der Hoek and A. L. Wolf, *A Characterization Framework for Software Deployment Technologies*, CU-CS-857-98, 1998.

[4] P.Y. Cunin and V. Lestideau and N.Merle, *ORYA: A strategy oriented deployment framework*, 3rd International Working Conference on Component Deployment, Grenoble, 2005.

[5] R. S. Aguilar-Saven, *Business process modelling: Review and framework*, International Journal of Production Economics, Elsevier, 90, 2004, pp. 129-149.

[6] M. L. Rosa and A. H. M. ter Hofstede and P. Wohed and H. A. Reijers and J. Mendling and W. M. P. van der Aalst, *Managing Process Model Complexity via Concrete Syntax Modifications*, IEEE Transactions on Industrial Informatics, VOL. 7, NO. 2, MAY 2011.

[7] D. Dietrich and D. Bruckner and G. Zucker and P. Palensky, *Communication and Computation in Buildings: A Short Introduction and Overview*, IEEE Transactions on Industrial Electronics, 2010, volume 57, number 11.

[8] W. Kastner and G. Neugschwandtner and S. Soucek and H.M. Newmann, *Communication Systems for Building Automation and Control*, Proceedings of the IEEE, 2005, June, Vol. 93, N. 6, pp. 1178-1203.

[9] A. Mahdavi, *Modeling-Assisted Building Control*, Proceedings of the CAAD Futures ’97 Conference (the 7th International Conference on Computer Aided Architectural Design Futures), Munich, Germany, 1997, pp. 219-230.

[10] A. Mahdavi and M. Schuss and G. Suter and A.S. Metzger and S. Camara and S. Dervishi, *Recent advances in simulation-powered building systems control*, Building Simulation 2009, L.B. McElroy, J.S. Turley (ed.), 2009, pp. 267-268.

[11] K. Brunner, *The Design of a Building Model Service*, TU-Wien, 2005.

[12] A. Mahdavi and M. E. Ilal and P. Mathew and R. Ries and G. Suter and R. Brahme, *The architecture of S2*, Building Simulation ’99: Proceedings of the Sixth International IBPSA Conference, 1999, 3, pp. 1219-1226.

[13] International Alliance for Interoperability, *Industry Foundation Classes IFC2x Edition 3*, http://buildingsmart-tech.org/ifc/IFC2x3/TC1/html/index.htm, 2012.

[14] A. Chazalet and P. Lalanda, *Deployment of Services-Oriented Applications Integrating Physical and IT Systems*, 21st International Conference on Advanced Networking and Applications(AINA’07), 2007.

[15] J. Bordogna, *Tomorrow Civil Systems Engineer - The Master Integrator*, Journal of Professional Issues in Engineering Education and Practice, Vol. 124, No. 2, 1998, pp. 48-50.

[16] R.S. Pressman, *Software engineering: a practitioner’s approach*, ISBN 9780073375977, LCCN 2008048802, McGraw-Hill Higher Education, 2010.

[17] S. Desikan and G. Ramesh, *Software Testing: Principles and Practice*, ISBN 9788177581218, Pearson Education Canada, 2006.

[18] T. Goschenhofer, *Energy performance of the office building ENERGYbase (heating, cooling, air conditioning, lighting*, UAS Technikum Vienna, 2011

[19] S. T. Bushby, *BACnetTM: a standard communication infrastructure for intelligent buildings*, Automation in Construction, volume 6, issue 5-6, pp. 529-540, 1997.

[20] Modelica Association, *Modelica - A Unified Object-Oriented Language for Systems Modeling Language Specification Version 3.3*, URL = ”https://www.modelica.org/documents/ModelicaSpec33.pdf, 09.05.2012.

[21] SIEMENS, *Siemens Desigo Insight - Desigo building automation: flexible and energy-efficient*, http://www.hqs.sbt.siemens.com/gip/general/dlc/data/assets/hq/Desigo-building-automation—flexible-and-energy-efficient_A6V10227660_hq-en.pdf, 22.09.2012

[22] LLC Thermal Energy System Specialists, *TRNSYS*, http://www.trnsys.com, 22.01.2012.

[23] ASHRAE, *Standard 189.1-2009. Standard for the Design of High-Performance Green Buildings Except Low-Rise Residential Buildings*, 2009.

[24] Google, *Google SketchUp*, http://sketchup.google.com/product/, 17.10.2012

[25] C. Hettfleisch, S. Ledinger, F. Dubisch: *Energiesparpotenzial eines Passivhauses unter Bercksichtigung von Wetterprognosen*, Vortrag: BauSIM 2010, Wien; 22.09.2010 - 24.09.2010; in: ”BauSim2010 Proceedings”, (2010).