Developing web-services for distributed control and building performance simulation using run-time coupling

Yahiaoui, A., Hensen, J.L.M., Soethout, L. & Paassen, A.H.C. van (2007).
Developing web-services for distributed control and building performance simulation using run-time coupling.
In Jiang, Yi (Ed.), Proceedings of the 10th IBPSA Building Simulation Conference, 3-5 September, pp. 1327-1333.
Beijing: Tsinghua University.

DEVELOPING WEB-SERVICES FOR DISTRIBUTED CONTROL AND BUILDING PERFORMANCE SIMULATION USING RUN-TIME COUPLING

Azzedine Yahiaoui[*1], Jan Hensen[1], Luc Soethout[2], Dolf van Paassen[3]

[1]Center for Building & Systems TNO-TU/e
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[2]TNO Built Environment and Geosciences
Postbus 49, 2600 AA Delft, The Netherlands
[3]TU Delft, Department of Mechanical Engineering
Mekelweg 2, 2628 CD Delft, The Netherlands

ABSTRACT

This paper reports the final results of a research project, which aims to achieve better control modeling in building performance simulation by integrating distributed computer programs. The paper focus on developing Web-services based SOAP/XML to run-time couple control and building performance simulation. Data exchange via Web-services allows system components to be loosely-coupled, rather than combined into an integrated building control systems. The paper also details an example apllication using this technology to configure a distributed simulation through an open protocol, like BACnet or LonWorks and so on. Particularity, this paper concerns the relevance and reliability of integrating Web-services with run-time coupling of control and building performance simulation environments over TCP/IP protocol suite. In addition, this approach provides a reusable patterns either for other similar projects or for real automated building applications.

KEYWORDS

Web-services; Building Automation Systems; Run-time Coupling; Building Performance Simulation; Distributed Control Systems.

INTRODUCTION

Recent developments in Building Automation and Control Systems (BACS which may also be known by a name of BAS) show that data communication plays an essential role in ensuring the operation between components of different network technologies. In general several Building Automation Systems (BAS) go by encompassing other names such as: Energy Management Control Systems (EMCS), Building Management Systems (BMS), Building Energy Management Systems (BEMS), Facility Management Systems (FMS) and so on. Nevertheless, in this paper the BAS's name is chosen and explicitly defined as a system comprising all products and engineering services for automatic control techniques used in buildings. Furthermore, those control techniques consist in their turn of monitoring, supervision, optimization for operation, human intervention and management to minimize energy consumption and safe operation of building services. While Building Automation Systems (BAS) starts to become advanced, well established and more sophisticated by offering a vast diversity of control functions for all the systems that operate within the building environmental performance, several network protocols (BACnet, LonWorks, Modbus, etc.) have been developed for a better exploitation. In BAS, an open distributed control systems composing of various components (sensors and actuators), control systems and one or more open network protocol interconnecting them, is used to ensure the communication of information through routers with the central computer. For assuring the control analysis and performance of various building equipments and components with the important data exchange on the network, as depicted in figure 1, a simulation based on distributed of control systems and building environmental performance is required.

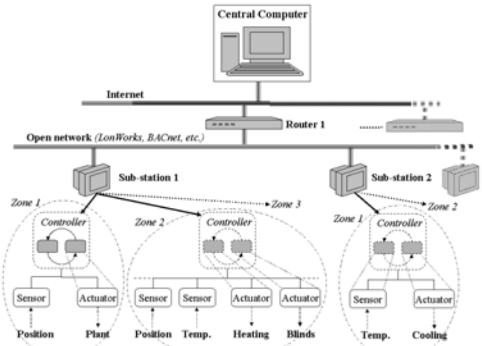

Figure1. Architecture of Building Automation Systems (BAS)

The development of automatic control systems has resulted in intelligent buildings with the automation

[*] Corresponding e-mail: a.yahiaoui@bwk.tue.nl

of HVAC (Heating, Ventilation and Air-Conditioning) and lighting systems in order to provide a healthy, comfortable and productive indoor environment with a minimum of energy use, and to control them in an efficient and rational way (see Yahiaoui et al. 2005). In fact, BAS exploits information between the central computer and sub-stations (or terminals) through different network protocols like LonWorks and BACnet, it is necessary to use a router since those protocols utilize different data format network representations (see e.g. Kastner et al. 2005). This data is therefore a means that BAS use to exchange information between management level and field level through an open protocol, like BACnet or LonWorks and so on. Although both significant protocols BACnet and LonWorks exploit different format of object-oriented data structure that BAS communicate among HVAC (Heating, Ventilation and Air-Conditioning) systems and building components, it is sometimes hard to find a very flexible data format that BAS can integrate when BACnet and LonWorks components are employed. To explore this potential, the paper focuses mainly on a practical approach for converting data and services between different network technologies used for building automation. This approach is based on gateways that can interconnect entirely different protocol stacks. Such gateways use a common format for data exchange, like XML (eXtensible Markup Language). But XML alone is not enough to clearly communicate such expectations between different technologies. although XML is only an exchange format, it is not an architecture oriented protocol. Consequently XML does not provide universal solution to exchange data between multi-protocols of building automation applications. On the other hand, SOAP is a simple XML-based protocol to let applications exchange information over an internet, more exactly over HTTP (HyperText Transfer Protocol). It is therefore, essential that BAS system utilizes a standardized way of integrating Web-services (or Web-based applications using both XML and SOAP) to exchange data between building components of different technologies.

For such a reason this paper presents a combination of XML and SOAP (Simple Object Access Protocol) to integrate distributed simulation where control modelling and building performance simulation environments are separated and work together through run-time coupling mechanism. Accordingly, a prototype of a distributed simulation mechanism of building automation and control systems has been implemented to offer a selective choice of all control functions possible for the entire systems that operate within building intelligent environments. This prototype is based on utilisation of Internet sockets in order to exchange information between control modelling and building performance simulation environments in distributed heterogeneous network.

Although a methodology developed in this paper is based in the encapsulation of data exchanged between ESP-r and Matlab with Web services, both control and building performance applications can run on different operating systems (Windows, Unix and Linux). As a result, an approach based on the simulation of building automation systems (BAS) can be enabled by similarity to run-time coupling of automatic control systems and multiple building performance simulation. In addition, various protocols used by BAS system are highlighted to be supported within distributed simulation.

The first part of this paper starts by giving a description of distributed control and building performance simulation. Then, it elaborates the reasoning behind our hypothesis that run-time coupling will facilitate integrated performance assessment by predicting the overall effect of innovative control strategies for integrated building systems. The major part of this paper is a discussion resulting in data requirements in view of integration in real building control protocols (BACnet and LonWorks). The paper finishes by indicating the future application of using Web-services to run-time couple control and building performance simulation.

DISTRIBUTED CONTROL & BUILDING PERFORMANCE SIMULATION

One of several key issues facing us when we want to simulate building and plant models plus control application is that frequently certain system components and/or control features can be modeled in one simulation environment while other components and/or control features are only available in other simulation tools. In other words, there exists a dependent domain relative for control modeling environments (CME), which is very advanced in control modeling and simulation features (e.g. Matlab/Simulink, Mathematica, etc.), but still not totally used for building environmental performance. On the other hand, there is a domain specific software for building performance simulation (BPS), which is usually relatively basic in terms of control modeling and simulation capabilities (e.g. ESP-r, TRNSYS, etc.). To alleviate such a restricted issue mentioned above, it is necessary to reason behind our hypothesis that marrying two approaches by run-time coupling would potentially enable integrated performance assessment by predicting the overall effect of innovative control strategies for integrated building systems. Although there is a quite number of tools developed for different domains, ESP-r and Matlab/Simulink are chosen for BPS and CME respectively. Furthermore, the strategy to run-time couple ESP-r and Matlab developed in this work takes the form of closed-loop control system, as shown in figure 2. Since run-time coupling between ESP-r and Matlab takes the form of closed-loop, distributed control and building performance simulation can be performed either in the feedback control loop or in the feedforward control structure. Both control loops can then be used for any simulation of control and building systems. In case an open-loop control is performed for building application, it would just necessitate to setting in Matlab side, the variables required passing back to ESP-r to zero. More exactly as defined in Mex-file language, those variables are the rhs (right hand side) arguments. However, both unidirectional and bidirectional communications can now be carried out within this developed strategy for run-time coupling.

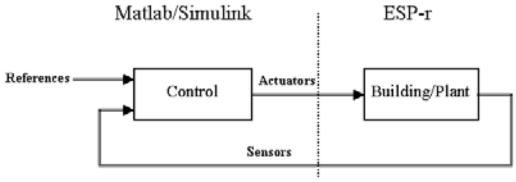

Figure 2. Structure based run-time coupling of control and building performance simulation

In previous work of (Yahiaoui et al., 2003 and Yahiaoui et al., 2005), it has been described that a promising approach to run-time coupling between ESP-r and Matlab/simulink is an IPC (Inter-Process Communication) using Internet sockets. Hence, this approach performs a distributed simulation through an open network protocol to exchange data between building model and its controller, as it almost happens in BAS system. Both building model and its controller which are separated and work together through run-time coupling mechanism can be located on a different kind of hosts (Windows, Unix or Linux), in which the performance simulation is much faster than using a single computer. Consequently, the development of this new advent would potentially enable new applications of building control strategies that are not yet possible.

However during the simulation, commands and data are transmitted between ESP-r and Matlab/Simulink. If for instance the building model (i.e. ESP-r) has to send its current measured variable to its controller (i.e. Matlab/ Simulink) with TCP/IP-stream format, a method called encodes and transmits them with a defined control sequence via TCP/IP to a method received. This then receives the control sequence, decodes data from TCP/IP-stream format and sends them to the recipient (Matlab/Simulink). When the controller has to send back the actuated variable to its building model via TCP/IP suite, a new procedure is executed again in the opposite way. Figure 3 illustrates a complete understanding on how data is exchanged between ESP-r and Matlab/Simulink by run-time coupling.

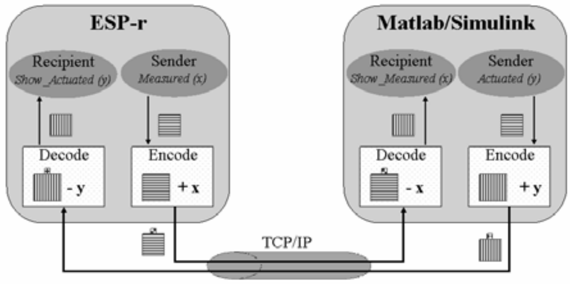

Figure 3. Distributed control and building performance simulation environments

In the current implemented approach of run-time coupling between ESP-r and Matlab, it is ESP-r which starts the simulation. Indeed, Matlab is a server of ESP-r, launched at every ESP-r time-step as a separate process. If the connection between ESP-r and Matlab breaks down, need data to be exchanged cannot be transferred till the communication between them is reconnected. More detail about distributed control modeling and building performance simulation by run-time coupling can be found in (Yahiaoui et al., 2004 and Yahiaoui et al., 2005).

As result of enhancing run-time coupling of control modelling and building performance simulation, there is a cooperative simulation (or a co-simulation) that uses Web services for communications in distributed building control systems. The originality of this work consists, by generality in terms of developement and improvement, of extending the infrastructure of run-time coupling between ESP-r and Matlab to simulate the environment of BAS system. This has for purpose to represent the entire BAS system in simulation, as shown in figure 4.

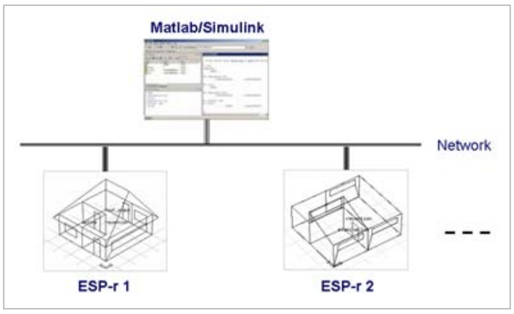

Figure 4. Run-time coupling between Matlab/Simulink and multi-ESP-r

Being physically distributed between Matlab and multi-ESP-r with run-time coupling is perphaps the major feature of BAS systems, with a key advantage derrived from simplified improved infrastructure. This infrastructure is developed and implemented with multitasking (threads) method in which a BAS application can simultaneously be executed across multiple processors or cores, as illustrated in figure 1. The creation of multi-ESP-r is, at this time limited to 10 and Matlab is server of those multi-ESP-r connected through network. We believe that such an infrastructure is suffisant to represent BAS system used for different buildings on big campus.

Data communication in a real building

In real building operations, it is desirable for control laws to communicate with building components or HVAC systems. The communication between controllers and building systems requires an interface or a gateway, due to their different communication speeds and data formatting. The proper operation of the gateway is dependent on the continued use at the corporate level to ensure the communication between controllers and plants and building components. In consequence, two important standard protocols "BACnet and LonWorks" (see AutomatedBuildings, 2007) are mostly exploited for building automation. Although BAS (building automation systems) can use TCP/IP as the underlying connection protocol for automated buildings, a data exchange protocol such as XML is dominating the future of interoperability among protocols. With the introduction of run-time coupling of ESP-r and Matlab/Simulink, the approach developed for exchange data between them provides a flexible TCP/IP connectivity to any application of building automation and control systems. By using this Internet protocol and a common Web documents, run-time coupling mechanism can play a role of BAS functions with more or less the same similarities.

Both protocols cover most or all layers of the OSI (Open System Interconnection) model used for network communications. The similarity of approaching a run-time coupling mechanism to BAS system concerns several ideas developed by considering basically the top-layer of the OSI model (i.e. application layer). The other layers deal with hardware specifications and the low-level details of the network communication. In our work, these details are already covered by the socket implementation. Though the application layer deals with the conceptual part (i.e. with the way of information is modeled), both protocols (BACnet and LonWorks) use different mode of exchange. What follows is a brief overview:

- BACnet (ASHREA 135, 2001) is a specification for a standard protocol published by ASHRAE organization. While the data exchanged in BACnet is encapsulated in objects accessed by services calls, BACnet can ensure the interoperability between devices of different industries. In consequence, the run-time coupling mechanism is implemented in order to create a communication protocol that complies with this specification. The standard defines protocol implementation conformance statements (PICS) that identify different levels of compliance. This meets a field level of building automation services (BEMS or BAS), in which run-time coupling can use various standardized set set of data types to communicate between ESP-r and Matlab as it is over BACnet protocol.
- LonWorks (Loy et al. 2001) is not specifically designed for automated building networks, but is also used in other industries. Now it is gaiting a high importance use in BAS architecture. While supporting communications on a variety of LANs or allowing different components to co-exist in the same network, LON (or LonWorks) communicates via Standard Network Variable Types (SNVT) to facilitate interoperability by providing a well-defined interface. Basically a proprietary communication protocol is called LonTalk, created by ECHELON Corporation. A chip is required for any device that uses LON. Consequently run-time coupling can also establish standard network variable formats to allow the transfer of data between Matlab and ESP-r by using the LonMark subset of LON capabilities to interoperate with each other.

It should be noted that integrating Web services in BAS system would provides a great flexibility to communicate between diverse devices and through various protocols designed with different technologies. In order to approach by similarity distributed control and building performance simulation using run-time coupling to BAS system, a data exchange format is then developed for assembing C/C++ Web-services as part of a service-oriented architecture. This format is composed of a typed model for encoding-based interoperability of protocol elements of different technologies.

Specifying different protocol objects encodings

Another important area of data communication in real buildings has to do with rules and mechanisms for encoding in their network object proprieties. So far, both BACnet and LomWorks protocols provide an elegant mechanism for specifying the encoding rules that apply to the object (or message) as a collection of data related to a particular function. Such a function can be uniquely identified and accessed over a network in standardization way.

All information in a BACnet protocol is represented by data structures, as described in figure 5. Each data structure consisting of one object that has a set of proprieties can be read from, and written to, by executing so called services. Although there are thousands of potentially useful object types that might be found in building automation.

Object_Name	Room Air Temperature
Object_Type	Analogue input
Present_Value	21 ºC
Status_Flag	Normal
High_Limit	25 ºC
Low_Limit	17 ºC

Figure 5. Example of BACnet object proprieties

On LonWorks networks, all information is given by LonMark objects using LON function definitions, as described in figure 6. Each object has the capability of defining multiple functional blocks by filling out the node self-documentation string, and the respective network variable. LON objects can be different in number of data structure which is based on executable specifications of a device.

SVNT_temp	
Data Length	1
Suggested Data Array Format	Float
Data Item 1	
Measurement	Temperature
Units	Degrees Celsius
Values Range	-274.0 .. 6,219.5
Invalid Value	N/A

Figure 6.Example of LonMark object proprieties

The object concept of different protocols (BACnet and LomWorks) allows one to organise information relating to sensors and actuators, plus non-physical concepts like software execution or calculations. All thes types have direct representation in XML schema so it is easy, through the use of SOAP attributes, to describe the type of data being passed in a message. For such a reason, Web services interoperability is merged with sockets API (Application Programming Interface) to exchange data by run-time coupling between Matlab/Simulink and ESP-r in practical logic applications of automated buildings.

Data exchange format in run-time coupling

The format for data exchange protocols exploited in this work is based on using both XML (eXtensible Markup Language) and SOAP (Simple Object Access Protocol) Web services in order to communicate between ESP-r and Matlab/Simulink with any data interchange format. Although ESP-r and Matlab/Simulink are intefaced with sockects internet by using C++ language, some compiler tools provide a SOAP/XML-to-C/C++ language binding to ease the development of Web services and client application writen in C and/or C++. The gSOAP (Engelen, 2006) is used in this work to generate efficient Web services and user-defined C/C++ data types. In reality, SOAP was created to exploit XML documents based protocol in order to communicate between different applications running on different operating systems, with different technologies and programming languages. Although most BAS architecture can use several protocols of different technologies, it is essential for the run-time coupling mechanism to adopt a certain capability of exchanging data between ESP-r and Matlab in heterogeneous distributed environments. Figure 7 illustrates how Web services based SOAP/XML is implemented to run-time couple ESP-r and Matlab on different operating systems.

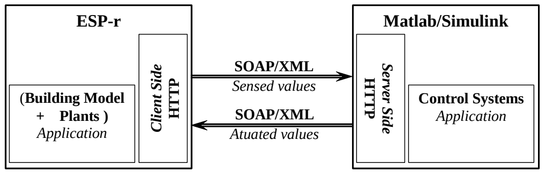

Figure 7. Implementation of Run-time coupling between ESP-r and Matlab using Web services

To communicate between Matlab and ESP-r in distributed heterogeneous environments, XML data binding is used to manipulate data easily with the XML-Documents via a set of simple objects. But to transfer XML-Documents using HTTP (HyperText Transport Protocol) between ESP-r and Matlab, it requires to also use SOAP in order to ensure packaging, encoding, and exchanging of structured data on run-time coupling. Figure 8 illustrates an example of programming code based SOAP message containing an XML-Document of two elements needed at least for data exchange between ESP-r and Matlab/Simulink environments.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="DataExchange"
 targetNamespace="urn:myuri:1.0"
...
<types>
SOAP-ENV:encodingStyle="http://soap/encoding/"
xmlns:SOAP-ENC="http://soap/encoding/"
xmlns:SOAP-ENV="http://soap/envelope/"
xmlns:xsd="http://2001/XMLSchema"
xmlns:xsi="http://2001/XMLSchema-instance">
xmlns:ns1="urn:myuri:1.0"
xmlns="http://2001/XMLSchema"

  <!-- operation request element -->
  <xs:element name="Air_Temp" type="ex:Celsius"/>
   <complexType>
    <sequence>
      <element name="TCTL" type="xsd:int" minOccurs="17"
      maxOccurs="25" nillable="true"/>
    </sequence>
   </complexType>
  </element>

  <!-- operation response element -->
  <xs:element name="Heat_Flux" type="ex:Watt"/>
   <complexType>
    <sequence>
      <element name="Q" type="xsd:int" minOccurs=0
       maxOccurs="unbounded" nillable="true"/>
    </sequence>
   </complexType>
  </element>
 </schema>

</types>

<message name="DataSent">
 <part name="parameters" element="ns1:Air_Temp"/>
</message>

<message name="DataReceived">
 <part name="parameters" element="ns1:Heat_Flux"/>
</message>
...
```

Figure 8. A SOAP message containing XML documents exchanged between ESP-r and Matlab with protocol object proprieties

This structured data is then described in the form of XML-Documents. In fact, SOAP is most commonly associated with Web services, serving as the de-facto standard message envelope (O'Reilly Media, 2006). For our purposes, it was essential to implement run-time coupling between ESP-r and Matlab is on heterogeneous network operating systems (Windows, Unix, etc), in which a SOAP message containing XML document is advantageous in developing a communication platform independent.

SIMULATION EXAMPLE

The distributed simulation environment described above, was developed and implemented to enable advanced control systems in building performance simulation (Yahiaoui et al., 2003 and Yahiaoui et al., 2005). In this project, we validated the concept of run-time coupling of control and building environmental performance by comparing simulated and experimental results (Yahiaoui et al., 2006). To demonstrate the utility of integrating advanced control systems in buildings, we simulated and tested within a test cell, designed for experiments, a complex application. The application concerns the optimization issues that lead to keep the indoor air temperature comfortable while the energy consumption is minimized.

A graphical user-interface has been developed both sides of run-time coupling between ESP-r and Matlab/Simulink. This interface facilitates to the user to choose one of communication modes (synchrone or asynchrone) and one of protocols (TCP, BACnet or LonWorks), as described above. Figure 9 shows a part of interfaces developed on Matlab/Simulaing side. Hence, the distribred mechanism simulation which developed to run-time couple ESP-r and Matlab/Simulink is flexible in terms of using any controller (or control system) for any building model with a possibility of data exchange through any communication mode and protocol.

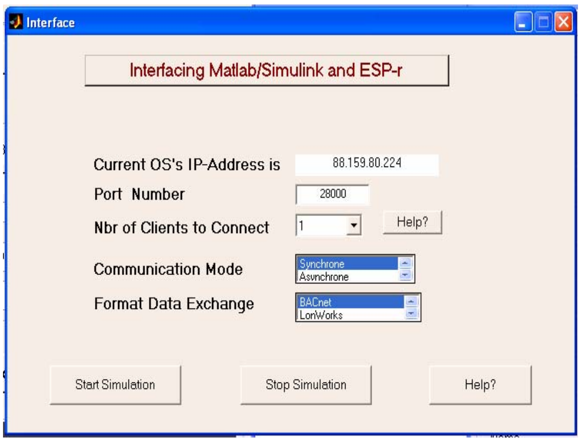

Figure 9. A first part of graphical user- interface developed on Matlab/Simulink side

Figure 10 details the measured time for different sizes of a SOAP message containing XML documents. This message is exchanged between ESP-r and Matlab and sent across network with different protocol object proprieties. This exchange of web services message between ESP-r and Matlab results in communication performance with reference to serveral operations. Those operations are data parsed to and data unparsed from XML document during the exchange. In this test, the internet connection is used as local network, in which it speed is 10Mbps.

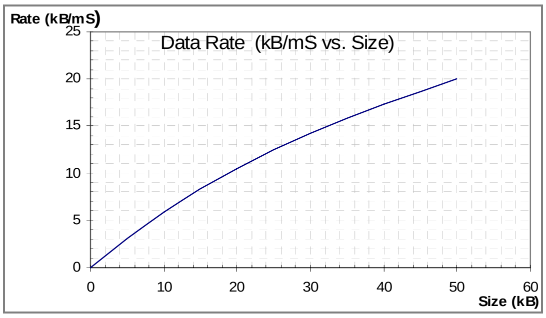

Figure 10. Response time for Web-services communication between ESP-r and Matlab

It is important to point out that the speed of internet connection, as shown in figure 10, would not influence the simulation results of parallel processing. For example some building plants require small time-step in simulation when the control of their responses needs to be faster for safety reasons. For 50 kB of Web-services data exchange requires about 21 mS of transmission time between two communicating parts. though when the speed of nework ligne is higher, the performance can be better for such an internet connection.

CONCLUSION

A new approach to the distributed simulation environments is presented with a methodology of developing Web-services based SOAP/XML to facilitate data exchange between components of different technologies. This approach would explore more control options in building environmental performance without modifying the physical simulation. The use of Web-services based SOAP/XML in BAS architecture allows us to integrate building components and equipements of different network technologies.

The ability of developping Web-services based SOAP/XML, in this paper, has for the purpose of making valuable components in a multi-protocol environment (i.e. having BACnet and LonWorks protocols on the sane network). SOAP is exploited to easily interface applications writen in different languages with a commun data exchange. XML is used to provide a common format for data exchange between different appalications.

Further objectives or our research include physical verification with validation of experimental results and utilitarian verification by means of practical application in realistic design studies. However, this research will generate associated knowledge for general and wider applicability.

REFERENCES

AutomatedBuildings, 2007. The Automated Buildings website, Inc. < http://www.automatedbuildings.com/news/jan02/art/hk/hk.htm>

ASHREA 135, 2001. BACnet –A Data Communication Protocol for Building Automation and Control Systems, ASHREA Inc.

Engelen, V. R., 2006. gSOAP 2.7.8 User Guide, Florida State University and Genivia Inc. < http://www.cs.fsu.edu/~engelen/soapdoc2.html>

Kastner, W., Neugschwandtner, G., Soucek, S. and Newman, H. M., 2005, Communication Systems for Building Automation and Control", IEEE journal, Vol. 93, n. 6, pp. 1178-1203

Loy, D., Dietrich, D. and Schweinzer, H. J., 2001. Open Control Networks – LonWorks / EIA 709 Technology, Kluwer Academic Publishers.

O'Reilly Media, 2006, O'Reilly's site < http://www.xml.com/pub/a/ws/2002/04/16/soap.html>

Yahiaoui, A., Hensen J.L.M. and Soethout L.L. , 2003, Integration of control and building performance simulation software by run-time coupling, IBPSA Conf. & Exhib., Vol. 3, pp. 1435-1441, Eindhoven, NL

Yahiaoui, A., Hensen, J., and Soethout, L., 2004, Developing CORBA-based distributed control and building performance environments by run-time coupling, 10th ICCCBE, Germany

Yahiaoui, A., Hensen J.L.M., Soethout L.L. and Van Paassen, D. , 2005, Interfacing of control and building performance simulation software with sockets, IBPSA Conference and Exhibition, Canada

Yahiaoui, A., Hensen J.L.M., Soethout, L., & van Paassen, D., 2006, Simulation based design environment for multi-agent systems in buildings, 7th International Conference on System Simulation in Buildings, Belgium.