

WEB SERVICES BASED DATA ACQUISITION FROM A PROCESS DATABASE

J. Cigler¹, J. Široký^{2,3}, M. Kulvejt³, M. Chlupáč³, D. Gyalistras⁴

¹Department of Control Engineering, Faculty of Electrical Engineering, CTU in Prague

²Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia in Pilsen

³Energocentrum Plus, s.r.o.

⁴Automatic Control Laboratory, ETH Zurich

Abstract

Advanced control techniques have been recently applied to building control in several pilot applications. Design and deployment of these advanced control techniques however require interface between numerical tools and building management system. In this paper, the complete software design of a web services Matlab client for a process database will be described. The main contribution of this paper lies, however, in the proposed alternative approach how to use web services within Matlab as the standard Matlab functionality is not capable to handle given problem. Discussion on why it is necessary to use another approach will be treated. The proposed strategy is based on Axis2 package (from the Apache Software Foundation), which provides a functionality to generate Java stubs based on the description of the web service. Java stubs are then compiled using arbitrary Java compiler into a form of jar package, which is dynamically added into Matlab Java path. Couple Matlab objects were proposed to make web services calls more convenient. Moreover, communication interface, properties of the process database as well as the proposed Matlab object scheme will be detailed.

1 Introduction

Advanced control techniques such as model predictive control have been recently applied to building control in several pilot applications [1]. Design and application of these advanced control techniques require interface between numerical tools and building management system.

In this paper, the complete software design of a web services Matlab client for a process database will be described. In this case study, we consider RcWare building control and management system. RcWare system consists of several levels, which are connected via standard interfacing technologies so that the topology is open for third party integration at any level.

There is wide variety of such process databases and RcWare system is only one of them. Each of them can/cannot provide users with a Matlab data acquisition tool which is usually realized by Matlab OPC Toolbox, Matlab Database Toolbox or possibly by a different software tool/driver. Hence, the main contribution of this paper lies in the proposed solution how to work with web services in Matlab and the resulting software tool is considered as a side contribution.

Currently, web services support in Matlab is based on Java library WSDL4J which stands behind the entry point for the web services usage i.e. function *createclassfromwsdl*. This function process url heading to Web Service Definition Language (WSDL) file and creates Matlab object having desired functionality based on the retrieved information.

From our experience, some general object structures in the WSDL file can cause problems, which WSDL4J is unable to tackle with. The function *createclassfromwsdl* cannot create the objects, which limits the work only to certain web services problems. Thanks to excellent Java support in Matlab, one can choose another java library capable to work with more complex objects and include it into project in a very straightforward way.

In this paper, we will present an alternative approach to create class from WSDL. Axis2 (from the Apache Software Foundation) provides a functionality to generate java stubs based on the WSDL description. The code is generated according to Axis2 Data Binding Framework (ADB) using Axis2 *wsdl2java* tool. The stubs are then compiled into jar package, which is added into Matlab Dynamic Java Path. However simple this procedure may look like, it brings about some cons. Lot of Java

objects is created, mainly due to strong typing of Java, which complicates work with the resulting stubs.

Apart from the detailed description of the aforementioned web services issue (content of Section 3.), communication interface and properties of RcWare system will be detailed in Section 2. Moreover, the proposed Matlab object scheme (which can be characterized as a layer between user and generated Java stubs) and performance comparison between various RcWare web services clients will be treated in Section 4. Finally, Section 5 will conclude the paper.

2 Process Database Description

Building managements systems (BMS) do store enormous amount of data. To illustrate this let us consider a modern office building where 2000 values are measured and stored each 3 minutes. That gives 9.6×10^5 values per day and 3.5×10^8 values per year. On the other hand, user of BMS requires access to any data in database in a reasonable time. This raises serious demands on the functionality of the process database, which is in the heart of every BMS. In this work, we consider RcWare Building Control and Management System and its process database RcWare DB. RcWare DB was designed with respect to BMS specifics. It enables to store billions of values and it provides quick access to the data without any special hardware requirements.

RcWare DB (<http://rcware.eu>) provides two additional features compared to the commonly used BMS. Firstly, RcWare DB can be used as an interface between BMS (or any other data interpreter such as Matlab) and SQL database (PostgreSQL, MS SQL, MySQL, etc.). In this case, RcWare DB optimizes data storage with respect to the selected SQL database engine. Customer can use his/her current database system and make use database engine capabilities thanks to RcWare DB interface. This type of installations is suitable for large applications where millions of values are stored annually. The matter of crux lies in saving signals in an incremental way, i.e. measured value of a particular signal is stored only if its change is bigger than certain threshold. The threshold is then viewed as a parameter of the *compression*. The bigger the threshold is, the smaller amount of data is stored in the SQL database. As the data is stored in the database in a compressed way, one data request can contain longer time frame of the desired data.

When user deals with smaller amount of data, then RcWare DB can be used without SQL server. Internal database library SQLite is used in that case instead of SQL server. Advantage of such solution is straightforward expandability of the data storage system.

Secondly, clients do connect to RcWare DB by standard web services interface (SOAP 1.1). Thanks to use of a standard technology, implementation of a new client is simple and can be done within a few minutes in modern programming languages. A client application does not have to manipulate raw data. Client application uses web services to call RcWare API methods that do manipulate the raw data. RcWare API provides methods as depicted in Figure 1. Apart from data acquisition and saving, client can operate with database variables and metadata associated with a particular variable.

RcWare DB API
<pre>+addMetadata(c:Credentials, amr:MetadataRecord []) +checkCredentials(c:Credentials) +deleteMetadata(c:Credentials, recIds:int []) +getAllVariables() (c:Credentials) +getData(c:Credentials, kvp:KeyValue [] [], from:Calendar, to:Calendar) +getMetadata(c:Credentials, kvp:KeyValues []) +getVariables(c:Credentials, kvp:KeyValue [][]) +saveData(c:Credentials, vr:ValueRecord []) +serverAlive()</pre>

Figure 1: List of RcWare API methods

Web Services Description Language (WSDL) describes the server side functionality and hence one can use standard tools to generate stubs in various programming languages.

3 Usage of Web Services from within Matlab

There are basically two elementary approaches allowing usage of web services from within Matlab. The former can be characterized as a simpler and more convenient approach because users just use function *createclassfromwsdl* with location of WSDL file as an argument. This function prepares all the web services functionality into a form of a Matlab class. The latter approach requires knowledge of SOAP protocol and user has to create, call and parse SOAP messages him/her-self. The class created by *createclassfromwsdl* actually uses SOAP messages but user does not have to know anything about SOAP communication [2]. In the following, we will focus only on automatically generated stubs and let the SOAP messages only for experienced users.

As already stated, RcWare API provides WSDL description of the methods it has. The API is completely implemented in .NET framework based on Windows Communication Foundation (WCF), which aims at rapid building service-oriented applications that communicate across the web and the enterprise. Typically, WSDL file is automatically generated based on the API program code. Although the resulting description files comply with W3C standards required for SOAP communication, they are not, however, interpreted correctly by the default Matlab WSDL parser i.e. WSDL4J (<http://sourceforge.net/projects/wsdl4j/>). The problem lies in interpretation of namespaces within the WSDL file generated by WCF [3]. In Figure 2, there is an example of WSDL file generated by WCF. According to the standards, *TestResult* element should lie in the same namespace as *TestResponse* (i.e. <http://tempuri.org> in this case), which, in case of WSDL4J interpretation, does not.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <TestResponse xmlns="http://tempuri.org/">
      <TestResult>hi Test</TestResult>
    </TestResponse>
  </s:Body>
</s:Envelope>
```

Figure 2: Example of WSDL file produced by WCF

Therefore, an alternative approach has to be considered. Moreover, as Matlab provides excellent functionality to call Java methods directly from the program code, Java based solution is the easiest way to tackle such an issue.

Apart from WSDL4J library, there exist several implementations of such as Java code generators. From the series, we mention for instance *wsdl2java* from Axis2 package (<http://axis.apache.org/axis2/java/core/index.html>) or the namesake from CXF package (<http://xf.apache.org/>). From the experiments, both of them are capable to handle WCF generated WSDL files and as Axis2 is the more active software project, the choice fell on it. The code was generated according to Axis2 Data Binding Framework (ADB). Resulting workflow is depicted in Figure 3. Note that generated stubs are compiled with arbitrary Java compiler into a form of Jar package, which is dynamically added into Matlab Java path. Matlab users can call web services methods then.

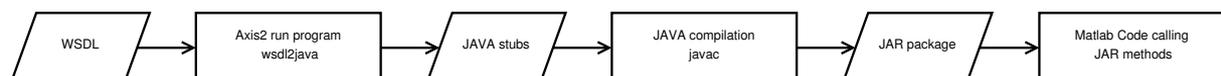


Figure 3: Workflow on how to use Axis2 code in Matlab

4 MatlabRcWare Client

In this section, having in hand compiled Java code generated by Axis2, we will detail the client side Matlab code in terms of introduced classes and we will also compare Matlab client performance to the other clients accessing RcWare DB.

4.1 Object Scheme

Since Matlab r2009a, we have had extended capabilities in Matlab object oriented programming, because of introduction of the new keyword *classdef* and related things. We adopted this concept, which helped us to develop objects more rapidly and create object hierarchy clearer than in the older object oriented programming framework.

The object scheme is designed in a more general way, due to possible extensions of the client for other process databases. Figure 4. shows the implemented object scheme; on the top-most level, there are abstract classes¹ *DbVariable* and *Database*. Moreover, the former one inherits from Matlab built-in class *Handle*, because it provides possibility to reference an instance of the object. The only common method for various databases is method for data obtaining. Every variable has this method too.

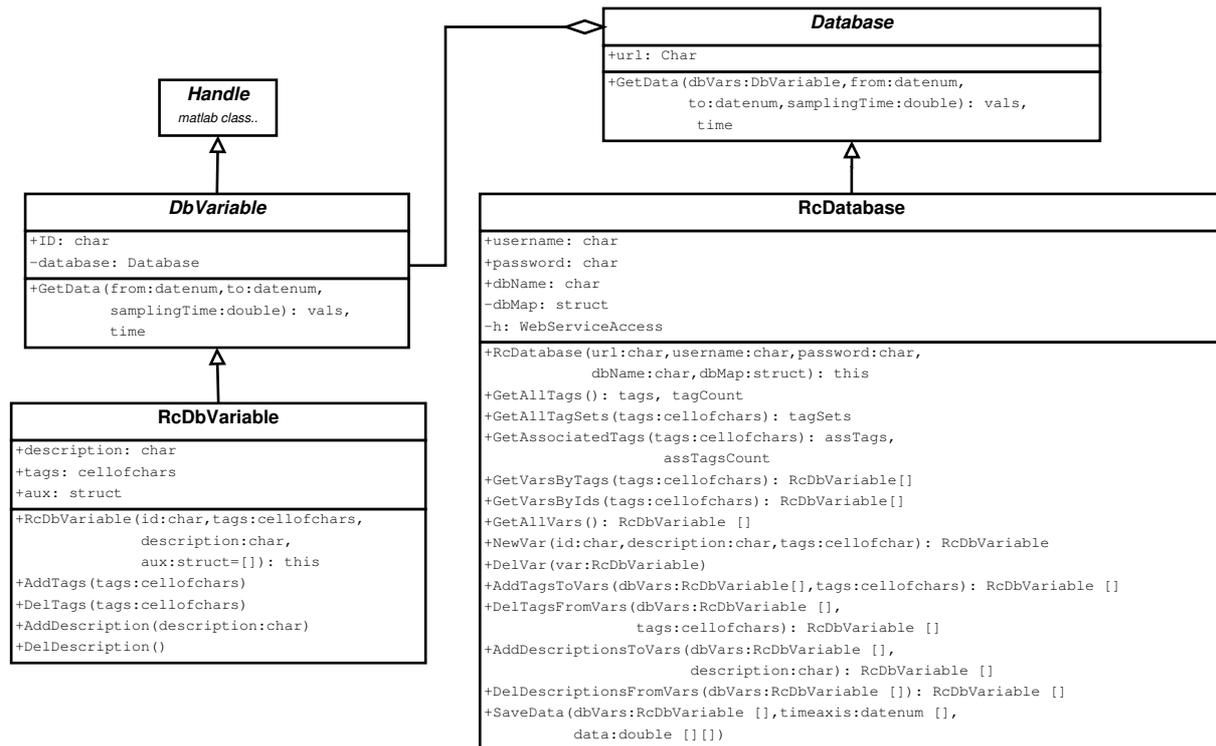


Figure 4: Matlab objects providing an access to RcWare DB

Moreover, *RcDatabase* specific methods extend inherited behaviour by possibility to work with metadata (tags, description) and data saving. Methods for metadata manipulation are suitable especially for a good orientation in a large process database when user can filter out variables only by specifying a list of tags that should or should not a variable have.

4.2 Performance of Data Retrieval

There exist several web services clients capable to access RcWare DB, they differ especially in the programming language they are written in, which, in fact, may affect the resulting performance. As Java is characterized by a strong typing, the stubs generated by Axis2 produced according to WSDL specification lot of additional objects, which would have not been needed (each server request or response is represented by individual Java object etc.).

In addition, each variable in database has its own pairs of timestamp and value. Having downloaded the pairs for couple variables, the Matlab client has to interpolate data to the common and equidistant time axis.

Finally, non-static Java method call ranks among the slowest (according to [4], it is hundred times slower than a Matlab function call), therefore one has to keep this in mind.

¹ Abstract classes have been allowed since the release Matlab r2009a

All the aforementioned reasons can cause some performance losses. We compared the performance of data retrieval against the native .NET client, which only downloads and plots the data values. The same computer was used for the data retrieval and the same variables were downloaded. In the first run, $N = 4\,500$ timestamps were downloaded while in the second $N = 70\,000$. Matlab client achieved performance slow-down about 34% against the native client, which can be considered as a reasonable value having in mind the discussion above. Table 1. summarizes the results.

Note that native .NET client provides possibility to communicate with server via g-zip compressed messages, which can even decrease by half the time needed to download desired data. On the other hand, compression of SOAP message takes some time, which may slow down the server response and, on very high-speed connections, cause comparable performance to the non-compressed one.

Table 1: COMPARISON OF DOWNLOAD TIME FOR NATIVE AND MATLAB CLIENT

Download set size	<i>Matlab Client</i>		<i>.NET Client</i>		Relative slow-down
	<i>text/xml</i>	<i>application/x-gzip</i>	<i>text/xml</i>	<i>application/x-gzip</i>	
N = 4 500	17.01s	N/A	12.71s	7.03s	34%
N = 70 000	107.29s	N/A	80.13s	45.73s	34%

5 Conclusions

In this work, we presented an alternative approach for using web services from within Matlab environment. We developed a Matlab toolbox providing functionality to acquire process data and thus allow users of RcWare DB to monitor and analyse the process in a more effective way.

In Section 3, we presented the reasons why is it necessary to, in some cases, reach for another procedure than the standard Matlab solution and hence the main contribution of this paper lies especially in the proposed workflow on how to use Java code generators in Matlab.

The toolbox is currently used in various industrial and research projects; e.g. predictive control of a building of Czech Technical University in Prague or optimal control of modern office buildings in Germany and Netherlands.

Further extensions of the toolbox are possible in *i)* the area of application/g-zip compression of the SOAP messages, which would help to increase the toolbox performance, *ii)* implementation of access to another type of process database, as it is straightforward with the proposed object scheme.

6 License of the Toolbox

The resulting toolbox is a result of a joined cooperation of Energocentrum PLUS, s.r.o., Czech Technical University (CTU) in Prague and Eidgenossische Technische Hochschule (ETH) Zurich during Optipremier project. Energocentrum PLUS, s.r.o, maintains the program. The program is licensed under the terms of GNU Lesser General Public License v3. Full text of the license is included in the program release. The source codes can be downloaded from the Subversion repository located on: <https://svn.rcware.eu:25443/svn/rad/SWTools/MatlabRcWareInterface>

References

- [1] J. Široký, F. Oldewurtel, J. Cigler, and S. Prívvara, Experimental analysis of model predictive control for an energy efficient building heating system, *Applied Energy*, vol. 88, no. 9, pp. 1-9, 2011.
- [2] Ways of Using Web Services in MATLAB :: Using Web Services with MATLAB (MATLAB®). [Online]. Available: http://www.mathworks.com/help/techdoc/matlab_external/br0203q-1.html. [Accessed: 13-Oct-2011].

- [3] Java - WSDL4J failed to parse WCF.NET service response with BasicHttpBinding - stack overflow. [Online]. Available: <http://stackoverflow.com/q/6042191/641274>. [Accessed: 13-Oct-2011].
- [4] MATLAB OOP: is it slow or am I doing something wrong? - stack overflow. [Online]. Available: <http://stackoverflow.com/q/1693429/641274>. [Accessed: 13-Oct-2011].
-

J. Cigler

jiri.cigler@fel.cvut.cz, <http://support.dce.felk.cvut.cz/pub/ciglejir>, +420 22435 7687

J. Široký

jan.siroky@rcware.eu,

M. Kulvejt

marek.kulvejt@rcware.eu,

M. Chlupáč

martin.chlupac@rcware.eu,

D. Gyalistras

gyalistras@control.ee.ethz.ch, <http://control.ee.ethz.ch/~dgyalist>