VIDIHIP – A Web Service based Integration Platform for Power Plant Control Systems

Mark Strembeck
Institute of Information Systems, New Media Lab
Vienna University of Economics and BA, Austria
mark.strembeck@wu-wien.ac.at

Otto U. Plhal
Electrical Engineering and Automation (ET)
Vienna District Heating Company, Austria
otto.plhal@fernwaermewien.at

Abstract

*European energy supply companies typically run a conglomerate of different, geographically distributed power plants. Unfortunately, the corresponding power plant control systems are based on proprietary technology and an integration of these control systems is a very complex task. We thus conducted a project to build an integration platform for power plant control systems that is based on open standards and technologies. In this paper, we describe the Vienna District Heating Integration Platform (*VIDIHIP*).* VIDIHIP *is based on Web Service technology and allows for the integration of arbitrary (heterogenous) power plant control systems. It provides a consistent interface to access different decentralized control systems and each standard Web browser can be used as a control front-end for* VIDIHIP.

1 Introduction and Motivation

Vienna District Heating (Fernwärme Wien[1]) is a subsidiary company of Vienna Energy (Wien Energie[2]), a public utility company providing power, heat, and gas solutions to private households as well as corporate customers in the City of Vienna. Currently, Vienna District Heating is running three incineration plants, five combined heat and power plants, and five peak boiler plants to serve more than 250.000 private households and more than 5000 corporate customers.

In Europe, plant sites of energy supply companies have a physical life of at least 20 to 25 years, and different plant sites typically have different commissioning dates and thus overlapping lifetimes. As a consequence, Vienna District Heating, just like any other energy supply company, uses a conglomerate of different plant sites in different locations and with different years of construction. Such different plant sites most often rely on a completely different operator environment (even if delivered by the same vendor). This, again, results in a very complex integration task where energy suppliers have to integrate several heterogeneous facilities (e.g. based on waterwheel generators, coal-fired power stations, wind-driven generators, solar cells, or nuclear technology) to get a comprehensive and aggregated overview of the different plant sites. This integration, however, is essential to allow for a centralized telecontrol of loads, to rapidly react on load peaks, and to provide support for preventive maintenance of plant sites, for example.

Many major vendors (as ABB or Siemens for example) offer modern IT-assisted *Instrumentation and Control Automation Systems* (I&C Systems, see also Section 2) which provide vendor-specific solutions to control the power generation process. These vendor-specific solutions, however, most often rely on proprietary company standards, e.g. for data storage or data transfer. Consequently, the coupling/integration of different I&C systems (especially systems purchased from different vendors) causes non-trivial integration efforts. Moreover, to date most vendors are not interested in a unification of I&C systems because it would open their systems to competitors and it would mean that customers (the energy supply companies) could comparatively easy combine power generation systems of different vendors.

Thus, given the present situation, power generation companies “integrate” their plant sites simply via special purpose control rooms providing monitors and operator panels for the different facilities. An integration of power generation facilities via a consolidated IT system

[1] http://www.fernwaermewien.at
[2] http://www.wienenergie.at

that abstracts from proprietary protocols and operating environments is practically never implemented, however.

In this situation, we started a project at Vienna District Heating for the integration of several proprietary I&C systems, including legacy systems, to provide a consistent user interface to access the different I&C systems and thereby to control the different decentralized plant sites. One of our main goals was to provide an integration platform for heterogeneous I&C systems that is based on open standards rather than proprietary technology and that should be accessible via standard software tools (esp. via Web-Browsers).

Section 2 gives some background information on instrumentation and control technology to ease the understanding of the paper. Subsequently, Section 3 describes our Web-Service based integration platform for I&C systems called VIDIHIP (Vienna District Heating Integration Platform). VIDIHIP is implemented on the basis of Apache Axis [1] and Apache Tomcat [2]. The platform serves as a generic integration layer for arbitrary I&C systems. Section 4 concludes the paper.

2 Background: Instrumentation and Control Technology

Instrumentation and Control Automation Systems (I&C Systems) are information systems that control power generation and power supply processes in modern power plants. I&C technology is an umbrella term for the technical equipment applied in power automation, the underlying functions, the transmission of signals between function units, as well as their interaction with human users. Typical examples of standard workflows that are supported through I&C systems are the control process for power generation depending on the actual power consumption, or the load balancing process for power generators to achieve an even work load of different generators.

Historically, I&C technology was developed on the basis of classical measurement and control technology (see, e.g., [3]). In recent years, however, the development of I&C systems was heavily influenced by standards and methods borrowed from the IT area. The primary task of I&C technology is to control the power generation process. The human operator only has to adjust setpoints and parameters and to monitor the system to be able to intervene in case of an error. A typical example of a setpoint adjustment would be to increase the desired power generation demand from 200 Megawatt to 300 Megawatt.

Most tasks and especially emergency procedures in the process of power generation are facilitated or automated via I&C systems. For example, in case of a critical error special predefined emergency-shutdown sequences automatically shut down plant components or even an entire power plant into a safe state of operation.

To connect an I&C system to other technical systems, it provides special purpose coupling interfaces. These interfaces, however, are proprietary low-level interfaces and cannot be applied to integrate different generations of power generation systems, let alone systems manufactured by different vendors.

Figure 1 depicts how the different automation levels are technically interconnected for one of the power plants run by Vienna District Heating. In particular, the scheme from Figure 1 shows the process control system of a Siemens power plant. Other power plants have very similar control structures, especially the triangle structure (also called: control pyramid), which consists of different automation levels, is a characteristic for power plant control structures.

The field control level (see Figure 1) includes a standardized field bus that connects both analog and digital input/output (I/O) controls for sensors and actuators which are connected to the actual power units of a certain plant, e.g. to heat or electricity meters, controls of electrical machines, or simple alarm contacts. The component control level, also called process I/O-level, is based on the plant bus that connects the I/O controls to a number of special purpose server computers. For example, archive servers provide chronological archives of all relevant data (especially sensor values, actuator values, and command sequences), diagnostics servers provide functions for online error analysis, and so called optimization servers allow for the optimization of the power generation process. An example for such an optimization could be load balancing of different power plants. For safety reasons the plant bus is two time redundant, to warrant access to critical system functions at any time.

Terminal bus and multimedia bus are coupled through the so called Human-Machine-Interface server (HMI server, see Figure 1). The HMI server hosts applications that provide human users with a (more or less) convenient interface to control the power generation process of this particular plant. The corresponding control functions then feed the plant control level which serves as the operating and monitoring center of a plant site.

The upper right segment of Figure 1 shows two grey boxes which indicate how VIDIHIP is integrated with I&C systems. In the subsequent section we describe the architecture and implementation of VIDIHIP.

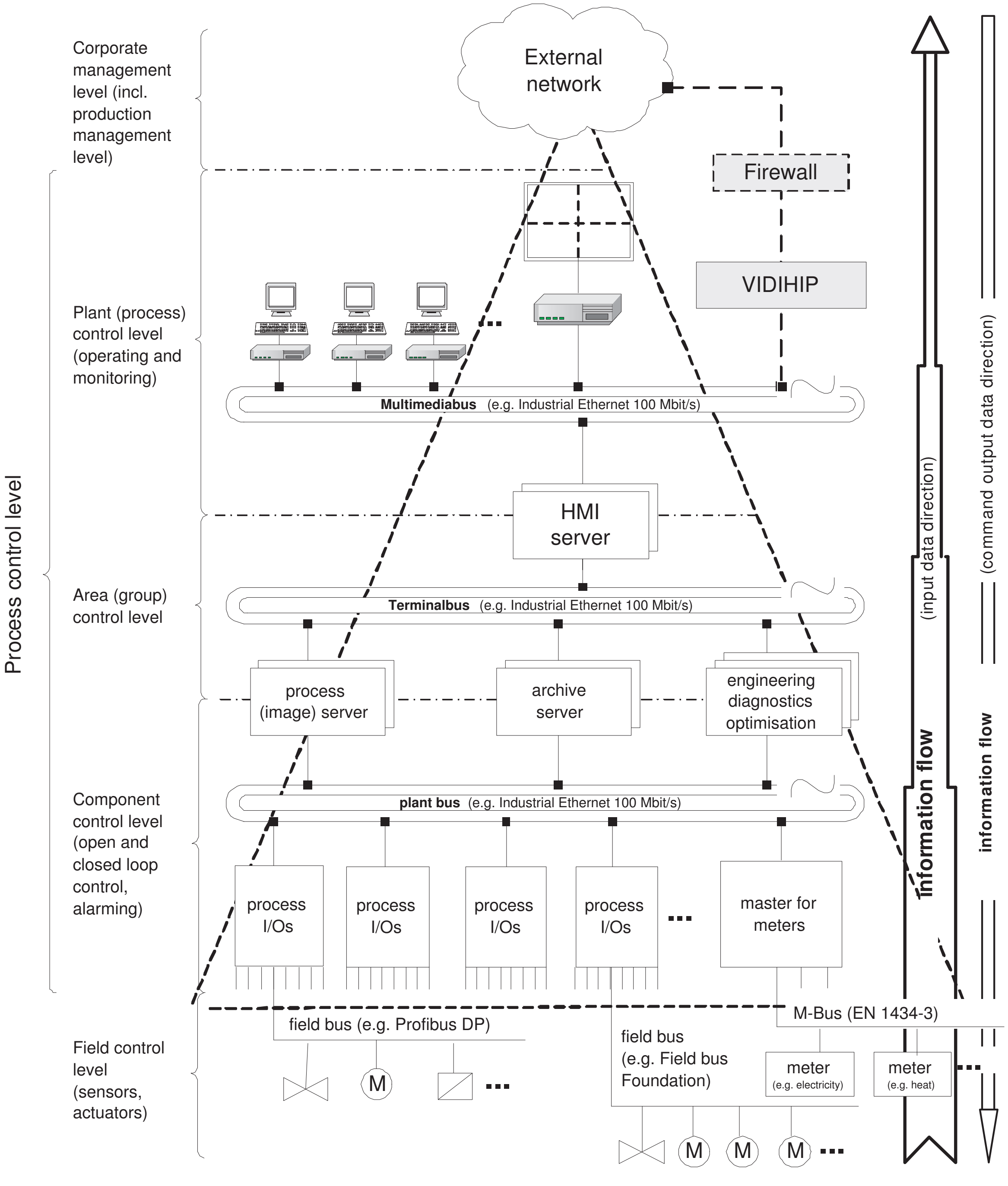

External network
Firewall
VIDIHIP
Corporate management level (incl. production management level)
Plant (process) control level (operating and monitoring)
Process control level
Multimediabus (e.g. Industrial Ethernet 100 Mbit/s)
HMI server
Area (group) control level
Terminalbus (e.g. Industrial Ethernet 100 Mbit/s)
process (image) server
archive server
engineering diagnostics optimisation
Component control level (open and closed loop control, alarming)
plant bus (e.g. Industrial Ethernet 100 Mbit/s)
process I/Os
process I/Os
process I/Os
process I/Os
...
master for meters
M-Bus (EN 1434-3)
field bus (e.g. Profibus DP)
field bus (e.g. Field bus Foundation)
meter (e.g. electricity)
meter (e.g. heat)
Field control level (sensors, actuators)
M
M
M
M
(input data direction)
(command output data direction)
information flow
information flow

Figure 1. Scheme of an IT-supported process control system at Vienna District Heating

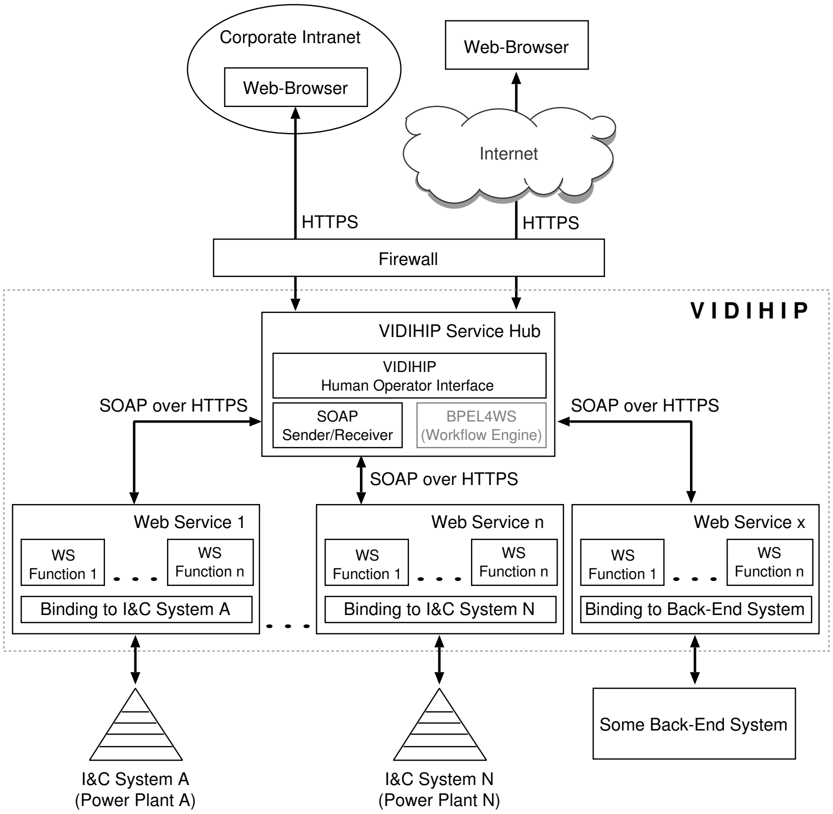


Figure 2. Base-Architecture of VIDIHIP

3 VIDIHIP: The Vienna District Heating Integration Platform

When we started our project to design and implement an integration platform for arbitrary I&C systems, several important factors where already predetermined. In particular, the resulting platform should be based on open, non-proprietary technologies and standards to be independent from individual vendors. Moreover, the platform architecture should be extensible to be ready for evolutionary advancements. Finally, the user clients should be standard software tools, preferably Web browsers – though it may seem of less importance at first glance, this was an important requirement because in the past the acquisition of different user clients (or end user software in general), including costs for user training, caused a significant amount of additional charges at Vienna District Heating. Because of the requirements sketched above and corresponding cost estimates, we decided in favor of a solution based on Web service technology [4].

3.1 VIDIHIP Base-Architecture

Figure 2 shows the base-architecture of VIDIHIP. Users need to log into VIDIHIP before they may access VIDIHIP functions, and all communication to, from, and within the integration platform is send over cryptographically secured connections, in particular using HTTPS [5, 6]. All VIDIHIP-specific code runs on the respective server machines (i.e. on the VIDIHIP Service Hub server and the servers hosting the different VIDIHIP Web services) and we do not require any additional client-side modules nor do we apply client-side scripting or Java applets. Therefore, each standard Web browser can serve as an end-user control front-end for VIDIHIP.

The VIDIHIP Service Hub (see Figure 2) acts as a Facade [7] to an arbitrary number of Web services. These Web services, again, act as wrapper/adapter [7] for (proprietary) backend-systems, especially for I&C systems. Due to the flexible nature of service-based systems, VIDIHIP allows to "plug-in" additional Web services simply by registering them with the Service Hub. The VIDIHIP Human Operator Interface component (see

Figure 2) then registers the different functions offered by the corresponding Web services and provides a unified GUI. In particular, the GUI consists of dynamically generated HTML pages.

Communication between the VIDIHIP Service Hub and its Web services is conducted via standard SOAP requests and responses [8] that are send over HTTPS connections (see also Figure 2). Each VIDIHIP Web service includes a customized binding to the backend system it encapsulates. This binding is responsible to translate Web service function calls to corresponding function calls in the backend system, and to present respective function results back to the Web service (note that the binding to a particular backend-system of course depends on the interface of the respective backend-system and must be specifically tailored to this interface).

In Figure 2 the VIDIHIP Service Hub includes a BPEL4WS [9] workflow engine. This component is depicted in light grey because it is not included in the first version of VIDIHIP but will be added as an extension in the next version. With the BPEL engine, the VIDIHIP Human Operator Interface component will offer additional functions to allow VIDIHIP users the definition of customized BPEL workflows based on the functions available from the VIDIHIP wrapper Web services.

3.2 Applying VIDIHIP

Since it is not possible to do test-runs on critical power supply facilities, of course, we had to find a partner who was willing to cooperate and had sufficient technical resources and an appropriate test environment to test VIDIHIP. Due to the good relations of Vienna District Heating with several manufacturers, we soon reached an agreement with the Siemens company. In particular, Siemens supported us in testing the VIDIHIP prototype and helped to evolve design and implementation details. We were able to thoroughly test VIDIHIP with the Siemens I&C system “Teleperm XP” (TXP) under realistic conditions in the Siemens test facility in Munich, Germany.

As mentioned above, the integration platform was implemented on the basis of Apache Axis [1] and Apache Tomcat [2]. Figure 3 depicts the setup of our test environment. The VIDIHIP Human Operator Interface component includes Java Servlets and Java Server Pages (see, e.g., [10, 11, 12, 13]) which implement internal functions and provide the dynamic Web pages that constitute the VIDIHIP GUI (in Figure 5 we see a VIDIHIP generated Web page showing historical/archived meter data). In Figure 3, the VIDIHIP Service Hub is connected to a wrapper Web service for the Siemens TXP system. The binding to the Siemens TXP system was implemented with a TXP-specific Java-API that allows to access the process data of the TXP I&C system. In particular, we apply the OPC-XML Data Access standard [14], defined by the the OPC Foundation [3] to access the functions of the underlying I&C system.

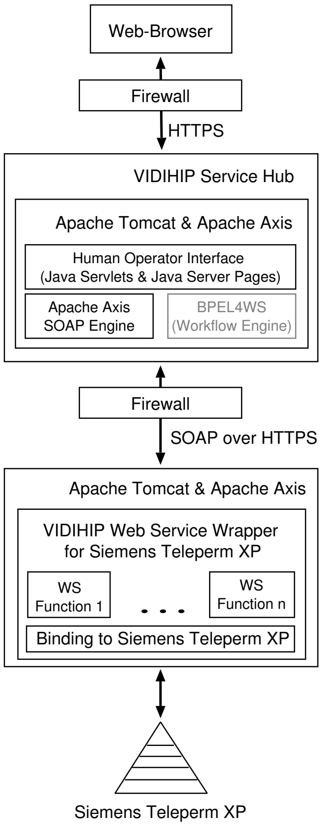


Figure 3. Setup of the test environment

Figure 4 shows a (simplified) message sequence chart for an access to the Siemens TXP I&C system. In this example, the user invokes the `doGet` function (e.g. by submitting an HTML form [15] offered to him via a VIDIHIP GUI Web page) which requests a number of parameters[4]. The invocation is received by the `WSCaller` servlet which parses the parameter list and invokes the `readTXP` function of the `OPC Client` component. Via the `TXPReadLocator` and the `TXPReadSOAPBindingStub` the `OPC Client` calls

[3] OPC is an abbreviation of “OLE (Object Linking and Embedding) for Process Control”, see also http://www.opcfoundation.org

[4] Instead of showing the actual (extensive) parameter lists we use `pm_list` as a placeholder in Figure 4.

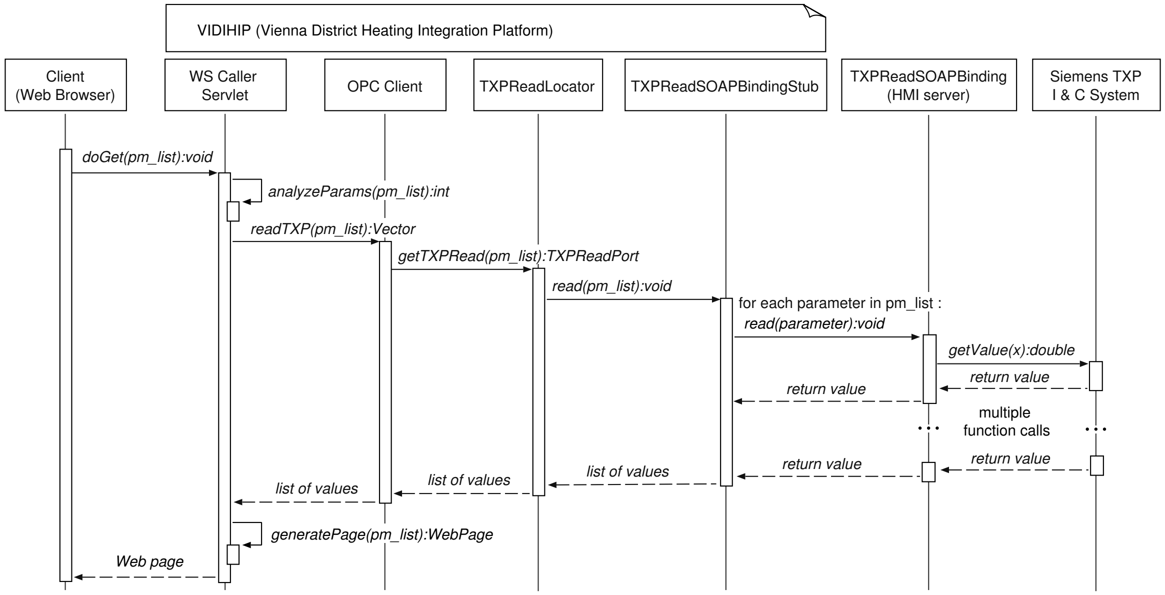


Figure 4. Example access sequence for the Siemens TXP I&C system

the `read` function of the `TXPReadSOAPBindingImpl` component. For each parameter in the list, the `TXPReadSOAPBindingImpl` then dispatches a corresponding call to query the respective parameter values from the `Siemens TXP system`. After all parameter values are received, the complete list is returned to the `WSCaller` servlet and finally back to the user, respectively his Web browser.

The example below shows a (simplified) SOAP request sent to the `TXPReadSOAPBindingImpl` component. This request includes a so called KKS-key (note that KKS is a power plant classification system developed by VGB PowerTech[5], in particular KKS is an abbreviation of the German term “Kraftwerk Kennzeichen System”), here the value of this key is: "1 ETI00 KE105". A KKS-key is a reference to a certain data unit of the corresponding power plant. In particular, the SOAP example below requests the submission of an error text.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
  <Read xmlns=
        "http://opcfoundation.org/webservices/XMLDA/1.0/">
   <Options
    ClientRequestHandle="B780F4618017F18C61110B9E3618DDBE"
    LocaleID="de"
    RequestDeadline="2006-09-20T11:04:27.029Z"
    ReturnDiagnosticInfo="false"
    ReturnErrorText="true"
    ReturnItemName="false"
    ReturnItemPath="false"
    ReturnItemTime="false"/>
   <ItemList ItemPath="Path for ItemList" MaxAge="0">
    <Items ItemName="1  ETI00 KE105"
           ItemPath="Path for item: 0" MaxAge="0"/>
   </ItemList>
  </Read>
 </soapenv:Body>
</soapenv:Envelope>
```

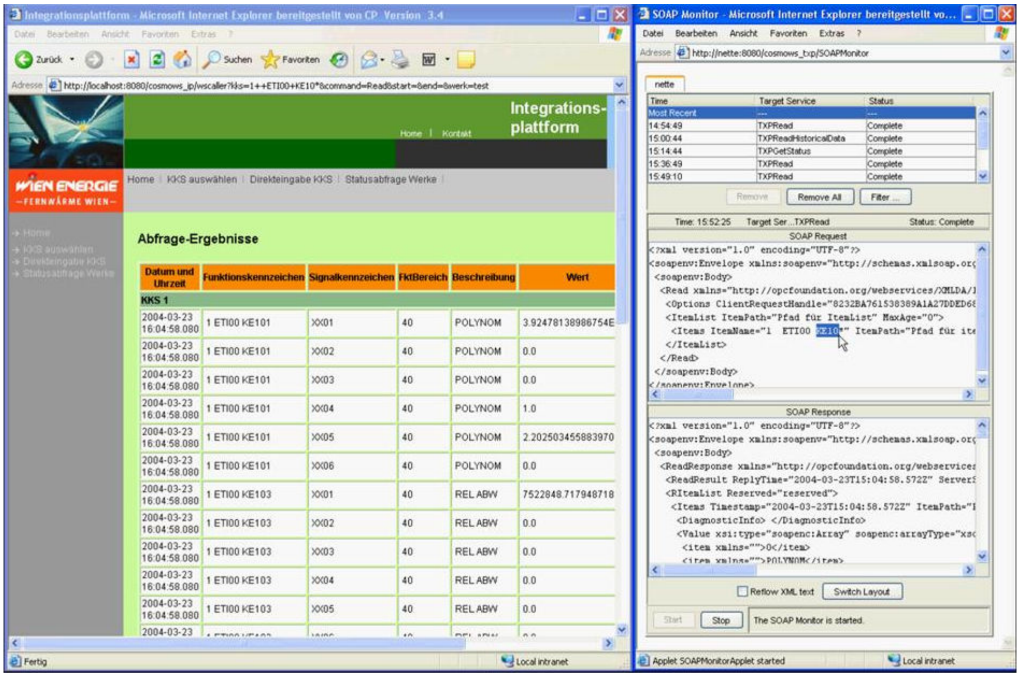
In general, requests to the `TXPReadSOAPBindingImpl` component include at least one `Items` element which contains the KKS-key. The user’s session ID is submitted via the `ClientRequestHandle` element. This session-ID is automatically generated by the Tomcat Server.

3.3 Adoption of VIDIHIP

The main focus of VIDIHIP is to enable the flexible communication with distributed multi-vendor I&C systems. The adoption of the platform can be regarded as an evolutionary step toward the opening of (proprietary) I&C system functions.

At Vienna District Heating we follow a two step procedure to connect VIDIHIP to (additional) I&C systems: In the first step, the VIDIHIP Service Hub is installed on a dedicated server machine which is then connected to the multimedia bus to enable the integration of (heterogeneous) I&C systems on the plant control level (see

[5] http://www.vgb.org/startpage.html



Integrationsplattform - Microsoft Internet Explorer bereitgestellt von CP Version 3.4
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück
Suchen
Favoriten
Adresse
http://localhost:8080/cosmows_jp/wscaller?kks=1++ETI00+KE10*&command=Read&start=&end=&werk=test
Integrations-plattform
Home | Kontakt
WIEN ENERGIE
—FERNWÄRME WIEN—
Home | KKS auswählen | Direkteingabe KKS | Statusabfrage Werke
→ Home
→ KKS auswählen
→ Direkteingabe KKS
→ Statusabfrage Werke
Fertig
Local intranet

Abfrage-Ergebnisse

Datum und Uhrzeit	Funktionskennzeichen	Signalkennzeichen	FktBereich	Beschreibung	Wert
KKS 1					
2004-03-23 16:04:58.080	1 ETI00 KE101	XX01	40	POLYNOM	3.92478138986754E
2004-03-23 16:04:58.080	1 ETI00 KE101	XX02	40	POLYNOM	0.0
2004-03-23 16:04:58.080	1 ETI00 KE101	XX03	40	POLYNOM	0.0
2004-03-23 16:04:58.080	1 ETI00 KE101	XX04	40	POLYNOM	1.0
2004-03-23 16:04:58.080	1 ETI00 KE101	XX05	40	POLYNOM	2.202503455883970
2004-03-23 16:04:58.080	1 ETI00 KE101	XX06	40	POLYNOM	0.0
2004-03-23 16:04:58.080	1 ETI00 KE103	XX01	40	REL ABW	7522848.717948718
2004-03-23 16:04:58.080	1 ETI00 KE103	XX02	40	REL ABW	0.0
2004-03-23 16:04:58.080	1 ETI00 KE103	XX03	40	REL ABW	0.0
2004-03-23 16:04:58.080	1 ETI00 KE103	XX04	40	REL ABW	0.0
2004-03-23 16:04:58.080	1 ETI00 KE103	XX05	40	REL ABW	0.0
2004-03-23					

SOAP Monitor - Microsoft Internet Explorer bereitgestellt vo...
Datei Bearbeiten Ansicht Favoriten Extras ?
Adresse
http://nette:8080/cosmows_txp/SOAPMonitor
nette
Remove
Remove All
Filter ...
Reflow XML text
Switch Layout
Start
Stop
Applet SOAPMonitorApplet started
Local intranet

Time	Target Service	Status
Most Recent	---	---
14:54:49	TXPRead	Complete
15:00:44	TXPReadHistoricalData	Complete
15:14:44	TXPGetStatus	Complete
15:36:49	TXPRead	Complete
15:49:10	TXPRead	Complete

Time: 15:52:25 Target Ser...TXPRead Status: Complete

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org
 <soapenv:Body>
  <Read xmlns="http://opcfoundation.org/webservices/XMLDA/1
   <Options ClientRequestHandle="8232BA761538389A1A27DDED68
   <ItemList ItemPath="Pfad für ItemList" MaxAge="0">
    <Items ItemName="1  ETI00 KE10*" ItemPath="Pfad für ite
   </ItemList>
  </Read>
 </soapenv:Body>
</soapenv:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org
 <soapenv:Body>
  <ReadResponse xmlns="http://opcfoundation.org/webservices
   <ReadResult ReplyTime="2004-03-23T15:04:58.572Z" ServerS
   <RItemList Reserved="reserved">
    <Items Timestamp="2004-03-23T15:04:58.572Z" ItemPath="I
     <DiagnosticInfo> </DiagnosticInfo>
     <Value xsi:type="soapenc:Array" soapenc:arrayType="xsd
      <item xmlns="">0</item>
      <item xmlns="">POLYNOM</item>
```

The SOAP Monitor is started.

Figure 5. Screenshot of a dynamic Web page generated by the VIDIHIP service hub

also Figure 1). In the second step, we then implement the VIDIHIP wrapper Web services for this particular I&C system (see also Figure 2).

In an optional third step, additional functions that are offered via plug-in services can be activated (see also Section 3.4), e.g. central maintenance functions for geographically distributed power plants, or integration with the (superordinated) data processing systems. Furthermore, due to its open and extensible architecture VIDIHIP can be extended on demand if necessary.

3.4 Plug-in Services for VIDIHIP

3.4.1 Central Data Management

A very important topic in power generation processes is central data management (and archiving) of all relevant process data as well as all calculated interim values resulting from the process data. In general, the VIDIHIP central data management has two main functions: On the one hand (heterogeneous) I&C systems can be coupled via VIDIHIP in order to integrate their respective archives (databases). The result is a central and uniformly accessible chronological data archive of several I&C systems. On the other hand, VIDIHIP provides access to power generation data for different user groups at Vienna District Heating. In particular, the functions offered via VIDIHIP abstract from and (to a certain degree) standardize access to specific data structures used by different I&C systems.

3.4.2 Maintenance and Operational Planning

Service and maintenance of complex I&C system equipment requires a comprehensive know-how of the different control system components and their long-term operating performance. Most of this knowledge originates from practical experiences of the respective system operators and long-term experiences of the manufacturers. Knowledge-based maintenance systems can structure and supplement this knowledge. As mentioned before, energy supply companies most often run widely distributed power plants which are involved in several power generation processes. The corresponding I&C systems usually need to be maintained by several mobile maintenance teams. We aim to ease maintenance tasks, via a central administration and management desk for all maintenance tasks of distributed power generation processes. The maintenance service offers additional functions as remote access for mobile users, request and acknowledgment functions for maintenance tasks, or a scheduling module to manage service personnel and spare parts.

4 Conclusion

We developed a Web-Service based integration platform that provides an unified, open, and extensible access layer for modern instrumentation and control (I&C) technology for power plants. The Vienna District Heating Integration Platform (VIDIHIP) presented in this paper was successfully tested in cooperation with the Siemens company. In particular, the VIDIHIP base architecture and a plug-in Web Service for meter data reading were tested together with the Siemens Teleperm XP I&C system. Based on this version of VIDIHIP we are enhancing VIDIHIP with additional components and plug-in Web services (cf. Section 3). For example, we will add a BPEL engine to enable the definition of Web-based workflows for I&C systems.

Siemens as well as other I&C vendors showed distinct interest in our ideas. Nevertheless, the transfer to an open system technology, the integration of the process technology in the business organization, and the opening of the classical control technology to standard Intranet/Internet technologies requires high initial efforts. This, again, requires a serious commitment of all affected stakeholders (especially vendors and power generation companies) to move towards open solutions (beyond the main automation-core of the I&C systems). At Vienna District Heating, the VIDIHIP project acted as a door opener for Web-Service technologies and in our experience, the possibility for a successive implementation of such integration projects via Web-Services is a central advantage.

Acknowledgments

The authors would like to thank Karlheinz Springer, Rainer Speh, Volkmar Pflug, Martin Glaser, Uwe Gerk, and Karl Jetter at Siemens Power Generation as well as Volkmar Sterzing, Michael Berger, Christoph Tietz, Robert Jasny, Stefan Hagen Weber, and Thomas Lemmerer, at Siemens Corporate Technology for helpful discussions and for allowing us to test VIDIHIP in the Siemens test facility. Last but not least, we like to thank Aarno Rapottnig and Veronika Wolf at Vienna District Heating for their support of the project.

References

[1] The Apache Software Foundation. Apache Web Services - Axis. http://ws.apache.org/axis/, January 2007.

[2] The Apache Software Foundation. Apache Tomcat. http://tomcat.apache.org, January 2007.

[3] M. Polke, editor. *Process Control Engineering*. VCH Publishers, New York, 1994.

[4] W3C Web Services Activity. http://www.w3.org/2002/ws/, January 2007.

[5] T. Dierks and C. Allen. The TLS Protocol - Version 1.0. IETF, RFC 2246, Standards Track, January 1999.

[6] E. Rescorla. HTTP Over TLS. IETF, RFC 2818, Informational, May 2000.

[7] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[8] M. Gudgin, M. Hadley, N. Mendelsohn, J.J. Moreau, and H. Frystyk Nielsen. SOAP Version 1.2 Part 1: Messaging Framework. W3C Recommendation, June 2003.

[9] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. Business Process Execution Language for Web Services, Version 1.1. Specification, BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems, May 2003.

[10] L. Moczar. *Tomcat 5*. SAMS, 2004.

[11] M. Hall and L. Brown. *Core Servlets and Javaserver Pages: Core Technologies (2nd Edition)*. Prentice Hall PTR, 2003.

[12] J2EE JavaServer Pages Technology. http://java.sun.com/products/jsp/, January 2007.

[13] J2EE Java Servlet Technology. http://java.sun.com/products/servlet/, January 2007.

[14] F. Iwanitz and J. Lange. *OPC - Fundamentals, Implementation and Application, Third Edition*. Hüthig Fachverlag, 2006.

[15] D. Raggett, A. Le Hors, and I. Jacobs. HTML 4.01 Specification. W3C Recommendation, December 1999.