LTM-SOLA – A Service-oriented Application to Integrate High-Tech Laboratories and Virtual Knowledge Spaces

Henning Sommerkamp*, Jonas Schulte*, Reinhard Keil*, Johann Rybka* and Ferdinand Ferber‡
*Heinz-Nixdorf-Institut, University of Paderborn, 33102 Paderborn, Germany
‡Department of Engineering Mechanics, University of Paderborn, 33098 Paderborn, Germany
Email: *{hsommer|schulte|reinhard.keil|jrybka}@uni-paderborn.de, ‡Ferdinand.Ferber@ltm.upb.de

Abstract—**High-tech laboratories are an important kind of knowledge source, nevertheless most of today's laboratory architectures consist of tight coupled devices with proprietary interfaces. Although there are several approaches to implement cross-functional and heterogeneous infrastructures, these solutions do not consider high-tech laboratories. Therefore, we developed a service-oriented architecture for laboratories called LTM-SOLA, an application to control the devices by a modern web interface based on JavaEE and Seam technologies. Furthermore, we show how to use field bus couplers instead of programmable logic controllers (PLC) to enable single laboratory devices to be used in different test environments.**

I. Introduction

The International Association of Online Engineering (IAOE) is an internationally acting organization with the objective of encouraging the further development, distribution and application of Online Engineering technologies. [1] Current research in this area, like considered by the International Conference on Remote Engineering and Virtual Instrumentation (REV), concentrate on virtual laboratories and the establishment of scientific workplaces for everyone. New comprehensive applications were developed and new infrastructures were established. [2], [3] These approaches work with topics like developing Grid infrastructures [4] or managing jobs more effectively [5] in virtual laboratories.

Current developments mainly concentrate on the creation of virtual engineering workplaces to make scientific engineering more touchable and accessible for everyone. We want to support researchers during their daily work processes in real laboratories and alleviate distributing their research results. In our understanding laboratories and particularly specialized high-tech laboratories are a significant source of knowledge development. In order to make new knowledge immediately available for everyone, we are working on a better integration of real laboratories into computer-supported cooperative work (CSCW) environments. Improvements regarding communication aspects and data exchange, that follow from this integration, help to neutralize the media discrepancy between knowledge development (laboratories) on the one hand and knowledge utilization (CSCW environments) on the other hand.

Most laboratory devices have their own proprietary interfaces and do not offer the possibility of return values[1], as well as the software to control these devices. Hence, our preliminary research and implementation concentrated on setting up a flexible laboratory architecture [6]. When setting up such a laboratory architecture two important issues have to be considered. First of all the architecture should offer common interfaces to specify instruction sets for the laboratory devices. Instead of proprietary interfaces and bitwise combinations to specify a command, it should offer self-explanatory functions allowing transparent access on the internal instruction set.

Based on this modern laboratory architecture, we developed an application for controlling laboratory components. In this paper we present LTM-SOLA[2], an application to integrate a laboratory into university wide e-learning and teaching infrastructures. As an example field of application we adapted our implementations to the requirements of the thermal shock laboratory at the University of Paderborn, Germany. In this laboratory steel samples are exposed to thermal stress by using an electrical induction heating with 80kW output power and water cooling. After a predefined number of heating and cooling cycles, exposing the sample to defined temperatures, the material is analyzed in terms of deformation, rigidity, and micro-cracks.

Not only the integration of the laboratory into a university wide infrastructure is an important task. Besides this the flexible arrangement of laboratory devices regarding the test specifications and requirements is essential. Therefore the monolithic and inflexible system composition has to be dehisced and divided into single independent parts. These parts have to be controlled and data has to be transmitted across a heterogeneous infrastructure. To solve these problems in the process, and to provide a smooth work flow, our new system can successfully be used. The laboratory architecture presented along with LTM-SOLA is justified by the following reasons:

1) Other laboratory software is not able to support close collaboration and publish test results in arbitrary repositories.
2) Current CSCW applications and e-learning environments do not include real laboratories.
3) Monolithic applications do not fit into dynamically changing experiment setups.

[1]Return values are any records of a laboratory device, developed during the test procedure.

[2]LTM-SOLA is an acronym for "Lehrstuhl für Technische Mechanik" (Chair of Technical Mechanics) and "Service-oriented Laboratory Application".

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8378
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8378

The paper is structured as follows. In section II we present our approach of a service-oriented architecture (SOA) allowing the modularization of laboratory components. Section III describes LTM-SOLA, a service-oriented application for controlling the thermal shock test laboratory and allowing the storage of test results in arbitrary digital repositories. The following section IV is about a Distributed Architecture based on LTM-SOLA we have developed. After this, our current research on this topic is concluded in section V. Furthermore we give an outlook on our ongoing work.

II. SOA for Modularization of Laboratories

Current software engineering has identified the desire for modularized systems to improve reusability and to achieve maintenance advantages in contrast to monolithic applications [7]. Existing software for operating laboratory equipment often does not allow a modularized setup, because of narrow and highly specialized application ranges.

A. Demand for a new approach

The laboratory that we consider for our developments is characterized by various software products, that are involved during the test process. Figure 1 depicts the current situation of the thermal shock laboratory and is representatively for various laboratories. This architecture is a typical pillow-architecture, referring to [8]. Each software product employed is not able to interact directly with other products. The first reason for this deficit are missing interfaces for contemporary programming styles. Most devices offer vendor specific protocols and instruction sets only. Secondly, most laboratory devices are not designed for aggregation to form a larger system. Hence we came to the conclusion, that the demand for modularization cannot be fulfilled by the current software architecture.

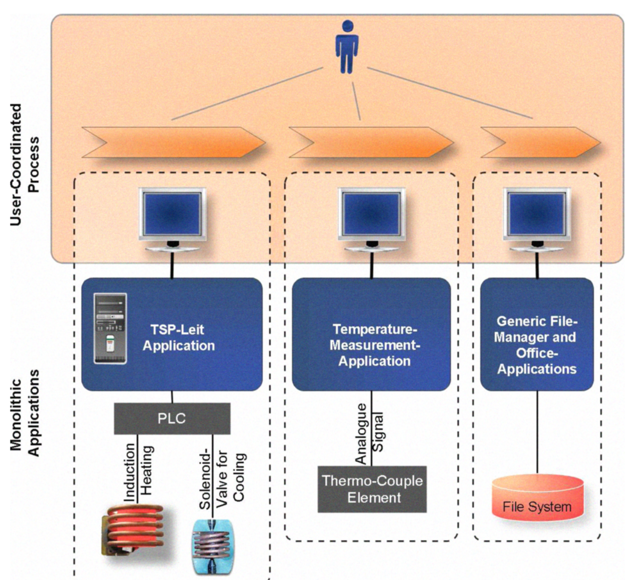

Fig. 1. Laboratory devices are controlled by specialized applications. Interaction between these applications is not intended. The user has to coordinate the laboratory process on his own.

As an example for such a situation, we describe the current software-architecture that will be improved. In the laboratory test-setup presented we have three applications. First, TSP-Leit is used for controlling a heating and cooling process. During this process it offers the interface for a Programmable Logic Controller (PLC), that controls the induction heating as well as the unit for cooling the metallic sample. A major disadvantage in this realization is the hard-coded PLC. Particularly with regard to the demand of using laboratory devices in various test scenarios, this realization, optimized for one specific task, is cumbersome. The interaction between the PLC and TSP-Leit is done by a proprietary byte-wise TCP/IP protocol. Second, a thermometer is used to record the sample's temperatures during the test procedure. Unfortunately this USB device is operated by a proprietary application, that runs on Microsoft Windows operating system only. The data recorded can only be saved on the computer's local file system. Any further post-processing or publication of these measurement-results is not supported by this application. For this purpose a third application is used. The engineer uses a generic file-manager to organize the test-data. For creating figures from the measured temperatures, a spreadsheet application may be used. This small insight into the current situation reveals high potential for improvements. Our aim is to integrate the laboratory test process into a modern service-oriented application, that can automatically perform most steps, e.g. publishing the test results, instead of requiring manual action.

In [6] a concept has been developed to split a monolithic laboratory software into a set of loose-coupled services. Each service can be implemented as a Web-Service that provides the functionality of one laboratory device (see figure 2). The Web-Service infrastructure offers self-describing mechanisms for each service and encourages the loose coupling of laboratory devices. Furthermore base services, that represent the instruction set of a device can be aggregated to high-level services, providing complex test cycles. In the following section II-B we give an example for an appropriate technology to implement this concept.

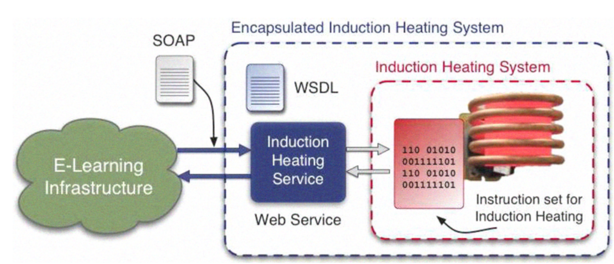

Fig. 2. Laboratory device ecapsulated by a Web-Service. This example shows the induction heating control.

B. Benefits of Java-EE Applications for Laboratories

Java-EE is an infrastructure for implementing modularized systems. Each semantic unit can be encapsulated in a so-called Java Bean. This ensures that each module has an open business interface, as defined by the Java-EE standard. Therefore, it

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8378
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8378

is possible to reconfigure the application by changing the encapsulated Web-Services, but not touching the application code. We can re-use certain services, i.e. for controlling the devices or to reduce the overall complexity of performing a single experiment. In a laboratory context this is critical because various test set-ups use the same devices time and again.

Another advantage of using Java-EE is the presence of ready-to-use frameworks. One of them is our WasabiBeans Framework discussed in section II-C. Furthermore, third-party frameworks like JBoss Seam[3] exist. They allow us to create web-based user-interfaces for visualizing and editing data managed by Java Beans.

C. WasabiBeans Framework

Documents generated during the experiment in the laboratory should always be stored for a longer period and shared with other users to gain the greatest benefit from the research results. For this purpose the room/document concept has been proven to be successful [9]. Recently, this concept was implemented by the WasabiBeans Framework [10], [11]. The WasabiBeans Framework is based on Java Beans running on the JBoss Application Server. Using this technology enables the framework to connect to various digital repositories used to store user-data for a longer period. Applications using WasabiBeans have access to Computer Supported Cooperative Work (CSCW) services, digital libraries, remote file-systems and databases transparently.

D. Adaptation of Various Repositories

It is possible to configure various repositories for storing data because we have integrated the WasabiBeans framework into LTM-SOLA. Currently, the room/document concept is used to make experiment descriptions and results permanent. But this is just one possible repository. For other applications it may be necessary to save some results in a separate file-system of a cooperating university. With WasabiBeans we can also connect LTM-SOLA to a digital library. In this way the system can be used to make experiments available for students as an electronic course reserve. The functionality has been successfully implemented with the document and publication server MILESS. MILESS has been developed at the University Duisburg-Essen, Germany [12] and is currently run at further education-related institutions.

III. LTM-SOLA – A Seam-based Web-Application for Thermal-Stress-Tests

In this section we present a sample experiment execution process controlled by our Seam-based application LTM-SOLA. Our approach is centred on the integrated process instead of being centred on the components and applications involved.

As an example we figured out a common work-flow as shown in figure 3. In step one a cooperation partner wants to perform a thermal stress test. Therefore, he or she creates a thermal stress profile with the LTM-SOLA characteristic line editor and defines the test parameters. Optionally, it is possible to store the profile in an external repository.

[3]http://seamframework.org

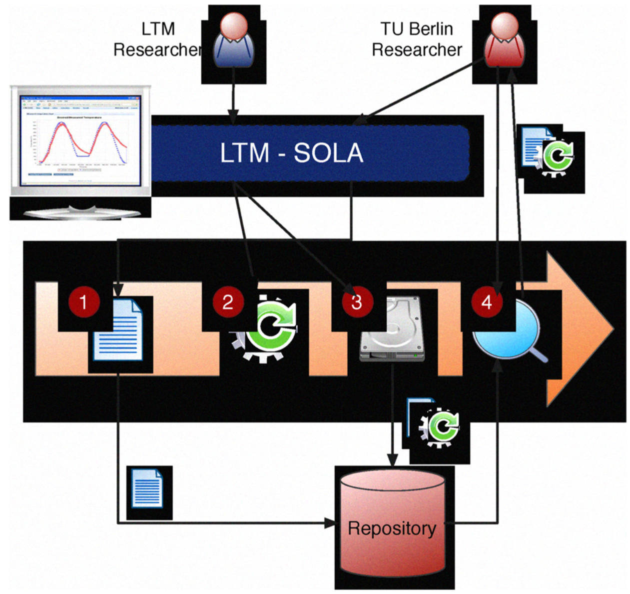

Fig. 3. Laboratory Business Process

If the thermal shock laboratory is available, a researcher from the Department of Engineering Mechanics can accept the test. In this second step a laboratory engineer is necessary because someone has to set-up the experiment, e.g. installing the sample and switching on the laboratory devices. Furthermore, the laboratory engineer ensures that nobody remains in the danger zone of the laboratory devices. Currently this is sufficient but also a more detailed restriction may be integrated into the system to support a more extensive safety precaution. Afterwards the laboratory engineer can execute the experiment by using the LTM-SOLA scheduler and the LTM-SOLA monitor for supervisory purpose. The LTM-SOLA monitor may be used by the laboratory staff locally, such as remotely, by cooperating researchers.

In step three, the experiment is finished and results are available. They are stored in the virtual knowledge space. This allows future browsing for documents by further users. Step four finally represents the handling of information gained by the laboratory. To sum up, the process consists of the actions: define, execute, store and find.

A. LTM-SOLA's Architecture

LTM-SOLA is based on a three-tier architecture. The principle of illustrating each tier within a SOA by horizontal beams was introduced in [8]. Conforming to this beam-approach, our architecture is shown in figure 4.

First, the base layer is composed of three Web-Services and the WasabiBeans Framework. There is one Web-Service for the thermocouple data acquisition, one for the PLC used to cool the sample and one more for controlling the induction heating. Furthermore, the base layer consists of different ser-

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8378
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8378

vices offered by WasabiBeans to access the virtual knowledge spaces for storing documents like temperature characteristic lines and result documents.

Second, the process layer, which implements the most important process in LTM-SOLA, the laboratory business process, shown in figure 3 and additionally, a process to execute a certain experiment flow. These are implemented as Java Beans. In a SOA it is not critical whether the base layer services and the service for executing processes are run on the same server. Loose coupling of services allows us to run services for the control of proprietary hardware interfaces on certain computers (in our situation the Heating Service and the Temperature Service), while others can be run on the same machine with the process service. In figure 4 these two kind of base layer services are depicted. Services with a remote interface may be executed on a specific machine, while services with a local interface have to be run within the Application Server running the process. The PLCService is one example for the latter case. Since it communicates with the PLC via a proprietary TCP/IP protocol on the Ethernet, there is no additional hardware interface necessary for this service. Therefore, it is sufficient to singly implement a local interface, which yields a better performance than a remote interface.

Third, the graphical user-interface, implemented as a Seam-based web application to visualize and manipulate the processes in the process layer eixsts. Using JBoss' Seam Framework, the application's individual pages are defined in an xml-like language allowing the user-interface elements to transparently access the data in the underlying process layer.

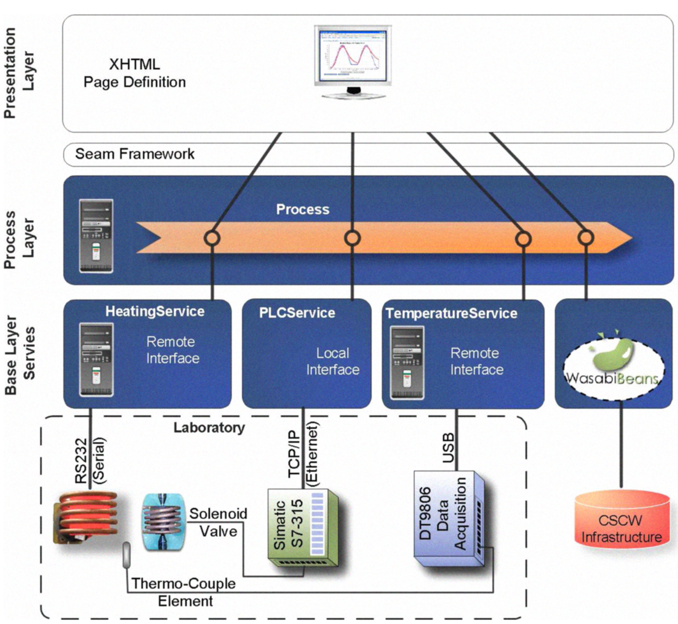

Fig. 4. LTM-SOLA's architecture: 1) **Base Layer Services** for controlling laboratory devices and binding storage. 2) A **Process Layer** to coordinate service invocations. 3) The **Presentation Layer** in terms of a web-application.

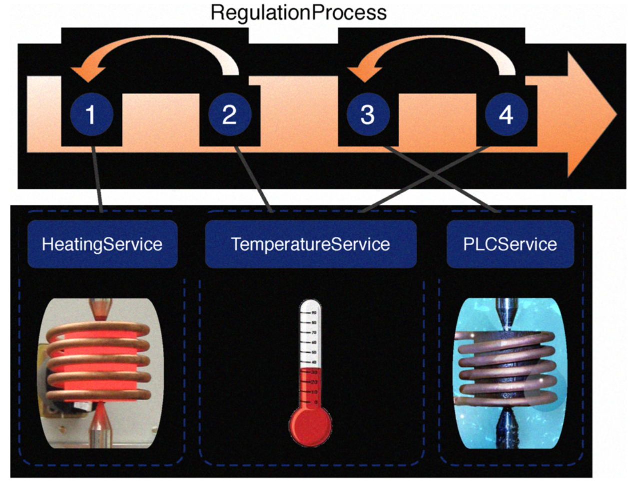

Fig. 5. *Experiment Execution Process* for regulation of a steel sample temperature. This process is executed at step 2 of the *Laboratory Business Process* shown in figure 3

B. LTM-SOLA's Business Logic

LTM-SOLA's business logic acts as "glue" between the base layer services like laboratory devices and digital library, and the presentation layer that provides the web front-end. The business logic implementation is located in the process layer. It integrates multiple services into a single service of higher value, which provides the methods needed by the web front-end. The experiment actually starts, after a user press the start button. The Seam Framework routes this event to our Java Bean's method within the process layer. This method initiates various calls to base layer services and collects data to assemble the experiment result. Currently, the service running the thermal shock process is implemented as Stateful Session Bean (SFSB). The experiment execution process used for demonstration purposes in LTM-SOLA is shown in figure 5.

The process performs a PID-regulation of the temperature by evaluating the actual temperature read by the thermocouple service, comparing it to the desired temperature given by the predefined time-temperature characteristic line, and setting the heating or cooling to the necessary power to achieve the correct temperature. It uses the scheduler Quartz, provided by the Application Server, to perform recurring asynchronous invocations of the regulation method. The collected data is stored in an Entity Bean. Entity Beans are an instrument to make data persistent within an Application Server. This way all our Java Beans involved in coordinating the process have access to the collected data. The Entity Bean serves as a preliminary storage mechanism for our test results. Upon process completion, this data may be stored to its designated location by the WasabiBeans framework and the Application Server can de-allocate the memory used by the Entity Bean.

Since our system conforms to SOA, it is easy to replace or update services in each of the layers mentioned above. Currently a fixed process with dynamic parameters is executed. But we are considering to execute a dynamic process instead.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8378
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8378

One possibility to describe and execute dynamic processes is the Business Processing Execution Language (BPEL). BPEL is an XML-based programming language to describe processes. Web-Service methods may be referenced in this process. After deploying a BPEL process on a BPEL engine, the process itself is exposed as a Web-Service and may be executed by Web-Service consumers. It is possible to employ a BPEL engine in addition to our process services described above. This BPEL engine executes user-defined sequences of base layer methods.

C. LTM-SOLA's Web Front-End

To support inter-organizational work and research, LTM-SOLA is accessible by a web front-end. For our implementation we rely on the Seam framework for reasons of best interaction with the WasabiBeans framework as well as strict modularization of the web application. The web front-end supports to load a predefined characteristic line from the user's local file-system, his private room in the WasabiBeans knowledge space, and also the creation of new characteristic line from the scratch by using the embedded editor. In addition a characteristic line can be modified and visualized by the editor. After finishing the desired characteristic line, that specify the test parameters, the engineer uses the scheduler page to set up the process parameters. These are the number of thermal shock cycles to run, the pause between the cycles, the regulation parameters, and special meta-data to identify the tested metallic sample. By starting the experiment, the process is scheduled and a monitor page gives information about the current process state and its progress.

After completion, the results are available in different formats. Figure 6 shows an overlay of two characteristic lines for comparing the target and actual temperatures. So far the result is stored as an image in PNG-format, but the storage as a text-based document (CSV-file) and a PDF export are under development. Especially the possibility to store test results in various repositories is essential to support flexible cooperative work. LTM-SOLA allows to publish the test results in the digital document and publication server MILESS with just one click.

D. Improvements in Using the Laboratory

The laboratory user benefits from a number of improvements provided by LTM-SOLA the previous architecture lacks. Test-description data and measurement results are accessible by one uniform web-interface. The web-application is presented identically no matter if is accessed remotely or locally in the laboratory. For example, this allows students to practice how to set up the experiment from a remote workplace.

Accessing our virtual knowledge space allows a convenient handling of documents and the transferring of documents between different users. Using the own home-room the user has all relevant documents at hand.

Test results can be published using a digital document and publication server. In this way a wider audience can be reached with the information.

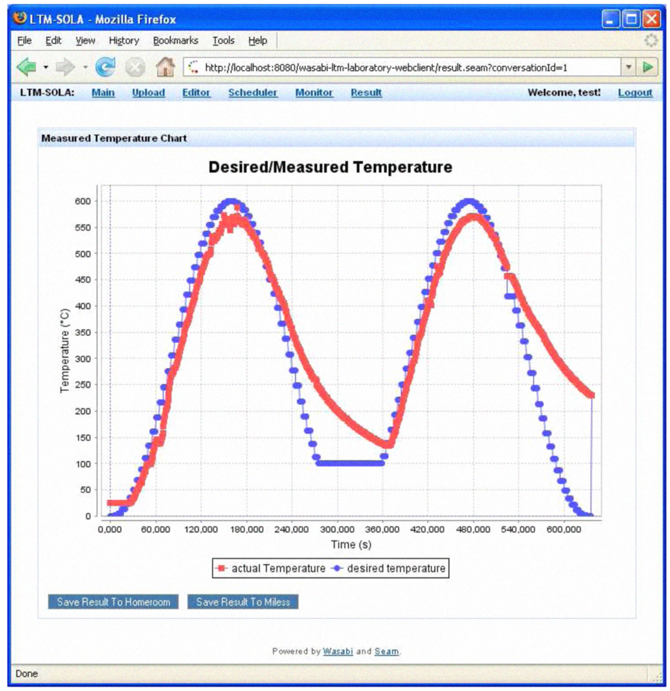

Fig. 6. Screenshot: LTM-SOLA's web front-end – Result graph with desired and measured temperature.

And finally the laboratory process, initially performed by the user, is now supported by the system. This relieves the user, so that he can concentrate more on the experiment itself. Supporting this process was made possible by exchanging data seamlessly between the individual application-modules.

IV. Distributed Architecture

LTM-SOLA is the fundament to transfer test results among various systems, that are involved in the cooperative work and research process. Its predecessor TSP-Leit was especially designed to control the devices of the thermal shock test plant. The further usage of results and the cooperation with other research institutes was disregarded in this first approach to develop a software for the laboratory. As presented above LTM-SOLA abrogates the monolithic and inflexible solution. By using a SOA, as presented in section II and III, we facilitate easy exchange of result documents between researches working at different locations as well as an easy further processing of test results, e.g. the storage in digital libraries. The basic service implementation and the WasabiBeans integration has considerably increased the flexibility of controlling laboratory devices. LTM-SOLA has proven, that the combination of WasabiBeans and SOA are suitable in a laboratory context. Its modular design reflects the requirements of software used in a laboratory.

However, the disaggregation of all laboratory devices as one complex connected network is another task to realize more flexibility in respect of the test possibilities. In this section we focus on a concept to achieve distribution not only in regard to the control software but also in the overall laboratory

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8378
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8378

architecture.

A. Distribution for Complex Test-Setups

Complex test scenarios often require diversified equipment involved in the test assembly. Thereby the presented laboratory setup in section III-A, where devices are connected by a centralized PCL appears to be insufficient. Different laboratory devices such as the thermometer for temperature data-acquisition or the eddy-current sensor to detect cracks on a sample's surface may be used in various test plants. In our new approach each laboratory device is intimately connected with its own controller, e.g. personal-computer or an field bus controller.

Implementing an infrastructure where each laboratory device comes with its own controller reveals a number of pleasant advantages. The software architecture can be mapped identically to the underlying system architecture. While LTM-SOLA realized modularization with regard to the software level, this concept aims to accomplish a modularization concerning the hardware level to allow a flexible rearrangement of the laboratory composition. The distributed architecture offers Web-Services for controlling the laboratory devices, that are running on a controller, that directly belongs to a certain device.

In addition this concept also improves the wiring in the laboratory. The controllers that are closely connected with the laboratory devices support a standard twisted-pair ethernet input. That means the whole connection and cabling in the laboratory can be reduced to simple ethernet cables. The major advantage is the possibility to change the local position of laboratory devices, since just one simple cable is necessary for the whole data transmission.

B. Fieldbus Controller Architecture

Our target is to simplify the handling of laboratory devices for researchers. Researchers in the laboratory should concentrate on the thermal shock experiment's development and execution. They should concentrate on setting the process parameters and the following test execution, instead connecting the laboratory devices, that are involved in the test cycle of an experiment. Equipping every laboratory device with a certain fieldbus controller, the target of modularization and distribution will be achieved. fieldbus controllers are usually used for industrial control tasks. They can be used to interface various proprietary buses to an Ethernet. We have evaluated the Linux fieldbus controller (LFBC) by Wago GmbH, Germany. Since it is running with the Linux Operating System, we can adapt or develop driver software for almost any kind of laboratory hardware. The device functionality can directly be shown by a Web-Service running on the LFBC.

Based on this concept, we can extend the LTM-SOLA Architecture to form a new fieldbus controller architecture shown in figure 7. While LTM-SOLA needs its base-services implemented as Java Beans, the fieldbus controller architecture does not need them. Base services are implemented on the LFBCs. This is one more step towards a wider distribution.

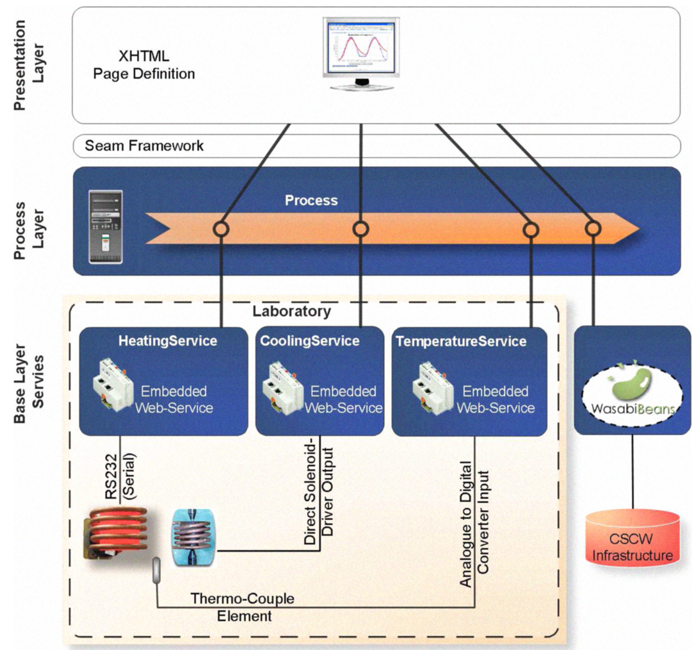

Fig. 7. Distributed Architecture: Base Layer services running on specific controllers located within the laboratory. Process Layer and User Interface can remain identical to those in LTM-SOLA's Architecture.

The base layer services are relocated from the Application Server running our Java Beans, into the laboratory itself. Our fieldbus controller architecture achieves a better reflection of the actual overall system concerning the spatial location of each Web-Service.

The process layer introduced earlier will still remain the same. Since we use standardized Web-Services, the device functionality can be referenced and transparently used by the process layer beans.

V. Conclusion and Outlook

Our approach of implementing a service-oriented architecture for laboratories has proven during LTM-SOLA's development. In this paper we emphasized two main reasons. First of all the increase in flexibility concerning the laboratory setup. Second, the closer interaction with existing infrastructures as well as the further processing of new knowledge exposed during the tests in those environments.

Our research and especially LTM-SOLA demonstrates the power of the SOA paradigm for laboratories. It is possible to integrate a variety of individual subsystems into one complex application (system convergence). Thus we not only eliminate many media breaches, but also avoid manual intervention to provide research results for further processing. Both aspects are relevant to quicken the wider distribution of research results and to allow immediate interaction among scientists. Additionally the SOA approach enables a faster re-configuration for the usage in other types of experiments. The architecture's dynamic character reflects the constellation in modern state-of-the-art laboratories.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8378
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8378

The future of modern research environments involves leading researches with collaborative work and modular control architectures. The automation and control with modular smart components ensures maximum efficiency and reduces hardware and system costs. The thermal shock process automation allows the researcher to concentrate completely on his basic work.

To conclude the results, section V-A deals with the first aspect to afford more flexibility within the laboratory itself. Section V-B immerses in the aspect of interoperability.

A. Integrating LFBCs into SOA

While LTM-SOLA certainly was implemented based on usual SOA-affine technologies (Java Beans, Web-Services, JBoss), our LFBC-architecture currently uses unorthodox technologies. Running Web-Services on embedded hardware is not widespread, but in the last years there has been research on this topic [13]. Currently there is no serious embedded out-of-the-box product for offering Web-Services. Therefore, we have to implement this feature on top of current embedded platforms or adopt available Web-Service libraries to run on these. Since the hardware performance on embedded controllers is a precious resource, it might be useful to implement the Web-Service in C/C++ in contrast to Java. Therefore, we are evaluating if existing C++ based Web-Service libraries like Apache Axis are suitable for our architecture.

Furthermore, SOA technologies like Universal Description, Discovery and Integration (UDDI) are not yet supported by our LFBCs. Our research pursues the possibility to connect a LFBC with its respective laboratory device to the laboratory Ethernet and its Web-Service will be automatically discovered by the laboratory's UDDI-registry. Once a service is registered, it can be incorporated into a laboratory process.

B. University-Wide Infrastructure-Binding

The primary goal of LTM-SOLA was to integrate the laboratory process into an university wide infrastructure. For further enhancing the learning conditions in mechanical engineering, processing of current research results in e-learning environments is a significant contribution.

The virtual knowledge space used by LTM-SOLA is accessible by all users of WasabiBeans-Applications and also by users of WasabiBeans' underlying repositories, like MILESS. For further broadening the accessible virtual knowledge space and making the laboratory result accessible to an even larger number of users, we are searching for additional interfaces within the university-wide infrastructure which are appropriate for a link to our new architecture.

References

[1] M. E. Auer, A. Pester, and H. Ressler. (2006) International Association of Online Engineering (IAOE). [Online]. Available: http://www.online-engineering.org/

[2] (2002) Poznań Supercomputing and Networking Center. VLAB - Web Portal. [Online]. Available: http://vlab.psnc.pl/index.html

[3] (2007) Monash University Information Technology. Virtual Lab (VLAB). [Online]. Available: http://vlab.infotech.monash.edu.au/

[4] R. Montella, G. Giunta, and A. Riccio, "Using grid computing based components in on demand environmental data delivery," in *UPGRADE '07: Proceedings of the second workshop on Use of P2P, GRID and agents for the development of content networks*. New York, NY, USA: ACM, 2007, pp. 81–86.

[5] M. Lawenda, N. Meyer, M. Stroinski, T. Rajtar, M. Okon, D. Stoklosa, and D. Kaliszan, "Effective job management in the virtual laboratory." in *EGC*, ser. Lecture Notes in Computer Science, P. M. A. Sloot, A. G. Hoekstra, T. Priol, A. Reinefeld, and M. Bubak, Eds., vol. 3470. Springer, 2005, pp. 414–423.

[6] F. Ferber, M. Gießmann, T. Hampel, and J. Schulte, "Bringing Together High-Tech Laboratories and E-Learning Infrastructures," in *Proceedings of the 50th International Symposium ELMAR-2008*, M. Grgic and S. Grgic, Eds., vol. 2, no. 978-953-7044-09-1. Croatian Society Electronics in Marine - ELMAR, Zadar, September 2008, pp. 361–364.

[7] L. Cardelli, "Program fragments, linking, and modularization," in *In ACM Symp. on Principles of Programming Languages*. ACM Press, 1997, pp. 266–277.

[8] I. Melzer, *Service-orientierte Architekturen mit Web Services: Konzepte - Standards - Praxis*, 2nd ed. Elsevier, Spektrum Akademischer Verlag, 2007.

[9] T. Hampel and R. Keil-Slawik, "sTeam - designing an integrative infrastructure for web-based computer-supported cooperative learning," in *Proceedings of the 10th International Conference on World Wide Web*, 2001, pp. 76–85.

[10] J. Schulte, T. Hampel, T. Bopp, and R. Hinn, "Wasabi Framework - An Open Service Infrastructure for Collaborative Work," in *SKG '07: Proceedings of the Third International Conference on Semantics, Knowledge and Grid*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 242–247.

[11] ——, "Wasabi Beans – SOA for Collaborative Learning and Working Systems," in *DEST '08: Proceedings of the Second IEEE International Conference on Digital Ecosystem and Technologies*. Phitsanulok, Thailand: IEEE Computer Society, February 2008.

[12] H. Gollan, F. Luetzenkirchen, and D. Nastoll, "Miless - a learning and teaching server for multi-media documents," in *Lecture Notes in Control and Information Sciences: Workshop on Wide Area Networks and High Performance Computing*. London: Springer, 1999.

[13] A. Rettberg and W. Thronicke, "Embedded system design based on webservices," in *DATE '02: Proceedings of the conference on Design, automation and test in Europe*. Washington, DC, USA: IEEE Computer Society, 2002, p. 232.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8378
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8378