2010

Alignment: A New Software Architecture Approach to Support Streamlining Business Processes

Susanne Altendorfer
University Salzburg, susanne.altendorfer@unileoben.ac.at

Thomas Aschauer
University Salzburg, thomas.aschauer@cs.uni-salzburg.at

Gerd Dauenhauer
University Salzburg, gerd.dauenhauer@cs.uni-salzburg.at

Wolfgang Pree
University Salzburg, wolfgang.pree@cs.uni-salzburg.at

Alignment: a new software architecture approach to support streamlining business processes

Susanne Altendorfer, Thomas Aschauer, Gerd Dauenhauer, Wolfgang Pree

C. Doppler Laboratory Embedded Software Systems
University Salzburg
Jakob-Haringer-Str. 2, 5020 Salzburg, Austria
firstname.lastname@cs.uni-salzburg.at
www.uni-salzburg.at/CDL

Abstract

Traditional business structures nowadays have to change fast to keep up with customers needs, which is often not possible due to monolithic software architectures and multiple software systems that do neither respond to process requirements nor interact well. Many existing software systems, however, are too complex and too unrelated to the business to support this change accurately. New ways of software architecture are needed to respond to changing requirements and support the business processes. Information systems have to be integrated into the organization's structures. It seems that a component-based software architecture, which supports the whole value chain, forms the basis for a business process reorganization to enable changes. In this paper we introduce a modeling approach based on Clabjects. We demonstrate how that approach can be applied to an industrial case in order to streamline and support the business processes. Further, this paper further describes the envisioned business process improvements.

Keywords

Business software, business process streamlining, IT business alignment, software architecture, model-driven system development.

INTRODUCTION

Companies strive for increasing productivity in their quest to stay competitive on the market. They reached a point where a structural change has to take place: Over the years organizations with traditional structures happened to create business departments as silos: They became big and shoreless (Oesterle 1995). This leads to blockades and information-silos. New ways of managing the organizations have to be found. Business Reengineering is one way to optimize the business processes according to the company's requirements responding to customer needs (Gadatsch 2008).

Business process management is an essential part for companies and becomes always more important. This productivity nowadays requires a business structure redesign with a process as well as a software redesign. Nevertheless, the software side is often left out of considerations. Although efficient software could considerably contribute to an increasing productivity. Especially companies, where software-intensive products are the main focus, have a high potential to profit from a reengineering project, where the software is also in the focus of the reengineering`s scope. In these days, computing and communication technologies are increasingly powerful. As the power and possibilities of technology expand, so does their impact on organizations and their potential to help them to improve (Kawalek 2008). As technologies enable new software architectures it is advisable to introduce not only standalone software systems but business software to support business processes. In this context Sundblad states that business software is often introduced for exactly one reason: It should support the business and its activities to increase the productivity and effciency of the business (Sundblad 2007). The advantage of business software lies in the fact that business software, contrary to standalone software, can be integrated in all relevant business processes to get a higher scale effect.

Information Systems (short IS) which lead to improvements in the organization's strategic and operational processes are not at all new in literature. However, empirical studies on the impact of business software are mostly restricted to Enterprise Resource Planning (short ERP) or Customer Relationship Management (short CRM) introductions, and are limited in scope and depth. Discussions on software modelling approaches are also limited with regard to the economical side effects the software could have on an organization's processes.

Especially in the automotive industry, where systems are very complex and consist of many software-intensive products, software restructuring by means of introducing a business tool would have manifold effects on the organization. According to Sundblad a reason for the hindrance of business agility is that IT people don't understand or just neglect business matters to be able to design systems that effectively support the business. Therefore a rethink has to take place to handle complexity also on the business process level and thus help streamlining the processes.

There are some problems faced by both research and industry in terms of restructuring. Many systems have been constantly growing over the years and had just been adapted when needed without considering the business requirements. Even if one could start from scratch it would not be easy to develop adequate software systems as most software developers do not consider the business area as relevant area.

Our research group, in contrast, tries to consider both issues—the software and the process reengineering. We argue that monolithic software architectures impede business processes to adapt to changing requirements whereas a so-called clabject-based modeling approach, enabling a business software approach, offers more flexibility and thus forms the basis of a business process reorganization that enables major improvements.

The paper therefore presents a research in progress with a twofold intention: Our research group is developing a new software tool, which should be used as business tool along the whole value chain. On the other hand we are analysing the existing process structures to integrate the new tool in a way to support the structures to respond to the market requirements. Nebst the analysis leads to statements how an organization can profit from a business tool introduction. The main idea is to adapt the process structure to the changing business requirements. Thus, the knowledge intensive processes have to be supported by a modeling approach, software architecture and tools which help to cope with the complexity.

The remainder of this paper is structured as follows: First the domain is described in detail to understand the intentions behind the reengineering approach. Next the clabject-based modeling approach is introduced and requirements are described in short. Then the process restructuring is presented and potentials for savings are sketched. This section outlines the feasibility of this modeling approach for test facility systems and its impact on the business process structure. The related work section deals with a special approach in a different area, where similar concepts are in use. The end of this paper is concluded by a section on the outlook on future work.

DOMAIN ANALYSIS

Our research partner is an engineering company that develops and sells engine test bed systems, with a focus on the automotive industry. These products are examples of automation systems. A test facility system basically measures, records, and visualizes numerous values provided by sensors according to test plans. The test facility requires appropriate parameterization for that purpose. Due to the various different use cases of test facilities, test facility systems have to be adapted according to customer-specific requirements. Typical test facilities consist of hundreds of thousands of components. According to Martyr “an engine test facility is a complex of machinery, instrumentation and support services, housed in a building adapted or built for its purpose. For such a facility to function correctly and cost-effectively, its many parts must be matched to each other while meeting the operational requirements of the user and being compliant with various regulations” (Martyr 2007).

In the ongoing research we are developing and evaluating a pioneering modeling approach that unifies the concepts of classes and objects. The goal of our research cooperation is to develop a framework for a model-driven development of test facility automation systems with the additional requirement of being able to refine the models from coarse-grained descriptions to ones that describe all the relevant details of the system. The final model radically simplifies the test facility’s configuration and operation.

This top-down refinement of the model allows the streamlining of the associated business processes as one tool is used along the whole value chain. Thus information loss and interface problems can be radically reduced or even eliminated.

Monolithic Architecture with Unstructured Parameters

A test facility system needs to be tailored to customer demands in a straight-forward way, which is not well supported by the current monolithic software architecture. “To build or substantially modify a modern engine test facility requires coordination of a wide range of specialized engineering skills; many technical managers have found it to be an unexpectedly complex task” (Martyr 2007).

The fact that the existing software, used to configure and parametrize the test facility system, has evolved over the past decades makes the situation even worse: The software comprises appr. 4.5 million lines of code, mainly written in C++ and C, making adjustments quite difficult, does no longer respond to business needs in an appropriate way. In the first years where it was developed it fulfilled the requirements best, but with the ongoing years requirements changed and fundamental software changes would have been inevitable. From a user's perspective, the current system requires error-prone editing of parameters (in the order of tens of thousands) in spreadsheet-like tables, as well as the adaptation of configuration files and scripts scattered in the file system (cf.

Figure 1. Thus the handling is considerably time-consuming and a training on the tool almost needs years.

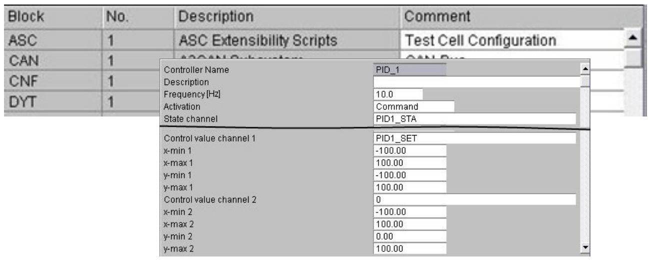

Figure 1: Current Parametrization Format

Using this representation and finding specific parameters is difficult as parameters associated with a physical entity of the test facility, such as the unit under test, are not grouped accordingly. If an entity is changed the user has to know by heart which parameters are affected. Configuration files and scripts exist in various standard and proprietary formats and different editors are required for each file type. Thus the handling is time-consuming and expensive. Furthermore the tool puts the company under the risk of a bottleneck, as only a few people understand the whole monolithic code conglomerat and the intentions behind different changes.

Due to its complex user interface and its focus on technical details, it is a system that can only be adapted and operated by experts. It is a standalone software tool that is not at all integrated in the overall business process chain, as consecutively described.

Current Business Process Chain

Figure 2 gives a coarse-grained overview of the most important business processes. All processes are considererd as a separate phase with several subprocesses, each having separate handover breaks.

Figure 2: Current Business Process Chain

The current software is restricted to the order fulfilment process, indicated in Figure 3, without having any connection to the up- and downstream processes, which are essential for the overall project success. Figure 3 illustrates this situation: Each process is a separate process and the handover points from one process to the other are without any tool support. Thus relevant information might be lost throughout the process chain, and information needed in the progression of the project might not be available, as the importance of these data might not be clear upfront.

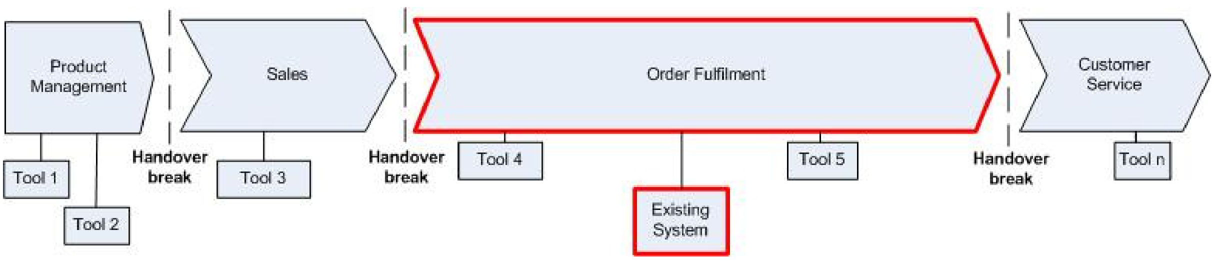

Figure 3: Position of the current software system

Currently, each process uses different tools, indicated as tool 1 – tool n, such as Excel sheets or proprietary tools, e.g. for the sales support, without dedicated interfaces. These gaps cause a significant information loss and cause extra manual conversion and transfer efforts. As the existing software systems have become complex artifacts and do not interact with each other, the actual situation can no longer be changed through incremental adaptations of the existing test facility software.

CLABJECT-BASED MODELING APPROACH

When defining the requirements to substitute the existing parametrization tool, it was decided to go for a radical new approach. And to support the business in the best way, an approach has to be found where the new tool could be integrated as business tool into the process chain. Thus we regard a modeling approach as adequate if it fulfils certain requirements which are relevant to establish a business tool in the research area.

Requirement 1: It allows the stepwise refinement of the description of a test facility.

This would allow a coarse-grained specification of a test facility system in the sales phase which is then continuously refined until the test facility can be deployed and later maintained.

Requirement 2: The modeling should be natural.

This means that domain experts without programming skills can accomplish it.

Requirement 3: It should be a visual tool.

Thus the parametrization can be done without endless spreadsheet-like tables but with 1:1 representation of real-world objects. What we basically came up with as modeling approach is what might be called a deep virtualization of the corresponding real-world test facility.

These are just some general requirements. Any description of further software related requirements would go beyond the scope of this paper.

For the development of software intensive systems, model-driven engineering (MDE) is a promising approach for coping with the inherent complexity of large systems. The basic idea of MDE is that models are the main artifacts describing a system under study and a model at a certain level of abstraction can be transformed into another model at a possibly different level of abstraction (Aschauer et al. 2009b).

By examining the examples of how to model an engine and its attached sensors, we identified problems of conventional modeling languages such as UML (OMG 2007). Instead we decided to go for what Colin Atkinson et al. (Atkinson and Kühne 2003) have called clabject-based modeling: an approach that unifies the notion of classes and objects. The components of the test facility model, which can be a couple of hundred thousands, are described as clabjects. The idea of unifying the concept of classes and objects goes back to an experimental object-oriented language called SELF (Ungar and Smith 1987). The advantage hereby is that the domain engineers do not have to concentrate on specific differences between instances and classes.

Clabjects—Uniform Representation of Classes and Objects

In the context of domain models, Atkinson and Kühne define accidental complexity as introduced due to mismatches between the problem and the modeling means to represent the problem (Atkinson and Kühne 2007). They argue that modeling languages that allow using only two levels, such as UML with the class and object level, induce accidental complexity when modeling domains that inherently require more modeling levels. Therefore they offer the concept of clabjects as the solution.

Figure 4 shows the clabject based approach for the domain of test facility automation systems. The notation used here is similar to that of the original clabject concept. Each model element has a compartment for the name, and a combined compartment for the type facet and the instance facet. The dashed arrows between the levels represent the "instance of" relationship. With a uniform representation of type facets and instance facets, our example can be modeled in a natural way. By definition, the clabjects at the top-level only have a type facet, whereas the clabjects at the bottom level only have an instance facet (Aschauer et al. 2009a).

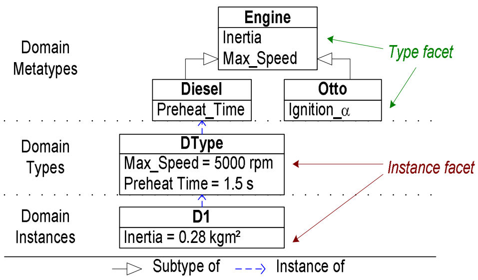

Figure 4: Engine hierarchy with clabjects (Aschauer et al. 2009a)

For the domain of test facility automation systems we extended the clabject notion for handling association inheritance and instantiation, which are essential for bridging the gap between theoretical research and industrial applications. Describing all modeling formalism would go beyond the scope of the paper. We refer to (Aschauer et al. 2009a) for a more detailed description.

Tool Realisation

The tool that we call DeepVTool, for deep virtualization tool, offers several views of the modeled test facility. For example, in the physical view, plugs and connecting cables are represented. The plugs have shapes but also types as in programming languages, so that it can be automatically checked, if they fit. If necessary, all pins of a plug and the associated signals are represented in the virtual test facility. With the clabject approach a library of all relevant elements is developed, as illustrated in Figure 5.

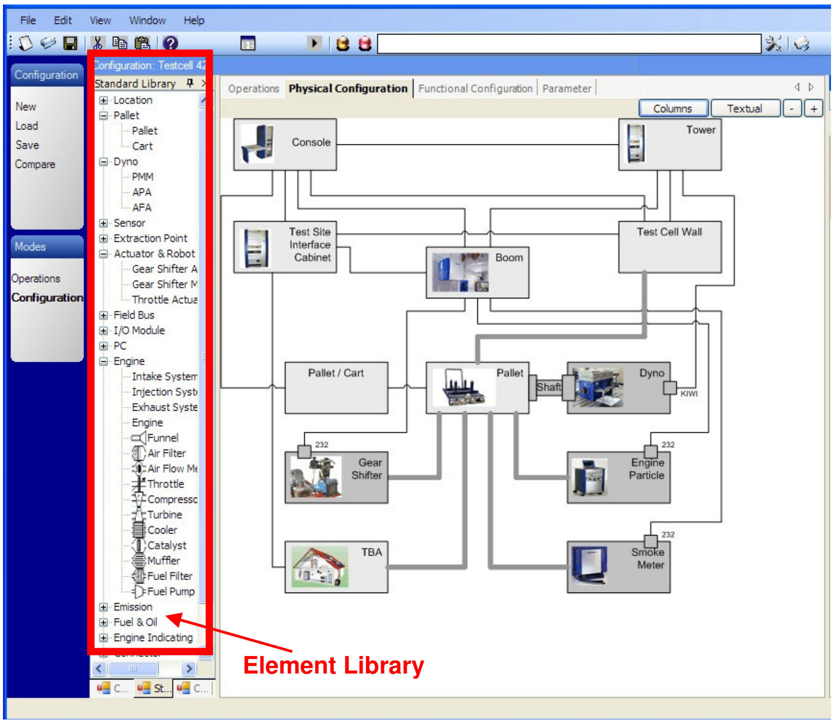

Figure 5: Real-world visualisation with element library

As a business tool, the tool has to support the whole business chain and therefore the architecture has to respond to the requirements of all processes.

BUSINESS PROCESS RESTRUCTURING

Top-down model refinement means that relevant information should be added gradually to the test facility model in each process phase according to the particular requirements. Thus the level of detail increases over the project's progression. A further advantage is that errors or dependencies in the design and development of test facilities can be identified already at an early stage in the sales phase, avoiding costs that are due to late detection and fixing of such errors in the subsequent project execution phase.

Impact on the Workflow

A typical business process chain covers aspects from product management, sales, project execution to the after sales support. Each process has its special requirements and associated roles on the tool. With the introduction of the new business tool, the process landscape could be streamlined in terms of execution times, which is shown in Figure 6.

In general these savings can be achieved primarily:

- by a reduction of cycle times via an elimination of process steps or

- by shifting process steps to upstream processes.

The first one can easily be achieved, as the system offers the possibility that process steps are automated and a manual work is no longer needed. The latter one enables a cost reduction as several steps can be done in house, which is obviously cheaper, than doing the same task at the customer's site. Although we are still at the beginning of our research first estimations show that savings between ten and fourteen per cent can be achieved alone in the project execution process. This first evaluation has been done with scenario techniques based on the existing process and work step descriptions. With a process-wide support even more savings can be achieved as Figure 6 indicates.

Figure 6: Process Chain Streamlining

The changes in the workflow also require a change in the role model of the existing processes. The granularity and level of detail of information however will be adapted to the role concept which is supported by the DeepVTool. Figure 7 shows the different roles and the scope of their rights. The smaller the circle, the fewer rights the role has to operate the tool. The sales engineer for example has least rights, as he is at the beginning of the process chain, where less information is available. By contrast, the technical engineer has more rights as he has to commission and operate the test facility, which requires more detailed information.

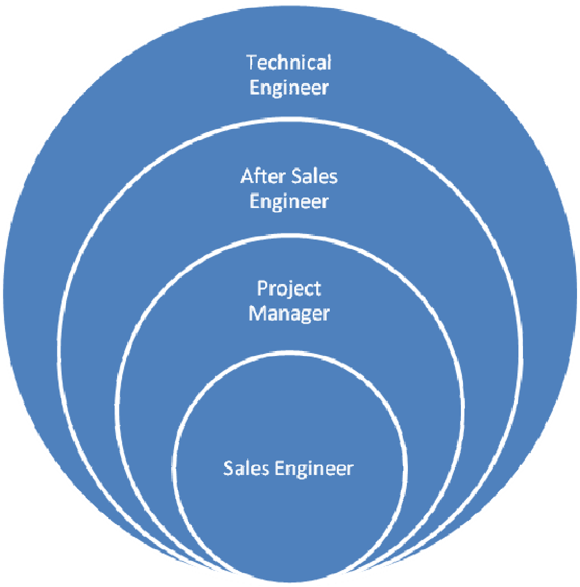

Figure 7: Role Model and scope of rights

Envisioned Processes

In the following we sketch the refactored processes using the DeepVTool. It is a coarse description of the tasks in the processes and how work is done. As the research is still in progress, we can just state how the DeepVTool will support the processes without giving detailed findings on savings.

Product Management

One essential task of the Product Development will be to define and maintain the library elements (test facility components) with the DeepVTool. Thus the model contains rich information on the elements, e.g. the component's requirement on the facility management. This "intelligent" model is then used by the upstreaming processes.

Sales

The sales engineer typically sells a test facility system to a customer. Since test facilities are individually customized to specific needs, the sales engineer starts with configuring such a test facility according to the customers' requirements. This first schematic view of the test facility, as shown in Figure 8, can then be discussed with the customer. With the DeepVTool the sales engineer can accomplish the modeling of a coarse-grained test facility. The DeepVTool allows a visualization of the test facility system from the very beginning in the sales process.

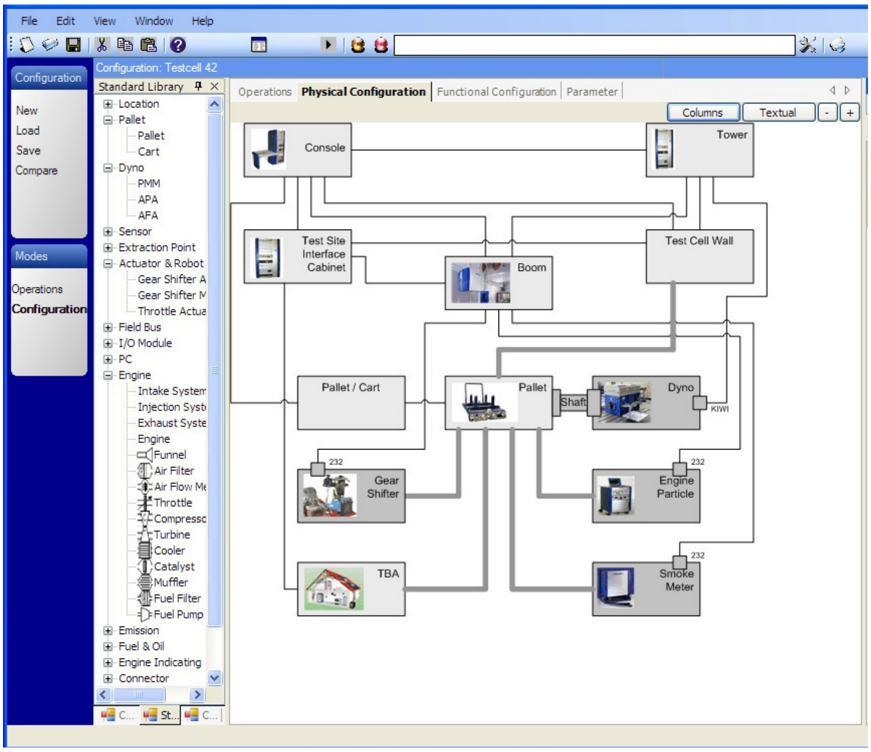

Figure 8: Schematic view of a test facility configuration

This visualization forms the basis for communication and project execution for the downstreaming processes. Another advantage for the sales process is that a configuration can be checked automatically for its consistency, as the sales engineer uses only pre-defined library elements. Thus, the risk for cost-intensive changes later on in the project can be significantly reduced.

By establishing interfaces to other tools used in the sales phase, such as SAP R/3, obvious system discontinuities can be avoided. Thus the sales engineer

- can save time until the offer is finished,
- does not have to fill in the same or similar information in different tools and
- can ensure that no information gets lost.

Order Fulfilment

This phase includes all parts of project execution from the time the contract is signed until the project is finished and handed over to the customer's responsibility and to the customer service. The project execution process is characterized by the fact that different departments are involved. The DeepVTool supports them by providing one consistent model. This can help to decrease execution time, thus saving costs. Furthermore at the beginning of the phase the DeepVTool model forms an ideal basis to discuss the project's scope with all stakeholders and to identify aspects that have a higher risk in the realization than others and where more work than usual is necessary.

Customer Service

In this phase the DeepVTool support is restricted to documentary tasks. But nevertheless it can help resolving problems as the customer and the after sales engineer have the same model of the test facility and thus the understanding for the problem should be more straight-forward. Thus the time for fixing problems could be shortened, which would positively effect the customer's satisfaction.

Changes in the Overall Tool Chain

Figure 9 demonstrates the core benefit of such a modeling tool. It shows the integrated approach of the DeepVTool across the process chain. Through the integration the tool becomes a business tool as it supports all essential processes and will have pre-defined interfaces to other tools, which are relevant for the processes and where an interaction is essential. While the DeepVTool is the technical backbone, other existing tools, indicated as tool 1 to tool n, are mainly used for calculations purposes. All technical solution components from the DeepVTool could be automatically exported to the coresponding tools. Thus, the workflow is well supported and needed information can be easily accessed at the right time.

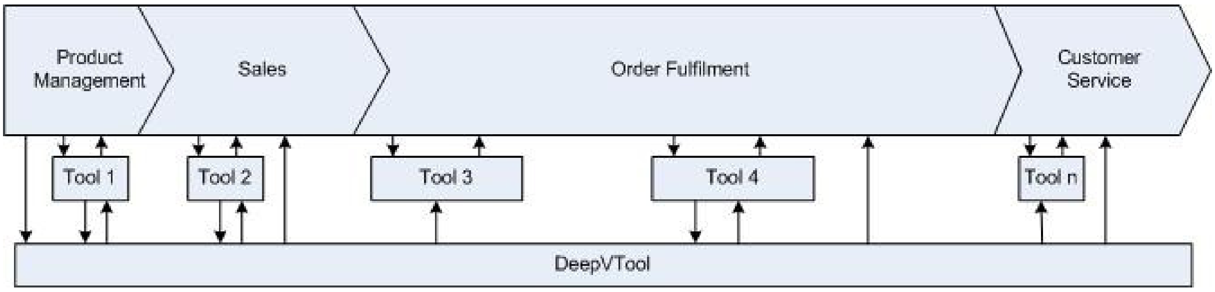

Figure 9: Envisioned tool supported process chain

RELATED WORK

Dealing with IT – Business Alignment considering the technical and economical / organizational aspects is very important. However, overall concepts are rarely described in literature. Literature on a business tool introduction and the effect on the organization are almost not available. There exists literature on enterprise engineering or parts of enterprise engineering, like (Bucher 2009) but those concepts are not that integral as our approach. Further approaches in the area of enterprise modelling are described and dealt with by (Frank 2002). Here the focus is mainly on the organization and its modelling and not so deeply on a specified alignment of IT and business aspects like our approach.

What we found quite similar from the idea of visualization is the work in the area of building information management which was first discussed in detail by Chuck Eastman in 1975 and then comparable research was conducted throughout the following decades (Eastman and Teucholz 2008). Concerning the idea of visualization we will describe this approach in a little bit more detail: In the building industry the visualization focused so far only on the visual look&feel of a building, including virtual walk- throughs. According to (The Economist 2008) Dr Eastman, professor of architecture and computing at the Georgia Institute of Technology in Atlanta, and others have championed an approach called building information modelling (BIM). The BIM approach is based on objects with their detailed properties. The model has an associated database including information about the various relationships between these objects. Thus any changes in the model imply that all related objects are automatically updated. Thus, any mistakes are identified in the design phase. Quantities of material needed and hence construction costs can be calculated at a high accuracy at an early stage. Once the model has been created, detailed plans of particular subsystems, such as water und electrical wiring can be extracted.

Figure 10 and 11 show different computer models of a building construction in different levels of detail:

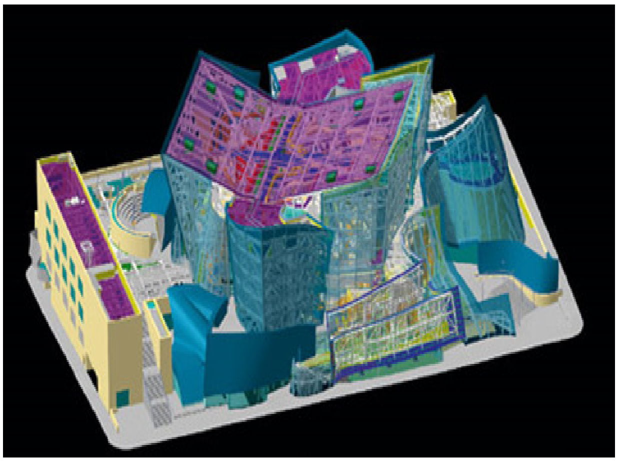

Figure 10: Computer model of a building (picture from (The Economist 2008))

Figure 10 shows a computer model of a whole building integrated in its surrounding, whereas Figure 11 sketches a detailed subsystem plan belonging to the building construction in Figure 10, such as water or electrical wiring. Depending on the construction plan's purpose the subsystem plan can differ.

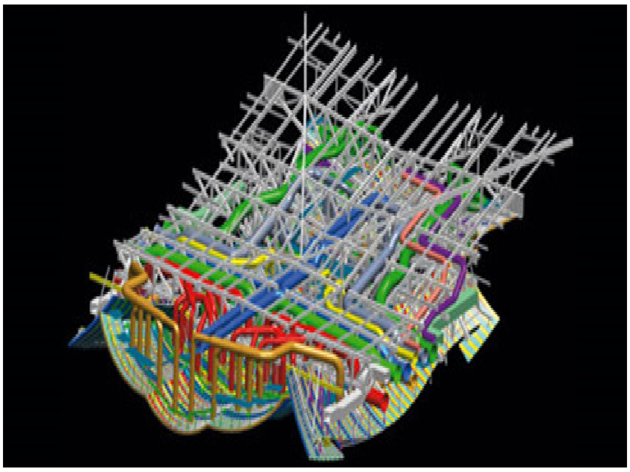

Figure 11: Computer model of a building with detailed plans of a particular subsystem (picture from (The Economist 2008))

CONCLUSION AND FUTURE WORK

In this research paper we have demonstrated the need for streamlining business processes according to business requirements and the need to adapt the software architecture and the modeling approach too. An appropriate modeling formalism to support model-driven engineering in the domain of test facility automation systems is therefore needed. We showed that with a process wide tool support a streamlining of the processes can be achieved. A first analysis indicates that in a final stage savings up to a two-digit percentage should be achieved.

We are currently working on the following issues: In close cooperation with our industry partner we are modeling real-world test facility examples, and first results corroborate that the modeling method as well as the DeepVTool is applicable in practice. The analysis of the existing processes is also almost finished and we are concentrating on the integration of the DeepVTool. Thus adaptations within the process structure have to be analyzed and evaluated. Furthermore, interface requirement specifications will be defined to establish interfaces to other tools.

A prototypical test facility case study will be used to demonstrate the feasibility of the new approach, both from a technical as well as a business perspective. This forms then the basis for verifying or falsifying the hypothesis that the new approach and tool set have a positive qualitative and quantitative effect on the business processes. Further to the evaluation we will retrieve a method to integrate and evaluate the introduction of a business tool organizationwide. This method should then be able to be used within different organizations and areas of applications.

As such a paradigma change (from standalone tools to business tools) might cause massively cultural and political resistance, the change has to be accompanied by an appropriate communication model, which has to be defined in this context. Savings can also be realized as the customer satisfaction can be directly affected by the DeepVTool. Thus the new modeling approach and the DeepVTool will quite likely have a significant impact on the whole company and its organizational structure. This will be further elaborated in the coming work.

REFERENCES

Aschauer T., Dauenhauer G., Pree W. 2009a: Multi-Level Modeling for Industrial Automation Systems. Proceedings of the 35th Euromicro Software Engineering and Advanced Applications (SEAA'09) Conference 2009, Patras, Greece, 27-29 August 2009.

Aschauer T., Dauenhauer G., Pree W. 2009b: Towards a Generic Architecture for Multi-Level Modeling. Proceedings of the 2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture WICSA/ECSA 2009. September 2009. pp 121-130.

Atkinson C., Kühne T. 2003. “Model-Driven Development: A Metamodeling Foundation”. *IEEE Software. Vol. 20. No. 5*. pp. 36-41.

Atkinson C., Kühne T. 2007. „Reducing accidental complexity in domain models”. Software and Systems Modeling, vol. 7. no. 3. pp. 345–359.

Bucher T. 2009. “Ausrichtung der Informationslogistik auf operative Prozesse”. Kovac Verlag. Hamburg.

Eastman C., Teucholz P. 2008. “BIM Handbook. A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors”. Wiley & Sons Inc. New Jersey 2008.

Frank, U. 2002. “Multi-Perspective Enterprise Modeling (MEMO) - Conceptual Framework and Modeling Languages”. *Proceedings of the Hawaii International Conference on System Sciences (HICSS-35)*, Honolulu 2002, (10 pp.)

Gadatasch A. 2008. “Grundkurs Geschaeftsprozessmanagement“. *Vieweg + Teubner, GWV Fachverlage GmbH*, Wiesbaden, Germany, 5., erweiterte und überarbeitete Auflage edition.

Kawalek J. 2008. “Rethinking Information Systems in Organizatons. Integrating Organizational Problem Solving”. Routledge. New York.

Martyr, A. 2007. “Engine Testing: Theory and Practice”. Butterworth Heinemann. 3rd revised Edition.

Object Management Group 2007. “Unified Modeling Language Infrastructure,” version 2.1.2.

Oesterle, H. 1995. “Busienss Engineering. Prozess- und Systementwicklung 1: Entwurfstechniken”. Springer. Berlin. 2nd Edition.

Sundblad S., Sundblad P.2007. “Business Improvement through better architected software”. *Microsoft Architect Journal.*

The Economist 2008. “From blueprint to database”, Online Edition. June 2008.

Ungar, D. and Smith, R. B.: “Self: The power of simplicity”. Proceedings of OOPSLA ’87, ACM SIGPLAN Notices, Vol. 22, pp. 227–242.

COPYRIGHT