TOWARDS PROCESS-AWARE ENTERPRISE SOFTWARE ENVIRONMENTS
A Framework

Bela Mutschler, Johannes Bumiller
DaimlerChrysler Research & Technology,Wilhelm-Runge-Str.11, 89013 Ulm, Germany

Manfred Reichert
University of Ulm, Department Databases/Information Systems, Germany

Keywords: process-awareness, evolving enterprise software environments, business process management and integration, economic-oriented assessments of process-oriented technologies

Abstract: To stay competitive at the market companies must tightly interlink their software systems in a process-oriented manner. While the business process paradigm has been widely accepted in practice, the majority of current software applications are still not yet implemented in a process-oriented way. But even if, process logic is hard-wired in the application code leading to inflexible and rigid software systems that do not reflect business needs. In such a scenario the quick adaptation of the software systems to changed business processes is almost impossible. Therefore, many software systems are already out of date at the time they are introduced into practice, and they generate high maintenance costs in the following. Due to this unsatisfactory business process support a software system's return on investment is often low. By contrast technologies which enable the realization of process-aware enterprise software environments will significantly contribute to improve the added value of IT to a company's business. In this paper we characterize process-aware enterprise software environments, describe benefits and present a conceptual framework outlining our theses.

1 MOTIVATION

A significant change has occurred in how value of IT for business is realized. Technology innovation, not long ago the dynamic driver of both business and technology infrastructure change, is thereby not seen as an important value driver anymore, but mainly as the source of high costs. Therefore, it becomes an important task to identify technology innovations that have a measurable positive impact on a company's added value. Concerning this claim one special area of high relevance is software since the development of new applications has to realize a positive return on investment (ROI) as well (cf. Tockey, 2004).

Designing and implementing enterprise application software a tight interweavement between software systems (e.g., a product data management system) and business processes (e.g., the process of developing a new car) is indispensable for economic success. However, very often there is a big gap between business needs and the respective business process support offered by current software systems.

Even more, today's enterprise applications are static and inflexible. Consequently, in many cases software systems are already out of date at the time they are introduced to the market or brought into operational use (e.g., if the business process originally to be supported has changed in the meantime).

Instead, software systems are needed that provide adequate business process support through their entire lifecycle. As software lifecycles are continuously increasing (with the goal to avoid the high effort of developing a new application or migrating a legacy system) software more and more has to face changes directly influencing it.

Changes can have their origin in both internal and external drivers (cf. Fig. 1) leading to new requirements (e.g., the company-wide use of new hardware or software technologies). The scope of changes can additionally be confined by both internal and external restrictions (e.g., laws).

Mutschler B., Bumiller J. and Reichert M. (2005).
TOWARDS PROCESS-AWARE ENTERPRISE SOFTWARE ENVIRONMENTS A Framework.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 374-378
DOI: 10.5220/0002544703740378

Drivers and restrictions influence a company's software infrastructure leading to a highly dynamic IT evolution. Therewith, we distinguish two basic evolutionary concepts: organization-driven IT evolution and technology-driven IT evolution (cf. Fig. 1).

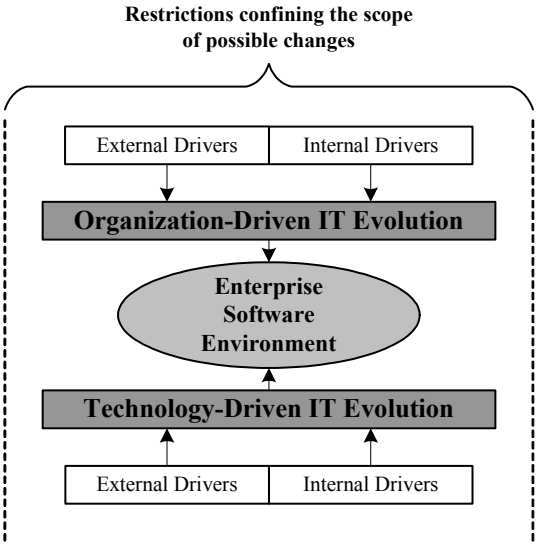

Figure 1: Drivers and restrictions influencing today's enterprise software environments´.

Applications lacking the capability to adequately deal with changes generate high costs. Thereby it is especially the operation and maintenance phase that significantly impacts a software system's economic efficiency (as well as its return on investment). In particular, the number of errors can be identified as a major cost factor. Adapting a software to emerging requirements when it is already in use is by orders of magnitudes more expensive than doing this in earlier phases (e.g., during design or implementation). Moreover, if process logic is hard-wired within the application code (as it is often the case in current software systems), it becomes even more difficult to rapidly and correctly adapt software to the changes. High costs may result as new errors are made during the necessary implementation of the changes.

In summary, today's enterprise software environments are increasingly faced with flexibility. Obviously, there is the need for adaptable software with a high return on investment.

Our new concept of *Process-Aware Enterprise Software Environments* (PAESE) illustrated in this paper offers promising perspectives concerning this request (cf. Section 2.2). Such an environment is implemented using various process-oriented approaches and software technologies. In our opinion, the PAESE concept can significantly contribute to improve the added value of IT for a company's business and strengthen its market position, its profitability, and its economic efficiency.

In this paper we analyze our theses and discuss relevant topics in detail. The remainder of this paper is organized as follow. In Section 2 we describe the correlation between enterprise software systems and the business processes they shall support. In particular, Section 2 sketches aspects of evolving enterprise software environments (including a description of drivers leading to this evolution). Section 3 presents a conceptual framework illustrating our theses. Finally, Section 4 motivates the accomplishment of economic-oriented assessments (e.g., cost benefit analyses) of process-oriented approaches and software technologies (e.g., workflow management systems) and finally gives an outlook as well.

2 EVOLVING ENTERPRISE-SOFTWARE ENVIRONMENTS

Enterprise software environments identify the main components of an organization's software infrastructure and the ways in which these components interact with each other in order to achieve defined business goals (cf. Sowa, 1992; Zachmann, 1987).

Realizing effective and efficient enterprise software environments with a high return on investment is only possible, if an adequate handling of a company's entire process map is achieved. The key to success will be the continued alignment of existing software systems not only to short-running, but also to long-running business processes.

Emerging requirements can be a serious problem in the effort to sustain the requested alignment. As already stated in Section 1 such emerging requirements can be driven by various internal or external events, as well as by a number of restrictions.

2.1 Drivers of Evolution

Dealing with changes in software can be seen as a daily challenge in many organizations (cf. Reifer, 2002). As motivated, it is not sufficient to provide only a static view on process-centric applications. In contrast, business processes, functions, and applications are continuously subject to change.

In the following we classify and describe relevant internal or external drivers in more detail. Doing so, we distinguish between events leading to a technology-driven IT evolution and events leading to an organization-driven IT evolution (cf. Fig. 1).

At first, we discuss technology-rooted drivers for process changes (and related restrictions).

One internal driver in this context stems from the effort to align business processes (e.g., development of a new car or treatment of a patient) to new information and communication technologies. Very often this is accompanied by process optimization efforts.

As an example take healthcare procedures. The introduction of mobile devices (e.g., a PDA to store patient data) in hospitals requires the solution of technical problems such as security, privacy, and reliability of services offered by those devices. These solutions can only work in practice if they align with the business processes employing these devices, and if these processes themselves are redesigned to work effectively with these devices. Even more ambitious is the introduction of ambient technology such as wireless body area networks for health monitoring. Apart from the difficult technical problems in making these devices communicate securely and reliably, healthcare processes must be redesigned to work effectively with them.

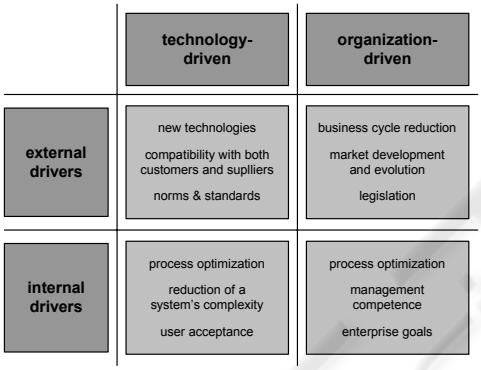

Figure 2: Internal/external drivers influencing enterprise software environments

Another internal driver stems from the effort to reduce the complexity of enterprise environments. This is a success-critical factor since control and maintenance of these environments become more and more difficult. The higher the complexity of a software system is the higher error rates and therefore costs are. Current development processes in the automotive sector, for example, are based on a multitude of heterogeneous applications.

A very important internal driver can be a bad acceptance regarding the usage of software systems. In large scale environments, hundreds up to thousands of users may work with an enterprise application (e.g., a document management system). While some of them are working with the system every day, others do this rather seldom. Additionally, users may have varying tasks, skills, and experiences. For the design of high-quality human-machine-dialogues it is thereby essential to consider for which kind of user a system is intended. This is important as different user classes have different requirements regarding interfaces (cf. Mutschler, 2004).

A typical external driver can be the lack of adequate rules and standards. For example, there are a lot of proposed process description standards (e.g., *Business Process Execution Language)*. But due to missing or too complex standard features a decision to use one of the existing standards can turn out as false in the future.

Another important external driver is the introduction of new information technologies like computing on-demand, grid-computing, and the semantic web. In many cases hypes arise around the market introduction of new technologies. This can produce significant pressure on enterprises to invest into such technologies without deeper reflection.

The third important external driver concerns a company's compatibility with both customers and suppliers (this is known as the *extended enterprise*). The customer's wishes (e.g., car features) as well as the supplier's requests concerning work collaboration and approaches of data exchange (e.g., the *Standard for the Exchange of Product Model Data*) have to be met. Original Equipment Manufacturers (OEMs) in the automotive industry, for example, have to distribute the well-developed solid models and documents of a vehicle (the managed product) to all people involved within the extended enterprise in a synchronized way.

After having described important technology-focussed drivers and restrictions, we now summarize organization-rooted drivers and possible restrictions.

Key internal drivers, for example, include the strategic objectives of the enterprise (e.g., to sell a certain number of cars within a given quarter). They are usually introduced and controlled by the management board and directly influence a company's organization. To achieve defined objectives it is necessary to adapt an organization and its business processes to these goals.

Business process optimization and reengineering are other internal drivers. The objectives of respective efforts may be to shorten process cycle times (e.g., in order to reduce time-to-market) or to increase product/service quality.

A powerful external driver, for example, can be change at the market site (e.g., the trend to drive Sports Utility Vehicles) requiring reactions from a company to defend its market position.

The most relevant external restrictions are changes in law. Each country can have different rules and laws concerning certain technology aspects. A globally sold product (e.g., a car) has to fulfil certain security aspects (e.g. concerning airbags) in one country in a more restrictive way than

in another country. Despite this variety every special law has to be met in the respective local markets. Whenever laws change, the respective business processes have to be adapted accordingly.

Only the capability of organizations to react to these dynamic drivers and restrictions assures that changes can be successfully and rapidly treated. The faster a company adapts its software to emerging requirements the better its position in the market will be.

2.2 Process-Aware Enterprise Software Environments

As stated before, business processes and information technologies have to be well orchestrated within an enterprise, but changes can complicate this request significantly. To simplify and fasten the adaptation of changes, existing applications must easily be aligned to the changed business needs. The new concept of *Process-Aware Enterprise Software Environments* (PAESE) supports this claim.

In particular, process-aware enterprise software provides more agility compared to classical, function-oriented software. It is realized by strictly following the process paradigm during software development and the intensive use of process-oriented approaches. While the ladder includes software technologies like workflow management systems or enterprise application integration tools, the former is based upon the strict separation of process logic from application code, and the system-supported modelling, execution, and monitoring of business processes by powerful process engines.

Process-aware enterprise software can be characterized by the following requirements. It

- has to support process-oriented perspectives and ways of thinking; this means that the basic building block is the process.
- must allow the fast and cheap realization and customizing of new/existing processes.
- must enable easy integration of self-developed as well as of bought-off-the-shelf software components.
- must support rapid business process changes as well as the propagation of these changes to current process instances (if possible and desired by the users).
- must provide an extensive support of process-oriented functions (e.g., process analysis or temporal constraint management).

Following the strict separation of business logic and application functionality, business process changes can be handled at a high semantic level and maybe without the need for recoding the complete application (or parts of it). This, in turn, helps to reduce the overall complexity of software systems. But if it is necessary to change an application's implementation, the reduced complexity makes it easier anyway.

Thus, the development of enterprise software can be realized faster and cheaper (and with reduced error rates) than with conventional software technologies (where process logic is still directly hidden in the application code). Altogether, the maintenance effort decreases while a software systems cost effectiveness increases. Only enterprises that meet these requirements are able to generate a short term return on investment and a long-term value.

In fact, process-orientation is even fundamental for enterprises as it enables the requested tight interweavement between business process support and the existing IT infrastructure requiring also a closer integration of the business processes with the business functions (cf. Katsma, 2004).

To provide a process-oriented baseline, various aspects reaching from modelling and analysis issues to the system-supported control and monitoring of processes have to be handled.

Our framework illustrated in the following chapter outlines the idea of a Process-Aware Enterprise Software Environment and provides initial thoughts about how to validate our theses.

3 THE FRAMEWORK

Figure 3 illustrates our conceptual framework to illustrate the requirements and challenges of realizing the new concept of an efficient and highly adaptable Process-Aware Enterprise Software Environment (PAESE).

Process-awareness represents the bridge between the business processes and the software systems used to support them. Both business processes and software systems are installed in a dynamic and fast changing environment influenced by various drivers and restrictions. To continuously meet real-world requirements, enterprises have to adapt both their business processes and software systems to these changes. Doing so, process-oriented approaches and software technologies offer promising perspectives.
We distinguish between business process technologies and software development technologies. While the ladder is used to handle the technology-driven IT evolution, the former supports the adaptation of the organization-driven IT evolution. To pool software technologies and approaches we envision two logical containers in the framework – each of them collecting process-oriented concepts and technologies.

Typical examples of process-oriented software concepts focussing on business processes (technology class 1) are enterprise application integration, workflow management, business process analysis, and business process performance management. We collect respective technologies in a container called *Business Process Technologies* (cf. Fig. 3). This container is additionally divided into two subclasses. Problems concerning the integration of business processes and software systems can be solved by *Business Process Integration* approaches and technologies (BPI technologies) like enterprise application integration. In contrast, problems concerning process management changes can be encountered with the idea of *business process management* approaches and technologies (BPM technologies) like workflow management.

Promising approaches regarding the development of flexible, process-aware software systems (technology class 2) are agile development methods (e.g., eXtreme Programming) or process-oriented usability engineering methods. These approaches are collected in an alternative container called *Software Development Technologies* (cf. Fig. 3).

4 THE NEED FOR ECONOMIC-ORIENTED ASSESSMENTS

From the business perspective enterprises are faced with an increasingly competitive, global marketplace. This situation forces them to streamline and accelerate their product development operations and their organizational business processes. To survive, they must look for better ways to do business (cf. Pisello, 2003). They must be able to realize changes in a quick and cost-effective manner. Such changes and the investments in promising approaches and technologies to handle the changes need to be justified: otherwise why make them? Only a systematic methodological approach can integrate relevant arguments into a meaningful economic-oriented evaluation baseline (cf. Boehm, 2003).

But despite the relatively intensive use of business ratios (like return on investment or net present value) in many IT departments, there currently exists no overall method which allows an integrated analysis or an economics-oriented assessment of process-oriented technologies (enabling process-awareness).

But regarding the assessment of process-oriented approaches, as already seen for example in (Horwitz, 2002) or (Sinur, 2004), it is exactly such an integrated method to assess process-orientation that is needed in our opinion. Therefore, the presented theses have to be validated by case studies, experiments and cost benefit analyses.

Doing so, special *process performance databases* could be used (cf. Fig. 3) to store any kind of economic-oriented characteristics and attributes (such as cost, benefit or risk factors or other business ratios) as well as external data to enable benchmarking. To be able to accomplish assessments on such a database, suitable data has to be derived from adequate value metrics at first (cf. Boehm, 2004).

Motivating companies to use process-oriented technologies can only be successful if the economic efficiency of investments in the described technologies is proven. Therefore, future research efforts address the goal to systematically outline the impacts of software technologies enabling process-awareness.

REFERENCES

Boehm, B., 2003. *Value-Based Software Engineering*; in ACM Software Engineering Notes, Vol. 28, No. 2, pp. 1-12.

Boehm, B.; Brown, W., 2004. *Value-Based Software Metrics*; in EDSER 6, Proceedings of the 6th Intl. Workshop on Economics-Driven Software Engineering Research, pp. 4-6.

Horwitz, S., 2002. *The Economic Benefits of BPM*; in EAI Journal, June 2002, pp. 37-39.

Katsma, P. S., 2004. *Business Function Support*; Pearson Custom Publications.

Mutschler, B.; Bumiller, J., 2004. *Improving Return on Investment of Product Data Management Systems using Usability Engineering*; in Proceedings of the PDT Europe 2004, pp. 215-222, Stockholm, Sweden.

Pisello, T.; Strassmann, P., 2003. *IT Value Chain Management – Maximizing the ROI from IT Investments*; in Information Economics Press.

Reifer, D. J., 2002. *Making the Software Business Case - Improvement by the Numbers*; Addison-Wesley.

Sinur, J., 2004; *Drivers for BPM – 11 Money-Relevant Reasons to start from*; Gartner Report, retrieved from: http://www4.gartner.com/DisplayDocument?doc_cd=119839

Sowa, J. F.; Zachmann, J. A., 1992. *Extending and Formalizing the Framework for Information Systems Architecture*; in IBM Systems Journal, Vol. 31, No. 3, pp. 560-616.

Tockey, S., 2004. *Return on Software – Maximizing the Return of Your Software Investment*; Pearson Professional Education, 2004.

Zachmann, J. A., 1987. *A Framework for Information Systems Architecture*; in IBM Systems Journal, Vol. 26, No. 3, pp. 276-292.