2005

Adaptability Concepts for Enterprise Resource Planning Systems - A Component Framework

Katja Andresen
University of Potsdam, katja.andersen@wi.uni-potsdam.de

Norbert Gronau
University of Potsdam, norbert.gronau@wi.uni-potsdam.de

Adaptability Concepts for Enterprise Resource Planning Systems – A Component Framework

Katja Andresen
University of Potsdam
Katja.Andresen@wi.uni-potsdam.de

Norbert Gronau
University of Potsdam
Norbert.Gronau@wi.uni-potsdam.de

ABSTRACT:

The research project CHANGE[1] aims to bring adaptability into Enterprise Resource Planning (ERP) software systems. Adaptability is seen as a quality to manage change. This could be a reaction to a need or a proactive push to leverage potential opportunities. In any case the process change should be optimally represented in the ERP application. One of the major problems in developing adaptable software systems is the lack of systematic methods during the process of software development. For that, a pattern-based approach has been developed which covers three identified dimensions of adaptability in ERP systems. In the next step a component framework is proposed for characterising adaptable ERP software systems, regardless of venture type.

Keywords

Adaptability, ERP systems, Framework, pattern

INTRODUCTION AND MOTIVATION

Development of adaptable software has been receiving much attention recently; as such software could better accommodate changes in user requirements as well as in needs of the developing organization. The role of Information Technology (IT) in the business process change is seen as dominant or as an enabler. However, insufficient support to optimally model into software systems hinders the business to be carried out as planned. Work arounds, and encapsulation are common procedures during the lifetime of a software system to accommodate change.

A case is often made for the socio-technical design approach which suggests a mutual, bidirectional relationship between IT and the organisation (Gronau, 2003).

An increasingly dynamic environment and the ongoing trend towards customized products are heightening the requirements which companies have to meet leading to constantly changing structures and processes.

In dealing with the challenge of creating and evaluating adaptable software the contribution begins illustration of adaptability. Next, requirements for a proper evaluation of an adaptable software system are derived. For that, a pattern-based approach is chosen. Also, the adaptability focus has been narrowed to ERP Systems as fundamental part of enterprise architecture. ERP systems are highly standardized software systems and they link all or many business functions and operating locations so that all have access to relevant information (Nick, 2001). ERP systems may be appropriate for some organisations but less for others. ERP systems themselves are limited in the processes they can model. As a result, an organisation is limited to the collection of functions delivered with the ERP system or has to modify either the business processes or the ERP program code. Thus, currently no single ERP packaged software can meet all company functionalities or all special business requirements. Therefore, companies must choose an adaptive ERP system (Gattiker and Goodhue, 2002). And, adaptability of ERP systems is to leverage the overall adaptability of a business organisation in a turbulent environment (Andresen, Gronau and Schmid 2005).

ADAPTABILITY

Adaptability is a new research field, which has moved into the centre of interest in the 1990´s. Adaptability (sometimes also referred to as transformability) has been related to factory planning before penetrating into the domain of information systems. It is to be seen as a new design goal in factory planning. Design guidelines for modular and

[1] www.change-project.de

adaptable factories were created under different aspects (Nofen and Klussmann 2002, Westkämper, Zahn, Beise and Tielbein, 2002, Eversheim, Lange-Stanlinski and Redelstab, 2002, Wirth, Enderlein and Hildebrand, 2003). Research in factory planning has defined so called adaptability enabling objects. Among them, the information system architecture (ISA) of a company which is seen as an adaptability enabling object holding strategic importance as well (Herndandez, 2002).

Definitions on adaptability are not consistent. The Webster dictionary defines adaptability as the ability to fit a specific or new use or situation, often by modification (Webster, 1988). Following Balve/Wiendahl/Westkämper (Balve, Wiendahl and Westkämper, 2001) it can be said that the term adaptability is only applied if the system under consideration is able to:

- Actively and quickly adapt its structures to changing and unforeseeable tasks and
- Develop itself according to evolutionary principles within the framework of relatively constant demand.

Adaptability enables information systems to follow changes which occur along the lifetime of a software system. In this spirit, adaptability is also a vision. The vision that business support is carried out almost automatically (autonomous) during build and runtime of the software system. By raising the question to bring adaptability into software systems the first step was to find criteria which allow adaptability in software systems. For that, the pattern approach seems well suited for documenting design techniques. Unlike a design document, a pattern reflects something that has been used in a number of situations and thus has some generality. Finally, patterns can express architectural considerations independent of language and design methodology. The pattern-based approach is subject of the following section.

FACTORS AFFECTING ADAPTABILITY – A PATTERN-BASED APPROACH

We understand all there is for us to understand through repeated parts and portions. We live by recognizing and using recurrences, by relying on what happens over and over. What repeats is important and reveals commonality. The recognition of repeatable parts is the idea of using patterns. Patterns are used in several areas. Christopher Alexander coined the term pattern language and explained the form well in his work "The Timeless Way of Building" (Alexander, 1979). As for software systems identified patterns embody repetitions and recurrences within software. Alexander structured his patterns hierarchically into a system he called a "pattern language". The first book on software patterns was published by the "Gang of Four", the nickname for the authors of the first book on software patterns (Gamma, Helm, Hohnson and Vlissides, 1995).

By approaching the research quest to design for adaptability observation and analysis showed some building blocks deserving a strong focus to design and build for adaptable ERP software systems.

Groups of identified patterns comprise system patterns allowing a neutral adaptability analysis of any ERP system and use patterns which are context dependent. The use patterns or the business dimension characterises the circumstances of usage for an ERP system. They reflect that adaptability enabling factors are also related to decisions referring to the deployment of the system. In that area patterns are for instance the capabilities of personnel (person-bound) knowledge; existing guidelines to proper deploy a software system. For that reason the second group which is not further within the focus of this contribution is termed use patterns.

System Patterns describe the immanent qualities of the ERP system itself. Independently from surrounding conditions system patterns show the latent adaptability of the system. The identified patterns of adaptability in software systems are mentioned below.

1. Self-organisation

Self Organisation is a basic characteristic of natural systems that adapt automatically to changing conditions. Self organization (auto-poiesis) marks the ability of a system to determine the system structure by adjusting and steering mechanisms to ensure the long-term system existence (Maturana and Varela, 1987). This quality is a fundamental criterion which for instance requires the software system to behave efficient and self-adjusting. The system elements or rather subsystems produce thereby their own order by taking up information about their environment and its interaction with the environment; what consolidates a "model" upon which further action is based in the real world. Self-organization of ERP systems is closely linked with mechanisms of adaptation as customization for instance.

2. Scalability

Scalability refers to the permanent state to effectively and efficiently operate at many different scales. It is the best pattern for situations where large volumes of operational data must be written. The increased loads my be random and unpredictable, or planned out over time [RoSr02]. Referring to the software applications have to be conceived to increase equally with rising demands. An example is represented by the sliding usability of organizational levels for cost calculation, developed by an ERP system manufacturer. The system makes always as much levels as needed available (Gümbel, 2004). An ERP system is described as scalable if it will remain effective when there is a significant increase/decrease in the number of resources as for instance data in parts lists. Usually the capacity is fixed and resource allocation is not optimized.

3. Modularity

Modularity or rather modularization in general means the structuring of a system into small, partly-autonomous and clear subsystems (Picot, Reichwald and Wiegand, 2003). These parts represent the modules. A module consists of a module-trunk and a module-interface. The latter contains a specification about the performance and the qualities, which are necessary for its surroundings. The module-trunk implements the definition of the module, which is specified in the interface. Therefore single modules can be removed without much expenditure, replaced by others or added to another system. In this way the modularity presents a possibility for efficient combination, recycling and fast change of informational applications. As for an ERP system, modularity is closely linked with component-based architectures. Modules are implemented so as to hide all the information about them except what is available through its interfaces.

4. Mobility

Mobility raises the question on spatial and temporal unlimited access to applications of the information system architecture and therefore to its functions and data by different technologies - for example via web-browser, terminal-server or a virtual private network, by means of these applications data can be accessed. Some ERP systems do provide a limited access; some are even fully web-based (web-ERP+2). A second dimension represents the platform independency of applications. This freeness covers for example the used hardware, the operating system, data bases or application servers.

5. Interoperability

Interoperability describes the ability of applications to work together. Independence of the used hardware, the operating-system, the network-technology and the realized application, cooperation between these applications can easily be established. Interoperability allows the uncomplicated access to different (also spatial) data- and processing resources within a workflow or rather the easy combination of different information systems. For cooperation between enterprises, interoperability means increased communication- and cooperation abilities. This indicator refers to the ability of system elements to make a high measure of compatibility available. Interoperability requires the use of well-established standards that define the behaviour of interfaces. It allows the uncomplicated access and coupling of different data- and processing resources within a workflow or rather the easy coupling of different ERP and information systems.

6. Self-Similarity

Design Patterns are traditionally related to architecture. Transferring this approach to information technology one fundamental pattern could be identified: Self-similarity can be considered as key feature being part of all mentioned system-based patterns as it is related to the inner structure of each system pattern. Self-similarity is a pattern repeating itself on different scales no matter the selected degree of abstraction. The self similarity is a phenomenon owned by many natural objects (clouds, plants, mountains etc.). In different size scales the same essential structures appear. Also, many chaotic systems show self similar behaviour. As an example of the advantages of self similarity a unique design philosophy of applications shall be mentioned resulting in the easier ability to learn and efficiently use the application on different platforms and levels.

DISCUSSION: ADAPTABILITY ANALYSIS OF WEB SERVICES

In this section an example for an adaptability analysis of web services is presented. Web services facilitate the integration of services. Web services are located at the application and service layer in the reference model. In the literature Web services are seen as new paradigm giving freeness to companies to create and reconfigure organisational competencies to sustain competitive advantages.

Scalability:

Web services facilitate the Integration of services seamlessly. Web services do allow organisations to link applications within and across enterprises. Organisations have the ability to add or drop services of other business partners without worrying about the implementation details.

Modularity:

[2] Available online: http://web-erp.sourceforge.net

Mobility: Web services and mobile services. Web services are loosely coupled software components delivered over Internet standard technologies. A Web service represents a business function or a business service and can be accessed by another application. From the point of view of the calling application a Web service is modular. The Web service is a modular entity that delivers services on demand through a well-defined programmatic interface.

Mobility:
On the one hand mobility is concerned with (data) access anywhere anytime. Another aspect of mobility is the independence of software and hardware. Since web services use the internet technology to become invoked and deliver their service they can be published, located and invoked from just about anywhere on the Internet or a local network. The provider and the consumer a Web service do not have to worry about operating system, language, environment, or component model used to create or access the Web service, as they are based on ubiquitous and open Internet standards, such as XML, HTTP,

Self-Organisation:
Web services are able to adapt to changes in its requirement and they are self-descriptive. Some features have been automated within Web services what makes them self-organising. The approach represents a major evolution in how systems connect and interact with each other. However, Web service is a concept – an envelop which can be deployed for a specific purpose. The collection of operations is individual as is the design and implementation of self-organisation which is determined by the programmer.

Self-Similarity:
The design principle of Web services is similar. The apply the same principles which means Web service is an interface that describes a collection of operations that are network-accessible through standardized XML messaging.

Web Services are actively supported by major IT companies as IBM, Microsoft, and Sun Microsystems. They represent a new IT application development paradigm. Based on XML, the different elements of the technology allow the integration of heterogeneous applications within and across organisations. With web services technology, business processes can span across departments, divisions and enterprise boundaries, allowing firms to integrate the services of multiple applications without concerning about the underlying technologies and implementation characteristics

CHANGE COMPONENT FRAMEWORK - AT A GLANCE

In order to apply the identified patterns on an ERP system we have modelled a layered systems structure. In seeking generalisability the extant perspectives tend to simplify a (real) ERP system into classes of services assigned to layers. The challenge is to produce a framework which serves the needs of the individual systems/system deployment. The illustration is shown in figure 1. The architectural model represents two layered ERP systems. Connectors mediate the interactions within the ERP system (numbers 1 - 4) and they govern component interaction outside the ERP system (numbers 5 – 9). The basic purpose of the framework is to allow for appropriate checking and analysis of ERP systems. For that, the layers provide the context of the patterns.

Accordingly, a framework is proposed that consists of specific levels of functionalities termed the “control”, “presentation”, “adaptation”, “application”, “service”, “data” and “infrastructure” layers. The need for levels, which are the result of iterative examination, reflects the different technological purposes of the model.

There is, at the control layer, a need to model business processes into the ERP system and to ensure that decisions are internally consistent. The control layer provides some sort of modelling language as for instance to be found within the ARIS suite of the SAP. At the presentation level the models purpose is to enable communication with the user as the user interface is located here. The application level is needed to structure and represent the software components. Part of this layer is the service layer which serves the handling of resources and transactions. The data area covers the data management. At the infrastructure level the decision is addressed how the system is distributed, what topology is chosen? An additional perspective approaches adaptability. Purpose of the adaptation layer is to cover decisions related to system-based patterns as mentioned before.

The usefulness of any model is limited, however, unless it provides specific guidance and discipline to operations. For that, technologies and standards where examined using the pattern-based approach. Thus, each layer was assigned one or more design considerations as standards, protocols supporting the service task. For instance, the control layer should allow acting on data modelled as business process into the system. On the one hand there is a need to represent the data what might be realized by protocols as ebXML or BPMI the Business Process Management Initiative. On the other hand, methods are needed to pull out relevant data of the control layer. The latter task is performed at the connector number 1 to mediate data within one single system, and also the connector 5 to allow external data exchange.

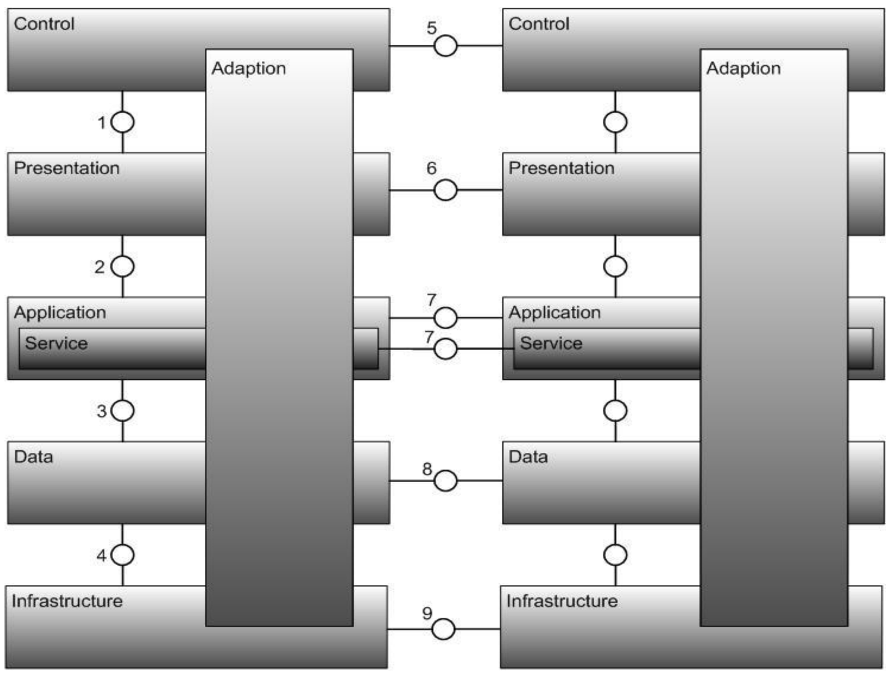

Figure 1: Adaptability Reference Model for ERP Systems

The matrix below (table 1) lists the given marks per technology providing first insights on adaptability measures. To classify the assessment in terms of adaptability a limited number of codes are applied which are shown in table 2. The usage of five degrees is the essence of several iterations. They have been proven practicable enough for us to classify technology and systems without implementing too much complexity.

The Codes taken:
„+“ full support for pattern (enables adaptability);
“-“ no support for pattern (breaks adaptability);
“?” weak support for pattern – some constraints exist

Table 1: Codes applied for evaluation

Pattern / Layer	Control	
Tecnology	BPMI	ebXML
Scalability	?	+
Modularity	+	+
Mobility	+	+
Interoperability	+	+
Self-organisation		+

Table 2: Example shows system-based patterns for Control Layer (cut-out)

A validation of layers and connectors deploying the patterns of adaptability has been carried out for a number of technologies per layer providing a foundation for the neutral adaptability analysis. The results reported are being developed jointly within the CHANGE project and they have been catalogued within a "Reference Guide" which is frequently under revision as technology gets evaluated. For a neutral ERP system analysis, each layer and its communication links of the framework is to be matched with the relevant ERP software architecture. The evaluation of each layer contributes to the over-all rating. The more adaptable the technique the better the system is evaluated.

NEXT STEPS

So far current technologies which are used in software systems and ERP systems respectively have been catalogued in terms of adaptability. The latter has been carried out be applying patterns as an artefact of choice to indicate the systems adaptability. As pointed out two dimensions of adaptability have to be considered on the one hand the neutral system evaluation and on the other the business dimension. The case for a neutral system evaluation with regard of adaptability is the logical next step. To demonstrate the usage of the framework four open-source ERP systems have been chosen, namely AvERP, Compiere, Lx-Office and webERP+. All systems mentioned had been installed at the ERP research Center attached to the the University of Potsdam for in depth studies.

CONCLUSIONS

In this contribution a framework was presented, which helps to design, describe, categorize and analyse ERP systems with regard of adaptability. Adaptability as a research domain in information science shall enable business organisations to deploy adaptable software systems which support business processes during change and modifications.

Future research in the field of adaptable ERP systems faces a couple of challenges which are addressed within the project CHANGE. Efforts have to be made to put the presented ideas into applicable tool-procedure models. Efforts have to be made to put the presented ideas into applicable tool-based procedure model. This involves also a further refined break-down of patterns and a procedure to weight single patterns. Another challenge involves the design of adaptable ERP systems leading to transfer of results to the area of software engineering. For this purpose and beyond the Centre for ERP systems was recently founded at the University of Potsdam.

REFERENCES

1. Gronau, N. (2003) Adaptable Architectures of Information Systems– Sustainability during organizational change (in German). Gito-Verlag, Berlin.
2. Nickerson, R., C. (2001) Business and Information Systems. Upper Saddle River, New Jersey.
3. Gattiker,, T., F., Goodhue, D., L. (2004) Understanding the local-level costs and benefits of ERP through organizational information processing theory, *Information & Management*, 41, 4, 431-443.
4. Andresen, K.; Gronau, N.; Schmid, S. (2005) Ableitung von IT-Strategien durch Bestimmung der notwendigen Wandlungsfähigkeit von Informationssystemarchitekturen, In: Ferstl, O.K./Sinz, E.J./Eckert, S./Isselhorst, T. (Ed.), *Wirtschaftsinformatik 2005*. Physica-Verlag, Heidelberg. (in German)
5. Nofen, D., Klußmann, J (2003) Adaptability by modular company structures, *Industrie Management* 18, 3, 49-52.
6. Westkämper, E., Zahn, E.; Balve, P.; Tilebein, Meike: (2004) Ansätze zur Wandlungsfähigkeit von Produktionsunternehmen. Ein Bezugsrahmen für die Unternehmensentwicklung im turbulenten Umfeld, *wt: Werkstattstechnik*, 90, 1/2, Düsseldorf, 22-26.
7. Eversheim, W., Lange-Stalinski, T., Redelstab, P. (2002) Wandlungsfähigkeit durch mobile Fabriken, *Werkstattstechnik online*, 92, 4, Düsseldorf, 2002, 169-170.
8. Wirth, S., Enderlein, H., Hildebrand, T.(2004) Plug and Produce. Ein Modularkonzept zur effizienten Gestaltung wandelbarer Produktions- und Organisationsstrukturen durch Vernetzung, *Industrie Management* 17, 5, 67-70.
9. Hernández, R.(2002) Systematik der Wandlungsfähigkeit in der Fabrikplanung. *Fortschritt-Berichte VDI*, 16, 149. Düsseldorf.
10. Webster's Ninth New Collegiate Dictionary.(1988) Springfiield, MA: Merriam-Webster,
11. Balve, P., Wiendahl, H.-H., Westkäpfer, E. (2001) Order management in transformable business structures—basics and concepts, *Robotics and Computer-Integrated Manufacturing*, 17, 6, 461-468.

12. Alexander, C.(1979) The Timeless Way of Building. New York: Oxford University Press.
13. Gamma, E., Helm, R., Hohnson, R., Vlissides, J. (1995) Desing Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison-Wesley.
14. Maturana, H. (1987) Varela, F.: The Tree of Knowledge. Boston.
15. Robertson, B., Sribar, V. (2001) The adaptive enterprise, IT infrastructure strategies to manage change and enable Growth, Pearson.
16. Gümbel, H.: Semiramis (2004) The Native Next-Generation Business Software, A White Paper.
17. Picot, A., Reichwald, R., Wiegand, R. T. (2003) Die Grenzenlose Unternehmung - Information, Organisation und Management; Wiesbaden.

.