Extending Lifecycle of Legacy Systems – An Approach for SME to Enhance Their Supported Business Processes through a Service-Integration-System

Florian Gruner and Stephan Kassel

Westsächsische Hochschule Zwickau, Institute for Management and Information
PSF 201037, 08012 Zwickau, Germany
{Florian.Gruner, Stephan.Kassel}@fh-zwickau.de

Abstract. Modern enterprises are supported by flexible and agile software systems and architectures to respond effectively and quickly to opportunities in a more competitive market. Even small and medium sized enterprises (SME) highly depend on appropriate software architectures to successfully participate in the growth of the economy. For this purpose they must be in a position to upgrade and enhance their ERP at low costs and with small manual efforts. A flexible middleware platform for the integration of future software systems is proposed as a solution to these challenges for innovation.

Keywords: Enterprise interoperability, SME, flexibility, agility, Service integration, software architecture, Enterprise Service Bus

1 Introduction

Small and medium sized enterprises are the largest employers in the European Union and in Germany (66,9 % of all jobs in the EU, 60% in Germany) [11] [12]. They have an important role for the gross value creation (58,4 % EU, 46% Germany) [11] [12]. These two facts are demonstrating the economical importance of SMEs for all countries of the European Union. But there are some difficulties for SMEs in handling complex problems of entrepreneurial development and cross-company collaboration. Only a small number of enterprises in this group have the financial and personal background to develop the needed instruments and knowledge to react to the changing business in times of financial and economic crises. It is important to utilize the knowledge of different independent business partners. These partners are providing best of class business processes, which have to be integrated into the processes of the SME.

Based on the high degree of dependence of enterprises from their information systems and the provided functionalities and business processes, we propose as our research issue "Is software interoperability a feasible way for SMEs to increase their flexibility in providing prime business functionalities?" From a technical perspective this research question seems to be solved (by using technologies like SOA or concepts like model-driven interoperability). But most of the proposed approaches need massive manual and financial efforts, which cannot be handled by SMEs [13] [14]

[15] [16] [17]. Taking these concerns into account, we try to find out, how these technological approaches can be tailored to SMEs to comply with the economical restrictions. To answer our main research question, we set five related hypotheses;

- There is a correlation between success and flexibility of SME
- Intense customizing is no option for SME to support business processes
- SME are depending on using different information systems
- A flexible integration system is able to handle different standard interfaces
- A service integration system can be designed for the needs of SME

The first four questions are answered by analyzing statistics, market figures and modeling use cases. To support the last hypothesis, there is a collaboration of the University of Applied Sciences in Zwickau and a regional service company for bakery crafts (especially bakeries and confectioners), named BÄKO Ost eG (BÄKO).

BÄKO as a medium sized trading company is primarily interested in enhancing their relationship to nearly 700 regionally based crafts. Caused by ongoing structural changes in the market BÄKO has to develop their business focus from a conventional trading company to a full service provider to sustain their own success and even the success of affiliated bakeries. But this transformation is not supported by their legacy system (ERP-System). There exist no standard functionalities to make this system interoperable with other information systems at low costs and low personal interventions. In summary, the change of the business model is causing a gap between new requirements and the provided functionalities of the ERP system. In consequence a solution is needed, allowing to integrate new business processes and the supporting IT systems to fulfill daily needs. An integration system should manage transformation and transportation tasks between integrated applications. It should further simplify the exchange of any application in order to meet changing market requirements.

2 Contribution to Value Vreation

This work is a first step in researching, how interoperability could be transferred into practical applications for the demands of SMEs. According to this objective, this paper is proposing some first steps for SME to achieve more flexibility, taking into consideration the special restrictions, as financial and personal resources. A first prototypical Service-Integration-System (SIS) has been built to connect a customer service supporting software system with the ERP-System. This work is prototypically showing, how such an integration system could be built and which further questions and problems need to be solved. This paper focuses on developing an integration scenario for software modules without programming direct interfaces to the legacy system and between several modules. Each integrated system as an information providing and needing system may be implemented independently as a standard application with a limited number of processes. The goal is that none of the applications is disturbing any other application with its business processes.

Considering the objective of flexibility of services for the future it is not helpful to implement a point-to-point integration, leading to an exponentially growing financial and manual effort with each further integration of software modules providing additional services. It seems more efficient to utilize a specialized middleware-system realizing the process integration. Thus, SIS serves as an enabler for a smooth growth of the application base of BÄKO.

The middleware should fulfill three main objectives; (i) it should support the flexibility and agility of BÄKO, by providing an easy mechanism to exchange software modules at low costs and minimal manual efforts, (ii) it should avoid the danger of a big bang scenario by exchanging major functionalities of the legacy system step by step if additional requirements have to be met (e.g. warehouse management, inventory management or even financial management) and (iii) it should enable the IT managers to follow the best-of-breed strategy, by selecting applications being excellent in their own specialties. To fulfill these objectives, the independence of each integrated software system has to be guaranteed.

3 Fundamental Approaches for Software Interoperability

In a technical sense interoperability is the ability of independent software systems or components to work seamlessly together to share and exchange business data across at least two involved systems in order to use the information that has been exchanged or shared in a defined manner without requiring special effort from the user in form of manual interventions [2][3]. This very generic technical definition requires further definitions of standards and solutions for collecting, exchanging, processing and transporting data [3], which should be converted into technical approaches. One powerful approach to achieve interoperability in enterprises is represented by Service Oriented Architectures (SOA) and a possible technical implementation in form of the Enterprise Service Bus (ESB). These two approaches can be used as models for SME to build appropriate middleware platforms.

In almost every market segment there is a loose change from hard-wired value chains, which were backed-up by long-running master agreements and stable production lines, towards flexible Business Value Networks [7]. This structural change first showed up and affected manufacturing companies, but trading and service companies are more and more affected, because the number of competitors is increasing due to globalization. For further success enterprises have to enhance their core business by developing product-supporting services. This combination of products and services has to be supported by the enterprise information systems to enable the variety and complexity of modern entrepreneurial tasks and duties. Service Oriented Architectures, as a technological solution for automating business integration, address the need of modern enterprises to respond effectively and quickly to opportunities in a more competitive market [4]. IBM defines SOA as »a business-centric IT architectural approach that supports integrating your business as linked, repeatable business tasks, or services [5]. The SOA definition of IBM argues, that a technical solution should focus on the economic concerns, resulting in a combination of both worlds in order to create a flexible and agile enterprise information architecture. Therefor this philosophy of infrastructure design is independent of any specific technology (e.g. Web Services or J2EE enterprise beans), by limiting the level of implementation restrictions to the level of the service interface. The transformation of the IT department to a business enabler leads to a growing and innovative enterprise. To support an enterprise with such useful advantages some prerequisites must be met – encapsulation of conventional software resources into *services*, »which are well defined [and] self-contained modules« [4]. Such self-contained services are independent of the state or context of other services in order to provide standard business functionalities [4]. Services are the basic technology to establish a Service Oriented Architecture, where they »communicate with each other requesting execution of their operation in order to collectively support a common business task or process« [4]. To realize such enterprise information architecture contemporary services have to utilize different Web service standards, such as Web Service Description Language (WSDL), Simple Object Access Protocol (SOAP) and Universal Description, Discovery and Integration registry (UDDI). One of the main objectives of SOA is to overcome the challenges of distributed enterprise computing, including application integration, transaction management and protocols and leveraging numerous access devices and legacy systems, so that different applications can be integrated easily and run seamlessly [4]. These objectives are achievable by defining coarse-grained services or single application components out of monolithic software packages, which are easier to manage. The main advantage of these services is their ability to be combined, orchestrated and reused. Furthermore they offer their functionalities as standardized Web Services in an expandable repository [7]. These typical properties of services allow facilitating ongoing and changing needs of entrepreneurial tasks, by a flexible and dynamic composition. Consequently companies are enabled to implement business improvements more quickly, while existing business processes will not be affected. Terzidis et al are providing a good summary of advantages of SOA [7] and Greiner et al are offering a overview of the essential properties of services in SOA as exposed pieces of functionalities [3].

These properties enable Service Oriented Architectures to provide users with a flexible architecture that unifies business processes by modularizing large applications into single services. This encapsulation of services out of applications is the key method to create a collection of services that are able to communicate with each other using standardized interfaces to pass messages from one service to another. By choreographing services to a unique composite of single services it is possible to create agile business processes, which are able to fit to changing needs.

SOA is as well a technology to enable enterprises to replace monolithic and grown IT architectures as well as application silos (self-contained applications) [9] [10] by composing and orchestrating functional components from a variety of services provided by different systems. Furthermore, an expansion and development of business activities inevitably requires an expansion of the functionalities of the information system, by installing new services or software modules. These requirements necessitate the implementation of middleware technology, which realizes data transaction and transforming by bridging different protocols, interfaces and platforms. These requirements of bridging different technologies will be met by the Enterprise Service Bus as an appropriate technical integration infrastructure. Furthermore it enables IT managers to provide access to distributed information across different IT systems and departments. The ESB focuses on the dataflow and provides basic functionalities for involved services and components of a SOA to create an IT infrastructure according to the current requirements of an enterprise. The concept allows crossing boundaries of interfaces, protocols and even different eras of technology development. The independence of used technology is one of the main advantages of ESB compared to the approach of Enterprise Application Integration (EAI) infrastructures. EAI is a point-to-point topology [4] and requires for each connection between every single system a separately developed interface. This is hard to manage and maintain (tighter form of system coupling) [4]. Consequently there are lots of difficulties if a system has to be exchanged, because this would have direct effects on the stability of the overall information system due to the hardly manageable complexity. With the implementation of ESB an integration layer is introduced to support easier interoperability among integrated systems and components. This integration layer allows a coexistence of different kinds of application platforms and directly follows the requirements on a capable and manageable integration infrastructure. The development of ESB is the basis for one of SOAs main properties. Loose coupling of systems taking part in an integration scenario is leading to an easily manageable enterprise information system. The ESB serves as the backbone of every SOA, because SOA properties and functionalities highly depend on its ability to support any data and information management task. Therefor, the ESB offers a number of functionalities going far beyond a pure mediation of data, by providing functionalities like services to be independently distributed and implemented (transportation, routing, transformation, adapter and business process services).

4 SIS Allows Combining New Software with Legacy Systems

Considering the business objectives of BÄKO IT managers should be enabled to follow the principles of the best-of-breed strategy by choosing specialized software systems in order to fulfill the requirements of the involved business departments in charge of their business processes. In a technical view it is easier for a SME to abstract from the SOA philosophy and its focus on coarse-grained services to complete applications, offering needed business processes. This property of the developed prototypical integration system leads to the possibility that the decision for a software system is no longer concentrating on provided interfaces, exchange file formats, programming languages and the question of in-house installation or on-demand SAAS-offers. The criteria catalogue for the decision towards a specialized software system has to be focusing on one single question – the capability to fulfill the business requirements. This priorization is chosen to simplify the selection process, but mainly to increase the degree of flexibility and agility.

For such a paradigm shift in the selection of specialized software systems, BÄKO requires an infrastructure to be enabled to choose software systems regardless of their system requirements and system properties. Without a suitable integration technology and appropriate management of the services this procedure would lead to a hardly manageable heterogeneous and complex IT environment with uncertainties regarding the exchange of data or uncontrollable financial efforts for system maintenance. Following the ideas of SOA and ESB the Service-Integration-System (SIS) was developed to provide BÄKO with a flexible, agile and business process oriented information system, where different software systems can be integrated and exchanged. The existing legacy system is interfaced to the SIS, being the inventory control system and the basis for all financial data and customer related data.

In cooperation with AGETO Holding AG, the Service-Integration-System (SIS) was developed to simplify the efforts of interoperating with further software modules of different technical characteristics, such as interfaces, file format or programming languages. The SIS serves as an independent and intelligent integration platform with interfaces to the existing ERP and future service systems to be installed, comparable with an Enterprise Service Bus, illustrated in Fig.1. In one regard, ESB and SIS are similar because both bridge the massive disadvantages of conventional point-to-point connections by introducing a hub-and-spoke architecture, which reduces the number of interfaces of every service providing module to only one interface to the SIS. This is the basis for an easier integration of new software systems, leading to a massive reduction of the efforts to code the interfaces and provide the data mappings. The SIS is acting as a mediator in a heterogeneous application environment, which will grow in future according to the entrepreneurial changes of the business of BÄKO. For this purpose the SIS provides a variety of functionalities for the intelligent management of interfaces for business process management and data management.

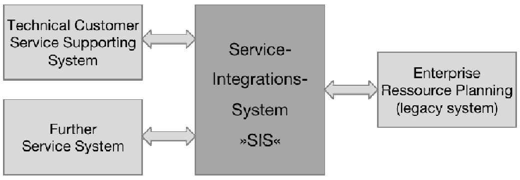
Technical Customer Service Supporting System
Further Service System
Service-Integrations-System »SIS«
Enterprise Ressource Planning (legacy system)

Fig. 1. Service-Integration-System

The main difference between SIS and SOA/ESB is the possible integration of independent standardized software modules, supporting a department with integral standardized business processes. Any aspect of entrepreneurial tasks may be covered by self-contained and specialized software modules, which can be intelligently connected and integrated with each other. That means basically that the mainly technical approach of SOA and its implementation with ESB is further simplified by changing the focus from primarily encapsulated, self-contained and fine-grained services, which can be freely combined to generate any possible business processes to integrated service-providing applications which are combined on a higher level. SOA is very useful to connect encapsulated services out of existing applications in an enterprise to create new business processes. But this requires massive efforts and know-how, which is not manageable for SME.

The approach of the SIS lies in the orchestrating of complex sub-processes, which can be performed by using the specialized services only. Thus, the overall business process has to be carried out by using several IT systems, but most steps can be done using the standard functionality of the specialized modules without company-specific programming. The aim is to develop an integrated enterprise information system out of (nearly) uncustomized standard software systems, thus reducing costs and risks for the SME resulting from extensive customizations. On the other hand, it reduces the danger of implementing software modules as isolated applications without any connections to surrounding systems. Avoiding this, the SIS will provide supporting tasks of data management, such as transportation and transformation, between the affected components. Therefor, IT managers have to define the processes to ensure data exchange to supply each single system with a consistent database. Especially the fact of data management will be interesting for SME. In the first stage, a module pushes data into the SIS, which should be transferred to target systems. The SIS is additionally storing these data in order to just push the changed data to the target system. Beside that, this database will be used to build up a cross-module data warehouse later.

Furthermore, the IT managers have to define the management processes for the independence of different sets of data. Every single system will be in charge for their own set of information and data required for the supported business processes. Thus the ERP system still has an important role in this architecture. Major data, such as financial data and customer related data are mainly managed in this system. Without a consistent database of these information a sound enterprise information system cannot be guaranteed. But other kind of data, such as technical customer service related data (customer machine data, customer service contracts) are managed by the corresponding software systems, reducing the need to change the ERP system in order to be able to handle these kinds of data as well.

In our case study an ongoing exchange has be illustrated between the ERP and the technical customer service supporting system (service system). This could not be done without focusing on the required data mapping processes. Because of the leading role of the ERP system for the customer data debtors data (customer data), creditor data and machine or rather spare part data has been exported from the ERP system to the service system. Because of the importance of these data they can only be updated in the ERP system. From the service system service reports are exported, being the base for accounting. In the next step, there will be a real-time query of spare parts from the service system to the ERP system to give an actual overview of parts in stock, in order to provide a better technical service to the customer.

5 Conclusions

Beside SOA, a Service Integration System (SIS) could be a useful approach especially for SME to gain more flexibility of their enterprise information infrastructure. The SIS is based on a hub-and-spoke topology that simplifies the exchange of modules according to changing necessities. This is quite similar to the SOA approach, but the SIS reduces the complexity of process integration to the level of standardized software modules. SOA is based on self-contained, encapsulated and fine-grained services, which makes it difficult to implement SOA in common SME, because most of them have insufficient financial liquidity, know-how and time.

The implementation of the SIS is a first step for SME to develop an appropriate level of interoperability between different software modules. This is achieved by the reduction of interfaces to be implemented to just one interface to the centralized middleware system. According to the paradigm of SOA it is possible to integrate application-oriented software solutions with standardized processes for the internal workflow. This will be achieved by expanding the actual SIS implementation according to the needs of the SME who should be able to control the flow of the business processes. Lastly, the SIS increases the reliability and stability of the system infrastructure, because system updates, suddenly occurring problems within one service system or an exchange of a system doesn't show effects to other systems.

References

1. Mo, T., Xu, X., Wang, Z.: A Service Behavior Model for Description of Co-Production Feature of Services. In: Mertins, K., Ruggaber, R., Poppelwell, K., Xu, X.: Enterprise Interoperability III. pp. 247-260. Springer-Verlag, London (2008)
2. Chungoora, N., Young, R.I.M.: Semantic Interoperability Requirements for Manufacturing Knowledge Sharing. In: [1]. pp. 411—422.
3. Greiner, U., Legner, C., Lippe, S., Wende, K.: Business Interoperability Profiles: Relating Business Interoperability Issues To Technical Interoperability Solutions. In: Gonçalves, R.J., Müller, J.P., Mertins, K., Zelm, M.: Enterprise Interoperability II. pp. 865—877. Springer-Verlag, London (2007)
4. Papazoglou, M. P., Heuvel, W.-J.: Service oriented architectures: approaches, technologies and research issues. In: The VLDB Journal. Vol. 16, Nr. 3. pp. 389—415 (2007)
5. IBM, http://www-01.ibm.com/software/solutions/soa/
6. Usman, Z., Young, R.I.M., Case, K., Harding, J.A.: A Manufacturing Foundation Ontology for Product Life Cycle Interoperability. In: Poppelwell, K., Harding, J., Poler, R., Chalmeta, R.: Enterprise Interoperability IV. pp. 147-155. Springer-Verlag. London (2010)
7. Terzidis, O., Sure, Y., Brelage, C.: SOA – Flexibility and Agility. In: Wirtschaftsinformatik, Vol. 50, Nr. 1. p. 76 (2008)
8. Touzi, J., Lorré, J.-P., Bénaben, F., Pingaud, H.: Interoperability through Model-based Generation: The Case of the Collaborative Information System. In: Doumeingts, G., Müller, J., Morel, G., Vallespir, B.: Enterprise Interoperability. pp. 407—416. Springer-Verlag. London (2007)
9. Billeb, M., Schäfer, A., Abbou, M., Niemann, M., Eckert, J., Repp, N., Steinmetz, R.: Einfluss regulatorischer Anforderungen im Internet of Services. In: IT-Governance, Vol. 3, Nr. 6. pp. 8—13 (2009)
10. Abramowicz, W.: Serviceorientierte Architekturen und Webservices. In: Wirtschaftsinformatik. Vol. 50, Nr. 1. pp. 1-3 (2008)
11. Wymenga, P.; Spanikova, V., Derbyshire, J., Barker, A.: Are EU SMEs recovering from the crisis? Annual Report on EU SMEs 2010/2011. 2011, p. 8
12. Statistisches Bundesamt Deutschland, http://www.destatis.de/jetspeed/portal/cms/Sites/destatis/Internet/DE/Content/Publikationen/STATmagazin/UnternehmenGewerbeInsolvenzen/2008__8/2008__8KMU,templateId=renderPrint.psml#Link3
13. Girard, P., Doumeingts, G.: GRAI-Engineering: a method to model, design and run engineering design departments, International Journal of Computer Integrated Manufacturing, Vol. 17 Issue 8, 2004, pp.716-732.
14. Blanc, S., Ducq, Y., Vallespir, B.: Evolution management towards interoperable supply chains using performance measurement, Computers in Industry, Vol. 58 Issue 7, 2007, pp.720-732.
15. Mertins, K.; Knothe, T., Jäkel, F.-W.: Interoperability – Network Systems for SMEs. In: [1], pp. 511-520.
16. Protogeros, N., Tektonidis, D., Mavridis, A., Wills, C., Koumpis, A.: FUSE: A Framework to Support Services Unified Process. In [1], pp. 209-220.
17. Xu, X.F., Mo, T., Wang, Z.J.: SMDA: A Service Model Driven Architecture. In [3], pp. 291-302.