Design of Services as Interoperable Systems - An E-Commerce Case Study

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

Abstract

In this paper, some foundations of a decision support model for a full-service e-commerce provider, providing a SAAS (software as a service) business model, are presented. The decision model is targeted to explicitly address interoperability issues, to provide a simple but powerful communication aid for the negotiations between service provider and customer, and to aim in automatically composing reliable software systems from service components.

1. Introduction

In the last years, e-commerce vendors changed their business model, from the original selling of technology like e-commerce systems and corresponding integration technology to the company’s legacy systems, to more complex technological services, like integration with public market places [1], or even business services, like warehousing, e-marketing, or invoicing.

This has led to more complex e-commerce systems with a richer set of features, but hand in hand with the need of enhanced customizing services for e-commerce platforms. In this increasingly difficult business, leading e-commerce vendors have adopted to customer’s need by two very different strategies.

One strategy lies in providing even more business functionality, leading to so called “full service e-commerce”. The e-commerce company not only provides software for selling goods on the world-wide web, but enhances this software with additional services. This is possible, because there are a restricted number of services, which are widely recognizes as important by most players on the e-commerce market. These services have a high degree of modularity, leading to an encapsulation metaphor, this defining individually usable business functionality. The processes, which have to be defined in an e-commerce application, can be synthesized bottom-up from the services, which should be integrated into the overall solution. Common examples for these kinds of services are

- hosting services, consisting of providing hardware systems for e-commerce, together with server rooms, internet connections etc.,
- inventory management, providing warehousing, storage management, packaging, and delivery of goods,
- online marketing, providing e-mail campaigns, individual marketing programs, affiliate programs, etc.,
- search engine optimization, consisting of keyword management, special marketing campaigns, link placement, etc.
- multimedia presentation of products,
- payment services and loyalty programs, etc..

The second strategy to deliver added value to the customers, lies in the provisioning of software as a service (SaaS). This means, that the e-commerce systems are capable of utilizing standards interfaces to be able to call and provide services, which adopt to the service-oriented architecture (SOA). Via webservices mechanisms, centralized “yellow pages” for service registration, and flexible provision / calls of services, the e-commerce systems are evolving to an integration platform for complex business processes. Thus, business process management (BPM) and e-Commerce are coming together to provide more complex services for the end customers of the e-commerce system, and to integrate the business of the end customers with the business of the company owning (or leasing) the e-commerce system.

Both strategies of e-commerce vendors lead to a higher demand for explicit interoperation between platform owner and customer, to seamlessly integrate the service processes, and to commonly define the needed level of interoperability. The business processes have to be crisply defined, at least those processes, which are directly part of the interoperation.
2. The case study

2.1. Business area of the industrial partner

The industrial partner of the project, which is now in the definition phase, is a medium-sized Canadian enterprise with a German subsidiary. The main business of the German branch of the company lies in integration and interoperation services for customers, who want to gain expertise in e-commerce. The founders of the German business have a strong background in e-commerce. The integration business of the company comprises of selling goods via different internet channels (dedicated shop systems as well as public marketplaces), by integration of the customer’s warehouse or ERP systems and defining the interoperation processes for the e-commerce component. Thus, the industrial partner provides some scalable e-commerce services, starting with just a technical integration, but reaching up to a complete set of business definition and interoperation services.

This kind of business demands for a dynamic pricing model, which has to take into account the complexity of the service, the number of services provided, the amount of business delivered on the e-commerce channel, the dedication of hardware and software to the special customer, and other business related information.

Although the pricing model is very complex, there is no support in finding the right price for a customer till now. The pricing is done in an iterative way with several sessions between e-commerce provider and the customer, who wants to use the e-commerce services.

2.2. A model for the pricing of services

The ad-hoc pricing of the services delivered by the e-commerce provider holds some serious business risks:

- the components are not priced well,
- the needed and paid performance of the e-commerce system for the customer has to be estimated very roughly,
- the pricing model is not very transparent, so the customer cannot gain trust into the service provider, and
- if there are any problems in the delivery of the service, the re-negotiation is always a negotiation of the complete set of services, because they cannot be separated at all.

There is a need for some delta pricing models, because most of the service constituents are already pre-built, they only have to be re-arranged and customized to the special customer.

So the research, which has to be done, consists of building a specialized decision model, which can be used in the early phases of a customization project. This decision model has to take into account that the software which has to be built, mainly consists of pre-built business components which are already suited to provide a rich set of services to the customer. These services definitely have to be tailored, but the variation of the services normally are not too numerous.

3. State of the art

There are a lot of research domains, which are touched by the task of the case study. So, the overview of the state of the art of these areas has to be very short, and only a few fundamental work is considered, which could be helpful in providing ideas and theories for the solution of the problem.

3.1. Decision support systems

Decision support systems are delivering a support in making semistructured or unstructured decisions. To achieve this, they consist of a flexible tool set with analyzing capabilities. There is a number of well-known systems, falling mainly into two classes [2].

One class of systems is model-driven. They are standalone systems, which are specifically designed for their tasks. They have a deep model of the overall problem, and they are most of the times standalone systems. Most of these systems are defined with artificial intelligence methods. Expert systems are a well-known example of this class if systems. The analysis capability of these systems is built on strong theories and models, combined with user interfaces designed for end users. The disadvantage of this system class lies in the complexity of the models. Building an explicit model with a strong theory is tedious, error-prone and often the users are not capable of building and maintaining these systems.

The second class of systems is data driven. Their purpose lies in analyzing large pools of data. These data are compiled into large data warehouses. The analyzing capabilities of these systems are based on flexible ways to view and combine mass data to analyse long-term trends. Additionally, the analysis of the data can be partly automated via data mining. This system class is very useful for strategic decisions and overall steering of a company, but it is not detailed enough to provide support for decisions like the
pricing of a customer-specific solution. (This is the disadvantage of lacking an explicit user model.)

Nowadays, there is a trend from decision theories and systems to decision aiding methodologies resulting in decision aiding systems [3]. In systems following this trend, not the decision itself is done by the system, but the system is assisting in making rational decisions. The decisions of the case study, which is presented here, has a simple, but powerful model. This model is based on some common methodologies widely used in economics and business. This kind of aiding system is needed to provide a solution to the issue of the pricing services. These issues are:

- to explicitly address the different levels from business processes to the programming level,
- to provide a negotiation aid for the business level,
- to directly connect business functionality and feature sets with pricing of the solution,
- to explicitly include business interoperability in the model of the business functionality,
- to provide an integrated view of the problem, leading to a consistent behaviour suited to the needs of the customer,
- to model user requirements as a mediation process,
- to automatically compose service components as well as reliable software systems, and
- to explicitly address interoperability.

3.2. Service interoperability

The area of service interoperability is not very well defined. Although there is a wide set of literature in the area of interoperability (e.g. [4], [5], [6], or [7]), the majority of the research still concentrates on the technical levels of interoperability. Even the roadmap of interoperability research of the European Union [8] still has an emphasis on technical interoperability levels, although the importance for defining the business level of interoperability is explicitly stated. The term of service interoperability is hardly defined, but we clearly have to differentiate between the (very technical) definition of service-oriented architectures (SOA), which only comprises of low-level, automated more technical-oriented services, and the business-oriented terms of service industries, comprising of immaterial “products”, which is traditionally termed as service outside of the IT-industry.

There is some research done in the enterprise interoperability group, which is a good basis for the definition of complex service interoperability. Especially the work in the area of model-driven interoperability, like [9], [10] and [11], provides some good basis for an enhancement to service interoperability. There are some models of service interoperability existing, (e.g. [12] and [13]), which are very useful as a fundamental basis to solve the pricing problem. However, these models are still lacking simplicity in the practical usage, and there are severe problems, which are typical for service interoperability, discussed in [14]. An adaptive approach to solve these issues is presented in [15], but it is still very generic, and has to be adopted very carefully to be used in a specific context like the case study at hand.

4. Solution proposal

To solve the pricing problem which is sketched in section 2.2, an architecture is used, which is similar to the idea of model-driven interoperability [9] with the enhancements for an overall service architecture, that have been proposed by Xu et al. in [13]. This architecture shows very promising characteristics, and therefore our idea is based on it.

But there still is a lack in the mapping from service demands to the pricing, which has to be done here. The classical way of the mapping goes from the demands to some fundamental service component specification, and later to the implementation.

We propose a different architecture, which is very useful, if the services can be pre-defined. This is the case in our example, so we introduce an additional modelling level, which is based on some very common economic model, namely decision trees.

The top level of our architecture consists of some kind of service grid, where the different business services, which are recurrent parts of an e-commerce platform, are presented to the customer, who can choose a subset, which he wants to be included in the service offering.

For each of the chosen services, a decision tree exists, which is a consistent way of defining business numbers or choosing special variations of the services. Thus the solution can be defined very well by walking through the decision trees and choosing a leaf for each tree, which is part of the offering.

At this point, there are two steps, which are following now. One step is the pricing, which can be directly derived from the chosen leaves. The other step is the construction of business processes, which can be partly defined from the leaves. By choosing from the service grid, and making decisions about the specific peculiarities of the services, a combined price for the overall solution is found and, at the same time,
the overall business processes can be constructed by the combination of the service process parts from the different leaves.

This leads to a model-driven way of defining the services, which can be enhanced in the common way of BPM. So the business processes are enhanced later to include the technical details of the service, and some workflow engine can automate these processes. In the concrete example, there is one more step, by compiling the services to be runnable with a high performance, if needed.
5. Conclusions

In this paper, we have discussed a serious business problem, coming out of an interoperability context. The business levels are explicitly addressed, and the interoperability of the business is part of the model. But also dynamic interoperability on a systems level can be included into the model.

The views of the model are integrated. The architecture is directly based on common standards like MDA and model-driven interoperability (MDI), as well as the SMDA, and it seems to fit well into these architectures. In the proposed architecture, the main focus is set on the different business levels, but there is an explicit proposal of some automatic mapping / model transformation between the different architectural levels (in a restricted domain, so we don't have to explicitly deal with semantic mapping of differing ontologies, which considerably simplifies the proposed architecture, compared to semantic-transforming models).

Our further research deals with a reference implementation of the upper system levels, to get feedback from practical use, and to be able to refine the proposed architecture. We have a special focus on the handiness of the system for practical purposes to achieve a real business value of the solution. Therefore, we want to utilize transactional cost theory to evaluate the theoretical appropriateness of the solution for some restricted usage areas. There is some chance that the restrictions of the domain can help to provide some useful system for business interoperability.

6. References


