Applying Model-Driven Integration Engineering to e-business – striving towards a framework concept *

Jens Schenderlein  
Godyo Business Solutions AG, Prüssingstraße 35, 07745 Jena  
Jens.Schenderlein@godyo.com

Sascha Sauer  
Truition GmbH, Leutragraben 107743 Jena  
saschasauer@truition.de

Tobias Krügel  
Mathema Software GmbH, Henkestraße 91, 91052 Erlangen  
mail@tobias-kruegel.de

Martin Gebauer  
University of Leipzig, Business Information Systems  
Johannisgasse 26, 04103 Leipzig  
gebauer@informatik.uni-leipzig.de

Abstract: Changing business requirements such as providing new business services lead to an ongoing need for fast and flexible adaptation of the underlying information systems and infrastructure. Thus integration is a recurring task to realize complex e-business processes. This paper outlines a concept for a model-driven framework for process based integration in the area of e-business. At first the results of domain engineering within three different companies are presented and used as a starting point to develop the framework concept with high-level domain-specific integration pattern as core. This concept will lead to the closure of the gap between business requirements and technical implementations and help raising the reuse of solution knowledge of system integrators.

Keywords: Model-Driven Integration Engineering, Systems Integration, E-Business, Framework, Integration Pattern

1 Introduction and Motivation

Changing business requirements such as providing new business services lead to an ongoing need for fast and flexible adaptation of the underlying information systems. Thus,

*This research was partly supported by the German Federal Ministry for Education and Research within the project EFIE under the grant sign 01IS08029
building information systems for business innovations means assembling new complex systems based on existing systems, possibly completed by new developed functionality. Information and communication technology (ICT) and innovations becoming inseparable. “The new series of sector studies confirms once again the critical role of ICT for introducing new business processes, such as organizational and process innovation in companies. The borderlines between implementing new ICT-based systems and introducing process innovation are getting blurred”[SLWA08, p. 10].

In the past there was a lot of research on integration done in fields like data integration [Len02], system integration [Has00] and enterprise application integration (EAI) [Lin00, TI03]. Themistocleous and Irani provided a taxonomy approach for application integration (AI) [TI02]. All these approaches do not treat integration problems from the perspective of an engineering science with an underlying methodology. ,,There are no standardized methodologies for EAI implementations”[LF02] is the top 3 failure reason. Because of rare EAI skills such projects ,,suffer from a failure rate of roughly 70 percent”[Tro03]. Bussler describes business to business (B2B) integration and examines concepts and architectures. Unless his efforts, he draws the conclusion “There is no academic field of ‘integration’ like there is for database managements systems or operating systems”[Bus03, p. vii] Integration Engineering is one step towards solving the methodological problem. It was presented by Thraenert [Thr08] being based on works from Rautenstrauch [Rau93]. The provided methodology consists of detailed definitions and provides clarity for the contradictory used terms in the sphere of integration. Furthermore it provides a procedure model which was evaluated within several cases. At this point only a few concepts should be introduced.

An important success factor of integration is its process orientation [LS04, Bus03]. That’s why we focus on process based integration. Integration is the action or process of integrating. The making up or composition of a whole by adding together or combining the separate parts or elements; combination into an integral whole: a making whole or entire [OX0]. An integration object (Iobject) is an element or subsystem which is or should be combined with other elements or subsystems to a larger system [Thr08]. Furthermore we distinguish between integration process and integration state. A state is a combination of circumstances or attributes belonging for the time being to a person or thing [OX0]. The integration state has different dimensions to describe its circumstances such as Iobjects, kind of integration, integration strength etc. The integration process is then the process of changing the integration state of two or more Iobjects.

2 E-Business Integration

The terms e-commerce, e-business and e-procurement are used in a very different way and often no clear distinction is made. “Electronic commerce includes electronic trading of physical goods and of intangibles such as information. This encompasses all the trading steps such as online marketing, ordering, payment, and support for delivery.”[Tim98] If we refer to e-commerce we focus on the trading and delivery side of a supply chain. In opposition e-procurement means the electronic support of the purchasing process [Pre03].
E-business integration is “essentially about the end-to-end integration of your business” [YA01, p. 7] and encompasses business-to-consumer (B2C) integration as well. This goes beyond EAI because company borders are crossed and is referred to as Supply Chain integration.

In the last few years empirical studies regarding the adoption of e-procurement gained attention [TWSQ08, WH07, PdRSA06, Bat07]. The asked questions range from “exchanging business documents completely electronically with their suppliers”[TWSQ08] over to “ordering more than 50% of goods electronically; have a complete Supply Chain Management and receiving e-invoices” [SLWA08]. This shows a very different understanding about the degree of IT support. Together with this understanding the need for integration differs too. Receiving e-invoices requires a much more complex integration with suppliers. A key inhibitor performing e-procurement is the integration with supplier systems [HSgL+07]. „It seems that only a few companies achieve a widespread roll-out integrating their suppliers into their electronic processes” [TWSQ08]. Unless these efforts none of the examined publications focuses on the integration methodology used to provide the solutions. Furthermore no distinction between direct and indirect procurement is made. Some studies claim to research the e-commerce field, but they do not make a clear distinction and use e-commerce instead of e-business such as Elie et al. [ELL04].

3 Problems for system integrators

„Integration projects are typically faced with a proliferation of standards and technologies, platforms and tools” [AVT06]. Thus companies providing system integration services often specialize in a few branches of business. In the past we examined three different companies which perform repeatedly integration tasks.

Company 1 is a system integrator specialized in e-procurement integration. Beside the offered system integration services the company sells an enterprise resource planning system (ERP system) for medium sized companies. It is based in Germany and employs not more than 50 persons and therefore is itself a medium sized company. Company 2 offers an on demand e-commerce solution and realizes full-service e-commerce solutions together with several partners which provide for example payment and fulfillment services. The company is based in Germany and Canada and serves more than 100 customers. Company 3 offers software development services and has successfully developed e-business and financial solutions. It is based in Germany as well. Integration was a need in every project and identified as the most risky task with the potential to be a project stopper.

Common to all examined companies is that they are medium sized companies working mostly for medium sized customers. A German study supported by the Federal Ministry of Economics and Technology (BMWi) reports that especially small and medium sized enterprises (SME) have a need to catch up with their e-business development to bigger companies [LMRP07]. Thus they have an increasing need for integration. Within the following two subsections we will describe the results of the domain engineering and present recurrent problems.
3.1 E-procurement integration

All e-procurement integration projects involve different roles of employees from the service provider site. In nearly all case there is a more business oriented role (consultant) and a more technical role (developer). The consultant communicates with the customer and collects the requirements. The developer is responsible for the creation of the integration solution. Similar roles can also be found at the customer site and causes several communication problems. Often technical details are forgotten to ask for by the consultant. This simple problem causes huge project delays especially if the technical role at the customer site is filled by a service provider. Figure 1 shows the schematic integration scenario between an ERP system and a shop system using a use case diagram. The shop system is used to implement a by-side e-procurement solution. Due to this, all further explanations originate in indirect procurement. As the use case diagram shows between both systems a product import and an order export process (shaded dark grey) must be realized. The development of these workflows is supported through reference workflows which are documented as activity diagrams. These activity diagrams are used for the communication between technically skilled project participants. Furthermore they serve as best practice for implementation. The term workflow is used to point out that the level of abstraction is an automatable process using different information systems (e.g. ERP and shop system). At the abstraction of business processes no best practices or the use of modeling tools were discovered. Integration problems involve business documents which must be exchanged electronically. The important business documents for electronic procurement according to Tanner et.al [TWSQ08] are Purchase order, Invoice, Order confirmation, Request for quote and bid, Dispatch advice. From this it follows that e.g. the order export is a recurrent workflow which must be solved through e-procurement integration.

Figure 1: Use Case.
3.2 E-commerce integration

The integration problems in the sphere of e-commerce differ from e-procurement especially because it is offered as an on-demand solution. As well as in e-procurement a differentiation of the involved roles was found. Therefore we conducted expert interviews with the managing director, the director services and a project manager. Figure 2 shows the schema of the integration solution. The schema shows the overall solution the customer needs to provide his new sales channel. The company we examined provides only the shop system. Hence all other components (payment, ERP etc.) must be integrated. The communication at the business level is supported by value chain diagrams. At the technical level in terms of workflows no modeling is used. Recurring integration tasks are conducted if a company wants to roll out this selling channel to different countries. This means not only offering a different set of products, but also solving integration with e.g. a different payment service provider. Thereby the Iobject shop system remains the same. Just as in e-procurement we found schematic redundancy, but some different Iobjects (e.g. payment integration is not needed in e-procurement).

3.3 Information needs for e-commerce and e-procurement integration

This section will summarize the observations from e-procurement and e-commerce integration. We showed schematic figures describing the integration scenarios. Within these abstract systems as Iobjects were presented. The classes of systems are ERP, Shop, Payment and Customer Service (CS). The provider of integration solutions abstracts apparently from concrete systems and uses the system type instead (see figure 3(b) and 2). From this it follows that a system type must be associated with a certain set of functionality. Figure 3 shows the abstractions used by the system integrators in cases of systems
Figure 3: abstractions used by integration service providers

and business objects. These types or abstract systems support business processes through abstract functionality such as providing information about orders. Thus an order is a business object with characteristics such as information about the customer, the products and their amounts regardless if we are talking about a SAP order or a Shop order. The order implemented through a SAP system will be referred to as Data Object (see figure 3(a)). Business processes and workflows were presented in an abstract manner too. As stated in the previous sections business process can be transformed into workflows. According to the targeted audience, system integrators use modeling tools to communicate at business process or workflow level. In summary the information needs for integration consist of business processes, workflows, abstract and concrete systems and the appropriate interfaces. Additional to these information reference processes or workflows must be captured and instantiated during the integration process.

4 A Framework concept for Model-Driven Integration Engineering

This section outlines framework concept. Starting with a brief introduction to model-driven software engineering (MDSE), we show based at which steps of integration model-driven software development will help mastering complexity.

4.1 Model-Driven Software Engineering and first steps

A new way of leveraging abstraction and handling complexity are model-driven methods [Sel03, MSUW04, Bez05, SV06]. Formal models in terms of domain-specific abstractions become the first-class artifact of software engineering which are transformed to concrete implementations [SV06, p. 4]. Model-driven software development starts with a handcrafted reference implementation to identify schematic, generic and individual code [SV06]. This idea was the base for our domain engineering outlined in section 3. We found the most recurrent tasks and tried to find domain specific abstractions. First attempts to ap-
ply MDSE to a part of the examined problems lead to an automation of mappings needed when sending data from one information system to another. The result was a development speed up. All other problems like communication still remained unsolved and made a broader approach necessary.

4.2 Why a framework approach?

Our goal is automating the process of developing integration solutions to fully support IE with model-driven technologies. This is named Model-Driven Integration Engineering (MDIE). MDIE aims at formal modeling and the usage of model transformation and validation to solve integration problems by consequently following an engineering paradigm. To provide clarification to the term framework we refer to the following definition: ,,A framework is a reusable ‘semi complete’ application that can be specialized to produce custom applications”[FS97]. Thus it provides a solution to a specific problem class. Combining this solution for a problem class with domain specific abstractions through a MDSE approach we are aiming at. The issues which must be solved are according to the terminology of France and Rumpe [FR07] and our identification of information needs:

- modeling challenges for: (1) business processes, (2) workflows, (3) information systems including interfaces and business objects and

- transformation challenges for: (1) model operators for the instantiation of reference models resulting in project specific instances, (2) business processes to workflows, (3) abstract systems to concrete systems.

Figure 4 shows our concept of a framework supporting MDIE inside e-business. Within integration several levels of integration (database, functionality and user interface) are distinct[Lin00]. Which role plays model-driven software engineering within the proposed framework? For each component a meta-model is provided which specifies the modeling constructs that can be used to express models (e.g. business processes). This is marked as $M2$ within figure 4. The $M1$ level are thus the modeled business processes. Accord-
ing to the terms of the MDA (see [MSUW04]) the abstraction levels are: (1): platform independent models (PIM), the abstract systems or system classes like ERP systems and (2): the platform specific models (PSM), concrete systems like a SAP R/3. The same distinction exists between the reference models as PIM and the reference implementations as PSM. A reference model transforms to a reference implementation using a reference transformation. This elements can be used to design own models, transformations and implementations.

4.3 Solving the modeling challenge

Solving the modeling challenge might lead to the development of domain specific languages. For integration purposes no suitable modeling language is available. Existing approaches originate from the sphere of enterprise architecture (EA). EA has a long research history starting in the 1970s and 1980s with CIMOSA and GIM-GRAI. Since then several Frameworks were proposed. A good overview about EA frameworks is given by Shekerman [She04]. An Architecture (e.g. an enterprise architecture) is “the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution”[Iee00]. The components of an enterprise architecture such as services or information systems can be seen as Iobjects (see section 1). Changing an enterprise-architecture means to change the components (Iobjects) or the relationships between them. The result is a changed integration state which might differ from the nominal state and leads to an integration process. The integration process is carried out to achieve the nominal integration state.

The most EA frameworks provide a modeling independent approach. While focusing on modeling languages we started to examine existing EA tools, investigating the implemented modeling languages. We identified 36 tools which claim to be EA tools. There are commercial as well as open source products. Even though the analysis is still ongoing we provide preliminary results. An outstanding modeling approach is the ArchiMate language which is described in detail by Lankhorst [Lan05]. It is maintained by the Open Group which is also responsible for the TOGAF framework and supported by BiZZdesign Architect, ARIS ArchiMate Modeler, Metis, Corporate Modeler, System Architect. No other language achieved such a broad tool support. Comparing ArchiMate with the information needs of system integration (see section 3.3) and the framework concept (see figure 4) it is obvious, that ArchiMate provides a starting point to solve the modeling challenge. The modeling components should be extended with standard attributes such as system name, version information. To cover variants in process descriptions a connector concept consisting of AND, XOR, OR connectors like in ARIS event driven process chains would be use full. Extensibility is an explicit concept of the ArchiMate standard. Hence the suggested extensions can be applied. Moreover the distinction in PIM and PSM models can lead to further extension needs of the language in terms of technical details.
4.4 Complex integration pattern

Identifying schematic redundancy and using generators instead of manual programming is a base idea of MDSD. Ways to capture schematic redundancy are patterns, which are widely discussed in software engineering. Some approaches tried to adapt the base work from Gamma et al. [GHJV95] to integration solutions [Lut00, AJ01, LOP02]. The focus on patterns concerning message orientation was presented by Hope and Wolfe [HW03]. A class library supporting these pattern is Apache Camel [AC] which could be used as a reference implementation. For SAP the 12 most significant message integration pattern were implemented using the SAP NetWeaver Process Integration [DS08]. This shows the adaptation of technical integration pattern. Despite the effort done, no approach raises the abstraction in a significant manner. Paul Allen proposes composite pattern as a mix-up of business pattern and integration pattern [All01]. Another approach concerning model-driven design and business pattern is the work from Hruby which is based on the resources, events, agents (REA) model. An example of his work is the exchange process pattern which aims at the interaction between an enterprise and its trading partners [Hru06]. We aim at completing the technical approaches and extending them with additional components like common business objects. Within section 3.1 and 3.2 we outlined our observations from e-procurement and e-commerce integration. Figure 5 shows a simplified part of an order example using the business and application layer of the ArchiMate language. The exchange process is an instance of the exchange process pattern at the business level. The exchange process represents a part of the sales channel of a company. It is triggered by the New Customer Order event. Check for new orders once a day represents a business requirement which affects the possible integration solutions which can be applied. Because the orders should only be checked once a day, a messaging approach is suitable. The
business layer shows the business object order which is accessed by the exchange process. Within the application two data objects (Shop Order and ERP order) implement the same business object. This indicates a mapping problem when it is necessary to exchange order information between the two systems implementing the data objects. In our case these are Shop and ERP, which expose their functionality (e.g. Stock management) through application services (e.g. Product management) so that these application services can be used by business processes. In figure 5 the integration solution is represented through an application component named integration solution. The modeling of the integration solution is not supported by ArchiMate and hence is not part of the figure. It might consist of a message channel and a message translator which solves the indicated mapping problem. Both are pattern and part of the enterprise integration pattern presented by Hope and Wolfe [HW03]. This shows how business pattern and technical integration pattern can be combined to achieve integration pattern at a higher level of abstraction such as the asynchronous order export between an ERP and a shop system. This pattern might have different variations such as create a new customer when it only exists inside the shop system or to refuse the order. This schematic solution will be supported by transformations to automatically create the integration solution.

5 Conclusions and Future Research

These are first steps towards solving integration problems. Using domain-specific concepts such as common business objects is a suitable approach squaring with the way of thinking of system integration providers. To verify the existing and to collect new pattern qualitative empirical research is needed. Therefore we prepare a panel e-business integration to achieve a stable set of domain specific constructs. The preselection using the Hoppenstedt [Hop] database resulted in a set of companies 139 companies. The database lists companies with more than 10 employees and an annual turnover of more than a million Euros. We selected companies which are classified as system integrators according to their business activity. A first analysis showed that 88 of the 139 companies are between 10 and 20 years at the market and 64 of them have between 11 and 100 employees. Hence the majority are medium sized companies. The results of the panel will be used to create a set of constructs needed to describe integration problems in alignment to EA approaches. Providing a base for transformations in terms of integration pattern it is a need to systematize the creation of integration solutions. Thus a complete catalog of patterns should be created with pre- and post conditions of their use. To align these technical patterns with business requirements and business processes we aim at applying them as showed in section 4.4. One main argument against the framework approach, in our opinion, is the expenses of modeling. Existing documentation models can be used through the concept of M3 level based transformations. This offers the possibility to open up existing model data bases such as ARIS or Meta Edit [Ker08].
References


