A Service Marketplace Consumption Platform for Extensible Enterprise Systems in an Internet of Services[1]

Matthias Allgaier, Markus Heller

SAP Research
Vincenz-Priessnitz-Str. 1
76131 Karlsruhe, Germany
{matthias.allgaier, markus.heller}@sap.com

Abstract: As services become tradable in an Internet of Services, the simplified consumption of services becomes a major requirement, especially when integrating services into standard enterprise systems. A key challenge for enterprise system providers is to deliver standard business applications with a rich functional spectrum. At the same time, dedicated flexibility and extensibility features need to be provided that allow the enterprise system owner to enrich the core system with complementary services, especially at a later stage in the software-lifecycle after shipment. In this paper, the architecture of a Service Marketplace Consumption Platform is introduced that opens up extensible enterprise systems in order to consume complementary (extension) services from service marketplaces.

1 Introduction

In the vision of an Internet of Services (IoS), service marketplaces are emerging as web platforms that enable service providers to offer and sell their services similar to physical goods ([JRS08], [WRK10]). Services can then be purchased and consumed by a variety of different channels. One of the most promising service consumption channels are standard enterprise systems (e.g. ERP, CRM or SCM systems) as these systems have been successfully implemented by a very large number of companies [LWG09]. These adaptors of standard enterprise systems constitute a high number of potential service consumers. On the other hand, service marketplaces provide the infrastructure to offer complementary services that can flexibly enrich and extend the functionality of standard enterprise systems. Such complementary services can be provided by partners or independent software vendors (ISVs) in a service ecosystem [BD06]. Figure 1 shows the Service Consumption Process: First the user of an enterprise system searches and selects a service on the service marketplace. Afterwards the service needs to be integrated into the enterprise system. Then the service can be used within the existing working environment. Finally the service usage is billed and the user can give feedback on the service quality. However, the integration of services into enterprise systems is typically carried out in manual integration projects (step C in Figure 1). In most cases, extension or adaptation of the enterprise system itself is required. Typically, enterprise systems do

[1] The work presented in this paper is embedded into THESEUS/TEXO which is funded by means of the German Federal Ministry of Economy and Technology under the promotional reference 01MQ07012. The authors take the responsibility of the contents.

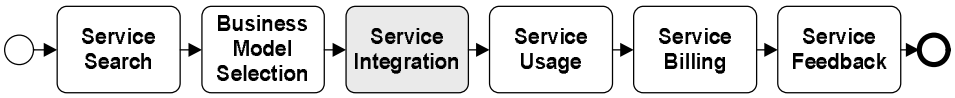

Figure 1 Service Consumption Process.

only provide extension mechanisms with a low level of abstraction (e.g. code-level interfaces), cf. [BHM01], [FS10]. Seamless and less complex integration of services into enterprise systems therefore becomes a key research challenge in the IoS context that we tackle within our work as part of the THESEUS/TEXO project[2].

In this paper we present the architecture of a *Service Marketplace Consumption Platform* that provides an integration layer between (1) an extensible enterprise system and (2) a service marketplace. It allows seamless service integration of internet-based services into business applications running within enterprise systems. This architecture has been prototypically implemented and is based on our previous work ([AH09], [AHW10]). Section 2 presents a motivating scenario. Section 3 describes the Service Marketplace Consumption Platform. Sections 4 and 5 describe shortly related work and a conclusion.

2 Application Scenario

In this section the integration of complementary services into enterprise systems is illustrated with an example from the automotive industry.

As a result of legal changes in export guidelines, a manufacturer of car seats has to certify his products to guarantee, that materials used within a car seat comply with ecological laws. The company runs a Product-Lifecycle-Management (PLM) application in an enterprise system that supports its core business processes (Figure 2). The core version of the system does not support the required calculation of eco values for a car seat. But a provider offers a service on the marketplace that allows the calculation of eco values for products including certification.

The service is consumed according to the consumption process from Section 1: The product designer of the company accesses the service marketplace from within his enterprise system and *searches* for complementary services that provide the missing functionality. As a result, services are suggested which are certified for the enterprise system. The product designer *selects* the service which best matches his needs, a service "Eco-Calculator" and purchases it on the marketplace. The service is automatically *integrated* into the enterprise system's business application without running a manual integration project: the user interface of the core business application is extended with (1) an *additional table column* ("Eco Value"), (2) an *additional button* ("Calculate Eco Value") and (3) an *additional field* indicating the total eco value for the car seat ("Entire Eco Value"). After integration, the service can be used. If the total eco value fulfils the legal requirements, a certificate is generated and passed to the consumer application. A bill is sent to the service consumer and questionnaire feedback is sent to the marketplace.

[2] http://theseus-programm.de/en-us/theseus-application-scenarios/texo/, visited on 23-04-2010.

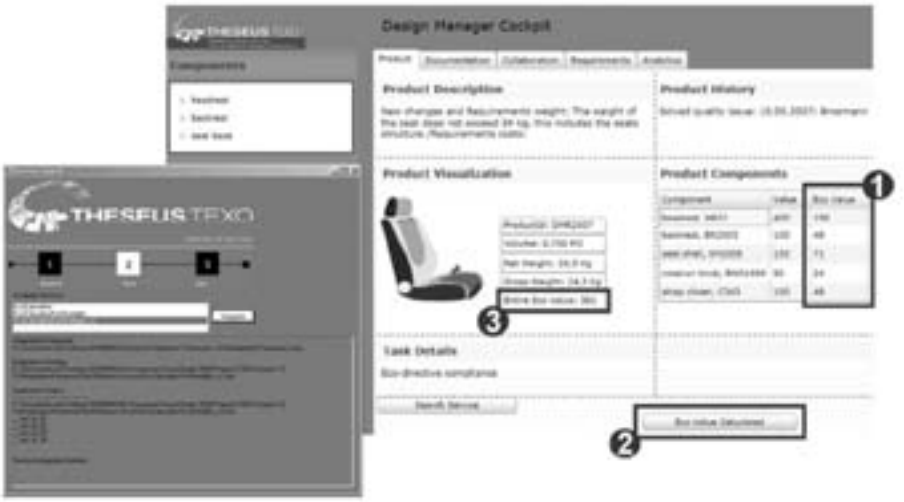

Figure 2 PLM Business Application extended with the complementary service "Eco-Calculator".

3 Service Marketplace Consumption Platform

The service marketplace consumption platform described in this section constitutes an important part of a *service integration framework* as outlined with research challenges, framework architecture and modelling support in [AH09] and [AHW10]. In brief, the service integration framework provides support for partners and independent service vendors (ISV) in a service ecosystem for extensible enterprise systems. It allows partners to develop integration solutions that realize the integration of a remote and complementary (web) service into an application of an extensible enterprise system. The framework offers an integration modeling environment for partners and a service marketplace consumption platform to automate service integration as far as possible.

Figure 3 illustrates the *integration modeling environment* with input and output documents to provide a basic understanding how service integration is modeled by partners in our approach (for details, please refer to [AHW10]). As a prerequisite, the enterprise system provider publishes an *application extensibility description* (left) that represents the extensibility capabilities of the enterprise system with its core business applications. Similarly, a service provider creates a *service description* (right) for a service that he offers, e.g. on a service marketplace. The service description describes the service's capabilities in detail. A partner or independent service vendor (ISV) acts in the role of a *service integrator*. He uses the integration modeling environment (middle) to define all relevant integration aspects for the selected service and business application (based on the loaded application extensibility description and the service description). Finally, an *integration description* with all defined adaptation and extension steps (including "add new table column", "add new button", "add new data field") is created and stored on the marketplace by the service integrator. The roles of service provider and service integrator can also be played by the same organization.

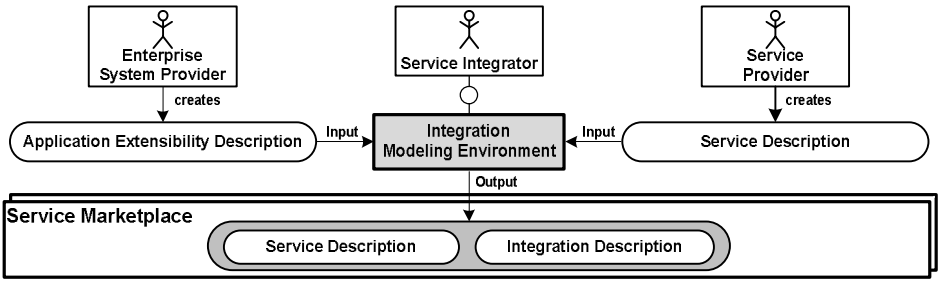

Figure 3 Integration Modeling Environment for Service Integration.

This paper focuses on the framework's runtime part. Figure 4 illustrates the architecture of the runtime *service marketplace consumption platform* that connects service marketplaces, service runtimes, and an extensible enterprise system.

A *service marketplace* (top left) offers a *repository* for service descriptions and integrations descriptions and several basic engines that can be used by providers or consumers. A *service discovery engine* receives queries to find and return service offers with specified characteristics on the service marketplace. A *feedback engine* collects and processes feedback data that is provided from the different service marketplace stakeholders. A *billing engine* is responsible to create invoices for service consumers according to the individual business models of the used services. The *service delivery engine* uses service descriptions and integration descriptions from the repository and delivers them to a consuming party.

A *service runtime* (top right) is responsible for executing one or more instances of services according to the service description (from the marketplace) and to guarantee all specified service level agreement characteristics.

The *enterprise system* (bottom) consists of one or more *core business applications* that can be adapted or extended according to customer specific needs to integrate a new integration solution service. Typically, a large set of native *extensibility features* is offered by the enterprise system that supports a wide spectrum of use cases and address various stakeholders. These extensibility features can affect several application layers, e.g. by adding new UI elements (presentation layer), adding a new process step (process layer), and adding services or business objects (service or business object layer).

The *service marketplace consumption platform* (center) is located in between the enterprise system and the service marketplace(s) and service runtime(s). The platform offers a *service marketplace access* component that can be used by either (human) service consumers or software components on the enterprise system. Service discovery engines and feedback engines from several marketplaces can be controlled and their results accumulated by this central service marketplace access gateway. A *billing integration module* on the consumption platform is responsible to mediate between the billing engines of the marketplaces and the invoicing subsystem of the enterprise system. A *service adapter* mediates the connection of the enterprise system executed service instances within service runtimes.

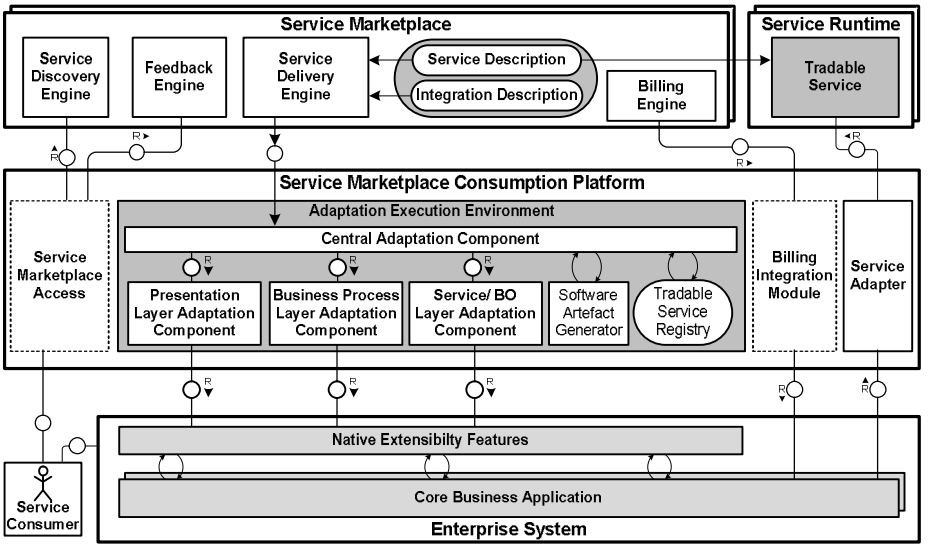

Figure 4 Architecture of Service Marketplace Consumption Platform.

The extension of the enterprise system with a new integrated complementary service is accomplished by the *adaptation execution environment* of the platform. The *central adaptation component* manages the overall execution of the service integration. It is triggered from the service delivery engine on the service marketplace (from where it receives a service description along with an integration description for the service to be integrated) and delegates integration-specific tasks to a number of layer-specific adaptation components, a software artifact generator and a tradable service registry. *Application layer-specific adaptation components* (for presentation layer, process layer, and services/business object layer) execute the adaptation/extension steps for the individual application layer of the enterprise system by using the native extensibility features of the target enterprise system as needed. The integration description can additionally contain data from which new software artifacts could be generated by the *software artifact generator* (e.g. help dialogs, configuration files). The *tradable service registry* contains information about all currently integrated services.

The described consumption platform has been implemented as a research prototype. The service marketplace with service repository and engines for discovery, feedback, and billing is implemented in Java and the extensible PLM business application system is implemented in Microsoft Silverlight. A PLM application user accesses the service marketplace engines via the consumption platform and buys the service (with integration description). The adaptation execution environment (in Java) receives the bought integration description from the marketplace and executes the adaptation/extension steps from Section 2. Figure 2 shows a screenshot of the PLM application after extension with the "EcoCalculator" service (implemented with Axis). For the presentation-layer extension we used XAML[3], process and service layer extension is under development.

[3] http://www.microsoft.com/Silverlight/, visited 23-04-2010.

4 Related Work

In general, our work is related to existing B2B Integration and Enterprise Application Integration (EAI) techniques that address structural or behavioural interface mediation techniques between two service interfaces, cf. [Bu03]. A wide spectrum of adaptation and extension techniques in the ERP domain are described in [BHM01]. In [BM10] federated ERP systems are presented that are composed of web services. Plug-in technologies allow the installation of components into a core application framework [Bi05]. In contrast to the aforementioned approaches our work focuses on the controlled extensibility of enterprise systems for service integration in the specific context of a service marketplace scenario.

5 Conclusion

In this paper we have outlined an architecture for the integration of an extensible enterprise system and a service marketplace. In future work, we will extend our prototype to cover the service integration into the process layer of the enterprise system.

References

[AH09] Allgaier, M.; Heller, M.: Research Challenges for Seamless Service Integration in Extensible Enterprise Systems, Workshop Industrial Experiences for Service Oriented Computing (IE4SOC).Stockholm, Sweden, 2009.

[AHW10] Allgaier, M.; Heller, M.; Weidner, M.: Towards a Model-based Service Integration Framework for Extensible Enterprise Systems. In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner and Arne Frerichs (Eds.) Tagungsband der Multikonferenz Wirtschaftsinformatik, Göttingen, 2010.

[BD06] Barros, A.; Dumas, M.: The Rise of Web Service Ecosystems. In: IT Professional Vol. 8, No. 5, 2006, pp. 31-37.

[BHM01] Brehm, L.; Heinzl, A; Markus ML.: Tailoring ERP systems: a spectrum of choices and their implications. In: Proceedings of 34th Hawaii international conference on system sciences (HICSS'01), Hawaii, IEEE, 2001.

[Bi05] Birsan, D.: On Plug-ins and Extensible Architectures. In: Queue Vol. 3, No, 2, 2005, pp. 40-46.

[BM10] Brehm, N.; Marx Gómez, J.: Federated ERP-systems on the basis of Web Services and P2P networks. International Journal of Information Technology and Management 2010 - Vol. 9, No.1, 2010, pp. 75 – 89.

[Bu03] Bussler, C.: B2B Integration: Concepts and Architecture. Springer, Berlin, 2003.

[FS10] Frick, N.; Schubert, P.: Flexibilität in ERP-Standardsoftware: In: Matthias Schumann, Lutz M. Kolbe, Michael H. Breitner and Arne Frerichs (Eds.) Tagungsband der Multikonferenz Wirtschaftsinformatik, Göttingen, 2010.

[JRS08] Janiesch, C.; Ruggaber, R.; Sure, Y.: Eine Infrastruktur für das Internet der Dienste. HMD - Praxis der Wirtschaftsinformatik, Vol. 45, No. 261, 2008, pp. 71-79.

[LWG09] Lo, H.; Wang, R.; Garbini, J.P.: The State of Enterprise Software 2009. Forrester Research, Cambridge 2009.

[WRK10] Weiner, N.; Renner, T.; Kett, H.: Geschäftsmodelle im Internet der Dienste - Aktueller Stand in Forschung und Praxis. Fraunhofer Verlag, Stuttgart, 2010.