AMCIS 2006 Proceedings

Americas Conference on Information Systems (AMCIS)

12-31-2006

Integration of IT services: Towards a pattern-based approach for eliciting service integration requirements

Michael Schermann
Santa Clara University

Tilo Böhmann
Technische Universität München- Germany

Helmut Krcmar
Technische Universität München- Germany

Recommended Citation

Schermann, Michael; Böhmann, Tilo; and Krcmar, Helmut, "Integration of IT services: Towards a pattern-based approach for eliciting service integration requirements" (2006). *AMCIS 2006 Proceedings*. 312.
http://aisel.aisnet.org/amcis2006/312

Integration of IT services: Towards a pattern-based approach for eliciting service integration requirements

Michael Schermann
Chair for Information Systems
Technische Universität München, Germany
Boltzmannstr. 3, 85748 Garching
michael.schermann@in.tum.de

Tilo Böhmann
Chair for Information Systems
Technische Universität München, Germany
Boltzmannstr. 3, 85748 Garching
boehmann@in.tum.de

Helmut Krcmar
Chair for Information Systems
Technische Universität München, Germany
Boltzmannstr. 3, 85748 Garching
krcmar@in.tum.de

ABSTRACT

Current industry trends suggest that IT services are being sourced more selectively. A prominent example of such an IT service is the user help desk that is often an integrated service, where different sub-services are being delivered by different service providers. Hence IT service providers face the challenges of integrating business processes and IT elements as well as service management tasks such as managing service contracts within a service delivering network more efficiently. Due to the dynamic nature of the IT services industry eliciting integration requirements is a crucial and recurring step in such endeavors. Hence we propose the foundation of integration patterns that supports eliciting integration requirements in two dimensions: process & IT and service management. Such patterns support reusing gained experience by capturing integration problems and proven solutions. Although our approach is still research in progress we present two pattern candidates.

Keywords

IT Services, service engineering, integration patterns.

CHALLENGES OF IT SERVICE INTEGRATION

Current industry trends suggest that IT services are being sourced more selectively (Cohen et al. 2006). Decomposition, modularization, and standardization of business processes, application systems, and infrastructure components have been established as common practices to allow integrating different service processes and resources (Böhmann et al. 2003).

Consider the following example of an IT user help desk (short: UHD) service (see Figure 1):

Company Gamma has outsourced its user help desk to the global user help desk service provider Beta. To maintain the existing integration of Gammas former user help desk with proprietary applications of Gamma such as monitoring and reporting applications, Beta is continuing to operate with existing applications of Gamma.

Furthermore Gamma is very sensitive about confidentiality of its data (e.g. the UHD knowledge base), thus Beta has agreed to operate the UHD knowledge base on servers that are hosted at Gamma. Solely, the incident management application is new and provided by Beta.

To gain flexibility and focus on knowledge intense services, Beta has contracted the local call center provider Alpha for delivering first level support. First level support personnel are receiving calls or e-mails containing incidents of the service customer’s users. Received calls are verified against service contracts and categorized by incident types (e.g. network, mail). First level personnel are trying to resolve as many common incidents as possible.

Beta core competence is delivering high-quality second level support. Here highly trained staff is resolving complex and difficult cases. Thus Betas employees are not in direct contact with actual users of the user help desk. First level support is escalating complex or non-standard problems that Alpha is not able to resolve within a certain period of time to second level support at Beta. Here incidents are finally resolved or further escalated to the vendor of the product that caused the incident. Thus Beta has support contracts with various vendors of enterprise application systems.

Further on, Beta is responsible to adhere to service levels for the user help desk. Likewise, Beta has agreed to ensure compliance of the user help desk to security and IT governance policies of Gamma. Figure 1 is summarizing the service network.

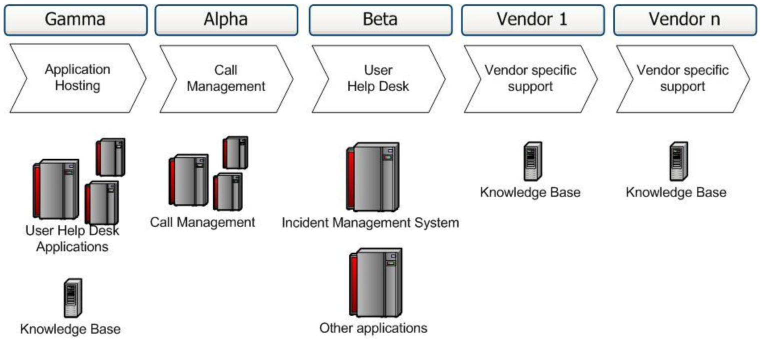


Figure 1: Service Network around Beta

As the example shows, selective service sourcing and delivery results in a network of services and underlying IT applications and infrastructure. Furthermore, facing saturated markets service providers are impelled to adapt their services constantly to service customer needs. As fast as businesses are changing, requirements of IT service customers are changing at the same speed (Cohen et al. 2006).

IT service providers therefore face two challenges:

Firstly, service providers have to cooperate in dynamic networks with other providers to deliver the requested composed service. Thus service integration subsequently requires *integration of businesses processes, applications, and infrastructure* among different subsidiaries and with other service providers (Cohen et al. 2006). Such requirements are generally subsumed by the term *enterprise application integration* (Conrad et al. 2006).

Secondly, *cross-service management* of services, service levels, and service contracts is required for deriving benefit of a selective sourcing strategy by enabling planning, control, and evaluation of different services and their interactions (Cohen et al. 2006). As the case of Beta shows, service providers are required to integrate their service management to embrace non-functional requirements such as security and compliance. Hence Beta has to ensure that Alpha is compliant to Gammas security and IT governance policies. Furthermore service levels and resulting obligations have to be propagated and managed throughout the service network.

As this case shows, IT service integration depends on integrating both IT and service management. Thus our first research question is: *Which organizational or technological elements of services and services providers have to be considered in service integration projects?* Due to the dynamic nature of the IT services industry IT services and with it service networks change often and repeatedly. Hence integration projects become an ongoing task of IT service providers. To enable reuse of knowledge gained in previous integration projects, we propose a pattern-based approach to eliciting service integration requirements. Thus, our second research question is: *How can patterns support eliciting integration requirements?*

The remainder of this paper is structured as follows: In the following section we develop a conceptual framework for IT service integration that is considering necessary integration objectives. Then we discuss integration patterns and present two service integration pattern candidates. The paper finishes with a conclusion and an outlook on future research activities.

A FRAMEWORK FOR IT SERVICE INTEGRATION

In this section we develop a conceptualization of IT service networks. First we clarify the term *IT service*. Based on existing research on information systems integration and service management we then construct an *IT service integration framework* that is conceptualizing necessary elements for integration projects. Based on this architecture we *discuss existing integration approaches in the context of our framework.*

As researchers and practitioners could not yet agree on a definition of service we have to make explicit our understanding of IT services. Services are typically seen as distinct from products because of the following characteristics: services are delivered and consumed at the same time, services usually integrate external factors such as locations, customer processes, personnel or equipment in service delivery, and services are intangible (Bullinger et al. 2003). We define *IT services* as services whose delivery relies on information systems and result in an IT-supported organizational element, i.e. a process or a function of the service customer (Böhmann et al. 2003).

To conceptualize the challenge of service integration we propose an IT service integration framework with two perspectives as depicted in Figure 2. Information systems are generally conceptualized in a layered approach separating business processes, applications, and infrastructure (Conrad et al. 2006; Dayal et al. 2001). Hence integrating IT services from an *information systems perspective* is required to consider necessary structural elements information systems: A business process layer represents business processes of both service customer and service providers. An application layer is grouping necessary application systems. An infrastructure layer encapsulates infrastructure elements, such as data centers and networks.

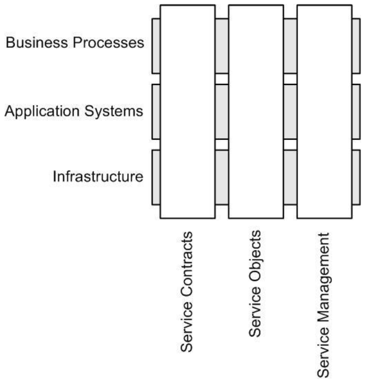


Figure 2: IT Service Integration Framework

While integrating information systems is focusing on loosely-coupled mechanisms that provide a maximum of information hiding, service providers often face a distinct challenge (Papazoglou et al. 2003): Customers seek to maintain control about data and applications used by external service providers. Service customers continue to maintain own infrastructures and service providers are delivering their services by operating on the customer’s infrastructure. This is a pertinent scenario if security requirements of the end customer do no permit transferring any element to service providers. Hence IT service integration requires integrating service management processes of all service network members.

As services are intangible and do not exhibit characteristics that customers can inspect prior to acquiring services, service contracts gain a pivotal role in specifying the features and quality aspects of IT services (Kaas 1995). Thus the *service dimension* of our framework is conceptualizing necessary elements of managing multi-sourced services: A service *contract layer* is grouping elements that are necessary to manage service contracts, such as contracts, offers, and service level agreements (Ludwig et al. 2003; Sturm et al. 2000). The *service object layer* consists of elements that are the building blocks of services, such as service modules (Böhmann et al. 2003) and service metrics (Fensel et al. 2002; Ludwig et al. 2003). The *service management layer* consists of methods and processes that enable engineering, delivery, and management of services, e.g. ITIL components (Office of Government Commerce 2001). In sum, the information system dimension and the service dimensions capture elements that are required to integrate IT services.

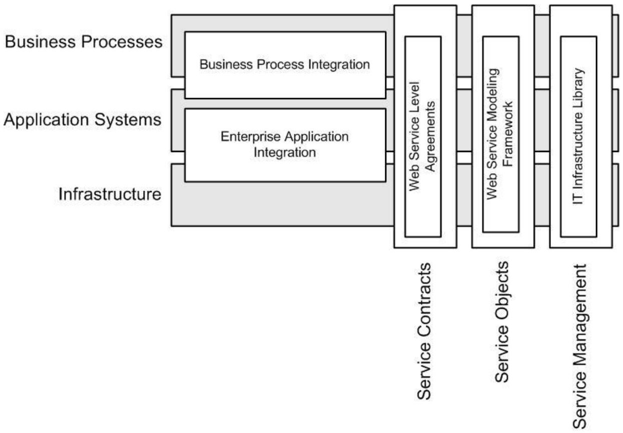

Business Processes
Application Systems
Infrastructure
Business Process Integration
Enterprise Application Integration
Web Service Level Agreements
Web Service Modeling Framework
IT Infrastructure Library
Service Contracts
Service Objects
Service Management

Figure 3: Examples of existing approaches aiming at integration various aspects of the IT service integration framework

Figure 3 maps examples of existing approaches of integrating information systems or services onto the IT service integration framework (Conrad et al. 2006; Dayal et al. 2001; Fensel et al. 2002; Ludwig et al. 2003; Office of Government Commerce 2001). As it can be seen various concepts, methods, and technologies for different layers exist. However, we are not aware of research considering both perspectives.

As we have mentioned introductorily managing service networks evolves into a recurring task. Hence designing and managing service networks depends on effective and efficient elicitation of specific requirements of IT service networks considering both the service perspective and IT perspective. Thus in the next section we propose a requirements engineering approach for IT service networks that is considering reusing experiences from previous integration efforts.

PATTERN-BASED REQUIREMENTS ENGINEERING IN SERVICE NETWORKS

Requirement engineering (RE) refers to the process of eliciting, analyzing, and systematizing requirements (Nuseibeh et al. 2000). The main task of RE is to reach consensus about the features, e.g. of an application system. Generally RE is differentiated in the following phases: eliciting requirements, modeling and analyzing requirements, communication requirement, agreeing on requirements, and evolving requirements (Nuseibeh et al. 2000). To abstract from specific integration projects, integration design experience and best practices are stored in a pattern repository. As various mature requirements engineering processes exist we focus on a pattern repository that may be seen as the distilled integration knowledge of service networks.

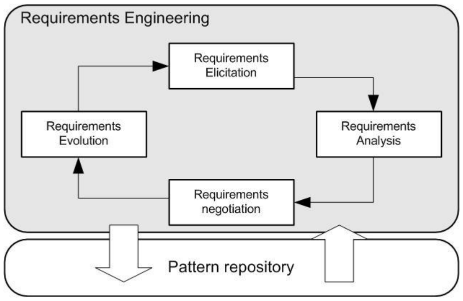
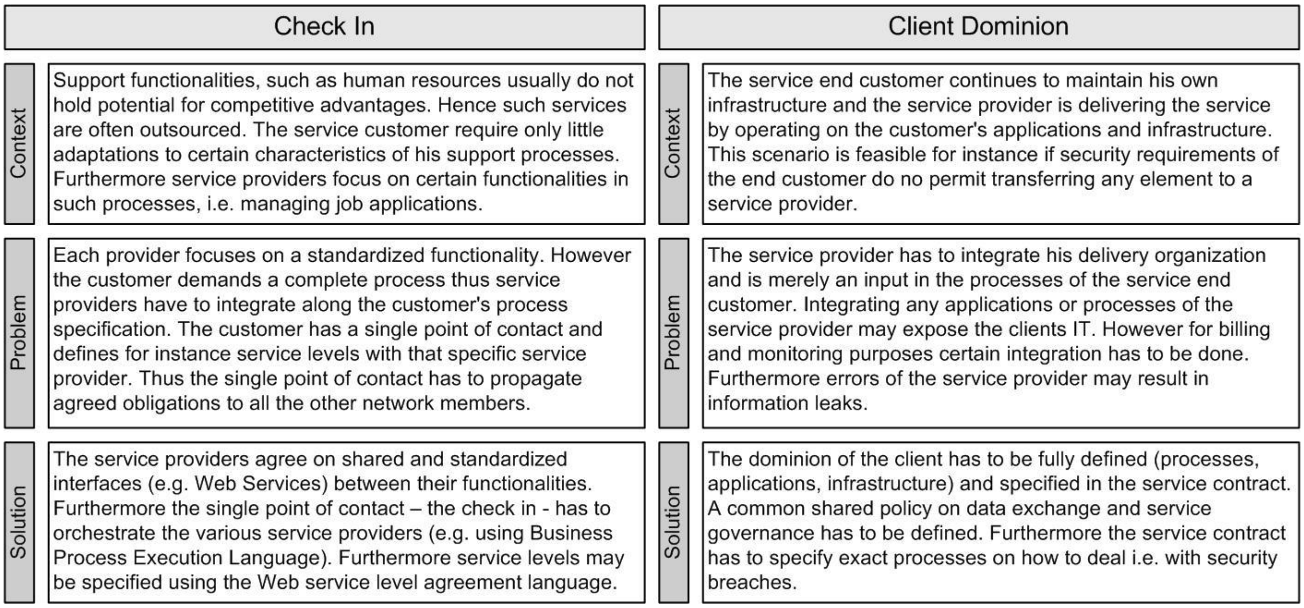


Figure 4: Pattern-based requirements engineering

The term pattern has originally been coined by Alexander (1979) in the field of architecture and has been transferred to software engineering by Gamma et al. (1994). A pattern can be defined as combination of a reusable solution to a problem in a way that the solution is applicable to many similar problems. A pattern generally comprises the following elements (Alexander 1979): the *context* comprises causes which lead to the problem described in a pattern and the conditions under which the problem occurs. The context should support acquiring the relevance of a pattern. The *problem* is to be described by explaining contradictions causing the problem in the context of the pattern. These aspects of the pattern's problem section are often called *forces* (Alexander 1979). The next section of a pattern explains the proposed *solution* by dissolving forces described before. An illustration of possible side effects is given as well. The closing section of a pattern is composed of *references to related patterns*.

Based on the introductory case, we present two integration pattern candidates that are depicted in Figure 5.



	Check In		Client Dominion
Context	Support functionalities, such as human resources usually do not hold potential for competitive advantages. Hence such services are often outsourced. The service customer require only little adaptations to certain characteristics of his support processes. Furthermore service providers focus on certain functionalities in such processes, i.e. managing job applications.	Context	The service end customer continues to maintain his own infrastructure and the service provider is delivering the service by operating on the customer's applications and infrastructure. This scenario is feasible for instance if security requirements of the end customer do no permit transferring any element to a service provider.
Problem	Each provider focuses on a standardized functionality. However the customer demands a complete process thus service providers have to integrate along the customer's process specification. The customer has a single point of contact and defines for instance service levels with that specific service provider. Thus the single point of contact has to propagate agreed obligations to all the other network members.	Problem	The service provider has to integrate his delivery organization and is merely an input in the processes of the service end customer. Integrating any applications or processes of the service provider may expose the clients IT. However for billing and monitoring purposes certain integration has to be done. Furthermore errors of the service provider may result in information leaks.
Solution	The service providers agree on shared and standardized interfaces (e.g. Web Services) between their functionalities. Furthermore the single point of contact – the check in - has to orchestrate the various service providers (e.g. using Business Process Execution Language). Furthermore service levels may be specified using the Web service level agreement language.	Solution	The dominion of the client has to be fully defined (processes, applications, infrastructure) and specified in the service contract. A common shared policy on data exchange and service governance has to be defined. Furthermore the service contract has to specify exact processes on how to deal i.e. with security breaches.

Figure 5: Patterns of IT service integration[1]

IT services are usually delivered in a mix of such integration patterns. Take for instance our user help desk case:

The check in pattern could apply to the incident management from Beta as this is a service that has to be integrated into Gammas processes. Here Gamma's employees want to submit incidents as they did before the outsourcing of the user help desk. Hence Beta has to provide a check in that allows Gammas employees to call for support without realizing the complexity of the IT service network.

The client dominion pattern may apply to the knowledge base of the user help desk that is maintained at Gammas site and Beta has to operate and integrate it. Here our IT service integration framework supports systematizing different forces in such patterns. For instance, from the perspective of application integration, web services may be a perfect solution to integrating heterogeneous applications. However, from a service perspective the idea of information hiding and encapsulating data processing is incompatible with IT security policies of most IT service customers. In case of a client dominion the usage web services have to be accompanied by a risk and security management as well as governance processes that define how data from the client dominion is used.

CONCLUSION & OUTLOOK

In this paper we discussed the challenge of selective sourcing in the IT service industry. Selective sourcing results in complex service network with various dependencies between service customers and service providers. Based on a running case we discussed elements of service integration and constructed an IT service integration framework capturing necessary elements of IT service integration. Here we have identified two pattern candidates of service integration. Then we have demonstrated the benefit of using such patterns in decomposing and describing complex service integration requirements. Although our research is still in progress, our pattern-based approach already supports decomposing complex service integration scenarios.

In our on-going research based on case studies (cf. Yin 2003) we are going to extract more integration patterns from integration projects and develop our pattern candidates to a pattern language (Alexander et al. 1977). Furthermore we are evaluating our approach in these integration projects and thus establishing our pattern language. Furthermore we aim at integrating our integration patterns with other pattern languages. For instance Hohpe et al. (2005) have collected various enterprise integration patterns. Here we plan to capture and combine existing academic and practical knowledge on integration IT services.

[1] A more detailed description of the patterns would go beyond the scope of this paper. For the same reason reference to other patterns are not depicted.

ACKNOWLEDGEMENT

This research is part of the MARIS project. The MARIS project is part of the research program “Innovative Services” that is funded by the German Ministry for Education and Research (Contract No. 01HQ0520). Members of the project consortium are Ruhr-Universität Bochum, University of Erfurt, Technische Universität München and various industry partners.

REFERENCES

1. Alexander, C. (1979) The timeless way of building, Oxford University Press, New York.
2. Alexander, C., Ishikawa, S., and Silverstein, M. (1977) A Pattern Language: Towns, Buildings, Construction, Oxford University Press, New York.
3. Böhmann, T., Junginger, M., and Krcmar, H. (2003) Modular Service Architectures: A Concept and Method for Engineering IT Services, IEEE Computer Society Press, *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS-36),* Big Island, Hawaii, 2003.
4. Bullinger, H.-J., Scheer, A.-W., and Grieble, O. (2003) Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen, Springer, Berlin.
5. Cohen, L., and Young, A. (2006) Multisourcing: Moving beyond outsourcing to achieve growth and agility, Harvard Business School Press, Boston, Mass., USA.
6. Conrad, S., Hasselbring, W., Koschel, A., Tritsch, R., and Pedersen, S. (2006) Enterprise application integration : Grundlagen, Konzepte, Entwurfsmuster, Praxisbeispiele, (1. ed.), Elsevier, München.
7. Dayal, U., Meichun, H., and Ladin, R. (2001) Business Process Coordination: State of the Art, Trends, and Open Issues, *Proceedings of the 27th VLDB Conference,* Rome, Italy, 2001.
8. Fensel, D., and Bussler, C. (2002) The Web Service Modeling Framework WSMF, Semantic Web Service Initiative.
9. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994) Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Publishing Company, Reading, Massachusetts.
10. Hohpe, G., Woolf, B., and Brown, K. (2005) Enterprise integration patterns: designing, building, and deploying messaging solutions, (6. print. ed.), Addison-Wesley, Boston.
11. Kaas, K.P. (1995) Kontrakte, Geschäftsbeziehungen, Netzwerke: Marketing und neue Institutionenökonomik, Verl.-Gruppe Handelsblatt, Düsseldorf.
12. Ludwig, H., Keller, A., Dan, A., King, R.P., and Franck, R. (2003) Web Service Level Agreement (WSLA) Language Specification.
13. Nuseibeh, B., and Easterbrook, S. (2000) Requirements engineering: a roadmap, Proceedings of the Conference on The Future of Software Engineering Limerick, Ireland, 2000, 35-46
14. Office of Government Commerce, G.B. (2001) Service Delivery : ITIL Managing IT Services, (6. ed.), TSO (The Stationery Office), London.
15. Papazoglou, M.P., and Georgakopoulos, D. (2003) Service-Oriented Computing, *Communications of the ACM,* 46, 10, 25-28.
16. Sturm, R., Morris, W., and Jander, M. (2000) Foundations of Service Level Management, Indianapolis, SAMS.
17. Yin, R.K. (2003) Case study research : design and methods, (3. ed.), Sage, Thousand Oaks, Calif.