Process-Oriented Integration of Applications for a Service-Oriented IT Management

Integrated IT Management Architecture

Christian Mayerl
Working Group ALFI at the Dean's Office
Faculty of Computer Science, Universität Karlsruhe (TH)
76128 Karlsruhe, Germany
mayerl@ira.uka.de

Frank Tröscher, Sebastian Abeck
Cooperation & Management, Institute of Telematics
Faculty of Computer Science, Universität Karlsruhe (TH)
76128 Karlsruhe, Germany
{troescher | abeck}@cm-tm.uka.de

***Abstract*—During the last years a shift from technology-oriented management to user-centered and service-oriented management of information technologies could be observed. Today the aim of service providers is not only to run necessary technologies but also to recognize user needs on service functionality and quality. In order to fulfill service levels agreed with customers, providers have to define and implement new processes for service management. To execute these processes efficiently in conjunction with traditional management processes, applications supporting service-oriented management have to be developed. This paper presents an approach how demands on applications can be derived from service management processes as extensions of traditional management processes. To extend traditional management applications with additional service management applications we propose an integration based on a service-oriented architecture.**

Keywords—Service Management, Process Modeling, Service Level Management, Management Application, Integrated Management Architecture, Web Service

I. INTRODUCTION

Information Technology (IT) and business are becoming almost inseparable. IT is a key driver for organizational efficiency and effectiveness and today's IT departments face great challenges. As IT and business are intertwined as never before IT failures are now business failures, suggesting the need for quality-conscious IT management. Today's business world demands an absolutely reliable IT. As such demands on IT and IT departments are very costly outsourcing IT is becoming more common. All of these circumstances inevitably led to the definition of roles involved in using and providing what no longer is called IT, but IT service. Now, enterprises are customers of IT departments, whether they are in house or outsourced. With this customer focus technology-oriented IT departments are being transformed into customer-centric IT service providers [1]. For a successful transformation it is essential that IT service providers have structured and proven IT management processes, which are subsumed under the term IT service management. In this paper the specification of demands on applications to support a service-oriented management of IT is presented. Thereby, it must be recognized that the paradigm change from traditional IT management to service–oriented management is not a revolution but an evolution. Keeping this in mind we have to investigate how existing management architectures and applications have to be extended supporting service-oriented management processes.

Therefore, the following questions have to be answered:

1. Which are additional activities and processes for a service-oriented IT management? To understand how management architectures have to be extended necessary processes of service management have to be analyzed at first. To derive demands on service management applications, processes should be modeled in a methodical way.

2. Which relationships and interfaces can be defined between traditional management activities and service-oriented management activities? After we have recognized that service orientation is not a revolution but an evolution, relationships and interfaces between new activities and traditional ones have to be clarified. Thereby, the flow of management information and messages along processes is of special interest.

3. How can traditional management architectures be extended to support service-oriented management processes? To support processes for service-oriented IT management demands on applications extending existing management applications have to be specified. Therefore, new service management applications have to be integrated with traditional IT management applications.

To answer the questions above, existing research, development and standardization achievements of the IT service management community must be taken into account. Best-practice processes are described in [2] as de-facto standard for the development of management applications.

Approaches to formalize management information can be found as part of management architectures.

Also in the area of Web service technologies formalisms to process service level agreements automatically are developed.

But there is no continuous approach how processes for service-oriented IT management can be mapped to adequate management architectures and applications. Understanding the shift to service orientation as an evolution, architectures must support the migration from traditional IT management and must be flexible enough to allow further adaptations.

In this paper we present our approach to specify demands on management applications derived from IT service management processes. Based on traditional IT management architectures service management applications have to be integrated according to the business processes of a service provider. To do this in a flexible way we apply a service-oriented integration on an architectural level. In the following section we discuss work most related to our approach. Some of the existing results are used as a starting point. Section 3 focuses on service management processes from which demands are derived and detailed in section 4. Section 5 describes an architecture integrating different management applications that we implemented and present as a proof of concept in section 6.

II. Related Work

The IT Infrastructure Library (ITIL) was initiated and developed in the late 1980s by and for the Central Computer and Telecommunication Agency (CCTA), a United Kingdom government agency, which today is the Office of Government Commerce (OGC). Since then ITIL has become a de-facto standard and the most widely used and accepted approach to define processes for service-oriented IT management [3, 4, 5].

Fig. 1 illustrates an ITIL-compliant provision of IT services whereby IT services enable customers to effectively and efficiently employ information technology to support and execute their business processes and are perceived by the customer as a coherent whole [6, 7]. As shown ITIL defines processes for Information and Communication Technology (ICT) infrastructure management and application management. Additionally, ITIL introduces and describes service support and service delivery processes. These processes are considered to be at the heart of service-oriented IT management. Both ITIL modules cover topics on service level management, financial management for IT services, capacity management, IT service continuity management, availability management, service desk, incident management, problem management, configuration management, change management and release management [2]. ITIL provides an overview of service-oriented IT management and defines processes, activities, roles and management information on a rather organizational level. This level is not sufficient to develop management applications which can be integrated along ITIL processes. Therefore, a common understanding needs to be established explaining what kind of messages are exchanged between processes and how these messages can be structured and formalized to be processed by management applications.

Before IT services can syntactically be formalized and correctly be processed a common semantic model has to be developed. Therefore, [8, 9] distinguish different dimensions of

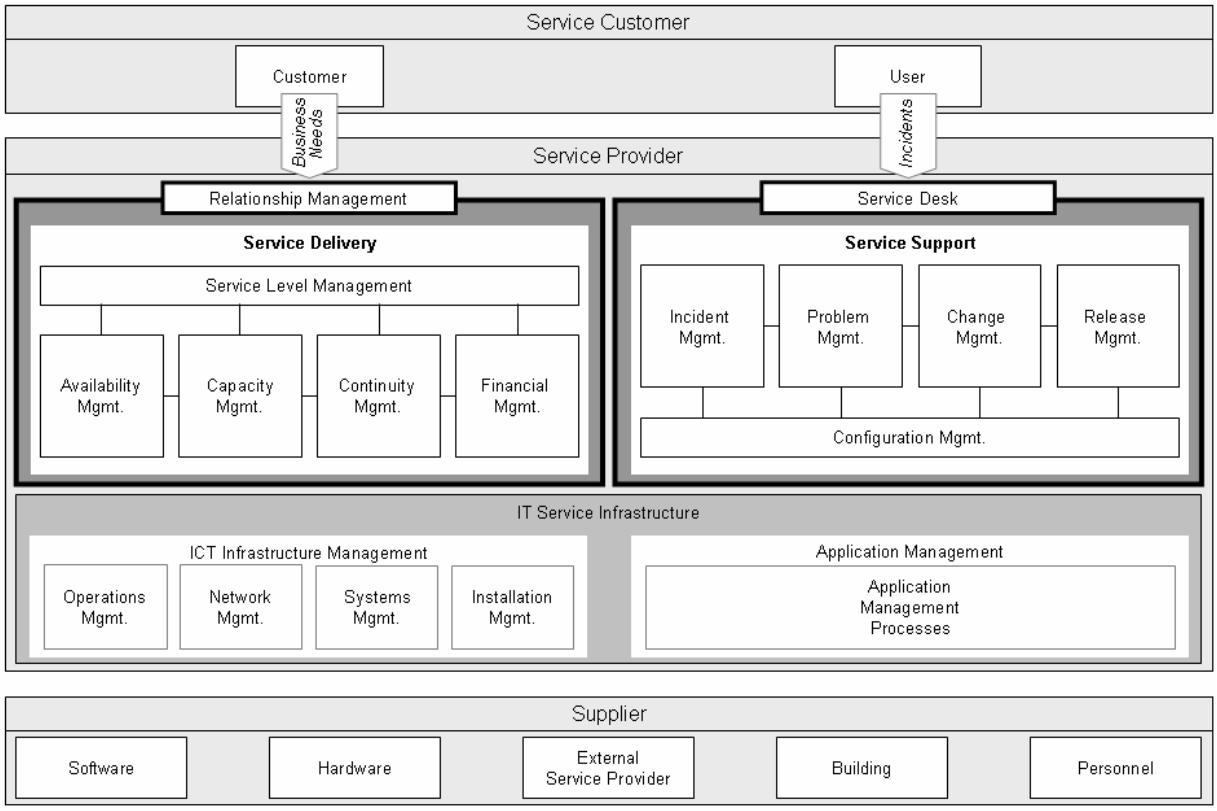

Figure 1: Structuring IT Operation Processes According to ITIL

a service model:

- The function dimension distinguishes primary and supporting functions. Primary functions directly support information processing along customers' business processes. Supporting functions would be additional functions provided to the service customer like training, help desk etc.
- The quality dimension refers to quality of service aspects. Quality can be managed following strategies in the range of best-effort provision up to SLA-compliant provision [2].
- The resource dimension is about the implementation of an IT service. Resources could be IT, personnel and others like buildings or patents [9].
- Depending on the view different IT service characteristics are important and focused on. These different views can be divided into end user view, customer view, provider view and supplier view [8, 2]. In order to process service management information and to develop adequate management applications it is especially important that IT service providers are aware of these different views.
- Adapted from [8, 10] the lifecycle of an IT service can be in one of the categories: design, build, deploy, operate and delete. An IT service can be in either of these stages. For example, an IT service would be in an operation stage after an SLA with a service customer has been established and the IT service is provided for the use by the service customer.

Generic service models have been elaborated in order to systemize services as development objects [11]. Looking for methods to structure and formalize management information different work can be investigated. As part of standardized management architectures the following examples are well-known: The Structure of Management Information (SMI) of the Internet Engineering Task Force (IETF) or the Common Information Model (CIM) of the Web-Based Enterprise Management (WBEM) initiative. Depending on the object which should be modeled and managed, one approach is sufficient or must be combined with another approach. Because information models like SMI and CIM are more technology-oriented and focused on network, system and application objects they do not support the modeling of IT services sufficiently. Recently, service level agreements and service level management protocols were developed associated with Web services. To automate the negotiation of SLAs [12, 13] use a XML-based language to formalize and process Web Service Level Agreements (WSLA). After agreeing on service levels, formalized SLAs can be processed for example to monitor the SLA compliance. Although the WSLA specification language was developed associated with Web services we adapt this language to formalize SLAs in general.

The described approaches answer our questions only partially. There is no continuous approach to derive demands from business processes of an IT service provider and to map these demands on an adequate architecture to integrate management applications along these processes.

III. Service Management Processes

To derive demands on applications and on an architecture that supports service-oriented IT management processes we focus on Service Level Management (SLM) as part of the ITIL service delivery processes.

SLM is the process of negotiating, defining, measuring, managing and improving the quality of IT services at an acceptable cost. Fig. 2 depicts a process diagram that illustrates SLM activities modeled with the Business Process Modeling Notation (BPMN). It also shows major deliverables created and exchanged during these process steps.

Three roles are involved in SLM activities: The service customer, the SLM department of a service provider, and suppliers. Service customers obtain and utilize IT services offered by the service provider. The SLM department is a representative of the service provider authorized to make agreements on the provision of IT services. Suppliers provide products and services necessary to provide IT services. Suppliers can either be external or internal ones, e.g. other departments within the IT service provider organization. As depicted in the process model (Fig. 2) SLM carries out the following activities and tasks:

The SLM department designs and catalogs IT services. The service catalog is made available to potential service customers who can choose necessary services. In order to be more efficient the concept of service modularization is employed. IT services are composed of service modules and service modules themselves again can be made up of other modules. Because of this modularization IT services can easily be adapted to different scenarios, as well as customized for different clients. Moreover, advantages of standardization and individualization can be utilized through this concept.

Triggered by a service request of a customer and based on the service catalog SLM establishes a Service Level Agreement (SLA) with the customer specifying how services are provided to the service customer. In the service measurement design as part of the SLA service levels are defined. Thereby both parties agree upon the quality of service provision by defining specific threshold values of IT services the service provider has to meet. In addition a report configuration is established that defines how the service provider communicates service level achievements to the service customer via SLA reports. As the SLA serves as a contract between both parties, representatives of the service customer and the service provider sign the established agreement. To substantiate SLAs SLM concludes underpinning contracts (UC) with external suppliers and operational level agreements (OLA) with internal suppliers.

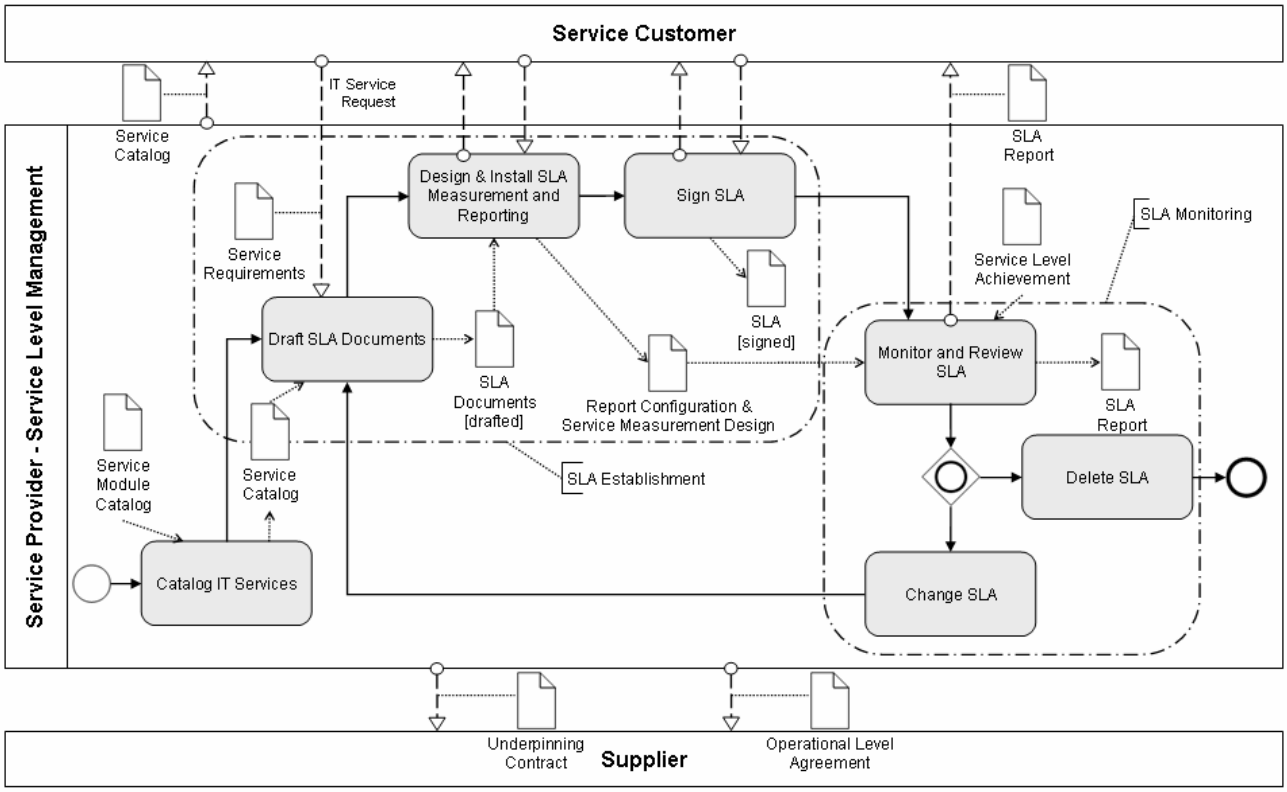

Figure 2: Service Level Management Process

Based on established SLAs, SLM monitors service levels agreed upon with the service customer to ensure the quality of IT services. Service level achievements are documented during this SLM activity. On the basis of service level achievement data SLM creates SLA reports to communicate the level of quality reached during the provision of IT services to the service customer. During regularly held SLA review meetings the delivered services are evaluated together with the service customer. Such meetings are held on the basis of SLA reports and the customer's perception of service provision. Any open issues are discussed and both parties agree on next steps including the possibility of SLA changes and even the closure of an SLA.

After establishing an understanding of the SLM domain and underlying activities requirements on applications that support the SLM process steps are derived.

IV. Requirements On Management Applications

To derive requirements on supporting applications, relationships and interfaces between traditional and additional service-oriented management activities are established in a methodical way. Based on the process understanding presented in the previous chapter, three major SLM activities can be identified: IT service cataloging, SLA establishment and SLA monitoring. To reduce the complexity of the SLM domain, requirements are established for each of the three areas separately. To establish an overview of the management information flow exchanged between SLM activities by messages, the three activities are further refined by modeling additional process diagrams. These diagrams zoom into activities and illustrate process steps of a lower abstraction level. Thereby they present more detailed process information, as well as additional process deliverables.

The process of cataloging IT services and making them available to potential service customers for example can be broken down into three process steps. In a first step service modules are described and collected in the service module catalog. Each service module description specifies properties of resources which are necessary to implement services. These service module descriptions (SMD) are compiled by the respective service module managers in technology-oriented departments. Then IT services, which are composed of service modules are described in service descriptions (SD) and put together in the service catalog.

In a third step the service catalog is published, so that service customers can choose IT services they want to obtain. Fig. 3 illustrates the outcome of the process refinement and depicts the overall management information flow between SLM activities. It also points to places where requirements on supporting applications have been identified and where distributed management applications have to be integrated.

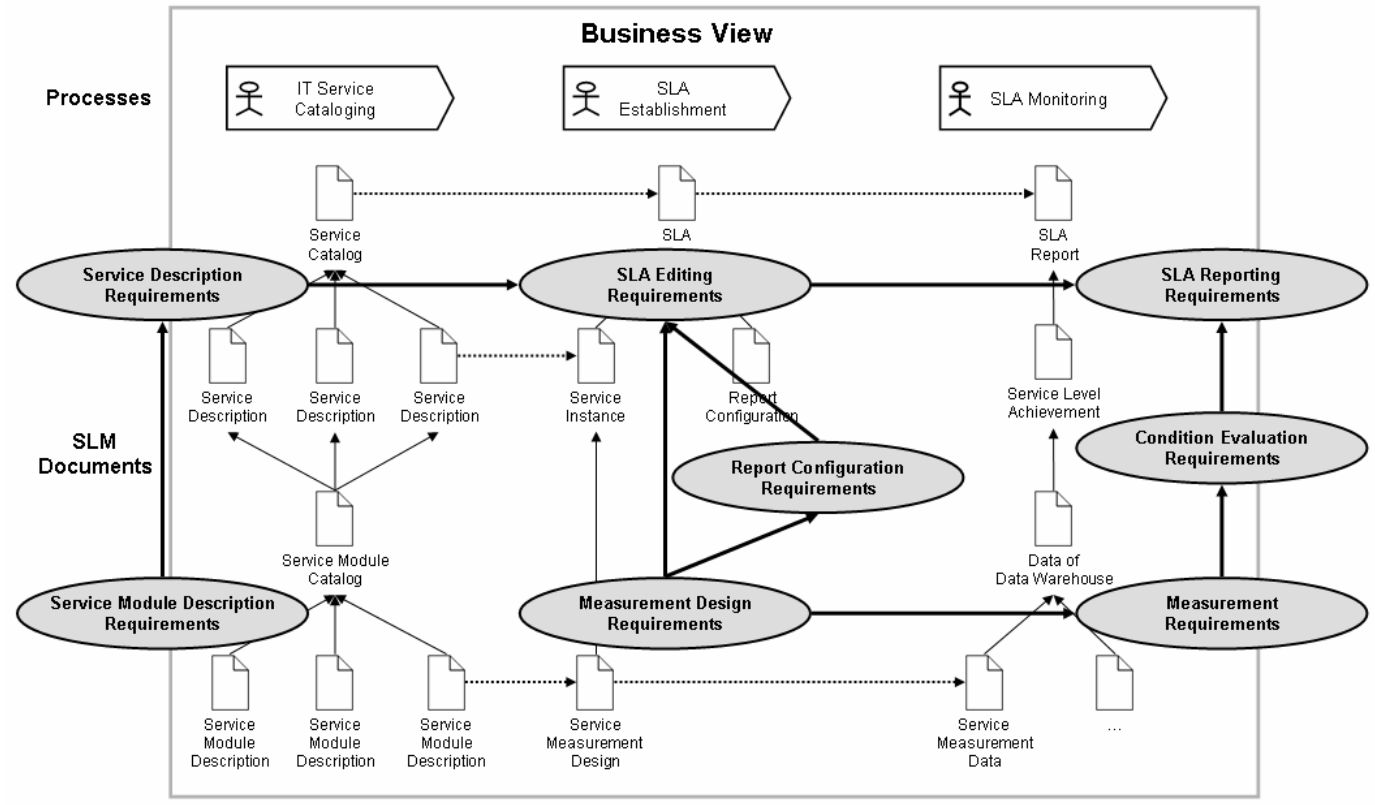

Figure 3: Requirements on Applications Processing Management Information

[Service Description] Service description functions support the SLM department during the creation, editing and deletion of services that are composed of service modules. They require interfaces to retrieve SMDs and to offer access to SDs.

[Service Module Description] To create and edit service modules, applications need to provide access to existing resource information to compile SMDs. They also need to offer interfaces, so that other applications can access information on service modules.

[SLA Editing] An SLA editing application offers IT support during the provision of SLA documents. It offers SLA templates and access to SDs to compose an SLA covering all services a customer wants to obtain. In addition editing tools allow SLM personnel to define threshold values of service levels and thus to describe service quality.

[Measurement Design] Measurement design tools offer assistance during the process of mapping SLA parameters to resource metrics. Resource metrics are available performance indicators of service modules and substantiate agreed SLA parameters. Therefore interfaces to traditional IT management systems that measure and hold the information of these performance indicators are needed.

[Report Configuration] On the basis of SLA parameters that have been agreed upon, report configuration tools enable SLM personnel to define how service level achievements are communicated to a service customer through SLA reports.

[Measurement] Based on measurement designs measurement support applications collect performance data from traditional IT management systems and associate this measurement information to services on a per customer basis. In addition to traditional measurement data, information of additional and service-oriented management activities like the service desk performance are gathered. Therefore interfaces to traditional and additional service-oriented management systems are needed.

[Condition Evaluation] Condition evaluation support applications compare the actual performance data with threshold values defined in SLAs. They trigger actions, e.g. sending notifications to the respective managers if threshold values are not met.

[SLA Reporting] Based on the service level achievement data gathered and compiled by measurement and condition evaluation support tools, SLA reporting applications assist SLA personnel by generating SLA reports. Thereby it creates these reports based on existing report configurations.

Besides these functional requirements support applications and an underlying architecture need to suffice fundamental requirements. Existing and additional support applications need to be integrated when shifting to service-oriented IT management. Management information from traditional existing applications are accessed and processed by new additional applications. In order to be fully aligned with the SLM process it is necessary that support applications can be

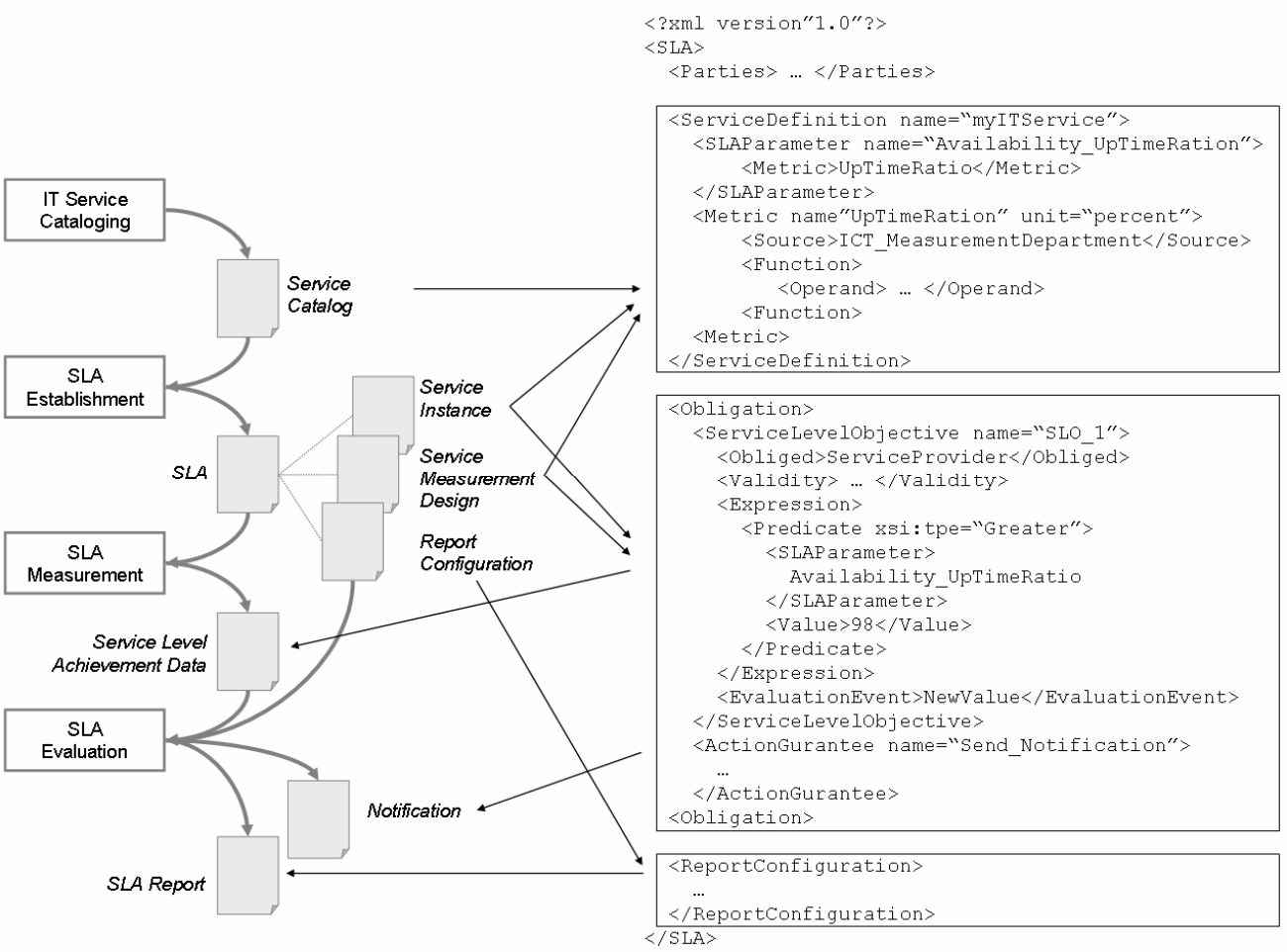

Figure 4: Relationships between Service Level Management Information

customized. An SLA editing tool for example needs to be flexible in a way that different SLA structures can be compiled. Workflow support is essential, as SLM is in the heart of service-oriented IT management, cooperating with various departments of a service provider. Often time critical actions to ensure SLA compliance need to be taken by SLM. Therefore SLM support applications need to provide workflow mechanisms to enhance the efficiency of SLM activities. Automation is another criterion that is important in the context of work efficiency. Especially in the reporting area, where SLM handles lots of data, automation can save plenty of time.

V. Integrated Service Management Architecture

To successfully cope with the information flow along management processes, it is important to develop a concept how this information can be exchanged between activities and applications by messages. Based on the WSLA approach of [12, 13] we propose an XML-based SLA description. Fig. 4 illustrates how this SLA description incorporates input data along the information flow presented in Fig. 3.

During the process of cataloging IT services, SDs are stored in the service catalog using XML. When establishing SLAs and specifying IT services, information from the service catalog can then be easily inserted into the XML-based SLA document. Furthermore, information of the service management design and the report configuration is integrated. Based on this SLA description performance data are measured, actual and threshold values compared and service level achievement data compiled. The report configuration stored in the SLA then determines how this data is communicated to the service customer.

An architecture supporting service-oriented IT management needs to be capable of integrating and composing management functionality along processes of an IT service provider. Furthermore, it should be highly flexible and adaptable to fulfill future needs on processes. To satisfy these requirements we propose an integration of management applications based on a Service-Oriented Architecture (SOA) [14]. Various characteristics of a SOA can be identified [15] and are reflected in the following definition from [16].

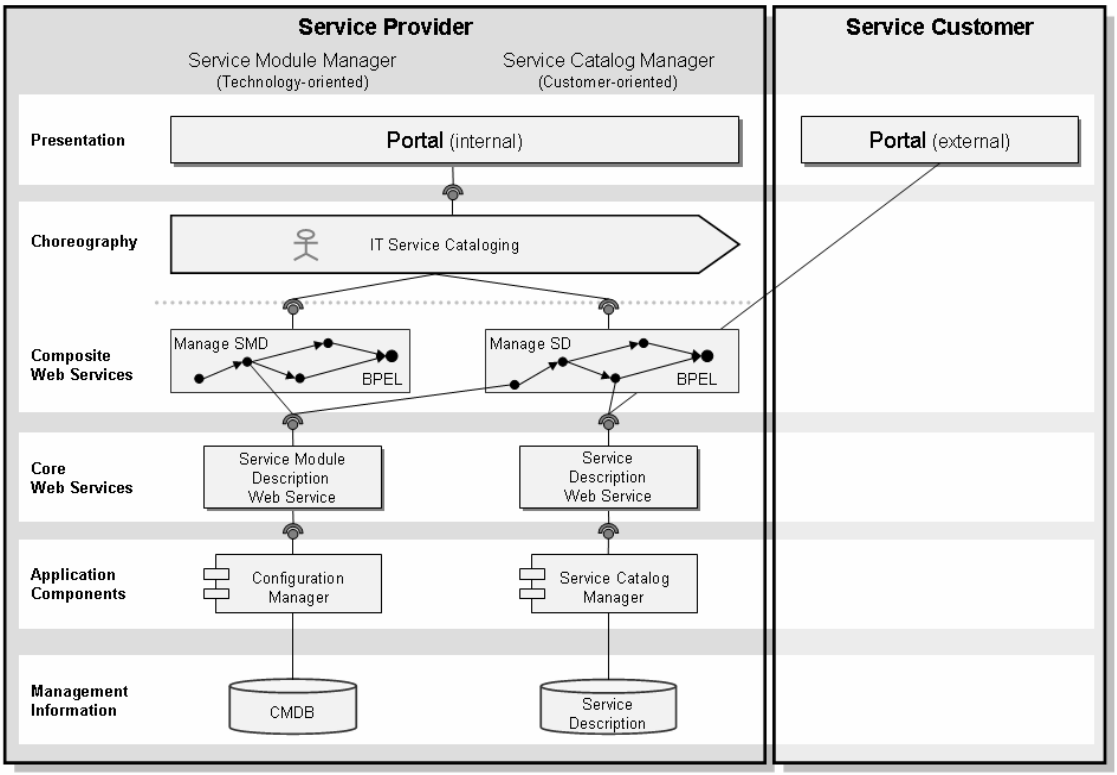

Figure 5: Integration of Management Applications Using Web Services

"SOA is the architectural style that supports loosely coupled services to enable business flexibility in an interoperable, technology-agnostic manner. SOA consists of a composite set of business-aligned services that support a flexible and dynamically reconfigurable end-to-end business process realization using interface-based service descriptions."

Fig. 5 illustrates the part of the SOA which focuses on the integration of management applications supporting the process of cataloging IT services. The architecture employs Web service technologies to offer and invoke functionality via service interfaces.

The SOA-based integration architecture consists of different layers: Traditional management applications, like a Configuration Manager together with a Configuration Management Data Base (CMDB), are extended by applications for service management, like a Service Catalog Manager (SCM) containing service descriptions. According to derived requirements on management applications (see Fig. 3) and because of the distribution of applications across different departments of the service provider, functionality of applications is offered as Web services. Examples are the Service Module Description Web service and the Service Description Web service (see Fig. 5). Via these Web services management applications can be loosely coupled at the Web service composition layer or at the choreography layer. Users can utilize applications in the context of their processes over a portal.

VI. PROTOTYPICAL IMPLEMENTATION

It can be ascertained, that the proposed integrated architecture for a service-oriented IT management contains different building blocks. To demonstrate the integration effort we paid special attention to the Service Catalog Manager (SCM) for example more attention [17]. In order to enable the SCM to participate in the SOA-based integration, adapters had to be added afterwards. In our scenario, Web services were chosen for implementing adapters. Enabling clients to interact with the SCM Web service, creating a corresponding interface description completed the adapter construction. In the case of Web services, a WSDL document contains necessary information required by clients to invoke the SCM Web service. Therefore, WSDL documents specify type definitions, message definitions according to SLM messages and operation definitions for input, output and fault messages. Involved management applications had to be wired along to activity sequences of service management processes. To perform the technical wiring, the activity sequence had to be mapped to a business process description and execution language. Therefore, the Business Process Execution Language (BPEL) is used to achieve process-oriented integration. BPEL specifies a formal XML-based description of automated business processes that can be composed of Web service invocations.

To implement the integrated IT service management architecture the BEA WebLogic platform has been chosen because of its comprehensive integration suite and free

availability for evaluation purpose. Furthermore BEA is one of the major players contributing to standards for SOA and process-oriented integration. The BEA WebLogic platform contains several components offering integration solution on different stages. In our case study the J2EE-compliant BEA WebLogic Application Server, the Beehive Application Framework and BEA WebLogic Integration have been focused upon. Additionally, the BEA WebLogic Workshop serves as an Integrated Development Environment (IDE).

Providing SOA conformity to the Service Catalog Manager (SCM), the BEA WebLogic platform offers two different approaches for adapter construction: First, Java controls provide high-level access to all kind of J2EE resources (EJB, JMS, JBDC, Java Mail, etc) concealing complex access logic. Recently submitted to the Apache Foundation within the Beehive project, Java controls allow establishing of native, high performance Java-based SOAs, but being intrinsically restricted to Java environments. Regardless of this fact, Java controls play an important role for establishing Web services in BEA. In doing so, BEA Workshop supports automatic derivation of Web services from Java controls. Web service communication involves some efforts, but ensures interoperability by leveraging well-established Internet communication protocols.

Therefore, the Service Catalog Manager is enhanced by a Java control and Web service interface to cope with both requirements, high efficiency in Java-based SOAs and highest possible degree of interoperability in Web service based SOAs. While the Java control is directly derived from SCMs EJB session bean façade, the corresponding Web service interface can map several Java control operations on one Web service operation for efficiency reasons.

BEA WebLogic Integration offers developer support at design and run time. During design time all kind of resources can be applied to build up an automated business process definition by graphical drag and drop Java controls and Web services. Behind the scenes Process Definition for Java (PD4J) is automatically generated, that is a precursor of BPEL and Java-enhanced BPEL/J.

Events and client requests can trigger business process execution. Furthermore process variables enable the definition of stateful processes. Finally, a mapping tool makes it easy to define data mappings between service calls by using XML technologies such as XML Schema, XQuery and XPath. BEA enables asynchronous Web service or Java Control invocations by callback mechanisms and achieves the loose coupling by buffering input and output messages.

VII. Conclusion And Outlook

With the evolution from traditional IT management to a service-oriented IT management we propose a SOA as integrating architecture for management applications. Therefore, a prerequisite is a sufficient understanding of service-oriented IT management processes. To map these processes to the architecture functional demands on applications have to be derived. Existing traditional management applications can be loosely coupled with additional service management applications according to processes of a service provider.

Further work will be done to model and formalize necessary management information inside management processes. On the one hand there are widely used standards like SMI and CIM to model IT-related management information about network, system resources and application elements. On the other hand there are approaches like WSLA to specify service level agreements at Web service interfaces. At a next step we want to investigate if it is possible to define a common model containing IT-related and service-oriented information. This model will be a prerequisite for the integration of further management applications for example to map resource properties to SLAs and to monitor SLA compliance.

References

[1] Axel Hochstein, Rüdiger Zarnekow, Walter Brenner: Evaluation of Service-Oriented IT Management in Practice, Proceedings of the International Conference on Services Systems and Service Management ICSSSM 2005, Volume 1, 2005.

[2] Office of Government Commerce (OCG): IT Infrastructure Library (ITIL) – Service Support (ISBN 0113300158), 2000; Service Delivery (ISBN 0113300174), 2001; Planning to Implement Service Management (ISBN 0113308779), 2002; Application Management (ISBN 0113308663), 2002.

[3] Jan van Bon, Georges Kemmerling, Dick Pondman: IT Service Management, An Introduction, itSMF Australia, May 2002.

[4] David Pultorak: An Introduction to IT Service Management, ITIL and MOF, Reprint from the September/October 2002 issue of DCM, 2002.

[5] Collin Rudd: The IT Infrastructure Library, An Introductory Overview of ITIL, Version 1.0a, itSMF, 2004.

[6] Hans-Georg Kemper, Evangelos Hadjicharalambous, Jörg Paschke: IT-Servicemanagement in deutschen Unternehmen – Ergebnis einer empirischen Studie zu ITIL, Praxis der Wirtschaftinformatik, HMD 237, 2004.

[7] Frank Niessink, Viktor Clerc, Ton Tijdink, Hans van Vliet: The IT Service Capability Maturity Model, Release Candidate 1, 2005.

[8] Gabi Dreo-Rodosek: Quality Aspects in IT Service Management, Proceedings of the 13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2002, 2002.

[9] TeleManagement Forum (TMF): SLA Management Handbook – Enterprise Perspective, Volume 4, published by The Open Group, 2004.

[10] Distributed Management Task Force (DMTF): Understanding the Application Management Model, CIM Version 2.7, Document Version 1.2, 2003.

[11] Hans-Jörg Bullinger, Klaus-Peter Fähnrich, Thomas Meiren: Service Engineering – Methodical Development of New Service Products, 2003.

[12] Alexander Keller, Heiko Ludwig: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services, Journal of Network and Systems Management, Seite 57 – 81, Springer, 2003.

[13] Heiko Ludwig, Asit Dan, Robert Kearney: Cremona: An Architecture and Library for Creation and Monitoring of WS-Agreements, ICSOC'04, New York, 2004.

[14] Mark Endrei, Jenny Ang, Ali Arsanjani, Sook Chua, Philippe Comte, Pål Krogdahl, Min Luo, Tony Newling: Patterns: Service-Oriented Architecture and Web Services, IBM Reedbooks, 2004.

[15] Jan-Peter Richter, Harald Haller, Peter Schrey: Serviceorientierte Architektur, Informatik Spektrum, Band 28, Heft 5, October 2005.

[16] Ali Arsanjani, Bernhard Borges, Kerrie Holley: Service-Oriented Architecture, Published in DM Direct, November 2004.

[17] Christian Mayerl, Tobias Vogel, Sebastian Abeck: SOA-based Integration of IT Service Management Applications, IEEE International Conference on Web Services (ICWS 2005), Orlando / Florida, 2005.