

On Tool Support for Service Level Management: From Requirements to System Specifications

Thomas Schaaf
Munich Network Management Team
Ludwig-Maximilians-Universität (LMU)
Munich, Germany
E-mail: schaaf@mnm-team.org

Michael Brenner
Munich Network Management Team
Leibniz Supercomputing Center
Garching/Munich, Germany
E-mail: brenner@mnm-team.org

Abstract—Service Level Management (SLM) is a vital discipline in customer-oriented IT Service Management. Covering technical as well as organizational and economic aspects, this multi-dimensional management area has become even more important against the background of the business-driven IT Management (BDIM) paradigm.

Today, tools and management solutions available for SLM face three major challenges: First, the lack of an established platform-independent model (PIM) for SLM entails the existence of hard-to-integrate “islands” of support tools and makes an integrated approach to SLM tool support all but impossible. Secondly, the current solutions are mostly focused on SLA-specific issues, neglecting significant SLM responsibilities such as the deployment of service catalogs. Lastly, the limited perspective of many existing SLM solutions disregards the important links between SLM-related management tasks and those of other management disciplines such as Fault or Performance Management.

This paper presents first results of ongoing research aiming at developing a practicable, integrated solution for SLM, addressing above mentioned challenges. To this end, essential requirements are pointed out, and four common modules of a management architecture for SLM are outlined.

I. INTRODUCTION

Service Level Management (SLM) is often regarded as the core management discipline in *IT Service Management (ITSM)*. Vital for customer-oriented provision of IT services, SLM is responsible for managing a provider’s IT service portfolio and monitoring and reporting levels of service performance. Through establishing a common understanding of expectations and achievements, SLM strives to improve the relationship between the IT provider and its customers. SLM covers the management of all types of *Service Level Agreements (SLAs)*, documentation of IT services and maintenance of service catalogs.

Through the years, numerous solutions for supporting SLM tasks have evolved in research and industry. These approaches differ significantly in scope, level of detail, the degree with which they address questions of tool support and how widely they have been adopted in practice.

In particular, best practice frameworks for ITSM like *IT Infrastructure Library (ITIL)*, *Microsoft Operations Framework (MOF)* or *Control Objectives for Information and Related Technology (COBIT)* are enjoying great popularity at present. Most of these frameworks are process-oriented, recommending

the adoption of specific IT processes to help providers become more effective in the delivery of IT services, and achieve a higher efficiency in the utilization of resources (i.e. technology, monetary and human resources).

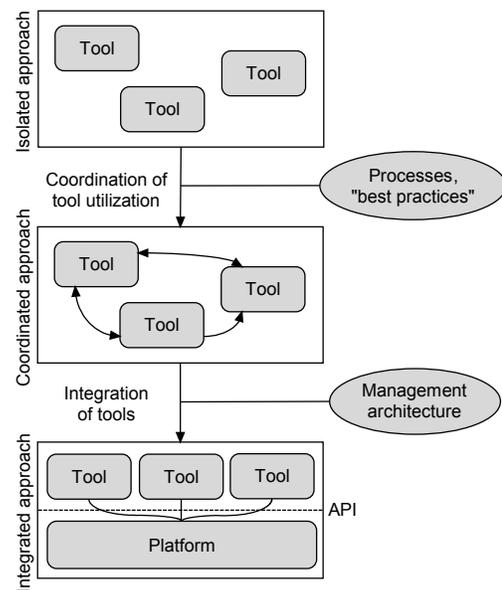


Fig. 1. Classification of management solutions with respect to their degree of integration

Establishing such processes – including those addressing SLM issues – helps to strengthen cooperation between the parties involved in ITSM and allow a *coordinated approach* towards SLM, as illustrated in figure 1. However, just adopting a process-oriented, coordinated approach does not guarantee that support tools share the same platform, that data formats are unified or that a standardized information exchange mechanism between tools exists. Consequently, realizing the full advantages of deploying SLM processes requires comprehensive and integrated tool support, for which these frameworks provide hardly any guidance.

This paper presents first results of ongoing research in addressing this gap, outlining a management architecture aiming at an *integrated approach*. The main objective of this

management architecture is to provide a framework for the development of concrete SLM systems. Figure 2 illustrates the general course of action that borrows some principles from the *Model-driven architecture (MDA)*, a software design approach published by the Object Management Group (OMG). Its main trait is the top-down forward engineering paradigm, starting with requirements and abstract specifications, and ending up in concrete platform-specific software modules. This paper presents results from the first four out of the six steps on the way to integrated SLM.

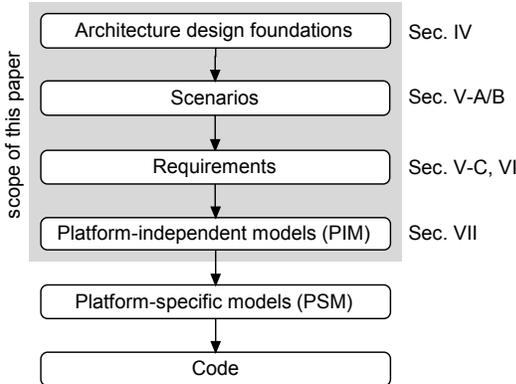


Fig. 2. Top-down approach using MDA-based techniques

In summary, the main contribution of the presented management architecture for SLM is threefold:

- 1) Provide a solid *requirements base* for an SLM-supporting management system.
- 2) Provide structured, platform-independent guidance for the *design of an integrated management system*, where integration means both: (i) integration of SLM-specific tasks in the context of a management system (ii) integration of SLM with the other ITSM disciplines (e.g. Incident/Problem Management, Configuration/Change Management, Performance Management)
- 3) Provide *guidance in transforming* the platform-independent models into software solutions¹.

The remainder of this paper is structured as follows: Section II shortly introduces selected terms and concepts in the context of SLM. In section III we give an overview of related work in the area of SLM, focused on management frameworks and architectures, tool support and information modeling. Section IV outlines the main concepts selected and applied within the design process towards our management architecture. Section V shows how the requirements on a management system for SLM have been derived, and in sections VI and VII selected contents of the approach are presented. The paper concludes with a short summary and an outlook on the next steps in section VIII.

¹to be addressed in future work

II. TERMS AND CONCEPTS

The following set of terms and definitions gives a brief overview of the “managed objects” from an SLM perspective. The definitions are mainly adopted from or in accordance with the IT Infrastructure Library (ITIL) [1] – a best practice framework and de-facto standard in ITSM – and the definitions given by Lewis [2] (see section III).

- **(IT) service – value-oriented definition (ITIL):** A means of delivering value to one or more customers by facilitating outcomes these customers want to achieve. Typically, an IT service is nothing an enterprise sells to its consumers (end customers). Instead, IT services support manufacturing of products or providing consumer services by either increasing the efficiency of the manufacturing process (e.g. by automation) or by removing natural or technical constraints from this process [3].
- **(IT) service – technical definition (Lewis):** An abstraction over components (including network devices, systems, applications, communications media) in an enterprise infrastructure. A more detailed examination of services is given by the MNM service model [4].
- **(IT) service provider:** Totality of humans, teams and functions in an enterprise, involved in the delivery and support of IT services.
- **Service level:** Range of allowed values for a set of (QoS) parameters representing the acceptable quality of an IT service.
- **Service portfolio:** The entirety of all IT services – requested, planned, or in operation – of a service provider.
- **Service catalog:** A structured document (or database) containing information about a provider’s services in operation or ready for deployment. ITIL differentiates between the *Business Service Catalog*, addressing a (prospective) customer and containing descriptions of services in a predominantly non-technical manner, and the *Technical Service Catalog* providing technical specifications of the services’ functionalities.
- **Service level agreement (SLA):** A contract between a service provider and a service customer/subscriber, containing a description of the service functionality in accordance to the service catalog, defining related service levels and declaring responsibilities of both parties. An SLA may also contain prices for service usage to pay by the customer/subscriber and penalties for service level violations to pay by the service provider.

These terms form the context of SLM that is relevant for the rest of this paper.

III. RELATED WORK

In the context of SLM, various efforts have been undertaken in the last decade. Although still today an integrated, tool support-driven management approach covering the entire field of SLM is not available, many pieces of the puzzle have already been addressed. In this section, we present completed and ongoing work in the area of SLM in three categories:

frameworks and architectures, tool support and information modeling.

A. Frameworks and architectures for SLM

ITSM frameworks provide guidance in the entire field of ITSM by describing management processes and principles. A popular and widespread framework is the *IT Infrastructure Library (ITIL)* whose third revision has been published in 2007. Service Catalog Management and Service Level Management are two of ITIL's core reference processes [5]. Besides ITIL, other frameworks or framework-like approaches provide relevant code of practice for SLM:

- *ISO/IEC 20000* [6], [7] – an international standard for ITSM, successor of the BS 15000 standard which was based on ITIL (version 2).
- *Microsoft Operations Framework (MOF)* – provides a process model, a team model and a risk management model for ITSM, based on ITIL (version 2).
- *Control Objectives for Information and Related Technology (COBIT)* – 34 processes covering about 200 control objectives for ITSM.
- *Enhanced Telecom Operations Map (eTOM)* – management guidelines, processes and data models especially for telecommunication providers, focused on, but not limited to IT-related management tasks.
- *The NGOSS SLA Management Handbook* [8] – guidelines for SLA management.

In a work presented on last year's BDIM workshop [9], two of the above mentioned frameworks have been analyzed and compared, namely ITIL (version 2) with its SLM reference process and the NGOSS SLA Management Handbook. A set of evaluation criteria has been derived from an IT scenario in order to elaborate possible areas of conflict as well as complementary fields and unaddressed issues. However, even a combination of these frameworks does not yield sufficient coverage of all requirements for an integrated approach towards SLM, especially with respect to automation and tool support (cf. section III-B).

Not least, an early examination of the field of SLM was performed by Lewis [2]. Based on case studies, it laid the foundation for a common understanding of SLM terms and concepts (cf. SLM conceptual graph [2, p. 43]). The presented SLM methodology and architecture are based on the idea of an SLM system consisting of collaborating intelligent agents. The focus is on aspects of artificial intelligence (AI), robotics and data warehousing. By contrast, the management architecture introduced in this paper aims at integrating SLM with other management disciplines and providing guidance in designing management systems from an ITSM perspective.

B. Automation and tool support in ITSM

Not specific to SLM, some recent work has addressed the issues of supporting ITSM processes in general [10]. The models and architectural blueprints developed there have however concentrated on support by workflow management tools. In this context, no architecture for SLM was developed.

In the taxonomy for classifying ITIL processes under tool support aspects [11], SLM has been characterized as a process with low recurrence, high lead time, a medium organizational complexity and very low structure – a challenging combination with respect to the development of support tools. This analysis is also attested by today's available range of SLM tools which is significantly behind the number of tools in other areas like Incident Management.

C. Information modeling in ITSM

Another important topic of interest – when having in mind the design of a management architecture – is given by the available approaches in ITSM information modeling. As we will see later on, a basic component of a management architecture is its information model.

In a research paper by Brenner et. al. [12], different information models from practice and research – among them the Common Information Model (CIM) and the Shared Information/Data Model (SID) – have been analyzed with respect to their capability to model ITIL information artifacts. The SID [13] has turned out as most promising, since SID has been developed to support and underpin eTOM's information artifacts and was the first information model tightly coupled to management processes. The SID model employs an object-oriented modeling approach and draws a clear distinction between the system and business view on management information. Section VII explains, how management model concepts from SID are being re-used in order to model SLM information artifacts in the context of our management architecture.

An approach addressing the specific information demands of SLA Management has been published by Debusmann et. al. [14], [15]. Similar to our approach, MDA-based techniques have been applied. But in contrast to the management architecture presented here, the work of Debusmann focused on SLA-centric data modeling. In this capacity, the results serve as an excellent input for the information model of our management architecture. Moreover, the relationships between SLM and other ITSM disciplines have not been considered. Since process frameworks as the ones presented in III-A enhance cooperation between different management disciplines, a supporting management system should take this circumstance into account as well.

D. Related Work: Conclusion

In summary, manifold issues related to SLM have been addressed in existing work. Existing work on SLM tools has so far not sufficiently incorporated the paradigm of process orientation, advertised by the best practice frameworks which are currently being adopted in many IT organizations. These frameworks in turn neglect to a large degree tool support issues. Conceptual approaches towards developing software solutions for supporting ITSM processes have so far largely concentrated on management disciplines other than SLM and are not directly transferable. An effective and efficient deployment of SLM in an IT organization demands a comprehensive

and tool support-oriented solution – this is the goal of our project.

IV. DESIGN FOUNDATIONS FOR A MANAGEMENT ARCHITECTURE

IT Management architectures aim at enabling tool support for their operational scope. Thus in the context of SLM, we regard a management architecture as the *entirety of requirements, models and design guidelines contributing to the development of a concrete IT-supported management system* covering the various tasks in SLM.

A. Components of a management architecture

A common and approved proceeding in designing a management architecture consists of developing an organizational model (OM), a functional model (FM), an information model (IM) and a communication model (CM) which altogether form the architecture’s core elements (cf. figure 3). Proceeding this way helps to reduce complexity and dictates a structured course of action in the development process towards the complete management architecture.

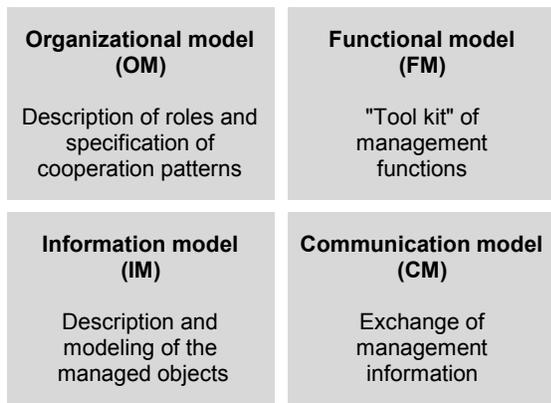


Fig. 3. Components of a management architecture

Originally, the partition of a management architecture into these four partial models comes from the OSI management framework for integrated management of networked systems [16]. Although other management architectures in the area of managing distributed environments have achieved greater popularity, OSI management provides a *reference architecture* and hence helpful development guidelines.

B. Views on a management architecture

When developing a management architecture in a field as complex as ITSM, it is useful to have different perspectives on each partial model of the architecture as depicted in figure 4. These perspectives differ in their scope and level of abstraction regarding the model’s content. From a process view, the organizational model for example contains organizational domains, roles and parties/stakeholders in the context of the regarded management process. From a system architecture’s view (platform-independent model, PIM), the OM should

indicate how the management system’s tool components to be generally set up in the management environment, i.e. which components should be centralized, which ones should be distributed, which components act as clients or servers, where peer-to-peer solutions are used and how the components can be interfaced by human operators. A system architecture covers any information needed to design a management system except those ones being specific for a selected platform (e.g. J2EE, web services, etc.). Platform-specific contents of the OM are covered by the technology model (platform-specific model, PSM).

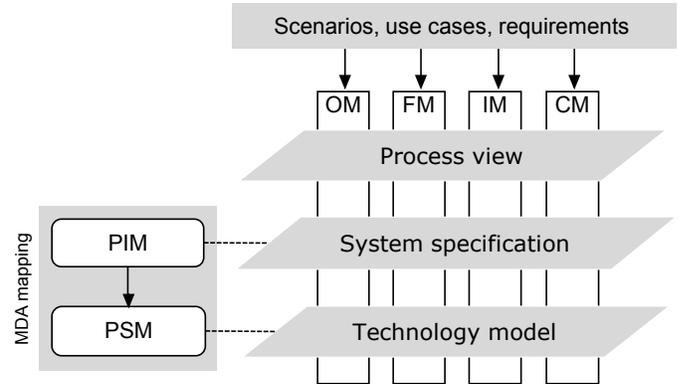


Fig. 4. Views on a management architecture

In the next section, we show in an outline of the requirements engineering process, how requirements on the SLM architecture have been derived from real scenarios and use cases.

V. REQUIREMENTS

A. A scenario-based requirements analysis approach

In order to ensure that the management architecture’s platform-independent model – and thus the system specification – becomes as generally applicable as possible, the elicitation of requirements is not limited to the examination of one single source. The following analysis considers SLM-related recommendations of the frameworks presented in section III as well as exemplary use cases identified in IT scenarios of different characteristics.

Afterwards, the use cases are analyzed with respect to organizational and functional requirements as well as requirements related to information modeling and definition of communication patterns. A generic management system model, borrowed from [16, p. 102] as depicted in the analysis tool level of figure 5, helps finding requirements by an analysis of the three general interfaces in an integrated IT-supported management system: The operator interface provides requirements on the management functions, the manager-agent interface provides criteria for the communication model, and information model requirements are derived by analyzing information needs of the agents operating on the respective information artifacts (MO access interface).

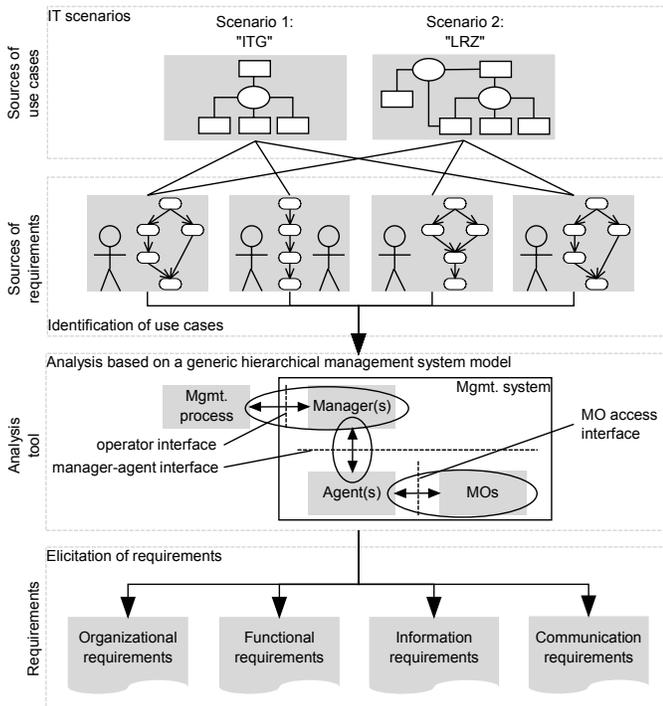


Fig. 5. Scenario-based requirements analysis approach

As a starting point, we shortly introduce the two scenarios employed for the requirements analysis:

Scenario 1: As a medium scale IT scenario, we consider a company in the field of logistics, providing business services in the area of transportation and shipping. The company employs about 2,800 people, amongst them 2,200 drivers (truck and bus drivers). The remaining 600 employees are located at three head offices and 20 branch sales offices all over Europe. IT operations are centrally managed from the main head office. The six main information systems (Admin, Shipping, Travel, Planner, Maintain, Office) in use have been either purchased from external producers and customized according to specific demands or have been designed and built by the internal software development team. While these information systems and the related IT services are operated and maintained by the internal technical operations team, some IT services (especially telecommunication services and a WAN service connecting the head and sales offices) are procured from external providers.²

Scenario 2: As a large scale IT scenario, we regard an academic and scientific data center providing both IT services for students and university employees as well as high performance computing services for the attached scientific institutes. Serving the majority of the Bavarian universities and colleges, among them the University of Munich and the Munich University of Technology, it is one of the biggest data centers in Europe, providing IT services to more than 100,000 users.

²This scenario is based on the "ITG" case study used for ITSM training purposes by the Examination Institute for Information Science (EXIN).

IT services include backup and archival storage, provision of groupware, file servers, web hosting, communication services like WLAN, e-mail, VPN and video conferencing as well as hosting special facilities like multimedia workstations and large size printers and scanners. All IT services are centrally managed.³

B. Identifying use cases

Each of these scenarios provides plenty of use cases that deliver numerous requirements on a management system for SLM. Some use cases may appear in both scenarios, others in just one of them. Evaluating the frequency of a specific use case may give a hint on how to weight and assess the requirements derived from this sample use case. The following is a description of one exemplary use case for each scenario.

Use case example 1 (from scenario 1): In order to allow end customers to track the status of their parcel transfer orders, the customer relationships department asks the IT for a new web application realizing this functionality. For that purpose, an online application needs to be developed, integrated with the web page, and data from the information systems Planner and Shipping need to be preprocessed in an appropriate way. In addition the web and application server capacities have to be reviewed with respect to the expected increase of working load.

Use case example 2 (from scenario 2): The institute of meteorology is taking advantage of the supercomputer for daily meteorological calculations. About one third of the 9,728 cores (6 of 19 compute partitions) of the SGI Altix 4700 high performance computer are dedicated for this purpose for a time frame of 1 hour per day. Since calculations are time-critical and results are required before 5 p.m., it is important that computing resources are not only delegated, but also highly available during this time frame and incidental failures are handled immediately. The institute thus wants to upgrade from the existing level of service.

C. Deriving requirements

Creating a requirements base according to the approach presented in section V-A means to analyze all identified use cases including the two exemplary cases presented above concerning their specific requirements on a management system in support of these use cases.

Example requirements 1 (from use case 1): A tool-based management system for SLM should support the customer in expressing requests for new services and enable the provider to extend its service portfolio in response to customer demands. Each new service has to be designed⁴, and basic options for service levels have to be defined, documented and offered to the customer. In order to extend the service portfolio, a close link to other ITSM disciplines – in particular Change and Configuration Management – has to be in place.

³More information about the Leibniz Supercomputing Center is available on its web site: <http://www.lrz.de>

⁴ITIL for instance proposes to create a *service design package (SDP)* which documents all aspects of an IT service through each stage of its life cycle.

Example requirements 2 (from use case 2): A management system should support the process of changing existing SLAs in order to meet new or changed business requirements. The management system therefore should not only support its operator in realizing the necessary changes to the related documents (contracts), but also take into consideration manifold constraints, dependencies and required information flows. For example, a (planned) change of a service level may require management action in the area of Performance Management, because new capacities are required or a higher service availability must be guaranteed. Moreover, Incident Management is dependent on information about any change in service levels in order to properly prioritize and efficiently handle upcoming future service incidents in accordance to the urgency determined by the SLAs.

Analogous, the following requirements have been derived from further use cases identified in the two scenarios:

- Handling of customer requests for new services
- Adding services to the portfolio
- Adding portfolio services to the service catalog
- Definition of realizable service levels and related costs
- Subscription of services by customers
- Customization of existing services
- Negotiation of SLAs for one or more services
- Closing of OLAs and Underpinning Contracts
- Contract administration including dependency discovery
- Handling of new/changed service level requirements
- Internal/external reporting of service level achievements
- Target-performance comparison and fulfillment reporting
- Periodical SLA and service catalog reviews
- ...

Basically, these use case-driven requirements describe provider and/or customer activities that should be supported by a management system. The next section applies the vertical partition of a management architecture (OM, FM, IM, CM) on the requirements in order to deliver a structured process view model as input for the system specification.

VI. SLM: PROCESS VIEW LAYER

The goal of this layer is to break down the high level requirements arising from the use cases, such as “*Handling of new/changed service level requirements*”, into specifications of organizational, functional, information and communication model requirements from a process perspective. In other words: The process view layer serves as the glue between the use case requirements and the system architecture models on the specification layer.

A. Defining organizational domains and roles

From an organizational view point, there are mainly two aspects that become visible in the scenario use cases and turn out as relevant for a constitutive management system’s design: *organizational domains* and *roles*. Figure 6 illustrates how roles taking on responsibility in SLM relate to the identified domains, which are the customer, provider and supplier domains. The roles determine human involvement in the context

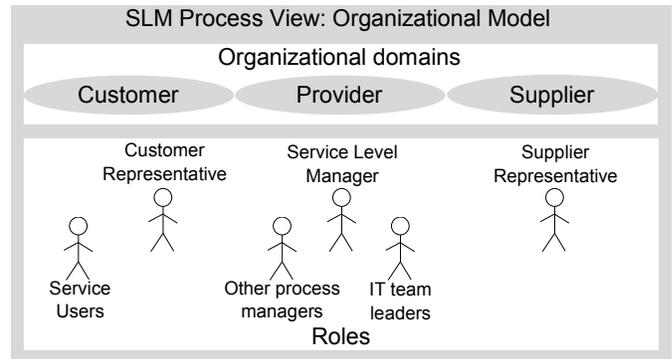


Fig. 6. SLM organizational model (process view layer)

of SLM and with that the “anchor points” of a management system in its environment. Where a specific role is responsible or involved into a certain SLM activity that shall be supported by the management system, a well-defined operator interface (e.g. a management console) has to be in place.

B. Defining management functions

SLM can be partitioned into two major functional areas: *IT Service Portfolio Management* and *Contract Management*. While IT Service Portfolio Management is responsible for managing information on all IT services being planned or actually operated, Contract Management is accountable for managing all kinds of contracts between the provider and its customers and suppliers including SLAs, Operational Level Agreements (OLA) and Underpinning Contracts (UC).

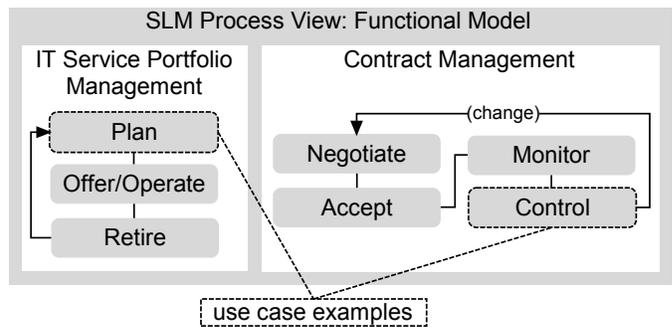


Fig. 7. SLM functional model (process view layer)

Figure 7 shows how these functional areas can be further broken down into more specific fields of activities according to a process- and lifecycle-oriented perspective. In the area of IT Service Portfolio Management these activities are:

- Plan: New services must be planned, related information documented and the respective service information entities added to the portfolio.
- Offer/Operate: Services ready for deployment are offered to the customer through the service catalog. Operational information on the service instance are documented here.
- Terminate: Services that are no longer provided have to be removed from the service catalog and service portfolio.

For the field of Contract Management, the following functional sub-activities have been determined according to the contract life cycle:

- Negotiate: Requests for IT services are expressed by customers, requirements and quality demands must be identified and negotiated.
- Accept: The offered service and related service level must be accepted by the customer and the provider.
- Monitor: Service delivery and the actual service level achievements must be monitored, deviations from the negotiated levels must be reported, countermeasures initiated and consequences (business impact, penalties) assessed.
- Control: Existing contracts have to be under permanent control of SLM, especially with respect to their compatibility with other/new contracts and required changes.

C. Defining information artifacts

In a management system, every management activity relates to or affects one or more information artifacts. According to Scherer [17], categories of information artifacts include *plans, documentations, reports, records, databases* and *“things”*. Contracts like SLAs or OLAs fall in the category of documentations since they document mutual intents concerning the delivery and subscription of IT services. From a process and requirements perspective, it is important to identify the required information artifacts, classify them, and define their goals and contents in a non-technical way. Figure 8 illustrates that the occurring information artifacts in the area of SLM, according to the functional areas, can be classified into *service portfolio artifacts* and *contract artifacts*. Inside each class, a categorization corresponding to the six categories mentioned above should be performed.

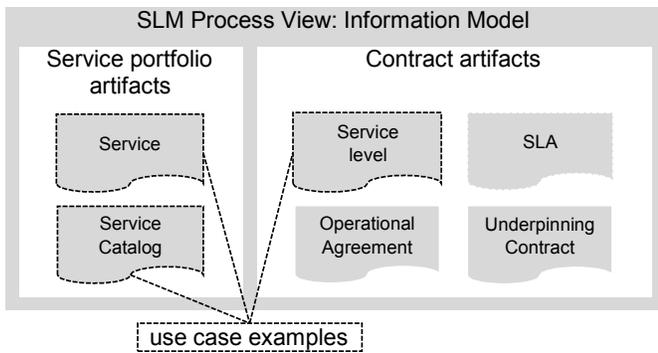


Fig. 8. SLM information model (process view layer)

At this stage the presented level of abstraction is sufficient to understand the basic requirements on the information model to be provided on the system specification layer.

D. Defining communication patterns

As mentioned earlier, one goal of our management architecture is to enable integrated management systems for SLM, where integration does not only mean to define a common

platform for different support tools, but also to embed SLM tools into the broad repertoire of ITSM tools. The process view communication model therefore plays an important role: It defines, how SLM relates to other ITSM disciplines. Thus, these disciplines have to be identified first.

Instead of presuming a certain set of management processes (like for example the more than 30 ITILv3 processes) being in place, we again borrow a concept from the OSI Management framework: As part of OSI’s functional model, the *FCAPS* provide a sufficiently detailed and at the same time non restrictive classification of IT management tasks [16]. Originally designed for pure network and systems management, we re-use the FCAPS for our purpose.

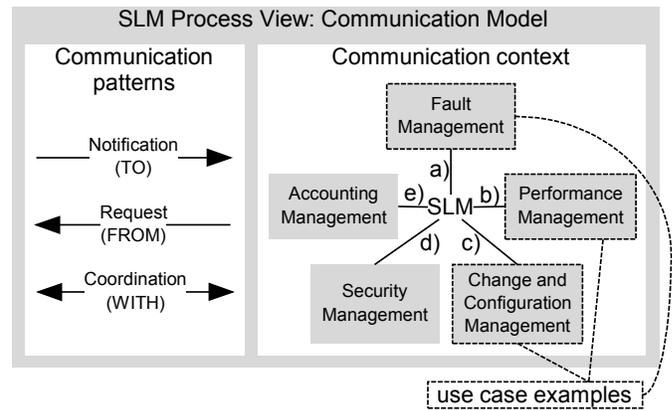


Fig. 9. SLM communication model (process view layer)

Table I shows an exemplary and intuitive mapping between the ISO/IEC 20000 management processes and the functional areas of Fault, Configuration, Accounting, Performance and Security Management (FCAPS). Fault Management for example encompasses Incident and Problem Management. Performance Management covers the processes of Capacity and Availability & Continuity Management as well as Service Reporting. Please note that OSI Configuration Management has a different scope than Configuration Management in ISO/IEC 20000 or ITIL. Nevertheless, this matrix shows that the FCAPS provide a self-contained set of functional management disciplines, helpful for analyzing relationships between SLM and its immediate environment. The process view layer communication model documents these relationships that serve as input for the later system specification.

The following gives a rough outline on the relationships, where \rightarrow means an information flow from SLM to the respective FCAPS discipline, \leftarrow means an information flow towards SLM, and \leftrightarrow expresses the necessity of coordination on the listed issues (cf. communication patterns in figure 9).

- a) Information exchange: SLM \leftrightarrow Fault Management processes
 - \rightarrow contracted resolution times, SLA-specific influences on prioritization of incidents
 - \leftarrow achieved resolution times (MTTR), first level success rate

ISO/IEC 20000 processes	Fault	Configuration/Change	Accounting	Performance	Security
Incident Management	✓				
Problem Management	✓				
Configuration Management		(✓)			
Change Management		✓			
Release Management		✓			
Budgeting & Accounting for IT services			✓		
Capacity Management				✓	
Availability & Continuity Management				✓	
Service Reporting				✓	
Information Security Management					✓

TABLE I
MAPPING: ISO/IEC 20000 PROCESSES AND FCAPS FUNCTIONAL DOMAINS

- ↔ realizable resolution times and cost dependencies
- b) Information exchange: SLM ↔ Performance Management processes
 - availability and capacity demands
 - ← achieved availability, actual capacity shortcomings
 - ↔ realizable IT service performance and cost dependencies
- c) Information exchange: SLM ↔ Change/Configuration Management processes
 - technical service demands/specifications, SLAs, OLAs
 - ← operation readiness notifications, test results
 - ↔ feasibility of requested services or changes
- d) Information exchange: SLM ↔ Security Management processes
 - customer-specific information security requirements
 - ← SLA violations affecting information security
 - ↔ realizable information security levels and related costs
- e) Information exchange: SLM ↔ Accounting Management processes
 - IT service accounting, charging and usage data
 - ← total costs of ownership for IT services
 - ↔ customer- and service-specific pricing

E. Process view layer: Summary

The four abstract models presented in this section have been derived from the scenarios by identifying use cases and analyzing these use cases by means of the requirements engineering approach presented in section V. The results build the first step towards a management architecture for SLM. So far, nothing has been said about concrete system design. To continue, the models of the process view layer have to be refined into system architecture models. Since these models become quite voluminous, for now we focus on the presentation of a small fraction of the system specification layer in the following section.

VII. SLM: SYSTEM SPECIFICATION LAYER

Revisiting the exemplary use cases from section V, this section gives a specific outline of the specification of a management system covering the presented requirements. Section

VII-A illustrates OM, FM and CM from a system specification point of view, and section VII-B illustrates a resulting management system's setup.

A. Extract of the platform-independent models

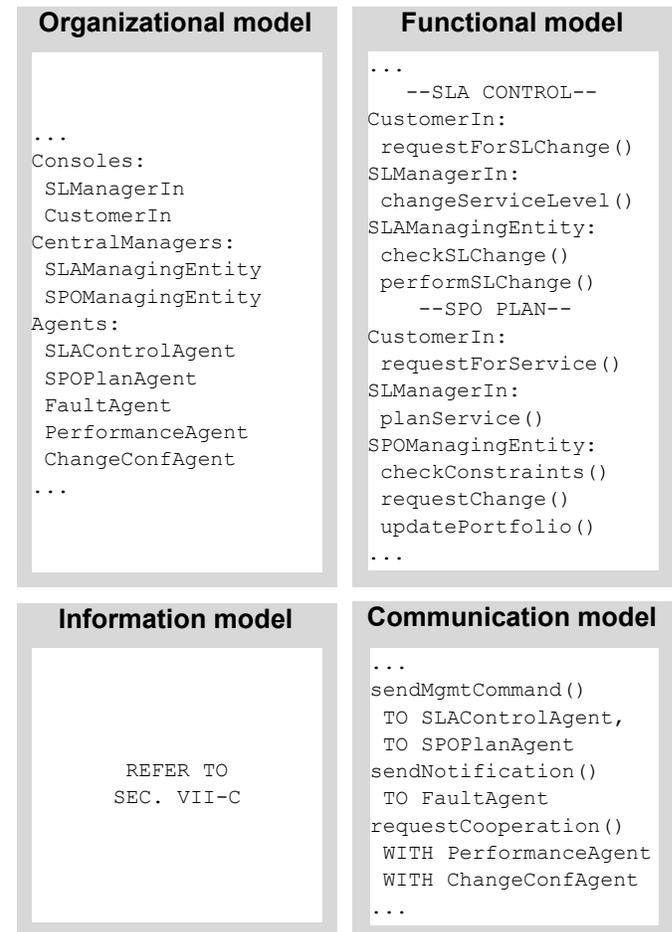


Fig. 10. Exemplary components of an SLM architecture (system specification layer)

According to figure 10, the organizational model defines three major types of entities: *management consoles*, *managers* and *agents*. In the regarded use case-specific partition,

two consoles are required: one for the provider-side Service Level Manager role and one for the Customer Representative role. Both consoles realize an interface to the central `SLAManagingEntity` of the management system. Agents are required to perform initiated management actions on the respective systems.

The functional model provides functions to be invoked from the two management consoles as well as functions to be called by the `SLAManagingEntity`. Invocations of the `requestForSLChange()` function from the `CustomerIn` console are collected by the central `SLAManagingEntity` and reported to the provider-side SLA manager by forwarding them to the `SLAManagerIn` console. The `checkSLChange()` and `performSLChange()` functions are called by the `SLAManagingEntity` after an invocation of `changeServiceLevel()` from the `SLAManagerIn` console.

The `sendNotification (TO)` and `requestCooperation (WITH)` functions of the communication model are needed to realize full integration of SLM activities into the whole ITSM management environment. In this example, a notification on the planned/performed change of service levels is sent to the `FaultAgent`, and a bi-directional information exchange is initiated with the `PerformanceAgent`.

B. Setup of a management system

To become more specific as regards concrete system design, we now show, how our use cases can be supported by a management system designed according to the described models. Having presented OM, FM and CM for the use case-specific part of the architecture's system specification layer, figure 11 shows how the defined model components form the respective part of a management system.

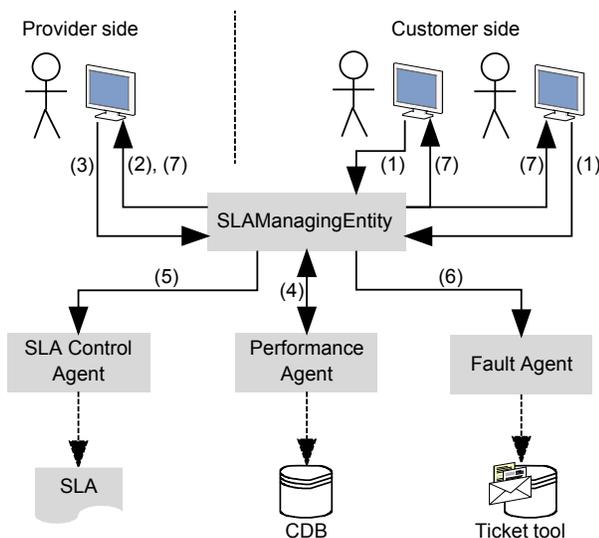


Fig. 11. Exemplary system design (small excerpt)

- (1) Requests for changes of service levels are collected by the central `SLAManagingEntity` from one or more customers' management consoles.
- (2) A report/collection of the requests is sent to the provider's management console.
- (3) Definitive service level changes are initiated by the `SLAManagingEntity`.
- (4) A coordination session is established between the `SLAManagingEntity` and the `PerformanceAgent`. In a specified way, the feasibility of the change(s) of service levels must be evaluated and related consequences and costs need to be determined.
- (5) When all constraints have been removed and the requested change of service level can be deployed, the `SLAControlAgent` is put in charge of performing the actual change.
- (6) Now, the `FaultAgent` is informed about the change and has to assess the associated impact on Fault Management processes. For example, data in the trouble ticket tool have to be updated to ensure correct incident handling in accordance to the (new) service levels.
- (7) Finally, all involved operators are informed about the successful completion of the entire management activity.

To conclude this exemplary insight into the system specification layer of our management architecture, let us now have a look on the information model which up to now has not been discussed in detail.

C. Information and data modeling

In a recently finished research project, a methodology for information and data modeling in ITSM – not limited to or specifically focused on SLM – has been developed [17]. In-depth results of this methodology applied to SLM will be published separately shortly after. The core steps of this methodology are (i) to identify information artifacts by analyzing management processes, (ii) to create an input/output/ownership matrix for each artifact, considering all relevant management processes, (iii) define the goal and essential attributes of each artifact, (iv) develop a detailed data model for each artifact, and (v) provide a bird's eye view on interdependent artifacts.

In order not to re-invent the wheel, the methodology considers the Shared Information/Data Model (SID, cf. section III) in steps (iv) and (v) and uses UML as a modeling language. If SID provides a model for one of the required SLM information artifacts, this model is analyzed as to its applicability with respect to the information requirements base. Where necessary, extensions are added to the SID element models, and a consolidated model is produced.

The information model is the most voluminous of the management architecture's four partial models, and at the same time forms a vital foundation for the realization of a tool-based management system according to the specifications coming from the system architecture layer. The requirements derived in section V tell us – based on the presented scenarios – where which information are needed. In the end, it is hard to give

evidence for the completeness and correctness of the resulting information model. This is why adaptability is an important requirement on an information model in the area of ITSM (cf. [12]) and should be considered during its design.

D. Modeling notation for the architecture components

Due to space restrictions, the system specification layer's excerpt in this paper has been presented without applying a formal modeling notation. For the final and definitive version of the SLM architecture, the Unified Modeling Language (UML) has been selected as a powerful, standardized general-purpose modeling language. UML does not only provide several graphical notations in terms of functional, structural and behavioral diagrams, but also an XML-based metadata interchange language (XMI) and solutions for transforming platform-independent into platform-specific models according to MDA.

VIII. SUMMARY AND NEXT STEPS

The following gives a summary of this work's main contributions and states the current state of research as well as the next planned steps.

A. Summary

As Service Level Management is gaining more and more attention in the IT management community, deficiencies of existing approaches become more apparent. In particular, the challenges of setting up an integrated, tool-based management system to support the tasks of SLM have not been addressed sufficiently.

In order to fill this gap, an architecture to effectively support the design and development of concrete IT-supported management systems has been presented. For this purpose an MDA-like approach has been selected to provide a platform-independent model (system specification) which can be transformed into platform-specific pieces of software. As a basis for this, requirements on the management architecture have been elicited from use cases identified in two different IT scenarios. The partitioning of the architecture into an organizational, a functional, an information and a communication model is oriented to the OSI management framework.

B. Where are we now? What comes next?

In this paper, we have outlined the hitherto achieved results of our ongoing research. These include models of the process view layer and – for selected use cases – platform-independent system specification models. Finally, a concrete use case-specific system design excerpt has been proposed.

Major pieces of work are still lying ahead: In particular, a methodology for transforming the system architecture models into platform-specific technology models, based on existing techniques, needs to be developed and applied. An implementation of a working subset of the management system's modules shall provide a proof of concept for the architecture and this methodology.

ACKNOWLEDGMENT

The authors wish to thank the members of the Munich Network Management (MNM) Team for helpful discussions and valuable comments on previous versions of this paper. The MNM Team directed by Prof. Dr. Heinz-Gerd Hegering is a group of researchers of the University of Munich, the Munich University of Technology, the University of the Federal Armed Forces Munich, and the Leibniz Supercomputing Center of the Bavarian Academy of Sciences. This paper was supported by the EC IST-EMANICS Network of Excellence (#26854).

REFERENCES

- [1] Office of Government Commerce (OGC), Ed., *The Official Introduction to the ITIL Service Lifecycle*, ser. IT Infrastructure Library (ITIL). The Stationary Office, 2007.
- [2] L. Lewis, *Service Level Management for Enterprise Networks*, ser. Artech House Telecommunications Library. Artech House, Inc., 2000.
- [3] Office of Government Commerce (OGC), Ed., *Service Strategy*, ser. IT Infrastructure Library (ITIL). The Stationary Office, 2007.
- [4] M. Garschhammer, R. Hauck, B. Kempter, I. Radisic, H. Roelle, and H. Schmidt, "The MNM Service Model — Refined Views on Generic Service Management," *Journal of Communications and Networks*, vol. 3, no. 4, pp. 297–306, Dec. 2001.
- [5] Office of Government Commerce (OGC), Ed., *Service Design*, ser. IT Infrastructure Library (ITIL). The Stationary Office, 2007.
- [6] *ISO/IEC 20000-1: Information Technology – Service Management – Part 1: Specification*, ISO/IEC, Dec. 2005.
- [7] *ISO/IEC 20000-2: Information Technology – Service Management – Part 2: Code of practice*, ISO/IEC, Dec. 2005.
- [8] *SLA Management Handbook – Volume 1: Executive Overview*, TM Forum, July 2004, GB 917-1, Version 2.0.
- [9] T. Schaaf, "Frameworks for Business-driven Service Level Management — A Criteria-based Comparison of ITIL and NGOSS," in *Proceedings of Second IEEE/IFIP International Workshop on Business-Driven IT Management (BDIM 2007)*, 2007.
- [10] M. Brenner, "Werkzeugunterstützung für ITIL-orientiertes Dienstmanagement – Ein modellbasierter Ansatz (Tool support for ITIL-oriented service management — a model-based approach)," Dissertation, Ludwig-Maximilians-Universität München, July 2007.
- [11] —, "Classifying ITIL Processes — A Taxonomy under Tool Support Aspects," in *Proceedings of First IEEE/IFIP International Workshop on Business-Driven IT Management (BDIM 2006)*, vol. 2006, 2006, pp. 19–28.
- [12] M. Brenner, M. Garschhammer, M. Sailer, and T. Schaaf, "CMDDB - Yet Another MIB? On Reusing Management Model Concepts in ITIL Configuration Management," in *Large Scale Management of Distributed Systems – 17th IFIP/IEEE International Workshop on Distributed Systems, Operations and Management, DSOM 2006*, ser. Lecture Notes in Computer Science, R. State, S. van der Meer, D. O'Sullivan, and T. Pfeifer, Eds., vol. 4269/2006. Springer, 2006, pp. 269–280.
- [13] *Shared Information/Data (SID) Model – Business View Concepts, Principles, and Domains*, TM Forum, Nov. 2005, GB 922, Version 6.1.
- [14] M. Debusmann, R. Kröger, and K. Geihs, "Unified Service Level Management using an MDA-based Approach," in *Proceedings of IEEE/IFIP Network Operations & Management Symposium (NOMS 2004)*, vol. 2004, 2004.
- [15] M. Debusmann, "Modellbasiertes Service Level Management verteilter Anwendungssysteme (Model-based Service Level Management of distributed systems)," Ph.D. dissertation, Universität Kassel.
- [16] H.-G. Hegering, S. Abeck, and B. Neumair, *Integrated Management of Networked Systems – Concepts, Architectures and their Operational Application*. Morgan Kaufmann Publishers, ISBN 1-55860-571-1, 1999.
- [17] A. Scherer, "Entwicklung einer Methodik zur Informations- und Datenmodellierung in IT Service Management-Prozessen (Development of a methodology for information and data modelling in IT Service Management processes)," Diplomarbeit, Ludwig-Maximilians-Universität München, Nov. 2007.