The MNM Service Model – Refined Views on Generic Service Management

M. Garschhammer, R. Hauck, B. Kempter, I. Radisic, H. Roelle, and H. Schmidt

Abstract: **In the last couple of years service management has gained more and more momentum in the research community. Nowadays, as complexity of IT–services is increasing, these services are usually composed of interdependent, layered services hosted by various providers. Therefore, a strong cooperation of providers involved in service provisioning is needed, especially regarding their management processes and systems. One important step is to establish a common understanding about service–related terms in order to be able to specify service functionality and the resulting management tasks. Additionally, means to analyze and identify the necessary actors and the corresponding inter– and intra–organizational relationships are needed.**

This paper presents an extended version of the MNM service model which provides a generic model defining commonly needed service–related terms, concepts and structuring rules in a general and unambiguous way. Furthermore, we demonstrate the application of the MNM service model by modeling a user help desk service based on a concrete scenario. The experience gained from modeling is used to discuss the benefits of service modeling in general.

Index Terms: **Service management, service model, provider hierarchy.**

I. INTRODUCTION

As the integration of IT- and telecommunication services has been intensifying in recent years, the need for service management solutions covering a broad variety of services is increasing. New requirements such as business process outsourcing and e–commerce extend the range of services from (classical) communication, Internet and application services to complex value–added services. Additionally, the emerging universal service market exposes all players to strong competition and forces providers to think in terms of services, quality of service (QoS) parameters and service agreements when talking to their customers rather than discussing technical details of network devices or end systems.

To deal with the increasing complexity of services, providers usually implement them by composing interdependent and layered services of different departments, carriers, Internet Service Providers (ISP), Application Service Providers (ASP) and Business Process Outsourcers (BPO). In consequence, all these organisations use terms like "service," "QoS parameters," and "service agreements," but lack a *consistent* and *common* understanding of what these terms mean.

This trend has serious implications on the management of IT and telecommunication environments: Despite the substantial work that has been carried out in the area of service management, a common and generally accepted understanding of terms and tasks associated with service management including their associations and dependencies has not been achieved yet. One of the major reasons is that most approaches focus on specific scenarios and management environments and thus, use a *varying terminology* regarding service management. In our opinion, the development of generic concepts and interoperable service management solutions is not possible until a *common terminology* is defined.

In order to address some of the problems associated with the above mentioned aspects of service management, we developed a generic service model (in the following: *"MNM service model"*) [1] that defines commonly needed service–related terms, concepts and structuring rules in a general and unambiguous way. The purpose of application of the service model is to model specific services and given service provisioning scenarios in order to analyze needs and demands regarding an appropriate service management. Although we use well–known UML class diagrams to define the MNM service model we have to emphasize that the MNM service model is a conceptual meta model *not* leading immediately to an implementation. Instead, it helps to analyze, identify and structure the necessary actors and the corresponding inter– and intra–organizational associations between these actors. Since it also covers the whole service life cycle, it helps to establish, enforce and optimize information flows between organizations and business units.

Meanwhile, we have applied the MNM service model in several industry cooperations to model and analyze concrete scenarios. Adopting the gained practical experience during modeling we modified our service model to improve its usability and applicability. Among other things, views were added. This paper presents these modifications which results into an extended version of the MNM service model. Furthermore, we demonstrate the application of the MNM service model by modeling a user help desk (UHD) service.

The remainder of this paper is organized as follows: Section II gives a brief overview of current research related to service management in general and to service modeling in particular. Section III presents the top–down methodology used to develop the generic service model. This section also outlines requirements that are to be posed on modeling concepts applied in the area of service management. Section IV introduces the modified MNM service model. To sum up the changes compared to [1], a basic service model as well as views are newly introduced and associations were altered on basis of a critical review. Then, in Section V we model a UHD service by applying the previously

Manuscript received July 17, 2001.

The authors are with the Munich Network Management (MNM) Team, University of Munich, Germany, e-mail: {garschha, hauck, kempter, radisic, roelle, schmidt}@informatik.uni-muenchen.de.

presented MNM service model. Finally, Section VI concludes the paper and presents further work.

II. RELATED WORK

This section gives a brief overview of the general research area of service management. It outlines the established concepts introduced by TINA-C, ITIL, TM Forum and DMTF as well as several new approaches that are being discussed in the research community.

Telecommunication Information Networking Architecture Consortium (TINA-C)

The TINA service architecture [2] introduces a set of concepts, principles, rules and guidelines for constructing, deploying, operating and withdrawing TINA services. TINA uses many valuable concepts like layering concepts, ODP viewpoints and a service life cycle, but its focus is set on specifying a software architecture in order to implement (telecommunication) services rather than on modeling services resp. service provisioning scenarios in order to analyze them. As the elements of the service architecture are specified in various models and a modeling method is missing it is difficult to use the TINA framework in order to model specific scenarios. Nevertheless, the business model introduced in [2] consisting of 5 roles is used to identify reference points and interfaces between participating roles and therefore can serve as a source for checking completeness regarding the service model we specified. Regarding service management TINA introduces a Management Architecture [3], which is based on well–known concepts like the OSI FCAPS, TMN layering concepts, computational aspects of management needs and life cycle issues. Additionally, new management concepts such as context negotiation and service transactions are introduced. The TINA definition of service management is mainly based on the concepts introduced by network and systems management of TMN/OSI, and most of the extensions towards service management are still unspecified or declared for further study.

IT Infrastructure Library (ITIL)

ITIL is a label used by the British Central Computer and Telecommunications Agency (CCTA) to publish various documents containing recommendations concerning the realization of IT service management. The goal is to establish a common foundation for a customer oriented, cost effective, high quality management of IT services for companies of any size and any structure. To achieve this goal, ITIL applies a bottom–up approach documenting the "*best practice*" in service management, i.e., practical experiences gained by organizations during service provisioning. Currently the ITIL documents are facing a restructuring process resulting in six volumes of which two are published up–to–now: service support [4] and service delivery [5]. The focus of ITIL is set on organizational aspects by describing participating roles and necessary (management) activities embedded in business processes. As ITIL's goal is to be applicable to all kind of organizations and services, most of the activities and recommendations are described on a high level of abstraction, explicitly not considering implementation details. Therefore, we experienced ITIL to be very useful in case of doing a reengineering of already existing processes. However, applying ITIL to establish service management from scratch not starting with a given foundation is in contrast difficult: A detailed model giving an overview over ITIL is missing so that it is hard to see connections between the management processes and activities described in different documents. In case of taking a closer look at similar activities described in different processes, even discrepancies can be noticed. However, ITIL claims to be the "de facto" standard for IT service management and it surely is one of the best available documents concerning this area. We used ITIL for structuring and identifying management activities during analyzing scenarios.

TeleManagement Forum (TMF)

The Telecom Operations Map (TOM) [6] introduced by TMF focuses on the end–to–end automation of communications operations processes and therefore uses a similar approach like ITIL. In contrast to ITIL, TOM is more formal by specifying a process framework that postulates a set of business processes that are typically necessary for service providers to plan, deploy and operate their services. However, in spite of the fact that TOM introduces the valuable concept of using business processes to deal with service management concerns, TOM is not intended and therefore not well suited to model a particular service or service provisioning scenario. The specification concentrates on giving service providers valuable hints and recommendations concerning what processes are necessary to provide an overall integrated service management covering several services. In further documents TMF gives for some processes implementation details as well as recommendations concerning the use of a specific technology.

Distributed Management Task Force (DMTF)

The Common Information Model (CIM) [7] introduces an approach that claims to integrate the information models of existing management architectures; CIM acts as an umbrella that allows to exchange management information in an unrestricted and loss–free way. CIM defines a basic meta-model, a syntax for the description of managed objects and two layers of generic managed object classes. The Core Model gives a formal definition of a service and allows hierarchical and modular composition of services consisting of other services. However, the focus is on technical details of a service implementation regarding its management part and therefore does not include a notion of e.g., domains, such as customer and provider, which would be necessary in order to model scenarios.

Current Research

Service management has been a hot topic in the research community for a few years now, and a lot of research has been carried out. Due to the complexity of service management, most approaches recently published pick out one particular problem only; most approaches focus on a specific scenario and try to develop a suitable solution: For example, [8] deals with the integration of network and service management, but is restricted to ATM services. In [9] a service information model is devoloped which also takes temporal information into account but which is restricted to VPN services. [10] introduces a QoS MIB along with some necessary management functionality, but restricts itself to multimedia services. [11] reviews various approaches to develop service management systems; however, the focus is on software engineering, rather than on conceptual aspects of service management.

Since all these approaches focus on one particular problem of service management, they do not provide a generic service model that can be used in different scenarios and environments. Even the approaches that try to build a general service model have certain limitations: For example, [12] develops a service management architecture, which is limited to the phase of service usage; it does not consider the complete life cycle, which is mandatory in order to establish an overall service management. Furthermore, it is not suitable to model service hierarchies. Although [13] presents a very detailed model for service management, it does not consider service hierarchies, the relationships between customer and provider and the interactions that take place between these roles. Finally, [14] presents an architecture that uses contracts based on service level agreements (SLAs) to share selective management information across administrative boundaries. However, the focus is on the definition of a language to formalize SLAs and does not cover the complete service life cycle.

III. APPLIED METHODOLOGY FOR MODEL SPECIFICATION

As the analysis of related work in Section II has shown, means for analyzing services and scenarios in order to specify the requirements for a service management implementation or to optimize an existing implementation, etc. are only of rudimental nature. Therefore, our goal is to define a conceptual meta model which exactly can be used for this purpose and thus helps concerned parties in establishing a well working service management. In the following, we present in Section III-A requirements which are posed on such a conceptual model. Furthermore, these requirements also lead to the analysis methodology presented in Section III-B which is used to define the elements of the service model. Afterwards, the analysis steps for identifying roles is carried out in Section III-C.

A. Requirements

The general advantages of object oriented analysis are of great importance especially for complex service hierarchies. A service model furthermore requires (according to [1]):

- **Generic and abstract service definition:** A service model needs an abstract definition of a service to provide a common understanding for describing services independently of a particular scenario or environment. This generic service definition ensures, that a model can be applied to all kinds of services, from communication services to complex value–added services including distributed services.
- **Integration of various dependencies:** One major goal is to model complex real life scenarios such as supply chains and provider hierarchies that are typical for the emerging universal service market. To be able to identify all important dependencies, e.g., functional, organizational or QoS dependencies, a modeling approach has to be capable to visualize these different aspects.
- **Separation of service and service implementation:** The separation of the abstract service description from a corresponding service implementation enables providers to realize services according to their local environment without restricting or implying a particular implementation.
- **Management as an integral part of the service:** As an offered service always requires management support for the provider side as well as for the customer side, a service model has to consider the management of services as an integral part of the service itself.
- **Consideration of the whole service life cycle:** By regarding the whole life cycle of a service, a service model has to support all phases of a service lifetime.

Taking these requirements into account, modeling helps to sufficiently control complex services. In addition, a service model helps to gain a common sense for services.

B. Top–down Approach

To understand how we have identified the elements of the MNM service model and why the model is reasonable, we present our approach for analyzing the service environment in this section. The analysis methodology leads to objects and relations which form our service model. As a consequence of the requirements mentioned in Section III-A, especially regarding the separation of service and its implementation, a top–down approach regarding the analysis methodology delivers a straight forward way to develop an appropriate service model.

As one of our main requirements is to develop a scenario independent service model, we use a methodology, which is following an object oriented development style. This results in the top–down approach depicted in Fig. 1, which is used to identify the elements of the MNM service model.

An inherent characteristic of every service is that it involves two major players: one offering and one requesting the service. Speaking in technical terms there is a *provider side* and a *customer side*. Both *interact* to accomplish a service. By solely examining these interactions, we are able to draw conclusions about the service functionality without the need to take the service implementation into account. Therefore, it is important to identify these interactions for which we use a *service life cycle*, which divides the lifetime of a service in several phases from its incarnation to its deinstallation.

As it is impossible to find every single interaction regarding all possible services, an abstraction of these interactions is needed. For this purpose we use a *classification* in order to group interactions. The life cycle phases lead to a first grouping of the interactions. To refine this rough structure, a *functional* classification (based on TMF's TOM [6] and OSI's Systems Management Functional Areas [15]) is performed in addition. The combination of these two activities leads to a classification *matrix*.

Finally, as interactions take place between a pair of *roles* representing e.g., organizational units on both sides, roles are as-

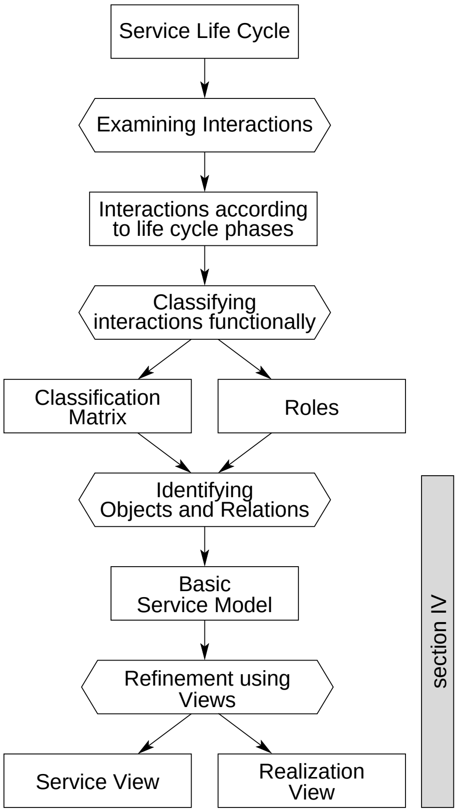

Fig. 1. Analysis method.

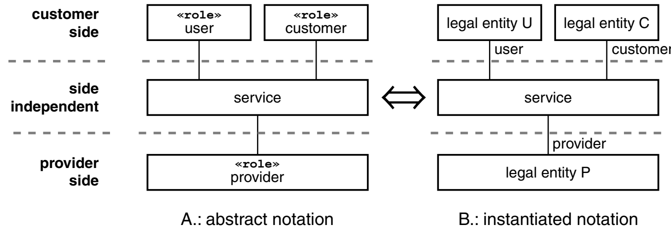

Fig. 2. Basic service model.

signed to interaction classes.

By examining the identified interaction classes and roles, we are able to specify interfaces as well as entities participating in service provisioning. To achieve this, we first introduce a *basic service model* containing the most relevant roles and associations. Afterwards, a refinement of the basic service model is done by using the concept of *views*: the *service view* concentrates on analyzing entities between provider and customer side, whereas the *realization view* is used to identify objects within the provider side. Finally, the result of our analysis method is a detailed service model which is split up into the service view and realization view.

To motivate the basic service model in Section IV, the remainder of this section concentrates on presenting the identified basic roles involved in service provisioning. A detailed description of the applied steps of the analysis methodology can be found in [1].

C. Roles

Besides identifying interactions, the main goal of the methodology is to clarify the participating roles. When modeling concrete scenarios, organizational units and/or persons are assigned to roles. Therefore, we need a set of generic roles which can be applied to every possible scenario.

Following the methodology described in the previous section, exactly one role on the customer side as well as exactly one role on the provider side is participating in every interaction. In order to reach a higher level of abstraction we group roles according to the specified interaction classes. A role is also associated with certain *rights*, which additionally contributes to the area of security management.

As shown in [1] we discovered two major interaction classes: usage and management interactions. Accordingly, we identified two main roles on the customer side: *Users*, who actually use the service and a *customer*, who is interested in maintaining a subscribed service and therefore performs all the management activities on the customer side. On the provider side all necessary activities for enabling service usage as well as service management have to be performed. Since these activities cannot be strictly separated on the provider side's internal processes we only introduce the role *provider* for the provider side at this stage which is sufficient for the scope of our service model presented in the following section. Of course a more detailed analysis leading to more than three roles is possible.

IV. THE MNM SERVICE MODEL

It is difficult to define the term "service" in a universal way, not restricting it to a small set of scenarios. The current research either leaves the term undefined or the definition is as narrow as needed to match the analyzed scenario.

Our approach is based on the top–down methodology presented in the previous section. Therefore, as a first attempt which will be refined in one of the following sections, a *service* is defined by the provided service functionality and is explicitly seen separated from its implementation. According to the analysis presented in Section III-B the service functionality is defined by a set of interactions taking place between the participating roles. These interactions can be mapped to protocol transactions, application transactions, business workflows, etc. The combination of the service with the previously identified roles leads to our basic service model. This basic model is then refined by the service view and the realization view. All parts together form the MNM service model. They are presented in the following sections.

A. Basic Service Model

The *basic service model* shown in Fig. 2 is a direct result of the analysis carried out in Section III, where three universal roles interacting with any *service* have been identified: *user*, *customer*, and *provider*. Considering these roles and their associated domains, the basic model is divided into three parts: the *customer side*, the *provider side*, and the *side independent* part where the service is located.

In a standard UML class diagram, to model these three roles and to express the fact that they are taken in by different players, three different classes have to be modeled. As all roles generally are carried out by legal entities, the corresponding classes have to be made distinguishable by auxiliary indices in addition to the class name "legal entity".

Instead of using standard role notation in the abstract model, a new *stereotype "role"* is introduced. The notion of this stereotype is to express that a certain legal entity acts as the specified role. Consequently, Fig. 2-A is equivalent to Fig. 2-B. As the notation using the stereotype role is much clearer for the abstract model, Fig. 2-A is called the *abstract notation of the basic model*. Fig. 2-B outlines the notation to be used on instantiation of the model, which will also be shown by example in Section V.

To analyze interactions between the service and the three roles in depth, the MNM service model offers two views: The *service view* and the *realization view*. These views are directly derived from the concept of service orientation, which separates functionality from implementation.

In case of the service view, functionality of the service is in the foreground, abstracting from details on its implementation. Its purpose is to define the service precisely and independently from special aspects of either the customer or provider side. Purpose of the service view is to serve as a basis for specifying service agreements.

When focusing on realization of a service, the realization view of the service model presents details within the provider side. It identifies objects involved and depicts in detail the possibility to use sub–services for service realization resulting in provider hierarchies or service chains. Purpose of usage is to serve as a starting point for the analysis phase in case of implementing a specific service.

B. Service View

The understanding of a service must be the same for customer and provider side. We follow the concept of service orientation which postulates the implementation independent description of the service from the perspective of the customer side. The side independent aspects can be found in Fig. 3 between the two domains symbolizing customer side and provider side.

Side Independent Aspects

According to the main interaction classes identified in [1], the service consists of usage *and* management functionality. Both types of functionality must satisfy a set of *QoS parameters*. These parameters define the minimum required service quality in order to be useful for the customer side. The QoS parameters are qualitative values.

The *usage functionality* covers the interactions needed by the user. These interactions represent the actual purpose of the service. Additionally, interactions beyond the service's purpose are needed to fulfill the customer's duties, to customize the service according to user's needs, to monitor and control the provider's service provisioning as well as for payment. The *management functionality* comprises these interactions.

The information presented up to this point describes the service bought by the customer side and provided by the provider side. To actually be usable, there must exist a service interface between these two sides. Service primitives, protocols and (where necessary) physical connectors are represented by the service interfaces. An interface can also be an interface of another service or communication system.

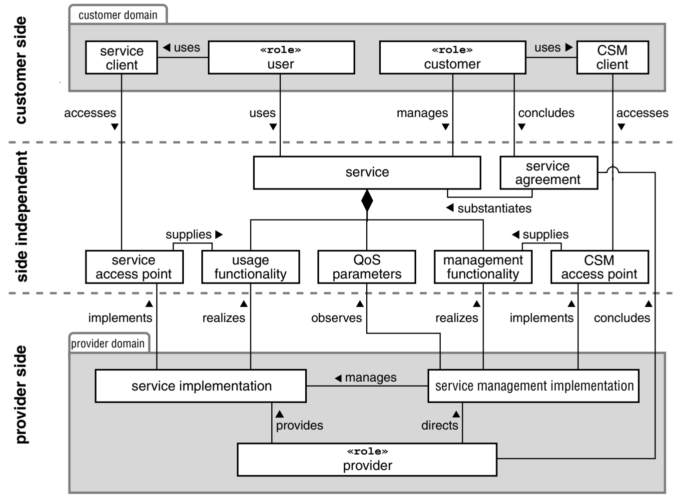

Fig. 3. Service view.

Just as the functionality was split up in usage functionality and management functionality, the interface is split up in a usage interface, called *Service Access Point (SAP)*, and a management interface, called *Customer Service Management Access Point (CSMAP)* [16], where the corresponding management functionality is accessible.

The *service agreement* substantiates the service by describing the usage and management functionality from the customer's perspective as well as the QoS parameters. Additionally, the service agreement defines the logical as well as physical aspects of the usage and management interfaces between customer and provider side.

Customer Side

In most cases some equipment is needed to access the service functionality from the customer side. Such *clients* allow user and customer to access the functionality at the SAP and the CSM AP respectively. Clients can be telephones, computers or applications.

These technical clients must be compatible to the physical and logical aspects of the service interfaces. The sole responsibility for the clients rests on the customer side of the service.

Provider Side

The main task of the *provider* is to make the service available. This includes all aspects of the service, namely the usage and management functionality of the customer side fulfilling the QoS parameters and the interfaces to enable usage and management of the service.

For this reason the provider needs a *service implementation* which realizes the usage functionality of the service. To allow the user to access this functionality, the service implementation also implements the SAP. The service implementation is the combination of all knowledge, staff, software and hardware needed to realize the usage functionality and the SAP.

Comparing the basic model to the service view, the provider's single association to the service in the former is split up into six subtler ones in the latter.

The provider is also responsible for the service management. That means he directs it in a way which is suitable to keep the service above the agreed quality level but also to optimize the service operation according to other goals like high efficiency and low risk.

The main purpose of the *service management implementation* is to ensure proper service fulfillment according to the service agreement. This means to care for keeping the QoS parameters in the agreed ranges by managing the service implementation. Additionally, it implements the management interface for the customer side allowing limited access to the provided management functionality.

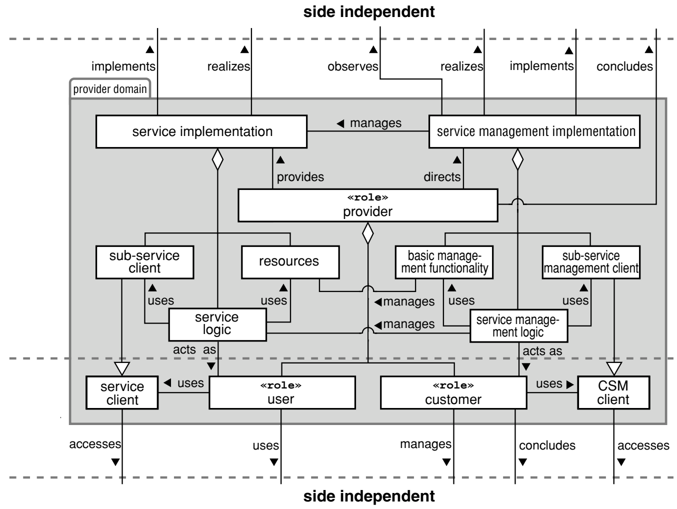

Fig. 4. Realization view.

C. Realization View

The service view presented in the section above focuses on single customer–provider–relationships from the point of view of the customer. Its purpose is to serve as a basis for specifying service agreements. In the following, we introduce additionally the *realization view* which we use for modeling the provider–internal realization of a service. Therefore, the realization view can serve as a basis for implementing a certain service. As many of the value–added services, that are now being offered by providers, are composed of services that are supplied by various sub–providers, the realization view has to be capable to express exactly this situation. Additionally, sub–providers in turn can apply the same principle and can contract other sub–providers, which leads to service chains and service hierarchies. Consequently, we use the realization view to model the resulting service hierarchies.

A provider contracting a service of another provider acts as a customer to the latter. This means that the provider domain embeds the tasks of the user/customer role and the provider role simultaneously. As such, we can reuse the already modeled associations between the customer and the provider domain in order to model the associations regarding the relation of provider and sub–provider. By expanding the provider domain with the entities of the customer domain, we are able to create an enhanced model of the provider domain containing the classes *service implementation* and *service management*, the roles *provider*, *user* and *customer*, and the *service client* as well as *CSM client* classes.

Fig. 4 illustrates the provider–internal realization of a service and its management. As already explained, all roles of the customer and the provider side of a service can reside within the provider domain. Thus, both, the *customer* role and the *user* role are part of the *provider* role. The *clients* used to access the subsidiary interfaces must be part of the service implementation and the service management respectively to permit an interaction with a sub–provider. As a consequence, there is a need of new elements within the service implementation and the service management.

The service implementation is composed of *resources* made available by the provider himself and *services* that are accessible through *sub–service clients*. Hence, we introduce a *service logic* to control both, the usage of services as well as the usage of the provider's resources. Thus, our class diagram shows the class *service implementation* consisting of the classes sub–service client, service logic, and resources. The sub–service client is actually just a refinement of the generic user client added to the provider domain.

The service management will use functionality of the traditional network, system and application management, the so called *basic management functionality* (BMF), along with the management functionality provided by subsidiary services. In consequence, there has to be a *management logic* controlling the BMF as well as the *sub–service management clients* for the subsidiary service management. The management logic treats the service logic as a managed object which leads to an association between the two classes service logic and service management logic. Corresponding to the service implementation we model the class service management as an aggregation of the three classes BMF, service management logic, and sub–service management client.

As both logics use corresponding clients to access the sub–service and/or their management respectively, they act in the role *user* (service logic) and *customer* (management logic). We model this correlation with two associations, one connecting the service logic and the user role, the other one connecting the service management and the customer role. Overall, only 5 associations are needed to realize a connection between a provider and his sub–provider.

V. MODELING A USER HELP DESK

In this section we demonstrate the application of the MNM service model by modeling a UHD service. We have chosen the UHD as a complex service in order to show the model's applicability to real life scenarios. The individual interactions of the UHD were analyzed with aid of UML use case and activity diagrams, interacting with members of the UHD staff. Result of

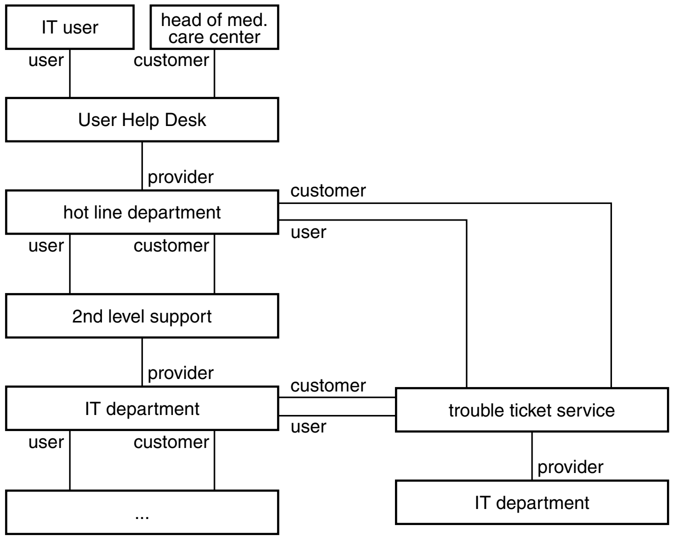

Fig. 5. Excerpt from basic service model of the UHD scenario.

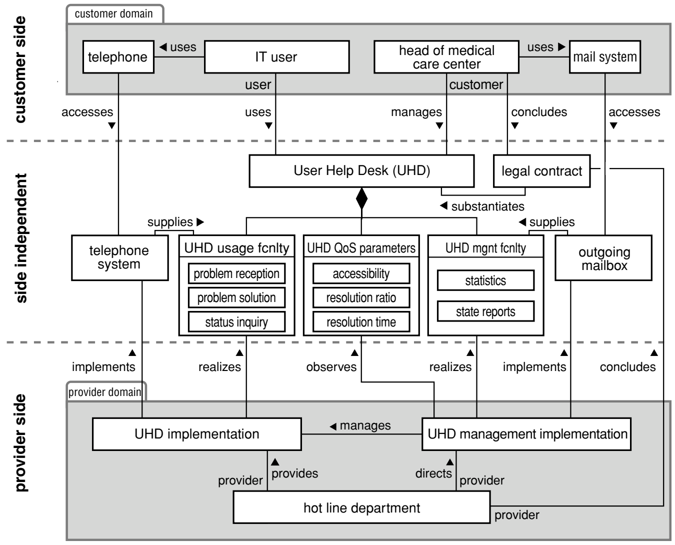

Fig. 6. Service view of the UHD service.

this analysis is a model of the current realization of the UHD service, of which parts are described in this section. The objective is to comprehend the current service allowing informed discussions on QoS, workflow optimization and other improvements.

A. Scenario and Basic Model

A computing center provides a UHD for the staff of a big medical care center. Up to now, neither a model of the UHD exists nor service oriented aspects like QoS parameters have been defined. Thus, the MNM service model applies as a reference model to ensure considering all important aspects of the service.

Some considerations about the structure and the environment of this service show its complexity. To offer a UHD service to users a variety of sub–services (e.g., a trouble ticket service) have to be accomplished. Fig. 5 shows an excerpt from the basic model of the UHD. The model is gained by recursively applying the abstract basic model of Fig. 2.

At first, the services are defined by specifying the former class “service” with concrete names (see Fig. 5). In the next step the roles involved in the services have to be determined. In case of the UHD service, users of the IT–infrastructure are represented with the *IT user* class while the head of the medical care center is regarded as the *customer* and the *hot line department* as the *provider* of the service. Doing so, all mandatory roles in conjunction with the UHD service are defined. In case of a provider using a sub–service (e.g., hot line department using 2nd–level–support), he simply also acts as the roles user and customer relative to the sub–service.

To fully show the applicability of the MNM service model, we present the model of the UHD service and of the 2nd–level–support. Modeling of other parts of the service chain can be done as well as we will show in Section V-D.

B. UHD Service

After applying the basic service model to the UHD service as a whole including all sub–services, the service view of the model is built to describe the UHD service more precisely (Fig. 6). First, a partitioning is done according to the trisection of the service view. Then, the side independent part, the customer side and the provider side are specified in detail.

Side Independent Part

Topics specifically related to the service are determined with the next steps:

- **Functionality:** The service functionality consists of the possibility to report problems, get these problems solved and to present the current state of the problem solution. From the management perspective, the provider supplies statistical information about provisioning of the service. Thus, the corresponding functionality classes of the UHD’s service view are specified by introducing a new class for each functionality. Combining these new classes to compositions leads to the usage and management functionality classes of the service.
- **QoS parameters:** After specifying the service functionality, its characteristic QoS parameters meaningful to the customer have to be specified. The *QoS parameters* for the UHD service are the problem resolution time, the problem resolution ratio and the availability from a users point of view.
- **Access points:** After the service functionality is defined, the corresponding access points have to be specified. The access point for submitting inquiries to the UHD service is the telephone system. Thus, the class *SAP* is named accordingly. Reports are sent via the email system to the customer so that the outgoing mailbox of the provider represents the management interface of the UHD service. Thus, the class *CSM access point* is named suitably. [17] and [18] present a detailed analysis of the topics related to customer service management.
- **Service Agreement:** After specifying all details of the service, the conditions of the service provisioning e.g., modalities of accounting, etc. have to be specified in a le-

gal contract which also contains SLAs. In [19] a methodology is presented to define SLAs which can be applied at this point.

Now modeling of all entities relating to the service itself is done. At next, the customer side is modeled.

Customer Side

To access the interfaces already defined, clients in the provider domain have to be specified. A client could be a classical application like a web browser as well as a telephone. In case of the modeled UHD service, reporting a problem is done by telephone (this could be implemented as a web form as well). Statistical data about the service is given to the customer (the head of the computing center) via email thus the *client* classes are modeled as depicted in Fig. 6.

Provider Side

The *service implementation* and *service management implementation* of UHD service are considered by assigning appropriate names to the abstract classes without going into detail about its concrete realization.

C. Realization View of UHD Service

In contrast to the service view illustrated in the previous section that defines the UHD service in a view common to both the customer and provider, this chapter shows how the UHD service is realized by the provider internally.

Modeling of hierarchical relations between (sub–)services is done in a further step using the realization view of the provider domain introduced in Section IV-C. Hence, a link between the UHD service provider domain and the customer domain of the sub–service(s) he uses could be established.

The new elements introduced in the provider domain by the next steps are shown in Fig. 7. First the implementation part of the realization view (left side of Fig. 4) is considered:

- **Sub–services:** The basic model of the UHD (Fig. 5) already shows dependencies between the UHD service and other services. E.g., the UHD service depends on the 2nd–level–support. In this case the corresponding sub–service client to access the 2nd–level–support is a *WWW browser*.
- **Resources:** In addition to sub–services a provider uses his own resources to implement the service. In case of the UHD service these resources consist of the *hardware* used (like PCs) and the *staff* responding to requests from users via the telephone.
- **Service logic:** The class *workflow* models any task needed to coordinate the usage of sub–services and resources. For example, it determines the cases when the 2nd–level–support has to be contacted by the staff. By now, only an informal description of the workflow exists at the medical care center. A formal specification of this abstract class can be done when all details of the implementation are laid down, thus specifying how functionality is distributed between sub–services and resources.

The service management part of the realization view (right side of Fig. 4) is symmetric to the functionality part, thus implying the same steps. From the management perspective sub–services are aggregated to form a value–added service (Fig. 7), too.

Defining the Customer's Roles and Clients

After extending the service implementation and management, the classes acting as the roles *user* and *customer* as well as their corresponding clients have to be included in the model to finish it. The clients have already been defined within the classes *WWW browser* and *SSL capable WWW browser* which both are refinements of their abstract counterparts. Hence, one only needs to define who is acting as user and customer to the sub–services. The head of the department 'hot line' is the *customer* of the sub–service. Every employee of the hot line department acts as a *user* (see Fig. 7).

D. Chaining of Services

In the same way as described in Section V-B, a service view for the 2nd–level–support service was developed. As the UHD service depends on the 2nd–level–support service, these two services have to be connected. Now, the structure of the MNM service model allows its application on hierarchically structured services not only within the basic model, but also by combining the realization view and the service view of a sub–service. This reflects the requirement of a structured representation of a problem. Service chains are not only modeled in the basic view as a whole, but also a detailed view on every part of the chain can be obtained.

As all entities of a service view's client side are present in the realization view and furthermore the roles user and customer are also defined, the realization view of the UHD directly can be connected to the service view of the 2nd–level–support as depicted in Fig. 7.

E. Benefits

By applying the MNM Service Model several benefits for the medical care center were achieved.

The instantiated basic model gives a full overview of interdependencies of internal departments as well as external service providers. By that, all relevant entities and services regarding the UHD service were clearly identified.

Furthermore, by means of the service view of the UHD service, it became clear which QoS Parameters the service could offer, clarifying on what parameters the internal quality management of the medical care center could count on. Besides of simply specifying the parameters itself, it is now obvious on what QoS parameters the UHD service itself depends by the sub–services it uses. For example it became clear that full time service of the UHD cannot be guaranteed as the IT department cannot assure availability of the trouble ticket service.

In case of the realization view it became clear that the workflows, which resulted from specifying the central logic classes, need formal specification. As these classes are responsible for the way how the UHD service is provided and thus are responsible for realizing and observing the QoS parameters, their refinement and further definition were given priority over other tasks.

Dealing with the question of extending the SAP of the UHD by adding a WWW interface, the model assisted immediately by

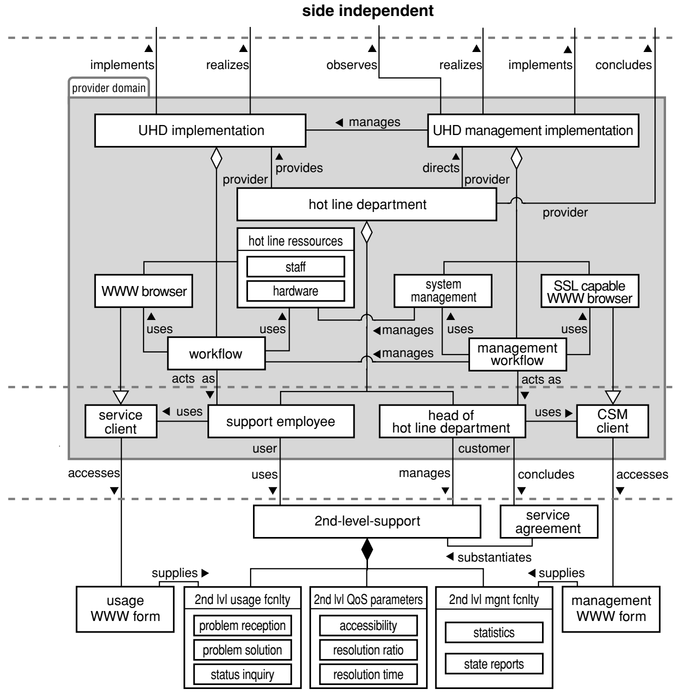

Fig. 7. Realization view of UHD service and chaining to 2nd–level–support.

showing possible options of realization. The IT department already runs such a system for the 2nd–level–support (see Fig. 7), but does not provide the UHD service itself. So two main possibilities were discussed: Definition of a new sub–service run by the IT department providing a generic WWW form service, which then is used by the IT department internally but also by the hot line department. The second possibility discussed was to restructure the whole UHD service, eliminating the hot line department and incorporating it with the IT department.

VI. CONCLUSION AND FURTHER WORK

In this paper we first developed a service model describing entirely the relations between customer and provider side of a service. In order to present the service model in more detail we introduced the service view and the realization view. In contrast to the service view, which illustrates the entities of customer–provider relationships from a customer point of view, the realization view is used to model provider–internal elements needed to realize a certain service. Additionally, the realization view is used to represent provider hierarchies and service chains.

The model itself determines commonly needed service–related terms by their associations in a non–scenario–specific, generic and unambiguous way. Thus, the service model can be used for any kind of service and scenario.

In the second part of the paper we presented the application of the MNM service model to a real life user help desk service. The service view revealed the benefits of modeling: all relevant functional, organizational and management related aspects of a service were integrated in one single model. Thus, previously undiscovered dependencies were identified. Furthermore, by using the realization view of the service model important parts of the service implementation, the service management as well as associations between these tasks were identified. Overall, a structural and straight–forward analysis of existing services and scenarios is possible.

Currently, our work, which is co–funded by several companies like Deutsche Telekom, Siemens, DeTeSystem and BMW, focuses on different aspects of the MNM service model: A refinement of the model is investigated for applications in the area of SLA management, accounting and customer service management. Furthermore, we use the MNM service model to support the development of service management solutions by identifying the locations of required adaption resp. optimization. In case of component–based applications we succeeded in developing management software which automatically constructs the service management logic on basis of the application logic [20], [21]. Likewise, the service model was used to analyze relevant management interactions taking place between provider and customer in order to specify and implement a generic customer service management interface [22].

For modeling guidance we have already succeeded to specify a modeling methodology which will be published soon.

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, and the Leibniz Supercomputing Center of the Bavarian Academy of Sciences. Its web–server is located at `http://wwwmnmteam.informatik.uni-muenchen.de`.

REFERENCES

[1] M. Garschhammer *et al.*, "Towards generic service management concepts–a service model based approach," in *Proc. 7th IFIP/IEEE Int. Symp. on Integrated Management (IM 2001)*, May 2001.

[2] Service Architecture Version 5.0. TINA Baseline, TINA Consortium, June 1997.

[3] Management Architecture Version 2.0, TINA Baseline TB_GN.010_2.0_94, TINA Consortium, Dec. 1994.

[4] Office of Government Commerce (OGC), *Service Support*, the Stationary Office, 2000.

[5] Office of Government Commerce (OGC), *Service Delivery*, the Stationary Office, 2001.

[6] Telecom Operations Map. Approved Version 2.1 GB910, TeleManagement Forum, Mar. 2000.

[7] Common Information Model (CIM) Core Specification Version 2.5., June 2001.

[8] D. Lewis *et al.*, "Integrating service and network management components for service fulfilment," in *Proc. 10th IFIP/IEEE Int. Workshop on Systems: Operations & Management (DSOM'99)*, Zürich, Switzerland, 1999.

[9] R. State, O. Festor, and E. Nataf, "Managing highly dynamic services using extended temporal network information models," in *Proc. 7th Int. IFIP/IEEE Symposium on Integrated Management (IM 2001)*, May 2001.

[10] J. Hong, J.-S. Kim, and J.-T. Park, "A CORBA-based QoS management framework for distributed multimedia services and applications," in *Proc. 9th IFIP/IEEE Int. Workshop on Systems: Operations & Management (DSOM'98)*, Newark, Delaware, 1998.

[11] D. Lewis, "A review of approaches to developing service management systems," *J. Network and Systems Management*, vol. 8, no. 2 pp. 141–156, 2000.

[12] C. Mayerl *et al.*, “Specification of a service management architecture to run distributed and networked systems,” in *Proc. 3rd IFIP/GI Int. Conference on Trends towards a Universal Service Market (USM 2000)*, Munich, Germany, 2000.

[13] G. Chen and Q. Kong, “Integrated management solution architecture,” in *Proc. NOMS 2000 IEEE/IFIP Network Operations and Managment Symp.–The Networked Planet: Management Beyond 2000*, pp. 217–230, Apr. 2000.

[14] P. Bhoj, S. Singhal, and S. Chutani, “SLA management in federated environments,” in *Proc. 6th IFIP/IEEE Int. Symp. Integrated Network Management (IM’99)*, Boston, 1999.

[15] Information Technology–Open Systems Interconnection–Systems Management Overview. IS 10040, International Organization for Standardization and International Electrotechnical Committee, 1992.

[16] M. Langer and M. Nerb, “An ODP enterprise specification of customer service management for connectivity services,” in *Proc. 3rd Int. Enterprise Distributed Object Computing Conference (EDOC ’99)*, pp. 94–103, Mannheim, Germany, Sept. 1999.

[17] M. Langer, *Konzeption und Anwendung einer Customer Service Management Architektur*, Ph.D. thesis, Technische Universität München, Mar. 2001.

[18] M. Nerb, *Customer Service Management als Basis für interorganisationales Dienstmanagement*, Ph.D. thesis, Technische Universität München, Mar. 2001.

[19] H. Schmidt, “Service Contracts based on Work ow Modeling,” in A. Ambler, S.B. Calo, and G. Kar, editors, *Proc. 11th Annual IFIP/IEEE Int. Workshop on Distributed Systems: Operations & Management (DSOM 00)*, no. 1960 in Lecture Notes in Computer Science, Austin, Texas, USA, Dec. 2000. Springer.

[20] R. Hauck, *Architektur für die Automation der Managementin- strumentierung bausteinbasierter Anwendungen*, Ph.D. thesis, Ludwig-Maximilians-Universität, at München, Oct. 2001.

[21] R. Hauck, “Architecture for an automated management instrumentation of component based applications,” in *Proc. 12th Annual IFIP/IEEE Int. Workshop on Distributed Systems: Operations & Management (DSOM 2001)*, Nancy, France, Oct. 2001.

[22] M. Langer and M. Nerb, “Customer service management: An information model for communication services,” in *Trends in Distributed Systems: Torwards a Universal Service Market. Proc. 3rd Int. IFIP/GI Working Conference, USM 2000*, Sept. 2000.

[23] 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.

Markus Garschhammer achieved his diploma in computer science (M.Sc.) at the Munich University of Technology (TUM) in 1999. Since then he is a member of the MNM Team and working at the University of Munich (LMU) as a research and teaching assistant. His work focuses on service management, especially the description and mapping of QoS in service oriented environments.

Rainer Hauck studied computer science at the Munich University of Technology, Germany (TUM) and received his diploma degree (M.Sc.) in 1996. Since 1996 he has been a member of the MNM Team, working as a research and teaching assistant. He received his Ph.D. from the University of Munich, Germany (LMU) in 2001. His major research interests are the areas of service and application management, with a special focus on the management of component-based application services.

Bernhard Kempter studied computer science at the Munich University of Technology, Germany (TUM). He joined the MNM team after receiving his diploma degree (M.Sc.) in 1999 and is currently a Ph.D. candidate at LMU. There he also works as a research and teaching assistant. His research interests focus on integrated network and system and management in general, with an emphasis on service management and resolution of QoS conflicts.

Igor Radisic studied computer science at the Ludwig-Maximilian University (LMU) Munich, Germany, and received his diploma degree (M.Sc.) in 1999. In April 1999 he joined the MNM Team. He is currently a Ph.D. candidate at LMU, where he is also working as a research and teaching assistant. His research interests center around integrated network, system and service management, especially regarding accounting management.

Harald Roelle studied computer science at the Munich University of Technology (TUM), Germany, and received his diploma degree (M.Sc.) in 2001. In July 2001 he joined the MNM Team. He is currently a Ph.D. candidate at LMU, where he is also working as a research and teaching assistant. His research interests center around integrated network, system and service management, especially regarding QoS and service management.

Holger Schmidt studied computer science at the Munich University of Technology and received his diploma degree (M.Sc.) in 1997. In 1998 he joined the MNM Team. He received his doctoral degree (Dr. rer. nat.) in 2001 at the University of Munich, where he is also working as a research and teaching assistant. His research interests comprise integrated network, system and service management, focusing on service modeling and SLAs.