Service Management Platform: The Next Step in Management Tools

Gabi Dreo Rodosek

Munich Network Management Team
Leibniz Supercomputing Center
Barer Str. 21, 80333 Munich
dreo@lrz.de

Abstract. IT service management is a new, upcoming area of research with several new challenges to cope with. The paper uses a systematic approach to identify new requirements to service management and analyzes some of them more precisely. As a result of the discussion, the introduction of a *service management platform* is motivated. The architecture of a service management platform is specified as well as the integration aspect discussed on a fault management scenario.

1 Introduction

A plethora of new challenges to management is arising due to the paradigm shift from network and systems management to service management. Trends of evolving electronic services, e-commerce, e-business, outsourcing, liberalization and open service market are the drivers of an IT service management. IT services are more and more treated as products with quality and price and it becomes more than ever important to achieve overall management solutions applied across all types of networks, computing systems and applications.

Network and systems management have a device-oriented view on the managed environment. In other words, network and systems management refers to the management of the infrastructure. Management information is specified in terms of MIB variables for a device (e.g. IfInOctets) or application. With the paradigm shift to service management, a new dimension is added. Managed objects are now not only network devices, end systems or applications but more complex objects, namely services. A *service* in our context is comprised of several other services, applications, devices and end systems. It is provided to a customer with a certain Quality of Service (QoS) as agreed upon in Service Level Agreements (SLAs). Thus, a service is provided to various customers with various QoS parameters. For example, a WWW service is provided to one customer with an availability of 99,7% and to another with an availability of 99,8%.

Managing a service means to deal with other services, several applications, devices and end systems as well as dependencies between them. One of the most challenging issues of service management is to recognize what are the service components and how a service is realized (i.e. provisioned) on an infrastructure.

A. Ambler, S.B. Calo, and G. Kar (Eds.): DSOM 2000, LNCS 1960, pp. 59–70, 2000.

In other words, the problem of mapping the service layer to the resource (e.g. network devices, end systems, applications) layer needs to be approached. Examples from the area of performance and fault management show the associated problems with this mapping more precisely. In the performance scenario the mapping of QoS parameters - as agreed in SLAs - to technical parameters and thresholds - as reported by management tools - is discussed. The fault management scenario deals with the questions how to aggregate events from network devices, end systems and applications in order to obtain the status of a service.

Service quality, external provisioning of services, supply chains are examples of other aspects of IT service management. A systematic top-down analysis and the identification of these new research questions are the first steps towards service management. As a result of our discussion, the need for a service management architecture and a service management platform is identified. The objective of the paper is to motivate the development of a service management platform.

Work in this area refers to topics such as the specification of a service architecture for telecommunication services, including some management aspects in the management architecture as specified by TINA [1], the specification of an information model [2], deals with service quality aspects (e.g. [3], [4]) or analyzes the problem area from a process-oriented view as described in Telecom Operations Map (TOM) [5]. Vendors have as well recognized the need to provide a service view on the device-oriented managed environment as demonstrated with Business Process Views from CA, Global Enterprise Manager from Tivoli or SLA reporting tools such as Network Health from Concord or InfoVista from InfoVista. However, existing approaches are only (rudimentary) add-ons to existing device-oriented network and systems management tools. They approach the problem of providing service views from a bottom-up side in terms of what service views can be represented based on the available device-oriented data. By approaching the problem area of IT service management from a top-down perspective, the need for a new management platform, the *service management platform*, becomes obvious.

The paper proceeds as follows: Section 2 identifies aspects of IT service management. Based on these requirements, some of the new challenges and research questions of service management are identified. A particular one, namely the architecture of a service management platform, is discussed in Section 3. In Section 4, the integration aspect of a service management platform is explained on a fault management scenario. Finally, in Section 5, some concluding remarks and interesting open issues of IT service management are sketched.

2 Aspects of IT Service Management

To identify the requirements of service management, we discuss the problem area on a scenario. Fig. 1 motivates and gives us a picture of the topics to discuss. As already stated, a service in our context is a functionality provided to a customer with an agreed QoS at the customer-provider interface. A service may be comprised of other services, applications, network devices and end systems. Services can be provided by external providers or internally within an organization. As depicted in Fig. 1, four types of aspects provide the basis for a systematic analysis:

- organizational aspects,
- service functionality aspects,
- service quality aspects, and
- management aspects.

Organizational aspects refer to several issues. Roles (customer, provider) are certainly one of them. An organization can act in a customer or provider role or in both roles simultaneously. In case an organization is fulfilling both roles simultaneously, it acts as a value-added service provider, using one or more services to set up the value-added service. A service can be provided (in whole or in parts) internally within an organization or externally by external providers. In the discussed scenario in Fig. 1 organization B provides two internal services (DNS and IP) and uses one external service (ATM) from organization C to set up the WWW service. Nevertheless, whether a service is provided internally within an organization - for example from another department - or externally by another provider, the service should be provided with an agreed QoS.

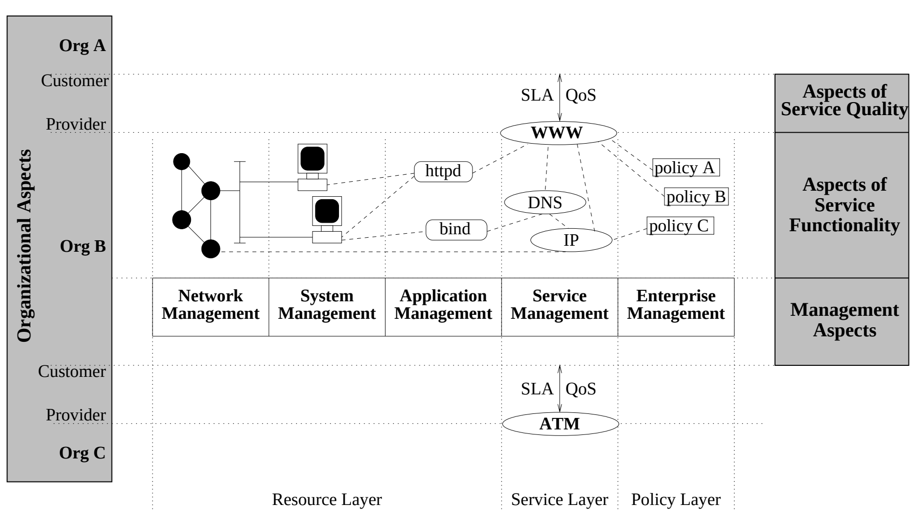

Fig. 1. Scenario

Customer-orientation is another important issue to consider. In our scenario, WWW services of organization B are provided to many customers. For each customer, individual SLAs and QoS parameters have to be specified and enforced. In case these Web services are set up on top of the same infrastructure (devices, servers etc.), a strict logical customer separation is necessary. In case a customer requests a strict physical separation, dedicated devices for each customer need to be provided.

Functional aspects refer to the

- functionality of a service and functionality of its components, such as other (sub)services, applications, end systems or network devices as well as
- dependencies between them.

An approach to describe the dependencies between services (e.g. WWW service depends on DNS and IP) is in terms of a service dependency graph as proposed in [6].

Quality aspects refer to service quality as agreed between a customer and a provider in a SLA. SLAs contain all service-related, organizational and operational information and parameters which are required to describe quality aspects of a service. Certainly, the specification of service-specific QoS parameters as well as the methodology for their measurement and verification are one of the most important ones. Open issues hereby are for example the building blocks of a SLA, SLA management as well as the management of the infrastructure to verify SLAs. One of the objectives of SLA management is to aggregate specific threshold values of devices in a dedicated way to obtain service-oriented parameters, like throughput or availability. Another topic is the specification of service catalogues where several services can be combined together and provided to a customer as one service. An example is the provision of an "Internet service" (including a Web service, DNS, proxy etc.). Classes of service quality (e.g. gold, silver) are other aspects to be consider as well.

As each object needs to be managed, there are several **management aspects** to consider for service management, too. The service lifecycle, including phases like planning, installation, operation and change of a service, is an appropriate way to identify these aspects. Another view to identify requirements to service management is with respect to the functional areas (FCAPS). An example from the area of fault management should demonstrate this. A typical problem of fault management is to map service-oriented trouble reports - as reported by users - to device-oriented events - as reported by management tools - in order to identify the cause of a fault. As already stated throughout our discussion so far such a mapping is not only relevant for fault management but also for other areas like performance or SLA management. The identified aspects provide the basis to systematically identify and refine requirements to service management. This is discussed in more detail for the planning and operation phase. A summary of the discussed aspects is shown in Fig. 2.

The planning phase is certainly one of the most interesting phases of the service lifecycle. Planning was always an issue, also in network and systems management. Although there was always a need for planning support to decide for example whether to expand an international link, change peerings, change the topology or expand the proxies, planning decisions were in most cases done based on experience and knowledge of network and systems managers. The reason is that the usage of planning tools (e.g. simulation tools such as [7]) is in most

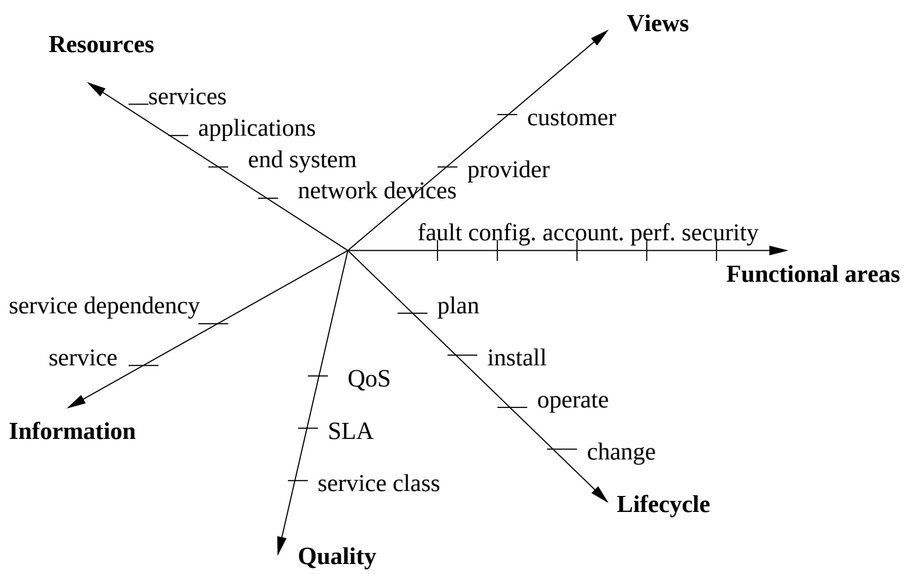

Fig. 2. Aspects of IT Service Management

cases associated with a lot of effort to collect all the necessary - sometimes even not available - data to obtain feasible results.

Nowadays, the importance of planning for the IP service has been rediscovered in terms of IP traffic engineering [8]. The objective of IP traffic engineering is to improve the perceived quality of network services and at the same time to maintain a high level of resource utilization. In spite of Gigabit speeds or even because of that dramatic increase in backbone speed, new precise control over traffic flows in terms of end-to-end management is essential. It should be noted that traffic flows represent *service-oriented* information of the IP service and are necessary to recognize the utilization of the backbone for the purpose of planning (e.g. [9]). What-if scenarios to experiment with configuration changes and traffic optimization are examples of planning issues.

Although planning was already a necessity for network and systems management, it is a must for service management. Adding a new service means to add new applications, network devices, end systems or change the existing resources to cope with the additional load. Examples of planning questions a manager is confronted with are as follows. Will the addition of a new service or new customers influence the quality of the already provided services? What are the potential consequences? What changes in the infrastructure are necessary (adding new network devices, enhance bandwidth, change topology, add new servers, change policies) in case new services with certain qualities need to be provided?

Due to the distributed realization of a service and the dependencies between its components and the resulting complexity, such planning decisions need to be tool supported.

To summarize, the planning phase has identified the following research topics:

- It is necessary to obtain *service-oriented information* for planning purposes (e.g. traffic flows for IP service). The associated problems refer to the questions (i) what is service relevant information and (ii) how to obtain the necessary raw data as well as how to aggregate them to obtain the necessary service-oriented information.
- Due to the complexity and service dependencies, the provision of a decision support tool is a necessity.

Another phase to analyze is *operation*. Examples of the research topics of this phase are the provision of a service view for the (i) service provider, and for the (ii) customer in terms of a Customer Service Management (CSM) [10]. The first point deals with the issue to provide a service view on the device-oriented management environment which requires to map the service layer to the resource layer. Some associated challenges to cope with are (i) to map service quality parameters to technical parameters and (ii) to aggregate device-oriented events to recognize the status of a service.

With the shift to service management and especially SLAs customers do request a view of their subscribed services as well. Customers want to have either scheduled reports about the quality of their services or even online access to the present quality information about their subscribed services. Besides, they want to report problems in a smooth and easy way. Such a customer view can be realized and provided to customers in terms of a CSM. CSM is a part of service management, and should be considered as a separate service.

Associated with the operation phase are policies. A service provider has to specify and enforce policies in order to describe constraints on the operation of services (e.g. some pages of a web server are only accessible from certain subnets). However, this is out of scope for our discussion.

From the discussion above, the following research topics can be identified:

- mapping the service layer to the resource layer (network devices, end systems, applications) and
- to provide (different) service views on the managed environment to providers and customers.

The analysis so far has identified requirements to service management with respect to the analyzed phases of the service lifecycle. In order to identify further research topics other identified aspects needs to be analyzed accordingly.

3 Architecture of a Service Management Platform

Although the previous discussion identified only some of the requirements to service management, the need for a service management architecture and as a consequence for a service management platform became obvious. The specification of a service management architecture is certainly a non-trivial task and is not the focal point of our discussion. The objective of this paper is to motivate the need for a service management platform in order to address the previously mentioned requirements and resulting research topics.

The architecture of a service management platform should be specified in analogy to the architecture of network and systems management platforms. With respect to the architecture of a management platform [11], a service management platform should have the following elements (Fig. 3):

- An infrastructure
  The infrastructure should consists of a repository where the service management information (i.e. the Service MIB) is stored and a standardized communication middleware such as CORBA. The term MIB should be understood in a more wider sense as a Management Information Base which contains complex objects such as services and not necessarily only SNMP MIB variables. A service MIB should describe the service functionality, the dependencies between the components of a service and the associated service-related QoS parameters.
- Basic applications such as:
  - a Service MIB browser,
  - an event management application as well as
  - a service level management application.

    A Service MIB browser provides the capability to load the description of a service into the platform. It is the same as loading a SNMP MIB into a network management platform.

    The objective of the event management basic application is to aggregate device-oriented events - events from the network and systems management platforms - in order to obtain and visualize the status of a service. A service could be for example in the status up, down or degraded in quality.

    The service level management application is concerned with the mapping of service quality parameters to technical parameters like thresholds of a resource. For example, the availability of a service is determined by an appropriate aggregation of technical parameters and thresholds of service components (i.e. resources).

Beside the infrastructure and the identified basic applications, several so-called management applications can be identified from the previous discussion. It should be noted that the reason if an application is considered to be a basic application is whether it is used by other management application. For example, event management provides information for the correlation of user trouble reports or provides the necessary information for the CSM GUI and SM GUI, as depicted in Fig. 3.

Service planning is one of the management applications. It resides on top of a service management platform and enables e.g. what-if scenarios by using for example constraint-based approaches to support planning decisions. Potential

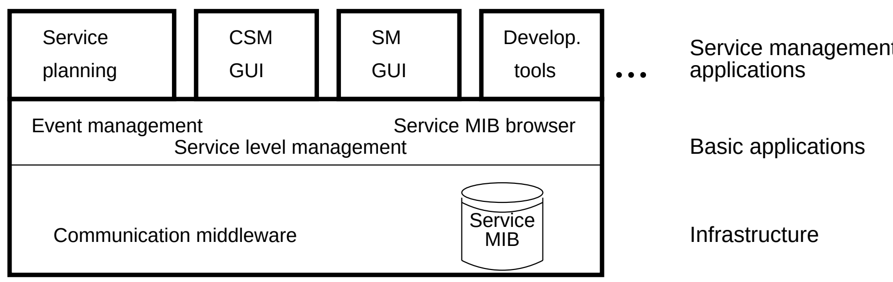

Fig. 3. Architecture of a Service Management Platform

problems and consequences could be identified in case new services and/or customers are added or changes on the service layer could request changes of the underlying infrastructure. Another planning issue is to "move" services around the infrastructure. In other words, it is necessary to support migration of services for example from one server to another. Of course, services are complex objects residing on several resources, thus it is necessary to consider that the migration involves several resources. Such a feature is associated with several issues from trivial ones (e.g. enough space on the target server) till complex planning issues and configuration changes. Constraint-based approaches are certain a reasonable way to address this problem area (e.g. [12]).

As already mentioned, a service management platform should provide **service views** on the managed environment to the provider itself and to the customers. Thus, on top of a service management platform (Fig. 3) the following GUIs need to be provided:

- the GUI for a service provider (SM GUI),
- the GUI for the customers (CSM GUI).

CSM as described in [10] realizes a lot of functionality which in fact belongs in a service management platform. The aggregation of device-oriented thresholds in order to obtain service-oriented parameters for the IP service is certainly an example of a basic application, namely the service level management. Because a service management platform is missing so far, the mentioned functionality has been realized as a part of the CSM application. As soon as a service management platform is developed, the current CSM application and implementation is reduced to a simple CSM GUI (as shown in Fig. 3), providing *one* interface to the customer. Customers use CSM to (i) inform themselves about the quality of their subscribed services, and (ii) can actively report problems over one interface. Thus, Web interfaces to trouble ticket systems or the Intelligent Assistant [13] should be part of CSM.

With respect to the discussion so far, the architecture of a service management platform can be specified as shown in Fig. 3. Examples of basic applications of a platform are event management, service level management and a

Service MIB browser. Examples of management applications are service planning, a CSM GUI and a SM GUI. The CSM GUI refers to the customer view of a service whereas the SM GUI refers to the service provider view.

Another important aspect of a service management platform are development tools, such as an editor for the generation of new service descriptions, descriptions of QoS parameters and the generation/updates of the service MIB.

To summarize the previous discussion, the functionality of a service management platform should provide:

- the ability to deal with service descriptions in terms of a Service MIB (i.e. it is necessary to load/unload the service MIB in the service management platform),
- the ability to describe services and dependencies between service components (i.e. development tools like editors, compilers),
- to provide service views on the managed environment to the provider and customers, and to
- map services to resources (e.g. to aggregate events from network and systems management to service-oriented events, to map QoS parameters to technical parameters of devices).

Throughout the discussion the central point was always the mapping from the service layer to the resource layer. Of course, such mapping needs to be done automatically and tool supported. A precondition for this is to describe characteristics of resources in an appropriate way (e.g. server with Gigabit Ethernet interface with several CPUs). Besides, the service requirements to resources need to be specified as well (e.g. a SAP service for a large enterprise with an availability of 99,9%). To approach this problem means to specify a framework for dynamic service provisioning. This is, however, out of scope for this paper.

The service management platform has an important role to play in the management environment. It has the role of a **master** for the network and systems management platforms. This means that the service management platform sets for example thresholds of devices, event configurations of the network and systems management platforms. In fact, it configures both device-oriented platforms. Therefore, it is necessary to standardize ("smart") interfaces to both device-oriented managers. The following section should demonstrate the integration of the service management platform with device-oriented managers on a correlation scenario.

4 Applicability of the Service Management Platform

With the introduction of a service management platform the integration issue becomes even more important than in today's management. An impression of the new challenges should be given by the following example from the area of fault management.

The necessity to map service-oriented views to device-oriented is a hot topic in today's management, especially in fault management. Users report their problems from a service point of view. They report that they can not send an email, have problems to access a web site or can not send orders in SAP R/3. On the other side, network and systems manager have a device-oriented view of the managed environment in terms of ups and downs of nodes, links or running processes on dedicated workstations. The objective of fault diagnosis is to map both views in order to recognize the cause of the fault.

To be able to map these views, several problems need to be approached. From the device-oriented view, the problem is the enormous amount of events to deal with as reported from management tools. To cope with the enormous amount of events, event correlation tools like InCharge from SMARTS or Tivoli Enterprise Console (TEC) are used to correlate events. On the other side, users report problems in most cases in an unprecise manner (an approach to deal with this problem has been presented in [13]).

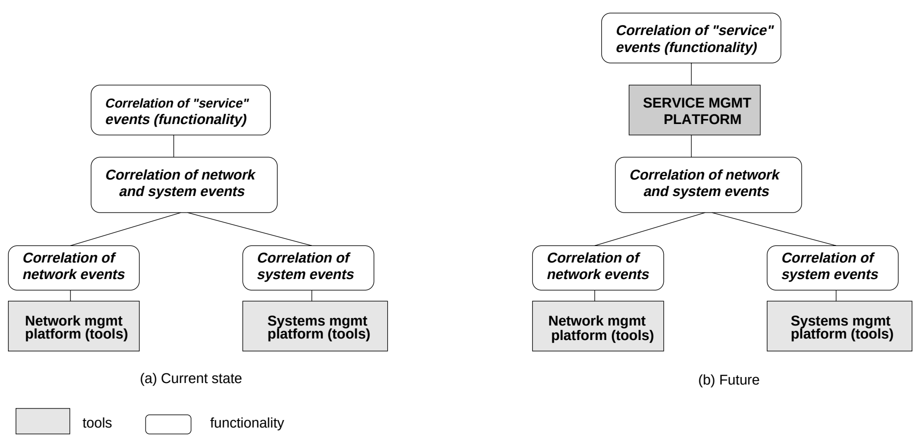

Fig. 4. Levels of Event Correlation

Event correlation is performed on various levels (as depicted in Fig. 4 (a)) with respect to topological, temporal and functional criteria. Mostly, the correlation of network events is performed as part of a network management platform (e.g. a down/up trap of a node within a certain time interval should be suppressed). Such correlation tools are mostly a part of a management platform (e.g. ECS from Seagate as a part of HP OpenView Network Node Manager and IT/O, InCharge from SMARTS as a part of Tivoli NetView). Another level is the correlation of system events (e.g. wrong passwords). The next level is the correlation of network and system events with respect to topology. For example, a reason that a workstation does not respond to a ping could be that the workstation is not working correctly or that there is a problem with the connectivity. To recognize the cause of the fault, network and system events need to be correlated and sometimes actively polled in addition.

The last level of correlation is done with respect to functional criteria. The knowledge about the dependencies between services is nowadays hardcoded for example in rules of TEC, based on the experience and knowledge of some experts. The generation of the rules (i.e. the knowledge acquisition), the maintenance and the update of the rules is in most cases a difficult, time consuming task.

After the events are correlated also with respect to the functional aspect, the resulting events generate automatically a trouble ticket. Fault diagnosis is afterwards performed according to the knowledge of the support staff who still has to correlate trouble tickets reported from users with the automatically generated trouble tickets. However, the task is certainly simplified.

With the introduction of a service management platform, the functional aspect and with this the service dependencies would be handled completely within a service management platform (Fig. 4 (b)). Such correlation would be performed based upon the Service MIB. According to this, an appropriate aggregation of device-oriented events would be possible in order to obtain the status of a service. If a customer reports a problem with his service, and the status of his subscribed service is visualized in a service management platform, fault diagnosis is simplified essentially.

The integration of a service management platform could coordinate also the usage of other tools. Currently, almost every management tool makes its own polling and/or autodiscovery (e.g. SLA tools like Network Health, management platforms, device-specific tools like CiscoWorks) and this in a more or less uncoordinated way. By using a service management platform such in certain extent unnecessary pollings and autodiscoveries could be omitted.

5 Conclusions and Open Issues

Service management is certainly the next upcoming area of research and the next dimension in management. To recognize what are the requirements of these new research topic, aspects of IT service management have been introduced as a basis for a systematic analysis. Some of the requirements have been discussed in more detail and the need for a service management platform has been identified. The paper points to the functionality and the architecture of a service management platform by describing also the integration issue on a correlation scenario.

Beside the mentioned benefits, the usage of a service management platform would also improve proactive management. Due to the knowledge, what impact a component of a service has on the quality of a service, appropriate preventive actions could be taken in advance before serious service degradations would occur.

There are several open issues of IT service management to deal with. To sketch a few, the specification of a service management architecture is one of the most important ones, including tasks of specifying the Service MIB, specifying a framework for dynamic service provisioning etc.

Acknowledgments

The author wishes to thank the members of the Munich Network Management (MNM) Team for helpful discussion and valuable comments on previous versions of the paper. The MNM Team, directed by Prof. Dr. Heinz-Gerd Hegering, is a group of researchers of the Munich Universities and the Leibniz Supercomputing Center of the Bavarian Academy of Sciences. Its webserver is located at `http://wwwmnmteam.informatik.uni-muenchen.de`.

References

1. "Service Architecture," TINA baseline, June 1997.
2. "Common Information Modell (CIM) Specification, Version 2.2," Tech. Rep., Distributed Management Task Force, June 1999.
3. P. Bhoj, S. Singhal, and S. Chutani, "SLA Management in Federated Environment," In Sloman et al. [15].
4. L. Lewis, *Service Level Management for Enterprise Networks*, Artech House, 1 edition, 2000.
5. "Telecom Operations Map," Evaluation Version 1.1 GB910, TeleManagement Forum, Apr. 1999.
6. G. Dreo Rodosek and Th. Kaiser, "Determining the Availability of Distributed Applications," In Lazar et al. [14].
7. `http://www.caci.com`, *COMNET*.
8. IEEE Network, *IP Traffic Engineering*, vol. 14, 2000.
9. G. Dreo Rodosek, Th. Kaiser, and R. Rodosek, "A CSP Approach to IT Service Management," In Sloman et al. [15].
10. M. Langer, S. Loidl, and M. Nerb, "Customer Service Management: A More Transparent View to Your Subscribed Services," In Sethi [16].
11. H.-G. Hegering, S. Abeck, and B. Neumair, *Integrated Management of Networked Systems – Concepts, Architectures a nd their Operational Application*, Morgan Kaufmann Publishers, ISBN 1-55860-571-1, 1 edition, 1999.
12. M. Sabin, A. Bakman, E.C. Freuder, and R. D. Russel, "A Constraint-Based Approach to Fault Management for Groupware Services," In Sloman et al. [15].
13. G. Dreo Rodosek and Th. Kaiser, "*Intelligent Assistant*: User-Guided Fault Localization," In Sethi [16].
14. Aurel Lazar, Roberto Saracco, and Rolf Stadler, Eds., *Proceedings of the 5th International Symposium on Integrated Network Management, San Diego*. IFIP, Chapman-Hall, May 1997.
15. M. Sloman, S. Mazumdar, and E. Lupu, Eds., *Proceedings of the 6th IFIP/IEEE International Symposium on Integrated Network Management, Boston*. IFIP, IEEE Publishing, May 1999.
16. Adarshpal S. Sethi, Ed., *Proceedings of the 9th Annual IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM'98), Newark*, Oct. 1998.