A DEVELOPMENT FRAMEWORK FOR THE PROVISION OF INTEGRATED MANAGEMENT SERVICES

Martin Hans Knahl
Network Research Group
University of Plymouth
Plymouth PL4 8AA
mknahl@plymouth.ac.uk

ABSTRACT

A rapid development of communication services over interconnected local and wide area networks can be seen on the market. This evolution of networked systems and services requires rapid and flexible development of management services. In this paper, we propose a development framework for the provision of management services that facilitates rapid and flexible construction of integrated network and system management services. The different stages and aspects in management service development are addressed to provide a fundamental basis for the planning and implementation of Integrated Network and System Management. The INSMware framework proposes a high-level logical model in order to accommodate the range of requirements and environments applicable to Integrated Network and System Management that can be used as a reference model. The INSMware approach has further been combined with the componentware paradigm for the implementation of a practical management system.

KEYWORDS

Network and System Management, System Development, Componentware

1. INTRODUCTION

As a result of the recent rapid evolution and development in the area of network and system technologies, heterogeneous environments using different communication and management protocols are omnipresent in Local and Wide Area Networks (Lewis, 2003; Mayerl, 2001). There are a number of different management systems and protocols and it can be expected that there will be further evolution in the future (Mayerl, 2001; Patel 2002). To integrate various network devices and services and to manage them altogether, we must be able to provide scalable management systems and to handle various network management protocols (Lewis, 2003; Naik, 2004).

This paper proposes a new Integrated Network and System Management (INSM) development framework for the provision of integrated management services. Section 2 reviews the state of the art of current network and system management related research and identifies critical aspects of current management approaches. Section 3 explains the principles of the proposed INSMware framework and illustrates key aspects of the approach. Section 4 discusses practical issues and experiences of the proposed approach. Section 5 concludes the paper.

2. ANALYSIS OF CURRENT RESEARCH

The overall aim of an Integrated Network and System Management (INSM) framework is the automation and delegation of management related services to enable a human administrator to concentrate on high level activities such as network planning and improvement (Mayerl, 2001).

Management platforms have been widely adopted for network and system management (Dornan, 2001; Jander, 1999). However, in practice management applications running on management platforms tend to be loosely integrated with each other. Management platform providers (e.g. Hewlett Packard) use proprietary approaches and implementations to provide a distributed management infrastructure. Furthermore, management platforms such as Tivoli or HP Open View usually require extensive customization to reflect specific characteristics of the managed environment and to integrate third party applications via proprietary management platform APIs (Knahl, 2002). In addition to standard services that must be provided by a management system the requirement to offer customised services (e.g. user controlled activation of communication services, integration with workflow management systems to facilitate distribution and automation of the various processes, increased mobility of users) for users with specific requirements (e.g. specified in a Service Level Agreement) will further increase (Patel, 2002; Naik, 2004). To cope with the challenges, management systems can be composed of a range of components that perform the required management services (Lewis, 2003; Knahl, 2004). However an exploration of the management services and technologies that are required for current and future environments is still the subject of research (Foster, 2004; Lewis, 2003). While traditional middleware architectures such as CORBA and DCOM have focused on achieving interoperability across heterogeneous platforms and software languages, they lack provision of end-to-end management (e.g. for QoS) for distributed applications or to dynamically adapt to changes in the underlying communications infrastructure (Kalogeraki, 2004)

Integrated Network and System Management development and operation is an area that is currently not well addressed by standards and design guidelines (Lewis, 2003; Mayerl, 2001). The IETF management related standards focus primarily on TCP/IP related management provision, the TMN family of standards focusing primarily on telecommunications networks (King, 2000; Patel, 2002). System management is commonly encountered on an application specific basis. General methodologies and approaches for the management of distributed applications and services are still at the development stage (e.g. license and performance monitoring) (Debusman, 2002).

The TeleManagement Forum (TMForum) has addressed integrated management development and has produced some guidance on how Management Systems can be planned and developed within their architectural frameworks. The TM Forum approach proposed open telecommunications management interface development that draws heavily on object-oriented analysis and design techniques and developed some specific management interfaces. A framework of telecoms management business processes (termed Telecoms Operation Map) can be used to identify which management tasks should be analyzed to develop industry interface agreements (TMF-OM, 2000). A novel management system should reflect this and combine it with an appropriate planning and operation strategy to integrate management into the business processes (Knahl, 2004). However the heterogeneity of today's communication infrastructures and services in which Management must be performed means that a common and accepted architecture for the provision of Integrated Network and System Management is currently not available (Dornan, 2001; Lewis, 2003; Patel, 2002).

The integration of management technologies and applications is an issue that has not been resolved satisfactorily (Lewis, 2003). The integration of management services is becoming increasingly critical because of new services and resources in networked systems and because of the diversity of management architectures. One challenge is at which level integration should take place (e.g. at the management tool level), what should be integrated (e.g. user interface, events or data) and what the integration interface should look like. Integration is an extremely important issue. Until now, only partial solutions have been available and the lack of integration of management services for existing IT infrastructures results in higher costs and personnel requirements. It is not only a system or technology oriented problem but also an organisational problem (Mayerl, 2001).

These trends inevitably have an impact on the development process. Since a management instrumentation of existing resources and services is complicated and expensive, there should be a move to incorporate the management requirements in the development process (i.e. to take the management aspects into account at the stage of resource and service development). Developers have to be aware of these management related problems. The main reason for this is that the dynamic nature in some areas of existing and future IT services with great potential is being slowed down or delayed considerably due to a lack of management concepts and services. Examples for this include:

- The state of current network infrastructures and communication services falls far short of what is technically possible of being achieved. One of the reasons for the limited uptake of state of the art enabling technologies can be found in the inherent management challenges of new technologies.
- The uptake of Quality of Service sensitive services (e.g. Voice over IP or videoconferencing) is partly being slowed down because of the lack of appropriate end-to-end management services. These are

required before organisations will adopt these services on a large scale (e.g. before Voice over IP will be a serious alternative to the established legacy phone services).

- The implementation, operation and maintenance of management services is complicated and expensive as current INSM tools and services do not automatically adapt to the environments being managed .
- management services are enterprise critical as they treat trust and privacy issues from an enterprise and user's perspective (Naik, 2004).

The development of management services must focus on end-to-end management service provision and the development technologies should facilitate rather than dominate the planning and implementation stages. The goal is to develop integrated solutions and to avoid technology conflicts (i.e. practitioners are concerned with integrating different existing technologies) (Knahl, 2002; Lewis, 2003).

3. INEGRATED NETWORK AND SYSTEM MANAGEMENT DEVELOPMENT FRAMEWORK

Figure 1 presents the INSMware high-level logical model that can be used as a reference model to identify and accommodate the range of requirements and environments applicable to Integrated Network and System Management (INSM) (Knahl, 2004).

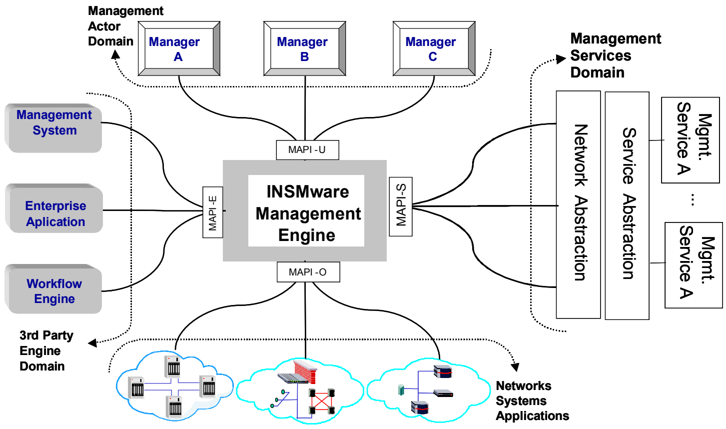

Figure 1: INSMware reference model

The interface around the central Management Engine is referred to as MAPI (Management Application Programming Interface) and is used to access and control the communication between the Management Engine and the different domains (Knahl, 2002). MAPI provides a general interface description containing the subset of interactions common to two or more domains. MAPI-Engine (MAPI-E), MAPI-User (MAPI-U), MAPI-Services (MAPI-S) and MAPI-Objects (MAPI-O) are domain specific extensions (see Figure 1). When building an INSMware management system by assembling building blocks, the various integration and distribution aspects need to be addressed and agreed upon by building block, building block Interaction Protocol and building block Interaction Broker providers so that different building blocks are interoperable. In an open building block market, this can potentially lead to the reduction of the development burden for common tasks. The 2 main areas of integration are horizontal integration and vertical integration that should be addressed by standardisation. In addition to the underlying services being shared (addressed by horizontal standards), building blocks must also inter-operate within a service domain in which building blocks will inter-operate (addressed by vertical standards).

The management service Domain of the INSMware framework contains the core services of the Management Domain and the Service Support Domain (Knahl, 2002). The Network Abstraction layer models the service resources and the Service Abstraction layer is responsible for the modeling of the overall management service. The specification of management services can be further subdivided into several categories depending on the type and scope of a management service. Provision of management services will coexist and must be integrated with other existing and future business processes. Therefore, a management system needs the means to integrate and adapt existing and new INSM and other services (e.g. integration of multiple Management Engines or integration with other business applications such as SAP R/3). The MAPI-E interface provides the mechanisms that are required to enable one Management Engine to make requests to another system and to effect the selection, activation and control of management services. Various managed objects exist that support a number of different management protocols that must be integrated to enable end-to-end management.

The proposed INSMware planning and operation methodology consists of building an initial model of a system based on the INSMware reference model followed by the addition of further implementation and process details during the design of a system. The proposed methodology emphasises the iterative nature of the development process and the pre-eminent place of the building block definition within it (Knahl, 2002). Representing building block at an analysis level as well as at a design level enables building block reuse to become much more central to the analysis process and to become directly relevant to the business process reengineering activity (Knahl, 2002). Given a building block interaction broker that a set of building blocks can "plug into," the underlying goal is to agree upon a set of principles which will create a sense of uniformity among building blocks on as many issues as is useful and practical for a given environment (Knahl, 2004).

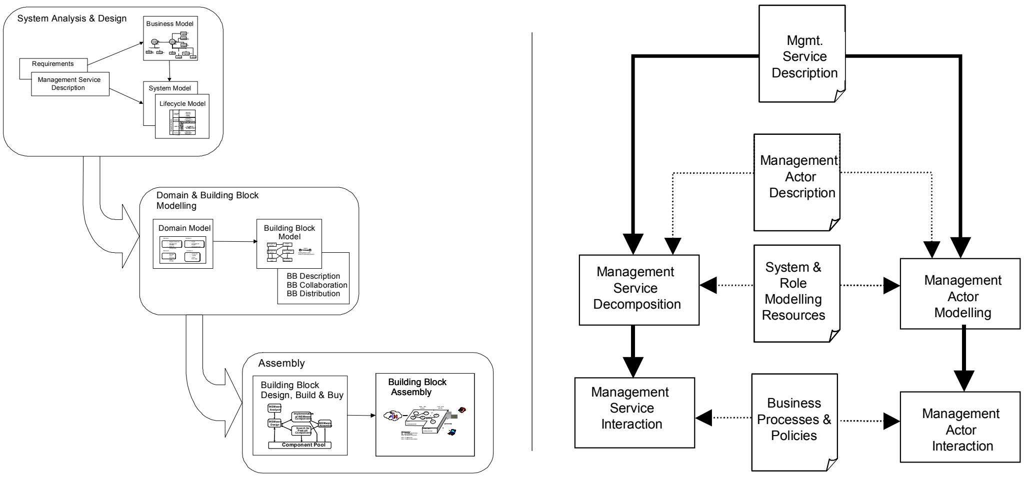

Figure 2: Overview of INSMware methodology and system modeling

The main activities of the INSMware development methodology are the system analysis and design (requirements capture), domain and building block modelling (management service specification) and the system modelling and assembly as illustrated on the left in Figure 2 (Knahl, 2002). The three phases are further subdivided into separate activities. Each activity may be influenced by, or take as input the results of another activity. The major relationships between the different activities are represented by arrows. However, the Methodology process is iterative (all activities may be repeated several times or the entire methodology may be repeated as a whole) and some phases can be carried out in parallel. The INSMware system modelling identifies the different stakeholders and Management Actors and specifies their roles and their relationships within the overall management system. The Management Service Description is the main input for the System Model. Furthermore, the System Model can be used to determine the interactions associated with the different Management Actors. The System Model provides an organisational context for the management services described in the Management Service Description. For the modelling of Management Actors, the Management Service Description can be used as a main input. Additional resources such as existing descriptions of business processes and policies or Management Actor descriptions in conjunction with investigations regarding the interactions between the relevant actors and other existing resources can be used to obtain a precise model. These Management Actors and their roles form a starting point for Management Actor modelling using the concepts described in the INSMware framework. In addition, the interactions between the Management Actors in the system model must be described. An example flow of the Management Actor modelling activities is shown on the right hand side in Figure 2. The system model can further be used for the refinement of the envisaged management services (e.g. establish organisational boundaries or context of management services).

4. FRAMEWORK DEVELOPMENT

The layers addressed by the INSMware development framework are illustrated in Figure 3 (i.e. collections of processing functions that together comprise a set of rules and standards for successful management service provision). The INSMware framework offers a conceptual model for an application architecture providing the means to group related functions into building blocks that use the underlying infrastructure in a regulated way (Knahl, 2002). On the highest layer of the model are the management services that support generic business processes. Below this layer are the layers addressing the INSMware framework and INSMware infrastructure principles that are utilised to implement the management services. The problem is how to provide deployable units of management building blocks that can inter-operate with other building blocks that can be developed by different building block providers independently from one another at different times in different organisations. Componentware can be used as an enabling technology for the INSMware development framework (Knahl, 2002). It can facilitate the implementation of building blocks and their design principles to establish a market for common building blocks to be used for different INSM implementations (Knahl, 2004). Some INSMware building blocks, such as those to provide user notifications and data storage, are useful broadly across all INSM environments (i.e. horizontal domain common building blocks). Other INSMware building blocks, such as those to provide configuration management of a proprietary managed object, are useful within a single (or limited number) of management environments (i.e. Vertical Domain Common building blocks).

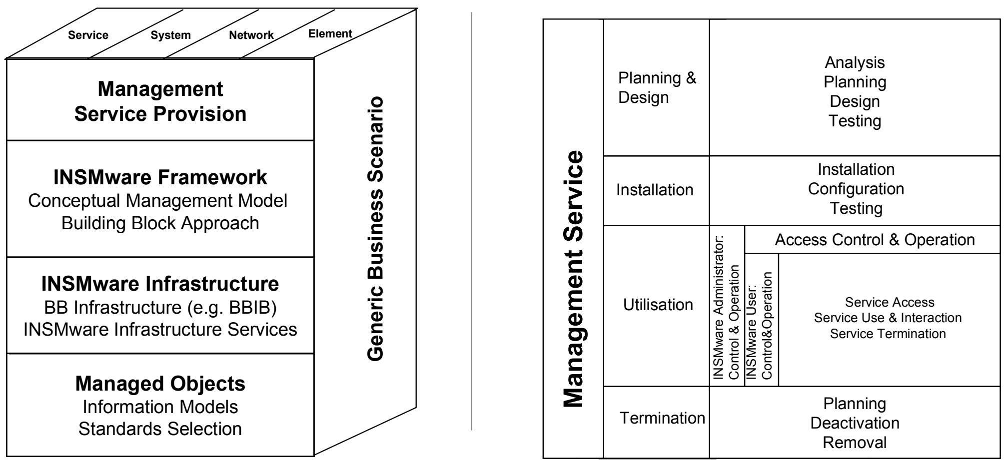

Figure 3: INSMware layers and Lifecycle Model

The INSMware Lifecycle Model defines the various functional areas and facilitates the identification of the dependencies between actors. Several management areas from different Management Actors' perspectives must be considered for the lifecycle phases of management services to be offered to different users. The different contents of the lifecycle phases are further refined in the proposed model (e.g. Installation, Configuration, Testing for the lifecycle phase Installation). The different lifecycle phases, namely Planning & Design, Installation, Utilisation and Termination, can be viewed from the perspective of the INSMware administrator providing the service and INSMware user using the service. This enables specific management functionality to be associated with a particular actor's management system and ensures that the management functions available in the INSMware administrator and user are complementary and consistent.

In a market of software components, different stakeholders will have different roles and requirements in the market context (Knahl, 2002; Lewis, 2003). The INSMware development context model identifies the different actors and the main relations within the INSMware development process (see Figure 4). However, further connections might exist (e.g. a management service User or INSM Service Provider may make direct requests to a Standard Provider or a Management Component Provider). The main aim of the INSMware Developer is to increase productivity and profitability (e.g. reduce development and maintenance costs) in order to secure continued development contracts from an INSM Service Provider. A number of limitations must be taken into account for the INSMware Developer (e.g. limited start-up resources for a new venture, limited development capabilities that may result in limited capability of initial release). However, it is required to provide full interoperability to gain market acceptance (i.e. to ensure that small INSM Service Provider systems can interact with large service provider systems). The INSMware Developer develops the management system for the INSM Service Provider. An INSM Service Provider may select different INSMware Developers for the provisioning of management services. The model separates these roles in the anticipation that the development, integration and deployment will be increasingly independent.

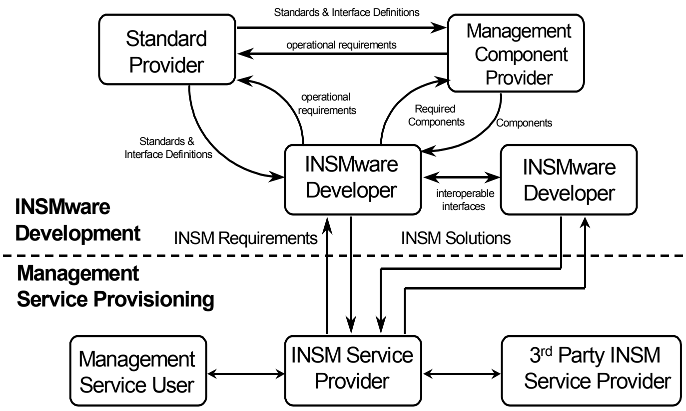

Figure 4: *INSMware development context model*

The INSM Service Provider is concerned with the provisioning of management services. The current lack of inter-domain network and system management standards and the lack of integrated products that support such interactions implies that implementation of interoperable interfaces between INSMware Developers must be addressed (Knahl, 2002). This relationship further introduces the requirement for INSMware Developers to interact with other developers to provide interoperable management services to the INSM Service Provider. The Standard Provider develops standards to establish new technologies. The Management Service User requires open service management interfaces that enable the integration of INSM Service Providers, thus encouraging competition between them. INSM Service Providers require management interface standards to simplify the implementation of third party relationships or to meet regulatory interoperability requirements. Both the INSMware Developer and the Management Component Provider have an interest in using standards that promote software portability and integration (i.e. Middleware technologies such as CORBA). The model highlights the need for open standards to underpin a competitive market both in the delivery of management components, integrated management systems and the delivery of management services. It provides an approach for the design of an integrated management framework that takes the requirements of standard providers as well as system developers into account (rather than being driven by architectural and technological considerations of a particular environment) and will potentially result in more powerful and interoperable management solutions.

Software reuse is widely seen as a key approach in meeting the challenges of developing software within cost and time constraints (Knahl, 2002; Lewis, 2003). One of the most promising techniques is the reuse of commercial off the shelf software components developed by a third party, here modelled as Management Component Provider. With software component reuse, the costs of developing and maintaining the INSMware software components falls on the Management Component Provider, who recoups it by selling the component to as many customers (i.e. INSMware Developer) as possible. In addition, the INSMware Developer can further practise software reuse internally by reusing solutions to frequently occurring problems or by implementing standard management solutions popular with management service providers.

A prototypical implementation of the approach has been developed using a number of INSMware software components running on top of industry standard distribution architectures (i.e. CORBA and DCOM) (Knahl, 2004). The design of the individual INSMware components is based on a domain specification that subdivides the entire application domain into sub domains and integrates SNMP managed objects. Figure 5 shows an overall view of the INSMware development process. After the INSMware domains and building blocks have been specified, software components implementing them have been searched for. A number of components developed in the Distributed Application Systems research group at the University of Applied Sciences Darmstadt, Germany could be integrated into the INSMware prototype (i. e. the Communication Component, Event-Controller and Database Component) (Knahl, 2002). The remaining software components have been developed for the INSMware prototype as no matching software components were found. When software components are used to build a system, existing components can have impacts on the design process (as the usage of existing software component may influence the design of software component to be developed). Also, method signatures, parameter names and types may have to be adopted for existing software components in order to work together properly. Furthermore, the INSMware software components need to be assembled (i.e. embedded into an operational infrastructure). In the case of CORBA or DCOM based distributed systems, an implementation specific Interface Definition Language Compiler is used to create component containers at compile time that can be utilised by a particular middleware infrastructure. The functionality of the illustrated INSMware system consists of the processing, filtering, and analysis of management relevant data, the presentation in a graphical user interface and user notifications at the occurrence of predefined states that represent important network states. The system allows that one or more users may be notified over varying communication channels.

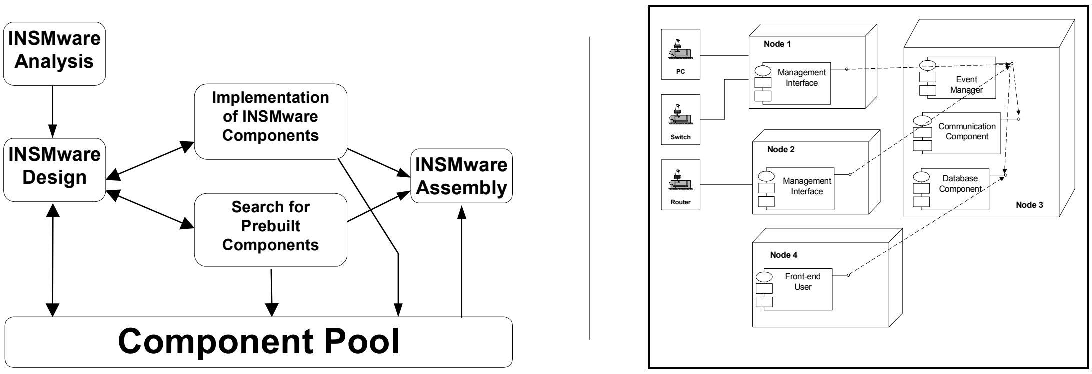

Figure 5: INSMware development process and implementation

Using the INSMware components several distribution scenarios have been realised. In the smallest configuration, all components are installed on one system. In this scenario, remote users connect to this system to receive all information. In the fully distributed configuration, all components are installed on different computers. However, in practical terms it is better to implement the Database Component, the Communication Component and the Event Manager on one system (see right of Figure 5). Using this distribution, a good performance can be achieved as this distribution situation avoids heavy traffic through the network connecting the different computers (Knahl, 2002). Furthermore the Management Interface may require a lot of hardware performance and could affect the performance of the overall system.

INSMware can be adapted to changes and evolutionary modifications through the adoption of componentware principles (Knahl, 2004). Applications that must be modified due to changing requirements can do so by replacing INSMware components with new component versions. Additional software components can be linked with the core system and can migrate to other systems thus allowing the system topology to evolve and be adapted to new management requirements and network topologies. The INSMware system is highly distributed, component based and service oriented and INSMware components have to run in a multitude of heterogeneous networks. Performance requirements further imply that the system must be adaptable to changes in network performance and workload intensity (Menasce, 2004). INSMware extensibility has been further facilitated by splitting the Management Interface component into several layers (i.e. management-domain specific component and generic Filter Component). To integrate managed objects that require different management connectivity (e.g. another Management Protocol), new “drivers” can be added. This mechanism allows extensibility in addition to facilitating application tailoring, which can be done by adding and distributing further components.

5. CONCLUSION

The INSMware development framework provides a comprehensive architecture that identifies of a set of building blocks (e.g. implemented using the componentware paradigm) that can be distributed over the network. It provides a basis for the provision of integrated services and facilitates the efficient deployment of these services. However, management related activities may slow down the introduction of new technologies and hence slow down the deployment of new services if management issues are not addressed carefully at the outset. Hence, it is essential to have an integrated management framework such as INSMware for the provision of flexible IT services. Management issues have to be taken into account during the design of each new technology and service. The same applies to the definition of management services in connection with the planning and operation of IT infrastructures. Any disregard of these basic rules will increase costs and will affect the viability of the provided services.

INSMware prototypes have been implemented for different middleware platforms (DCOM and CORBA) to demonstrate that the architecture is independent of specific technologies. Further work will consider the adoption of the Web Services paradigm for distributed management services (Vinoski, 2004). Developers demand architectures that render their task easier and today's users are accustomed to application ease of use. Hence, a development framework must allow applications to be built that can be customised by the final user rather than only by developers. The separation of the Front-End System into Front-End User and Front-End Administrator enables to the configuration of the system.

Integrated management systems must be changed in order to keep up with the rapidly and continually changing world of the IT infrastructure. We propose a componentware based INSM that can be applied flexibly and efficiently to meet today's management requirements. Further development of the prototype is undertaken at that stage together with an adaptation to various application scenarios.

REFERENCES

Debusmann, M. et al., 2002. Generic Performance Instrumentation of EJB Applications for Service Level Management, *Network Operations and Management Symposium (NOMS) 2002, Florence, Italy.* 2002

Dornan, A. , 2001. Network Management In A Crisis. *In Network Magazine.* July 2001.

Foster, I. et al. , 2004. End-to-end quality of service for high-end applications. *Computer Communications.* Volume 27, Issue 14, 1 September 2004.

Jander, M. , 1999. Management Framework Survey - Framework Fraud? *Data Communications International.* September 21.

Kalogeraki, V. et al. , 2004. Managing Distributed Objects in Peer-to-Peer Systems. *IEEE Network..* January/February 2004.

King, A. and Hunt, R. , 2000. Protocols and architecture for managing TCP/IP network infrastructure. *Computer Communications.* Vol. 23.

Knahl, M. , 2002. *A Generic Network and System Management Framework.* PhD Thesis, University of Plymouth.

Knahl, M. , 2004. Componentware based Network and System Management Development, *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN 2004),* Innsbruck, Austria, 17-19 February 2004.

Lewis, D. et al, 2003. Towards the Technology Neutral Modelling of Management Components. *In Journal of Network and Systems Management,* Vol. 11, No. 1.

Mayerl, C. , 2001. *Eine integrierte Dienstmanagement-Architektur für die qualitätsgesicherte Bereitstellung von Netz- und Systemdiensten.* Shaker Verlag, Germany.

Menasce, D. A. , 2004. QoS-Aware Software Components. *IEEE Internet Computing.* March/April 2004.

Naik V. K. et al. , 2004. An architecture for the coodination of system management services. *IBM Systems Journal.* Vol. 43, No. 1, 2004.

Patel A. , 2002. Current status and future directions of software architectures for telecommunications. *Computer Communications.* Vol. 25.

TMF-OM, 2000. *TeleManagement Forum - Telecom Operations Map.* GB910/v2.1. Tele Management Forum Specification.

Vinoski, S. , 2004. Web Services Notifications. *IEEE Internet Computing.* March/April 2004.