51. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

PROCEEDINGS

11-15 September 2006

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION SCIENCE

INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING - DEVICES AND SYSTEMS, MATERIALS AND TECHNOLOGIES FOR THE FUTURE

Startseite / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=12391

TECHNISCHE UNIVERSITÄT
ILMENAU

Impressum

Herausgeber: Der Rektor der Technischen Universität Ilmenau
Univ.-Prof. Dr. rer. nat. habil. Peter Scharff

Redaktion: Referat Marketing und Studentische Angelegenheiten
Andrea Schneider

Fakultät für Elektrotechnik und Informationstechnik
Susanne Jakob
Dipl.-Ing. Helge Drumm

Redaktionsschluss: 07. Juli 2006

Technische Realisierung (CD-Rom-Ausgabe):
Institut für Medientechnik an der TU Ilmenau
Dipl.-Ing. Christian Weigel
Dipl.-Ing. Marco Albrecht
Dipl.-Ing. Helge Drumm

Technische Realisierung (Online-Ausgabe):
Universitätsbibliothek Ilmenau
ilmedia
Postfach 10 05 65
98684 Ilmenau

Verlag: isle
Verlag ISLE, Betriebsstätte des ISLE e.V.
Werner-von-Siemens-Str. 16
98693 Ilrnenau

ISBN (Druckausgabe): 3-938843-15-2
ISBN (CD-Rom-Ausgabe): 3-938843-16-0

Startseite / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=12391

51st Internationales Wissenschaftliches Kolloquium
Technische Universität Ilmenau
September 11 – 15, 2006

H. Kärst, R. Böringer, A. Mitschele-Thiel

Workflow for Autonomic Reconfiguration of Wireless Access Networks (ARN)

ABSTRACT

The complexity of traditional network management increases due to the rising heterogeneity of wireless networks. Future wireless access networks will contain network elements supporting different types of radio technologies, e.g. WLAN, WiMAX, UMTS, GSM. Therefore resource discovery, integration, configuration, monitoring, failure discovery, failure recovery, adaptation to environmental changes of network resources and network services will require enormous management resources.
Nowadays, network management resources already cause a large portion of the initial and operational cost of a network infrastructure. This portion will increase again due to the rising management complexity of heterogeneous wireless access networks.
Self-management of access networks minimizes human interventions during the management process. The network monitors the resources and the environment itself. Partially it autonomously decides which management processes have to be executed. Self-management allows to minimize operational cost of network management and to increase the flexibility of network services provided by heterogeneous wireless networks. This paper presents a workflow to describe the required tasks for the implementation of a self-managed wireless access network, referred to as Autonomic Reconfigurable Network (ARN). The workflow simplified the development of self-managed architectures.

INTRODUCTION

The classical management methods are based on a centralized network management system (NMS; e.g. the SNMP manager [1]). The NMS is able to monitor and perhaps to configure several network parameters, e.g. amount of traffic or utilization of network interfaces. The parameters will be distributed by means of management clients (e.g. SNMP-Client). For some network parameters, certain threshold values can be defined. If the threshold value of an observed parameter exceeds, the NMS produces an event and informs the human network administrator. The administrator decides which management processes have to be activated. In some cases, the NMS is able to react without human interventions by the use of simple predefined policies or rules, when certain events occur (e.g. restart of a faulty network element).

Due to the different network technologies, a future wireless access network will include various technology-dependent features. Everything is integrated in a common managed network environment. Therefore, a huge amount of different network parameters has to be considered, to be able to manage the network in an optimal fashion. It will overstrain the capacities of classical network management.

Additionally, the classical network management assumes that the network conditions are static and often quite well predictable. However, due to the mobility of customers the conditions of future heterogeneous networks will be dynamic in a wide range. Therefore, the classical network management will not be able to adapt network resources fast enough compared to changes of the network environment.

To overcome the complexity and the restrictions described above, several proprietary solutions have been developed to support automation of network management, like policy-based management [2], [3], [10], middleware-based approaches [13] or automated monitoring using mobile agents [4],[11]. However, human makes the management decisions either - due to predefined policies - or by classical alarm reactions. The NMS assists the human administrator during management decisions. Perhaps, these solutions may decrease the complexity of network management. The announced approaches allow representing complex management tasks through simple, abstracted policies. However, the NMS cannot reach a management decision if unforeseen events occur. Therefore, new systems are required that are able to find several management decisions autonomously. In that case, the correlation between human administrator and NMS changes. Now, the human administrator assists the NMS. Thus, several research activities can be noticed dealing with self-management of networks [5], [6]. Self-management promises new possibilities to minimize the management complexity as well as the operational costs of network management. Furthermore, it increases the flexibility in term of changing customer needs, as new services and applications.

AUTONOMIC RECONFIGURABLE NETWORK

In the following, a self-managed network is referred to as Autonomic Reconfigurable Network (ARN). The ARN monitors the network elements and the network environment itself. Partially it autonomously decides which management processes should be executed to adapt the network elements to the overall network conditions. Environmental changes, as changes in network traffic or changes in QoS requirements of network services, occur due to the mobility of users and provisioning of new network services. Therefore, network elements of an ARN have to be adaptable from Layer 3 up to Layer 8 of ISO/OSI model. Thereby, the term self-reconfiguration can be categorized as a specification of the term self-management. The ARN autonomously discovers and integrates new network elements required to extend the network capabilities or the network size. Beside the integration of new network elements, the ARN is able to exclude unnecessary network elements self-dependent. The ARN also provides functionalities to autonomously administrate and manage network services, e.g. kinds and locations of network services. It supports autonomous error recognition, failure management and failure recovery, e.g. self healing as well as runtime updatability to increase reliability. Automation always goes along with the necessity of extended security architectures.

To achieve the requirements described above, this paper presents a workflow classifying and describing the required tasks for the implementation of an ARN. The workflow includes processes, which consider the integration of new network elements and the reconfiguration of the network elements of an ARN as well.

ARN-WORKFLOW

The workflow is partitioned into three phases, namely *Information Collection, Information Evaluation and Decision-making* and *Reconfiguration.* Figure 1 shows the three phases of the workflow. The phases are subdivided into numerous processes.

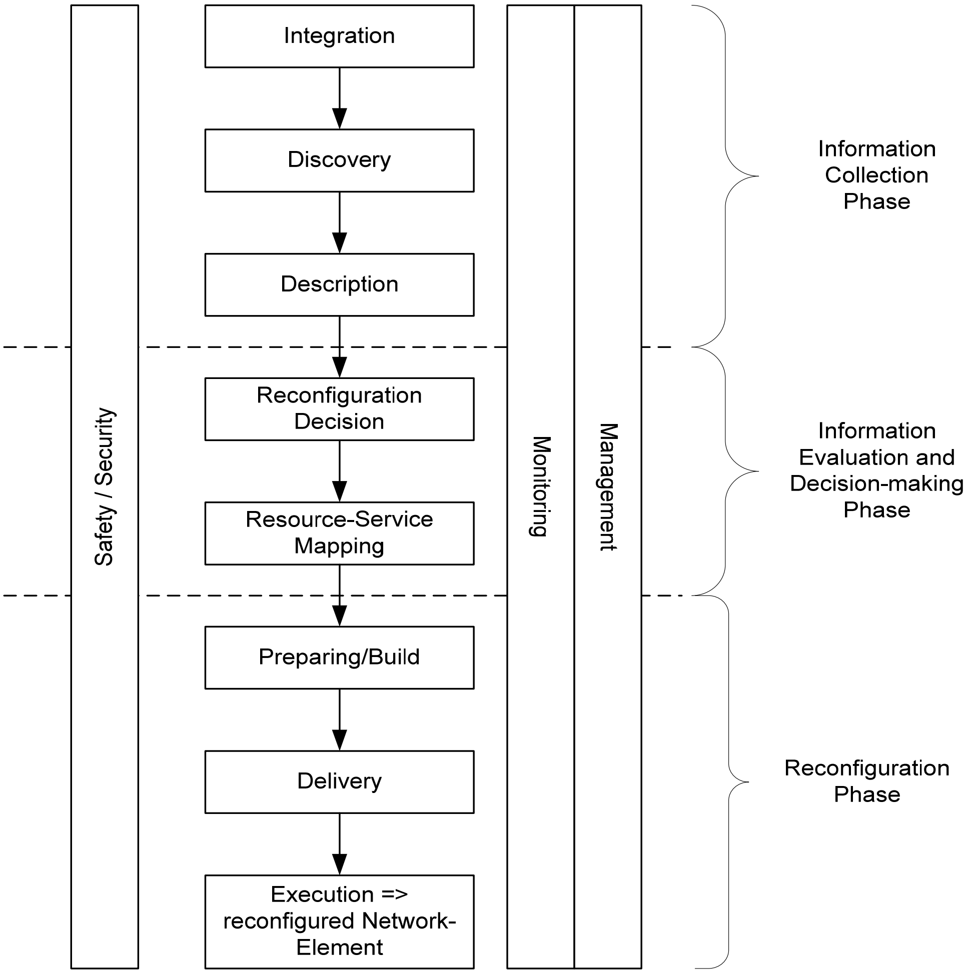

Figure 1: ARN-Workflow

Information Collection

Information Collection includes processes of autonomic integration of network elements as well as integration of network services.

The notion "network services" includes services and auxiliaries, who can be classified to Layer 3 up to Layer 8 of ISO/OSI-Model, like transport services, management services or application services. Network services could be designed modular, containing several independent distributable service functions.

After the integration process, a discovery process follows. The major task of this process is to discover the capabilities of the network elements and accordingly the resource requirements and dependencies of network service functions. For example, resource requirements are minimum required memory size, CPU consumption, QoS parameters, etc. The capabilities of network elements contain performance information, like available memory size, type of CPU or type of hardware architecture. Up to now, several discovery mechanisms have been developed. The mechanisms are based on measurement or estimation procedures like benchmarking. In addition, mechanisms are known using inquiry procedures, e.g. discover capabilities based on requests of information structures, like SNMP-MIP [1].

Services are based on software mostly. Several approaches have been announced to estimate resource requirements of software components with or without explicitly knowledge of the implementation (source code). Solutions, which do not need information about implementation details, propose to use statistical information and similarity comparison for estimation of resource requirements [7]. However, they cannot guarantee the correctness of the estimated values. An estimation of resource requirements of software components even during the design process was already proposed in [8]. The approach requires special methods of software development and software design. These methods allow developing services as a combination of independent and distributable service functions. For every service function, the resource requirements as well as the dependencies to other service functions are described. The description could be integrated into the service function itself, but can also be given in terms of “meta information”. The determined capabilities and service requirements have to be described in a standardized structure, e.g. like a structure which is defined by W3C [9]. For the following phase of Information Evaluation and Decision-making it is important to describe capabilities and service requirements at the same abstraction level.

Information Evaluation and Decision-making

A decision entity uses the collected information to determine whether reconfiguration of the network is required or not. The decision entity has information about available network elements and desired service functions. It could decide to reconfigure one

network element, a group of network elements (subnet) or the whole network to provide the desired service functions.
In all cases, the main task of the decision entity is to determine an optimized mapping of service functions to available network elements. The determination is possible if the discovered service requirements and capabilities of network elements are described on the same abstraction level, by use of the same description language and by means of the same information structure.
The mapping process is partitioned in two steps. At the first step, the decision entity determines which of the available network elements are able to fulfill the resource requirements of the service functions. The result of the first step is a mapping list, which describes what service functions may be executed by which network elements. The second step of the mapping considers additional constraints like QoS requirements or locality restrictions. It leads to a final mapping list, from which the reconfiguration processes can be derived.

Reconfiguration

The third phase of the workflow executes the reconfiguration processes, which are defined and described as a mapping list by the decision entity.
The service functions will be prepared and configured corresponding to specific capabilities of the target execution environment (network element).
The reconfiguration processes have to be sorted chronologically accordingly to their functional dependencies (service dependencies). The result of the sorting is an execution list, which defines what service functions should be distributed to what target network element at which point of time. The accomplishment of the execution list leads to the delivery of service functions to the determined target network elements. Several delivery techniques may be utilized, e.g. client-server-based or agent-based delivery mechanisms. The distributed service functions will be installed and executed at the target network elements. It leads to the reconfiguration of the network elements and at least to the reconfiguration of the network.

Monitoring, Safety/Security

Important tasks during all the phases of reconfiguration processes are monitoring and safety/security. Monitoring comprises known techniques and mechanisms to discover and collect run time information (alarms, errors) and statistical information about environmental parameters, like network traffic or utilization of services. Such techniques could be based on client-server-mechanisms [1], middleware (CORBA, RMI, RPC) as well as agent-based mechanisms. Additionally, techniques have to be developed to monitor special tasks of autonomic reconfiguration, such as monitoring of reconfiguration states of the network and in particular of network elements. The reconfiguration state describes the current phase and process of reconfiguration. Runtime information, statistical information and information about reconfiguration states are required to control and manage an autonomic reconfigurable network.

Safety/security mechanisms ensure the protection of the autonomic reconfigurable network against unauthorized activities, like unauthorized access or spurious reconfiguration of network elements. Specific security requirements arise during new characteristics of autonomic reconfigurable networks. New entities, like the decision entity described above, and new states of network elements, e.g. reconfiguration states, enhance the security requirements. Known security mechanisms have to be extended to accomplish the special security requirements. Furthermore, the Safety/Security task includes techniques for failure recovery. After failure-recognition it uses information collected by the Monitoring task to determine which failure-recovery mechanism has to be activated. Additionally, it provides failure redundancies and failover processes.

DISCUSION

The presented workflow contributes to outline the requirements of the implementation of autonomic reconfigurable wireless access networks. The workflow points out the algorithms and methods, which are required to afford self-management.

In the domains of information collection and representation as well as delivery and execution of decided reconfiguration processes several methods are already known, e.g. client-server-based, middleware-based or agent-based technologies. These techniques have to be analyzed carefully to determine their applicability with respect to the identified requirements. If necessary, the techniques have to be extended or new approaches have to be developed.

However, an increasing demand for research can be observed in the area of software development. Available estimation methods to determine resource requirements of software are too complex and at least not useable yet. New technologies have to be investigated to simplify the discovery of resource requirements of software modules. Simplification could be reached by estimating and describing resource requirements even during the software design.

Moreover, self-management opens new research areas, especially by the phase of Information Evaluation and Decision-making. Nowadays, a definition does not even exist for the term *self-management*. Indeed, different proposed approaches are called „self-managed“. The approaches assist administrators of networks to reach management decisions, but mostly they are not able to achieve decisions autonomously. Therefore, it has to be investigated whether usually management systems can be regarded as assisting or as self-managed. In [12], a first attempt for a definition of the similar term *self organized* is presented. However, due to the lack of definition, it is impossible to classify systems as self-managed or to determine the degree of self-management of a system yet.

In addition, there is no common definition of the degree of self-management. At least, human always want to define the management goal. In order to find an optimized degree of self-management, fundamental investigations of systems of different degrees of autonomy are necessary.

Generally, the described workflow summarizes the processes, which are necessary for the realization of self-management. Therefore, it allows to develop future self-managed heterogeneous wireless access networks more simply. It emphases the research areas, that should be investigated carefully.

CONCLUSION

This paper discusses the advantages and the necessity of self-management of wireless access networks. It explains that automation of management and reconfiguration processes are important to enhance the usability and management of those. The paper classifies phases and processes, which will be required to automate several management and reconfiguration tasks. The classified phases and processes are ordered chronologically into a workflow. The workflow identifies the requirements and the research areas to realize autonomic network management of future wireless access networks. It simplifies the comprehension of the term *self-management*.

Currently, an overall framework is implemented in our Wireless Lab, following the workflow defined in this paper. The framework allows utilizing the represented phases and processes of the workflow. It will lead to new cognitions offering the opportunity to improve the definition of the phases and processes. Furthermore, using the implementation of the framework several techniques for the domains of information collection and representation, decision making as well as delivery and execution of reconfiguration processes will be compared.

References

[1] J. Case, M. Fedor,M. Schoffstall,J. Davin,"Simple Network Management Protocol" (SNMP) - RFC 1157, MIT Laboratory for Computer Science, 1990

[2] Y. Nikolakis, E. Magana, M. Solarski, A. Tan, E. Salamanca, J. Serrat, C. Brou, A. Galis, "A Policy-Based Management Architecture for Flexible Service Deployment in Active Networks", IWAN03, 2004

[3] S. Beyer, K. Mayes, B. Warboys, "Dynamic Loading in an Application Specific Embedded Operating System", International Workshop on SASO+STEPS, 2005

[4] N. Guo, T. Gao, H. Zhao, "Distributed Plug-and-Play Network Management Model Based on Mobile Agents", IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04), 2004

[5] B.R. Haverkort, "Model-Based Self-Configuration for Quality of Service," SELF-STAR: International Workshop on Self-* Properties in Complex Information Systems, 2004

[6] G. Di Marzo Serugendo and M. Deriaz, "Specification-Carrying Code for Self-Managed Systems" SelfMan 2005 IFIP/IEEE International Workshop on Self-Managed Systems & Services, 2005

[7] M. de Jonge, J. Muskens, M. Chaudron, "ScenarioBased Prediction of Run-time Resource consumption in component-based Software Systems" 6th ICSE Workshop on Component-Based Software Engineering: Automated Reasoning and Prediction, 2003

[8] G. Barthe, M. Pavlova, G. Schneider, "Precise analysis of memory consumption using program logics", Proceedings of SEFM'05, pages 86-95. IEEE Press, 2005

[9] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, L. Tran, "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0", W3C Recommendation, 2004

[10] J. Paterson, P. Goldsack, P. Anderson, "Autonomous Reconfiguration with Central Policy Control", Lisa 03 the 17th large installation systems administration conference, 2003

[11] J.L. Marzo, P. Vila, A. Bueno, L. Fabrega, E. Calle, "Automatic self-configuration of the logical network using distributed software agents", IEEE Global Telecommunications Conference GLOBECOM '04, 2004

[12] K. Herrmann, M. Werner, G. Mueh, H.U. Heiß „Ein methodischer Ansatz zur Klassifizierung selbstorganisierender Softwaresysteme", Selbstorganisierende, Adaptive, Kontextsensitive verteilte Systeme (SAKS), 2006

[13] J. P. A. Almeida, M. Wegdam, L. Ferreira Pires, M. van Sinderen, „An approach to dynamic reconfiguration of distributed systems based on object-middleware", 2001

Authors:
Dipl.-Inf. Holger Kärst
Dipl.-Ing. Rene Böringer
Prof. Dr.-Ing. habil. Andreas Mitschele-Thiel
Technical University of Ilmenau, Gustav-Kirchhoff-Str. 1
98693, Ilmenau
Phone: +49 (0)36 77 69 13 38
Fax: +49 (0)36 77 69 12 20
E-mail: {holger.kaerst|rene.boeringer|mitsch}@tu-ilmenau.de