10
Nokia SCE - An Architecture for a Lightweight SCE

Mikko Kolehmainen
Nokia Research Center
P.O.Box 45, FIN-00211 Helsinki, Finland
E-mail: Mikko.Kolehmainen@research.nokia.com

Abstract

This paper describes the architecture of a lightweight Service Creation Environment (SCE) that is developed at Nokia Research Center. The discussion focuses on an example of the future service creation architecture and on solutions that are considered to be worth further research. The aim of this article is to point out those architectural core capabilities that are essential for an effective Intelligent Network (IN) service creation environment. The core features of the architecture include a method to define service logic programs for several Service Control Point (SCP) platforms, a strong support for SIB based service creation approach, a possibility to define service logic programs with textual or graphical editors, a concept of using intermediate format of service definition language as a source format, and finally a possibility to use a network simulator via Core INAP (Intelligent Network Application Protocol) protocol.

1 INTRODUCTION

Nokia has a strategic alliance with Hewlett-Packard (HP) on IN products and platforms. As a result of this alliance Nokia delivers to its customers among other things HP based tools for service creation. The HP originated SCE has proven to be an effective tool for implementing new IN services. It provides capabilities for service creation for both fixed and mobile IN applications. Even though HP SCE is a powerful tool in the hands of a capable software engineer, it demands expertise beyond that of a person with a non-technical background on software engineering. This is mostly due to the relatively high expression power of the Service Logic Execution Language (SLEL) used in HP SCE. Bearing in mind that one of the basic purposes of IN service creation was the ability to produce services in a relatively easy man-

D. Gaïti (ed.), *Intelligent Networks and Intelligence in Networks*

ner, without deep knowledge of SCP programming, it is considered that there is a need for a lightweight service creation tool. Furthermore, since there is always a certain degree of trade-off between the ease of use and the capabilities of the language even regardless of the application area, the problem in the concept of SCE architecture is to determine the correct level of expression power of the service definition language.

Nokia Research Center has initiated a project to produce a prototype of an easy-to-use service creation environment. The architecture of the prototype SCE should be scaleable, fulfill the requirement of producing services to different platforms and conform to the ideas presented in IN recommendations. The architecture under construction has been named NOKIA SCE. The illustration below describes how the NOKIA SCE architecture can strengthen the current service creation approach based on HP SCE. When considering the service creation product line, it is important to realize that an SCE implemented according to the NOKIA SCE architecture is a lightweight SCE that is targeted towards the needs of second or third operators.

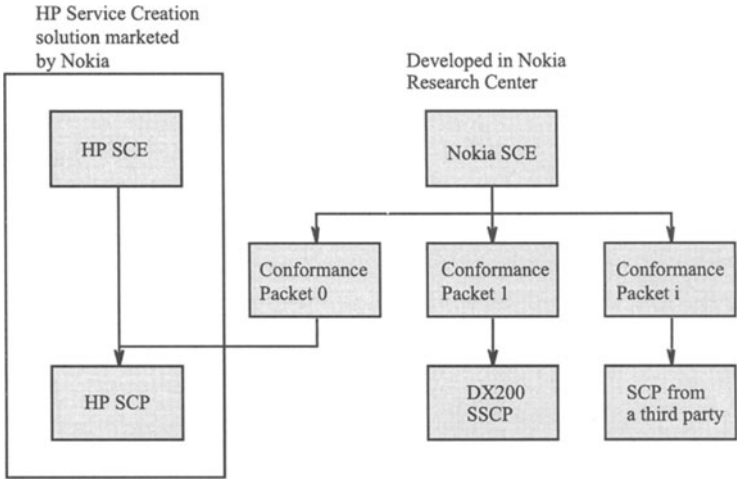

Figure 1 The goal of making NOKIA SCE a multitarget environment is achieved with the aid of conformance packets.

The idea is that services defined with NOKIA SCE could be ported to different SCP platforms. The NOKIA SCE provides the basic set of service creation functionalities that are expected to be present in most of the third party SCP platforms. The platform specific capabilities are added by means of a conformance package; each package is designed for a specific platform according to common guidelines.

2 THE GOALS OF NOKIA SCE ARCHITECTURE

The goals of the NOKIA SCE project can be divided into two classes: internal research work and a basis for future productization of a lightweight SCE. Both of these classes of goals are of equal importance and must be met.

In respect to research goals there are two issues that should be mentioned. First, there is a need for a tool that can be used for rapid service creation for Nokia's internal network simulation tool. Second, an SCE is needed for service logic definition for intelligent broadband network demonstrators.

The goal of initiating a product is to design and implement a prototype of an SCE that can be used as a starting point for a commercial lightweight SCE. That SCE will extend Nokia's IN service creation product line from the lower end. In the figure 2. there is an illustration that shows how the objective classes relate to each other and to the NOKIA SCE project.

The mission of Nokia Research Center is to concentrate on research work and to find out new solutions and technologies to support operation of Nokia business units which will make the ultimate decision on productization of individual architectures and concepts. Thus the design process of the NOKIA SCE architecture has been concentrating mostly on the research issues and ideas for features that extend the usability of the NOKIA SCE concept. Therefore the act of productization itself may appear somewhat distinct from the approach and decisions taken during initial design and prototyping phase.

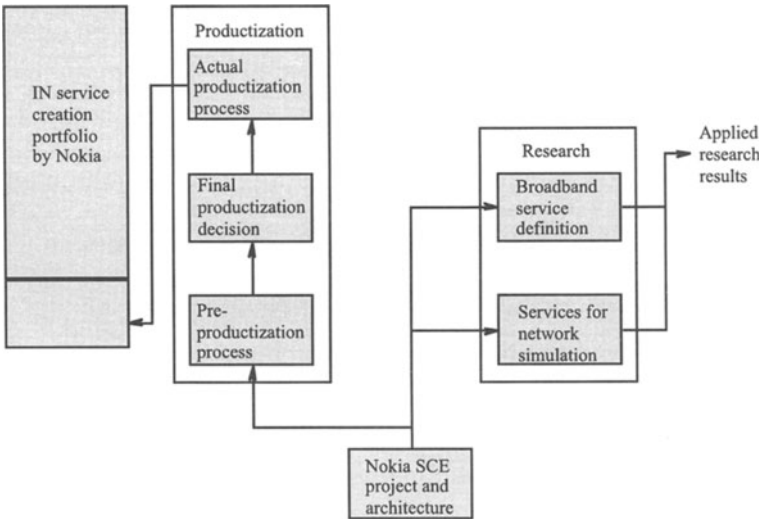

Figure 2 Objective classes of NOKIA SCE project and architecture design process.

3 THE OVERALL NOKIA SCE ARCHITECTURE

3.1 The set of NOKIA SCE units

The NOKIA SCE architecture consists of four basic elements, named *NOKIA SCE units*. The most visible NOKIA SCE unit is the graphical *NOKIA SCE Service Editor*, which acts as the main user interface towards the service designer. The purpose of the *NOKIA SCE Core* is to provide the functionalities related to service animation, validation, and Service Logic Program (SLP) database access for editing needed by the graphical editor. NOKIA SCE Core is also responsible for providing the tools required for processing the NOKIA SCE service logic language. The name of the language is *NoLo - **No**kia **Lo**gics*. NoLo is a scripting language that consists of an extendible set of SIB calls. Thus one SIB is considered to be one NoLo command. Besides tasks listed above, NOKIA SCE Core takes care of the communication with Inesim tool (*Inesim - **I**ntelligent **Ne**twork **sim**ulation*) for SLP testing and demonstration purposes.

The third NOKIA SCE unit is the *NOKIA SCE mini-SMS* (Service Management System) that is used for service animation management and service data management. The fourth unit is one of the distinctive features of the NOKIA SCE concept: *NOKIA SCE conformance unit.* The unit consists of conformance packages that ensure the portability of SLPs created with NOKIA SCE. In the figure below there is an illustration that depicts the overall NOKIA SCE architecture.

In the following sections there is a short description of each NOKIA SCE unit. From end user's point of view the division into NOKIA SCE units is not visible. Also, the mapping between NOKIA SCE units and host operating system processes is not direct; in a workstation one process may be responsible for more than one NOKIA SCE unit.

3.2 The NOKIA SCE Service Editor

The NOKIA SCE Service Editor is used as a tool for service definition and as a user interface for service animation and testing. In the current prototype the main functionality of the NOKIA SCE Service Editor is to provide adequate tools for service creation. Therefore the set of tasks assigned to NOKIA SCE Service Editor is as follows:

- To provide the basic SLP graph editing functionalities.
- To ensure the linguistic equivalence between textual and graphical NoLo representation.
- To provide a dynamically configurable set of SIB icons and Service Support Data (SSD) definition routines.
- To provide a graphic user interface for service animation, testing and debugging.

The set of interfaces to the NOKIA SCE Service Editor bases on the tasks specified above. In the figure 4. there is an illustration that describes those interfaces.

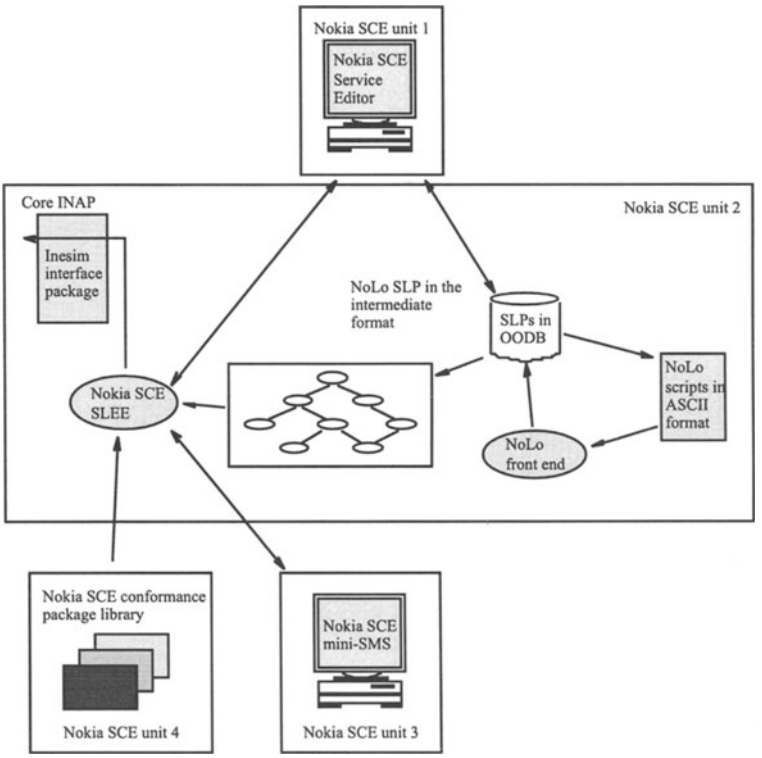

Figure 3 Overall NOKIA SCE architecture.

As can be seen, the editor has three interfaces to other parties of NOKIA SCE: one towards NOKIA SCE SLEE via socket based communication protocol, one towards NOKIA SCE Core database functionalities including NoLo interfaces and one is a computer-human interface towards the service designer. The sockets are needed in the communication between the NOKIA SCE Service Editor and NOKIA SCE SLEE since they will be executed in different processes. However, even though the editor and NoLo SLP database functionalities belong to different NOKIA SCE units they will be mapped to the same process in implementation.

3.3 NOKIA SCE Core

NOKIA SCE Core is the center of NOKIA SCE architecture. The core consists of four modules that are logically and functionally joined together. Together these modules provide the capabilities that are needed for service animation and testing. The modules are as follows:

- NOKIA SCE Core database (both runtime database and the one used for SLP storage).

- NoLo compiler front end and back end for ASCII NoLo scripts.
- NOKIA SCE SLEE with SIB implementations.
- Inesim interface package.

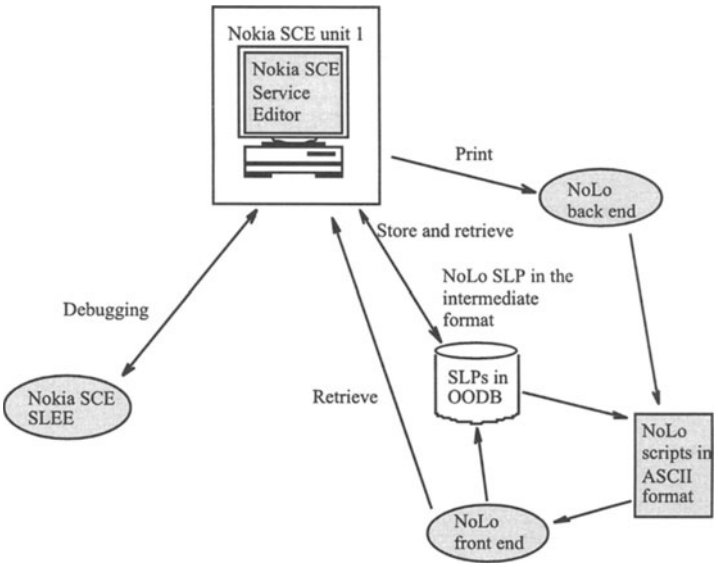

Figure 4 The interfaces related to NOKIA SCE Service Editor.

Both of the NOKIA SCE Core databases are modelled as object oriented databases. The reason for choosing OODB is twofold: on the one hand object modelling technique makes it possible to store tree structured objects, like service logic programs, into a database in an effective manner. On the other hand object database is a promising approach for implementing run-time datastore for service logic programs at least in the SCE itself. Of course, if runt-time database is object oriented in the SCE, and e.g. relational in the target SCP platform, a separate presentation transformation layer is needed in the database access components.

One feature of the NOKIA SCE architecture is the possibility to define services with both textual and graphical editor. In order to provide two kinds of service definition methods, two front ends of the NoLo-compiler are needed. Both front ends produce identical intermediate format of NoLo language. This intermediate language is then stored in the NOKIA SCE Core SLP storage database. There are tools for producing both textual NoLo scripts and graphical representations of intermediate format representations that are stored in the database. NoLo front end for graphical editor is considered to be conceptually a part of the NOKIA SCE Service Editor.

The execution of the service logic program in NOKIA SCE is the responsibility of NOKIA SCE SLEE interpreter. The commands of the NoLo language have been implemented in the SLEE so that there is an one-to-one mapping between NoLo commands and SIB implementations. The set of SIBs designed for NOKIA SCE architecture is flexible and extendible.

From software engineering point of view the structure of NOKIA SCE SLEE bases on parallelism achieved with usage of threads. This means that each SLP is executed in a thread of its own. This is beneficial in cases when NOKIA SCE SLEE runs on multi-processor computers.

NOKIA SCE SLEE uses as input the tree-like intermediate format of NoLo scripts. Thus there is no linear binary representation of NoLo scripts. The advantage of using a memory resident structure as an input format of an interpreter is the direct mapping between the graphical and the input representations. Thus it is possible to implement an easy debugging facility between the SLEE and the Service Editor. Furthermore, since the information is a graph structure that can be traversed, the interpreter is possible to be implemented in quite a straight-forward manner.

Inesim interface package provides a communication means towards the Inesim tool. The idea is that communication takes place via Core INAP protocol stack thus allowing the usage of different SSP simulators instead of Inesim tool. The interface between NOKIA SCE SLEE and Inesim tool can be implemented in a simplified manner; part of message parameters can be obtained from the default information base that is configuration specific.

3.4 NOKIA SCE Conformance package library

Even though IN standards for CS1 do not specify architecture of SCE they contain an idea of portable services. One of the goals of the NOKIA SCE concept is to provide a possibility to produce services that can be reused in several SCP platforms; for example the platforms may be delivered by Nokia, HP or some other supplier. The key portability tool in NOKIA SCE is the concept of conformance package (see figure 5).

The conformance package is divided into two separate logical parts. The first part is responsible for ensuring that the service is able to execute on the target platform. Usually this means that there exists a SLEE that can execute or interpret platform's native programming language. This implementation should also include the suitable set of SIBs implemented e.g. as macros or subroutines depending on the capabilities of the implementation language. The SIB library implementation is an additional component to the target platform's SLEE.

The second part of a conformance package is dedicated to service testing. The idea is that conformance package supplements NOKIA SCE SLEE with capabilities that are platform specific. This is due to the fact that from the functional point of view NOKIA SCE SLEE is the least common denominator of the functionality set needed for implementation of certain set of benchmark services. The terms "MIB" (Management Information Base) and "External channel" in the figure 5. refer to HP SCP specific capabilities that may not be present in other platforms.

It must be noted that there are service classes that cannot be reasonably ported to another platform. Most of such services perform operations that are not sensible in other environments.

For example certain SCP specific management related service logic programs might fall into this class. However, porting of the most common services can be considered worth the trouble. Furthermore, since service management issues are solved with very heterogeneous solutions in different platforms, they must be considered case by case anyway.

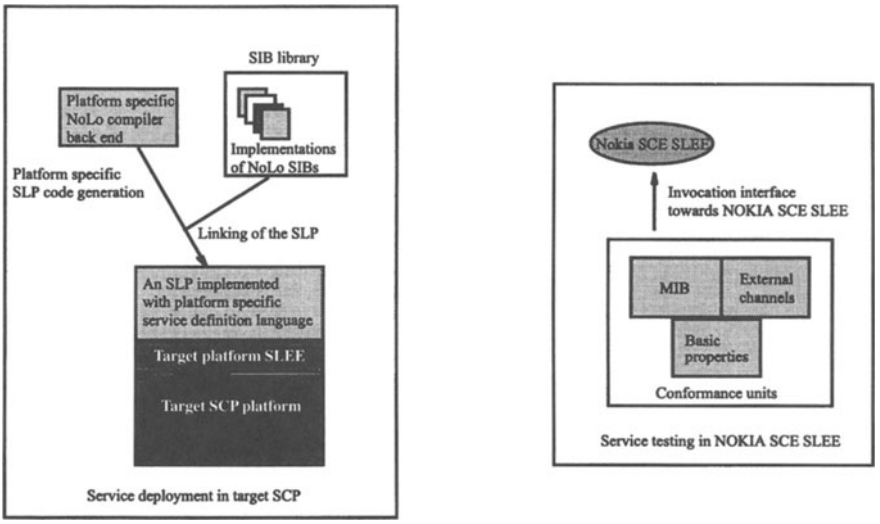

Figure 5 An example of a conformance package.

3.5 NOKIA SCE mini-SMS

According to current ideas about NOKIA SCE architecture, mini-SMS is not very interesting subject within the SCE architecture. The duties of mini-SMS are quite straight-forward: loading and stopping of SLPs, SLEE execution and network simulator management and data management.

4 THE NoLo-SERVICE DEFINITION LANGUAGE

NoLo-language is a simple scripting language that is not dependent of the application-area. This means that the structure of the language does not take any attitude to the question whether the language is intended for a certain application area or not. The functionality and the expression power of the language is based on the set of commands of which the language itself consist.

NoLo-commands are called SIB invocations. The set of SIBs is extendible. However, the extensions must be made in three locations: editor configuration file, NOKIA SCE SLEE and the corresponding conformance packages. The idea is that there exists a basic set of SIBs that

can be used when creating feature entities by combining SIBs into macros. From these feature entities it is relatively straight-forward to create services. If the basic set of SIBs appears to be inadequate, new SIBs can be designed and implemented.

As said earlier, there are two possible methods to create NoLo-scripts: the graphical and textual. Both of these ways are equivalent in a sense that scripts created with one tool can be processed with the other tool. Thus the user has the benefits of both approaches without any trade-offs.

Due to the fact that NoLo-scripts are interpreted and stored in the intermediate format it is possible to produce Flexible Service Logic (FSL) scripts as defined in Bellcore generic requirements (GR-1280). In NOKIA SCE SLEE this is implemented plainly just adding the FSL script in the interpreted intermediate source format when the FSL invocation has been met. In the SCP platform level, the idea is quite similar: both NoLo scripts and FSL scripts are compiled to the native language presentation. From the structural point of view FSL scripts are not different from plain NoLo scripts, except for the routines needed for data definition.

5 DATABASE ASPECTS

When considering creation of portable IN services and database aspects, there are two things that must be taken care of: how run-time database operations can be designed to be portable, and what type of database should be used. These two issues set certain boundaries to the concept of portability.

In order to achieve the possibility to port service programs from a platform to another the basic set of database operations (Retrieve, Store, Delete, Insert) ought to be designed in such a manner that it would be relatively easy to change the database system that is used as a basis for database routine implementation. The figure below illustrates the idea of the design of routines.

According to the NOKIA SCE concept there can be unlimited number of basic SIBs and thus unlimited number of SIBs that access the database. Within each one of the database related SIBs there is an inner interface towards the common database access routines. In that common interface there is one thing that has to be decided: whether the database used is relational or object-oriented. This decision is imperative and must be done before compilation of the SIB implementations.

Below the common interface there is a presentation transformation layer that takes care of the actual database access. If database is object oriented, the access takes place via ODMG compatible interface. If it is relational, a data presentation transformation is performed, if needed. After that, database specific access routines are invoked.

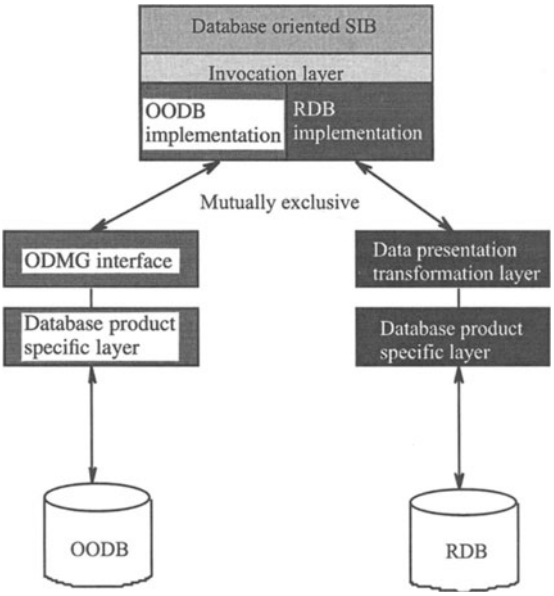

Figure 6 Database routine layers in a database oriented SIB.

6 CURRENT STATUS

A project that is heading towards implementing a lightweight SCE according to the NOKIA SCE architecture has started at Nokia Research Center. The project's aim is to achieve the specified research goals and both to evaluate the portability issues and to implement an exemplary conformance package. When the first phase of the project will finish, the second phase which concentrates on productization and portability will start. Precondition for the second phase initiation are the results received from the work done that far; if NOKIA SCE architecture will live to the expectations that have been set to it, the continuation of NOKIA SCE will take place.

7 SUMMARY

NOKIA SCE architecture is a concept of a lightweight SCE that is intended to supplement Nokia's IN product line as an easy-to-use service creation tool from the lower end of SCE product family. The main issue within the NOKIA SCE architecture is the possibility to de-

fine portable services in a relatively rapid manner either graphically or as a textual script. The NOKIA SCE approach is based on SIB philosophy as expressed in IN standards.

8 REFERENCES

Cattell, R.G. G.; ed.; "The Object Database Standard: ODMG -93"; Morgan Kaufmann Publishers, San Mateo, California, USA; 1994.

ETS 300 374-1; "Intelligent Network Capability Set 1 (CS1) Core Intelligent Network Application Protocol (INAP) Part 1: Protocol Specification"; ETSI; 1993.

Genette, M.; "The Service Creation Environment for Intelligent Networks"; Alcatel Telecommunications Review, 1st Quarter 1996; Alcatel Alsthom Publications; France; 1996.

GR-NWT-001280; "Advanced Intelligent Network (AIN) Service Control Point (SCP) Generic Requirements"; Issue 1; August 1993; NJ, USA; 1993.

Hino, K.; Tani, H.; Nishida, T.; "Object-oriented Service Creation Environment for Multimedia Communication Networks"; 3rd International Conference on Intelligence in Networks; ICIN; Bordeaux, France; 1994.

International Telecommunication Union, Standardization Sector; Q.1200 Series; ITU-T, Study Group XI; 1993.

Knight, C.; "Service creation from IN to mobile and broadband"; Intelligent Networks; Harju, J.; Karttunen, T.; Martikainen O.; ed.; published on behalf of IFIP; Chapmann & Hall, London; 1995.

Ricken, R.; Sevcik, M.; Eder, R.; "IN Service Creation - How to Make the Service Independent Building Block Concept Fly"; XV International Switching Symposium; Berlin, Germany; 1995.

Slutsman, L.; Lu, H.; Kaplan, M, P.; Faynberg, I.; "The Application-Oriented Parsing Language (AOPL) as a way to Achieve Platform-Independent Service Creation Environment"; 3rd International Conference on Intelligence in Networks; ICIN; Bordeaux, France; 1994.

9 LIST OF ACRONYMS

CS1	—	Capability Set 1
FSL	—	Flexible Service Logic
IN	—	Intelligent Network
INAP	—	Intelligent Network Application Part
Inesim	—	**I**ntelligent **N**etwork **Sim**ulation Tool
MIB	—	Management Information Base
NoLo	—	**No**kia **Lo**gics
SCE	—	Service Creation Environment
SCP	—	Service Control Point

SIB	—	Service Independent building Block
SLEL	—	Service Logic Execution Language
SLP	—	Service Logic Program
SMS	—	Service Management System
SSD	—	Service Support Data
SSP	—	Service Switching Point
ODMG	—	Object Database Management Group
OODB	—	Object Oriented DataBase
RDB	—	Relational DataBase

10 BIOGRAPHY

Mikko Kolehmainen was born in Raahe, Finland, on August 6, 1966. He received, in 1992, the M.Sc. degree in computer science from the University of Helsinki, Finland. After graduation he joined Nokia Research Center in 1993 and is currently working as a project manager in Communication Systems Laboratory. He has specialised in issues related to service creation in the area of Intelligent Networks. Currently he is also a graduate student in the University of Helsinki.