Author:	Pasi Nummisalo
Document:	IN
Date:	1/18/99
Version:	1.0
History:	

IN Service Creation and Execution

TOVE IN (Intelligent Network) service creation and execution environment is build on the DCF (Distributed Component Framework, see document DCF). The goal was to provide a simple and extendable prototyping environment for IN services.

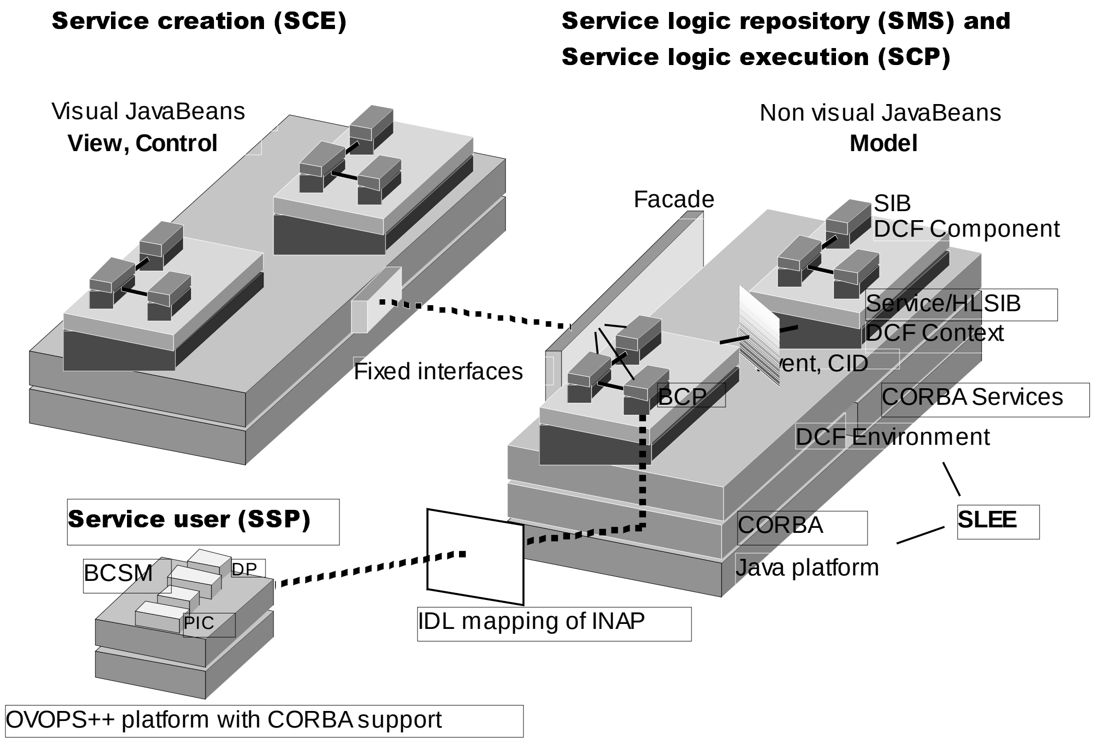
1 Introduction

IN services can be created from SIBs (Service Independent building Block). SIBs are seen in this implementation as DCF components. With this approach services can be created from user defined components in a visual editor. Ready made services can be copied to execution environment which is basically same as the service repository/testing environment, but doesn't have to support GUI connection functionality. The TOVE IN environment consists of a service repository node and possible several service execution nodes. The repository node with the GUI connected includes similar functionality as SCEF (Service Creation Environment Function) and SMF (Service Management Function).

2 Architecture

Pictures 1 and 5 show overall architecture of TOVE IN environment. The service execution environment uses CORBA 2.0 compliant ORB (Orbacus) to communicate with SSP (Service Switching Point). The IDL interface between these two points is based on an IDL mapping of an INAP (Intelligent Network Protocol). The mapping is defined in OMG's document *Interworking Between CORBA and TC Systems* (telecom/97-12-06). A method call (INAP operation) originating from SSP is translated to a generic event, which is filled with CID (Call Instance Data). Service components are organized as a tree hierarchy (see appendix 1, picture 3). Services are at top level. Services contain triggers and service subscribers. Service subscriber contains the individual service logic. Service logic includes BCP (Basic Call Processing) components (in and Out) and required SIBs. SIBs are connected to form the individual service logic. SIB properties (SSD, Service Support Data) can be altered to customize the logic of one SIB.

As said services and service subscribers are stored in repository node from where they can be deployed to execution node. When the service and its subscribers are installed also required BCSM (Basic Call State Model) triggers are activated at SSP.



Picture 1. IN service architecture.

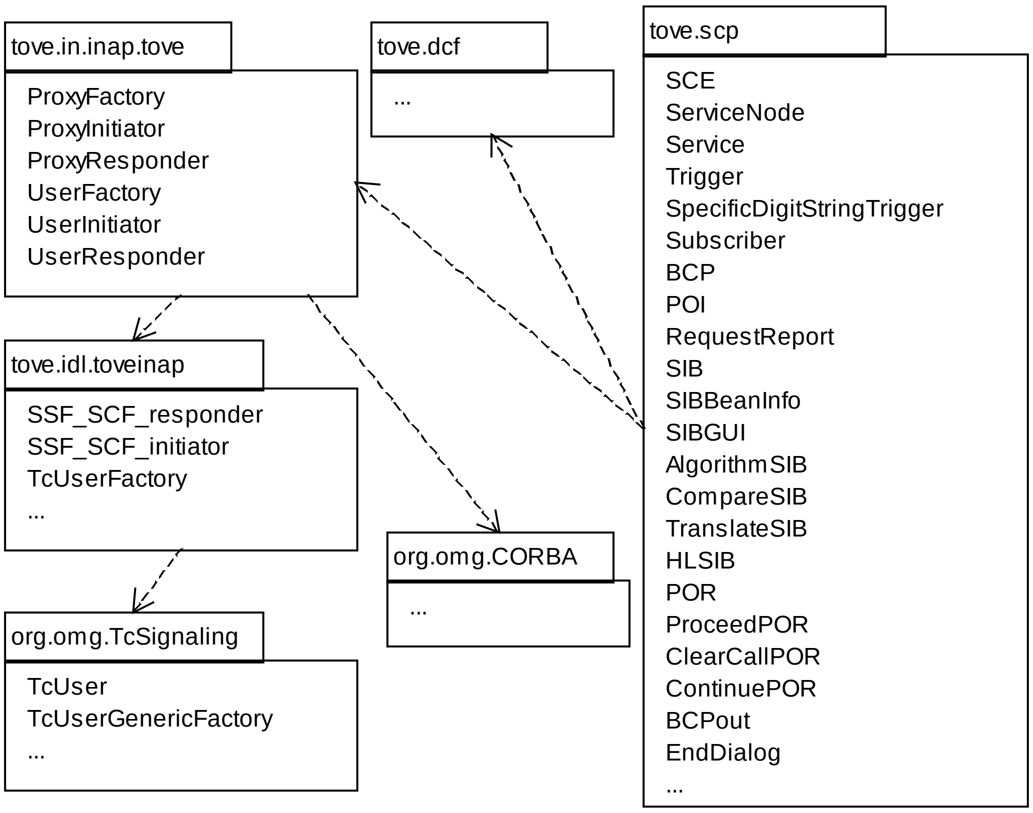
3 Implementation details

The implementation includes following visual components (figure 2): service, subscriber, trigger, BCP in with 9 POIs (Point Of Initiation), request report, translate SIB, algorithm SIB, compare SIB, end of dialog, HLSIB (Higher Level SIB), connection point, and BCP out with 3 PORs (Point Of Return). Service component presents certain service category e.g. free phone (0800 xxxx) service category. The service includes trigger components that are used when the service is deployed. Corresponding BCSM triggers are activated with parameters from these trigger components. Service is identified by service key property. Service subscribers live in the service context. Subscribers present companies how have subscribed certain service. Subscribers are identified by their number e.g. 0800 007 where 007 is the subscriber number and 0800 service category number.

Sessions are created from UserFactory object, which is bound to CORBA name service. When the first message (initialDP) arrives from the SSP a session (UserResponder object) is marked active. The session is in an active state as long as the SSP or the service logic ends the session (end assocation message or end dialog component).

3.1 Packages

Package scp includes SIBs and other component classes. Package in/inap/tove includes INAP classes. Package idl/toveinap is generated from toveinap IDL file.



Picture 2. Package dependencies.

4 Features implemented

INAP interface, trigger activation, service, subscriber, BCP with POIs and PORs, translate SIB, algorithm SIB, compare SIB, end of dialog, HLSIB (Higher Level SIB), request report.

5 Future development

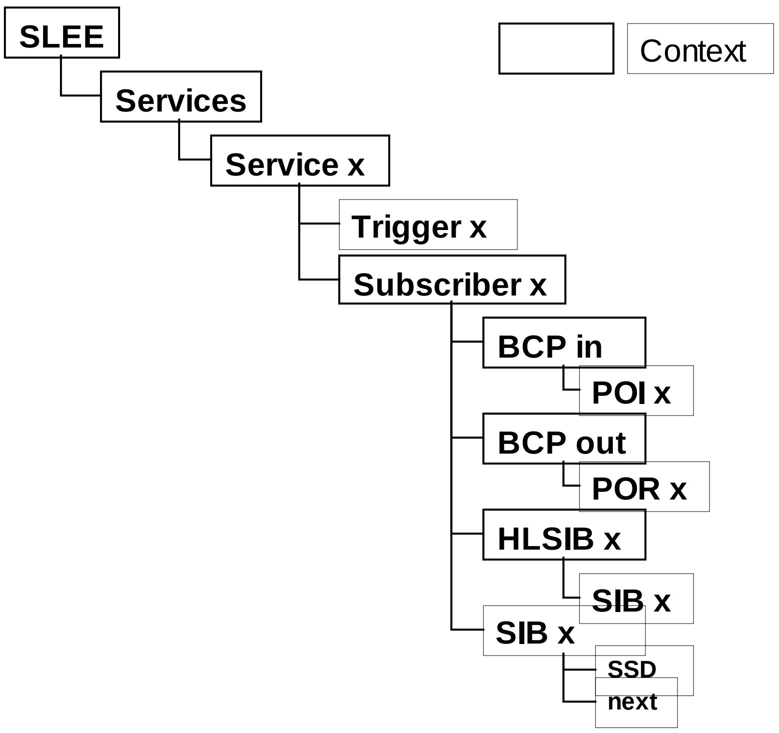
This solution doesn't include database access. It can be done with JDBC (Java Database Connectivity) API or with some another technique.

6 References

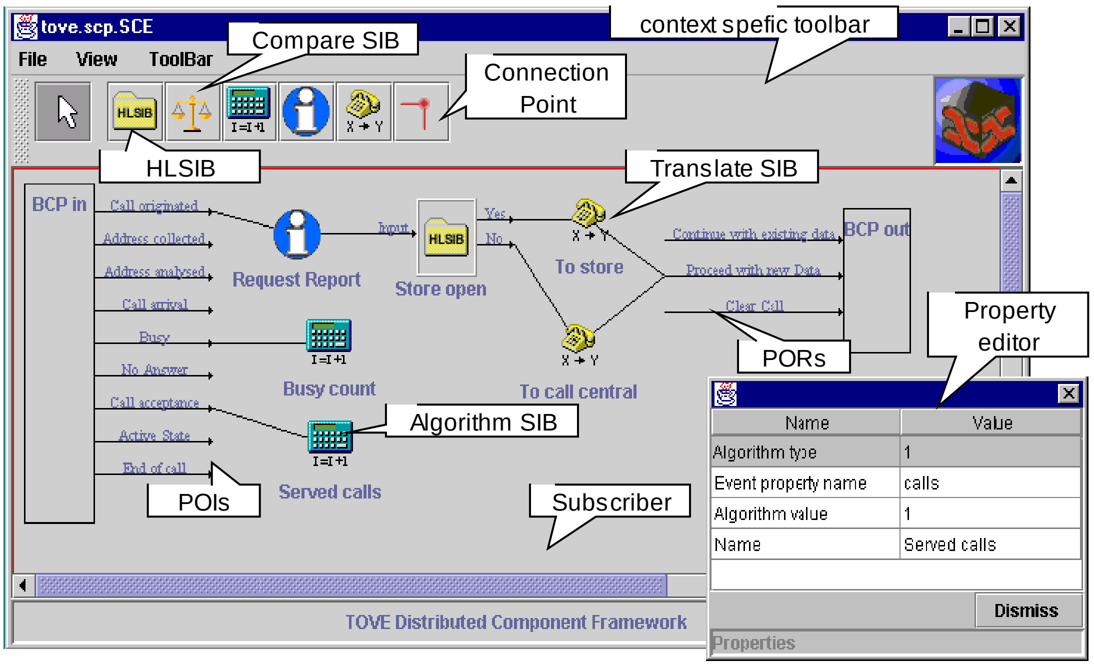
1. ITU-T: Q.1223, Q.1213

2. TOVE: DCF documentation

APPENDIX 1



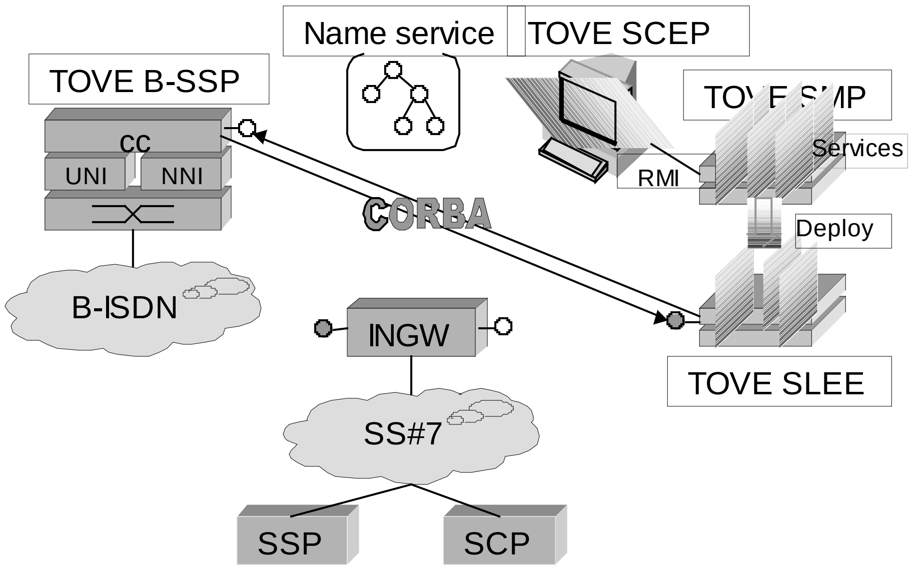
Picture 3. Component hierarchy.



Name	Value
Algorithm type	1
Event property name	calls
Algorithm value	1
Name	Served calls

Picture 4. Example service.

APPENDIX 2



Picture 5. Overall architecture.