

Intelligent networks for personal communications

*Hans Bisseling, Ericsson Telecommunicatie BV
etmjobi@crosby.ericsson.se ,
Jos den Hartog, Ericsson Telecommunicatie BV
etm.etmjdha@memo.ericsson.se ,
Ericsson Telecommunicatie BV
IN Application Laboratory
PO Box 8, 5120 AA Rijen, The Netherlands
Tel: +31 1612 29911, Fax: +31 1612 29699*

Abstract

Service mobility and the personalisation of services are main concerns of a modern telecommunication environment. In addition, co-operation among services and a simple presentation form to the end-user are of extraordinary importance as well. To fulfil these requirements, a user defined environment called Personal Services Communication Space (PSCS) is under development by the RACE II project Mobilise (R2003). This concept has to take into account all involved players and their specific requirements. The main features of PSCS are: personal mobility based on UPT, personalisation to have personal working environments for end-users, and interoperability to have effective interworking between different services offered on heterogeneous networks. The PSCS Conceptual Framework is primarily based on the Intelligent Network Conceptual Model (INCM) with extensions taken from Open Distributed Processing (ODP).

During the development process of PSCS, difficulties were encountered in composing PSCS services and services features in an implementation independent way by Service Independent Building Blocks (SIBs) of Capability Set 1 (CS-1). As a result important enhancements are proposed:

The introduction of recursion and parallelism into the concept of SIBs to allow IN service engineering to be more object-oriented.

To break down service features into a chain of SIBs that are executed sequentially is difficult, and will become much more difficult for future complex features. To handle these difficulties more effectively, recursion is introduced which enables grouping of SIBs into bigger SIBs, i.e.,

SIBs can be nested. In addition, SIBs are allowed to trigger new service logic which can be executed in parallel.

The introduction of domains in the GFP.

The IN network being regarded as a single entity neglects the different stakeholders involved in IN services. Different network operators and service providers are already visible in for instance mobility services such as UPT: originating, terminating, and home domains can be related to different stakeholders. Services may also cover more than one domain in a sense that a management service is partly implemented in an IN network, and partly in a TMN network.

Even end-users and subscribers will have their own domain in the future. This idea is reflected by the notion of the PSCS Flexible Service Profile (FSP), in which end-users have their own personalised service logic.

1. INTRODUCTION

The RACE II project Mobilise (R2003) is a four year project (1992 - 1995) and its objective is to define a concept for personal communication [Mobilise D4]. Service mobility and the personalisation of service conditions are one of the main concerns of a modern telecommunication environment. In addition, smooth co-operation among diverse services and a simple presentation form to the end-user are of extraordinary importance as well. To fulfil these main requirements in new telecommunication systems, a user defined environment called Personal Services Communication Space (PSCS), is envisaged to be developed. The main features of the PSCS are:

Personal mobility

Personal mobility means that an end-user can use any network access point and any terminal while being identified through the same number (identity) and charged to the end-user's personal account. PSCS is considered to be an extension of UPT [ETSI NA7], which offers a personal mobility service.

Personalisation

The PSCS concept for personalisation is that end-users have personal working environments, that can easily be managed by subscribers and configured by end-users. Subscribers can control the service delivery to their end-users, and define limitations on the service usage, depending on the situation the end-user is in at a certain moment. End-users are then allowed to configure their personal environments within these limits.

Interoperability

Interoperability describes the capability of the system to support effective interworking between different services, offered on heterogeneous networks.

The PSCS Conceptual Framework [Mobilise D4] is primarily based on the Intelligent Network Conceptual Model (INCM) [ITU-T Q.1200] with extensions taken from Open

Distributed Processing (ODP) [ISO ODP]. This paper discusses enhancements related to the IN's Global Functional Plane (GFP). [ITU-T Q.1203]

2. ACTORS AND ROLES IN PERSONAL COMMUNICATIONS

The PSCS conceptual design starts at an *enterprise modelling* stage which has similarity to ODP's enterprise view. Enterprise modelling is needed for PSCS in order to put the service features into the right context, to identify on-line and off-line (contractual) relationships and to relate the services to the domain interfaces.

A key feature of Mobilise is to approach the PSCS from a user's perspective rather than from a technological viewpoint, to ensure that final results will be readily acceptable. However, it is recognised that any set of requirements must consider technological constraints. While the conceptual work on PSCS is user-driven, consideration is also given to how it can be implemented.

Requirements are placed on a framework for personal communication from all the different stakeholders involved in the deployment and operation of PSCS: End-User, Subscriber, PSCS Service Provider, Application Service Provider, Network Provider and Access Provider. These PSCS stakeholders are characterised by their roles and the mutual relationships between them, see Figure 1.

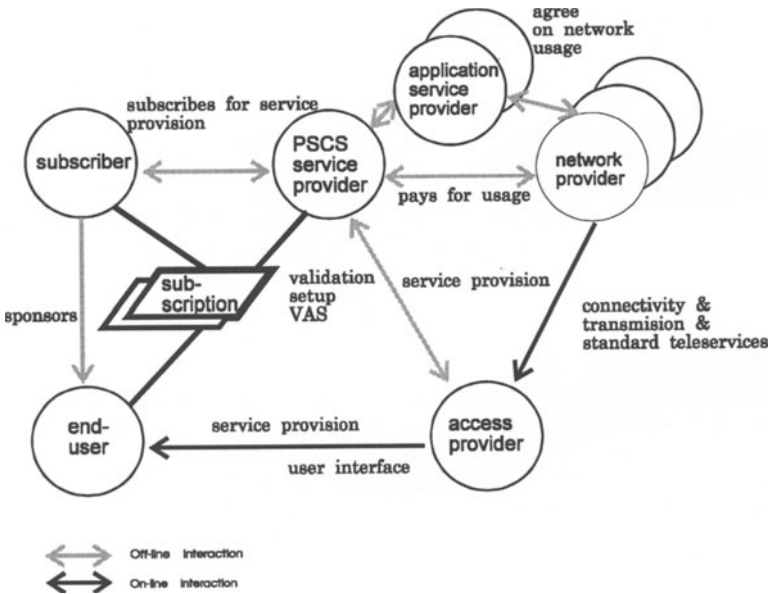


Figure 1. The PSCS Enterprise Model.

It is important to understand that this model relates entities in their business roles and not corporations. The same corporation (or parts thereof) might act in different roles in this model, simultaneously or at different times. This distinction is often referred to as a distinction

between *actors and roles*, whereby a certain role can be played by one or several actors; and whereby a certain actor might play one or several roles simultaneously. Therefore, Figure 1 shows the mutual relationships between the different roles that are relevant for PSCS.

Every entity in this model is linked via a contractual relationship to each other. A central role plays the PSCS subscription which links the PSCS service provider with both the subscriber and the end-user. Some of the relationships materialise during actual service use into physical interconnections. The increasing demand for flexibility pushes increasing parts of the relationships into the on-line state. Therefore business relationships for service provisioning have to be taken into account carefully in service design.

End-Users define their personal service environment using different roles and services, adapted to their personal needs. To be allowed to use services, the end-users have to agree with the subscriber on the particular rights which will be associated with them. These are given in a form of a *PSCS subscription*.

The *Subscriber* subscribes to and pays for services offered by the service provider, or offered by different service providers. The subscriber is generally an organisation with an interest in maintaining a balance between quality and economy of service on behalf of the real end-users of the services, e.g. its employees.

It is possible that the end-user and the subscriber are the same person; this would characterise most of the current residential market for PSCS. There the need has been identified to relate the cost of communication more clearly to the usage of the particular person as a family in fact is a small organisation.

On the other hand, the end-user could be an employee of the subscriber, which would be a corporate organisation. The middle ground is characterised by a many-to-many relationship between end-user and subscriber, such as a consultant working for several clients, possibly even within the same corporation. This relationship is modelled by the provision of various subscriptions linked to the same end-user.

The *Network Provider* provides the infrastructure for the interconnection of several sites with specified basic delivery conditions. He is providing standardised services, so-called 'network services', such as basic telephony and data communication.

The *Service Provider* is generally an organisation that commercially manages services offered to subscribers. He is responsible for agreements with network providers on network usage, and with access providers on usage of access facilities.

In PSCS, two classes of service providers are identified:

- The *PSCS service provider* offering a PSCS framework service which is a platform for other services enhancing and unifying network services.
- The *application service provider* offering one or many (tele-) communication services on top of the PSCS service platform, e.g. personalised information services, message service, directory service, etc.

The *Access Providers* ensures that the end-users can get access, through the provision of appropriate terminals or access networks, to the public network in order to be able to use the services.

3. ENHANCED SERVICE ENGINEERING WITH INTELLIGENT NETWORKS

To break down IN services and service features into a set of service components is difficult to do; therefore, the concepts to identify IN service components need to be powerful to ease the process of *service engineering*. Especially for PSCS type of services this service engineering is very difficult when it is based on Global Functional Plane (GFP) concepts for Capability Set 1 (CS-1) [ITU-T Q.1213]. Regarding CS-1 the following problems were identified:

1. The CS1 SIB definition is not recursive and CS1 SIBs are too low-level.

The (de-)composition process to implement services and service features based on CS1 SIBs is difficult due to the one step mapping of services features to SIBs. It is not possible to decompose services and their features into high level functions or processes and refine these later on.

2. The execution of a chain of SIBs is only sequential, whereas parallel execution is needed as well.

At a very low level one can regard the execution of a service as a sequence of consecutive actions, but on higher levels there is a need to distinguish from these fine-grained sequential activities and rather talk about parallel interworking activities or processes.

3. The introduction of non-call related service execution implies the need for a well defined process definition.

In CS1, IN services are only call related in which the BCP triggers GSL. In CS2 non-call related service execution is introduced for mobility services, for instance for location updating. To be able to model this in the GFP a well defined process definition is needed.

4. No domains exists in the GFP: the IN network is regarded as a single entity.

The IN network being regarded as a single entity neglects the different stakeholders involved in IN services. Different network operators and service providers are already visible in for instance the DFP regarding mobility services such as UPT: originating, terminating, and home domains can be related to different stakeholders. And even end-users and subscribers will have their own domain in future, in which they have their own unique service logic which is adapted to individual needs.

To be able to map effectively IN service features onto the Global Functional Plane, composition and decomposition techniques, sequential execution but also parallelism are necessary. To be able to perform composition (bottom-up) and decomposition (top-down), high level SIBs are introduced. And service processes are introduced to allow for parallel execution. But within each service process, the service logic is executed sequentially. To have a clear understanding of these concepts, the notions of *service processes*, *high level SIBs (HLSIBs)* and of *SIBs* were identified. SIBs are considered as the smallest service component, they are not further refined in the GFP. HLSIBs, however, are SIBs which are composed out of other (HL)SIBs. And service processes, that can be executed in parallel, encapsulate a chain

of (HL)SIBs that are executed sequentially. Furthermore, by using service processes, HLSIBs and SIBs for GFP modelling gives also good opportunities to further decompose/refine SIBs and service processes at a later stage.

These solutions identified by Mobilise have resulted in concrete proposals to ITU-T SG11 (Melbourne, 1 -10 March 1994) and ETSI NA6 (Vienna, 25-29 April 1994), who have adopted these concepts to enhance the GFP concepts for CS-2 [ITU-T Q.1203, ITU-T Q.1290].

3.1. Modularity

One of the basics of object orientation is modularity [Meyer, OO]. Whereby the level of modularity depends on criteria such as modular decomposability, composability, understandability, continuity and protection. In addition, principles such as explicit interfaces and information hiding are to be observed to ensure proper modularity. In IN the basic notion of modularity is the Service Independent Building Block (SIB). This section shows that this modularity can be applied recursively by introducing the notion of High Level SIB (HLSIB).

HLSIBs, that are as normal SIBs executed sequentially, support abstraction mechanisms as composition and decomposition. By composition, SIBs can be defined out of smaller SIBs, forming a HLSIB. On the other hand, the decomposition technique provides for a top-down refinement of a SIB (i.e. a HLSIB), allowing to partition the granularity of a HLSIB in smaller building blocks that can be reused, see Figure 2. A HLSIB stands for abstraction by hiding service logic and parts of the Service Support Data (SSD) that is considered to be local to the HLSIB. Call Instance Data (CID) is considered to have a more global nature. HLSIBs have the following additional characteristics:

- HLSIBs can be composed out of other HLSIBs and SIBs only.
- The lowest level of HLSIBs contains SIBs only, i.e., no further detail is visible on the GFP.
- One of the (HL)SIBs within a HLSIB is the first to be executed; therefore, HLSIBs have only one entry point (logical start), the same as with normal SIBs. But, as with normal SIBs as well, HLSIBs can have one or more exit points (logical ends).

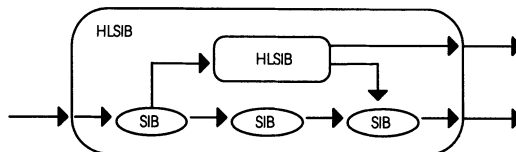


Figure 2. Recursive Concept of High Level SIBs.

So, by using HLSIBs the initially one-step functional decomposition of services into SIBs, interconnected via Global Service Logic (GSL), becomes much more flexible and more object-oriented.

3.2. Communicating Sequential Service Processes

For many current and especially future IN services parallelism is needed. It is even inevitable when parallelism is inherent to the service. And it is often wanted to improve Quality of Service (QoS). To allow parallel processing of activities in a service, a new type of service component is needed which is called service process. This means that a service may now comprise of more than one service processes (each of them containing a chain of SIBs) that are executed in parallel at a given time. This parallel execution is illustrated in Figure 3.

The spawning of a new service process is achieved by a Point Of Initiation (POI) and synchronisation between parallel service processes can be achieved via Points Of Synchronisation (POSS). Therefore, interprocess communication capabilities are needed: a Spawn SIB to spawn new service processes and Send and Wait SIBs for synchronisation purposes and data exchange of CID.

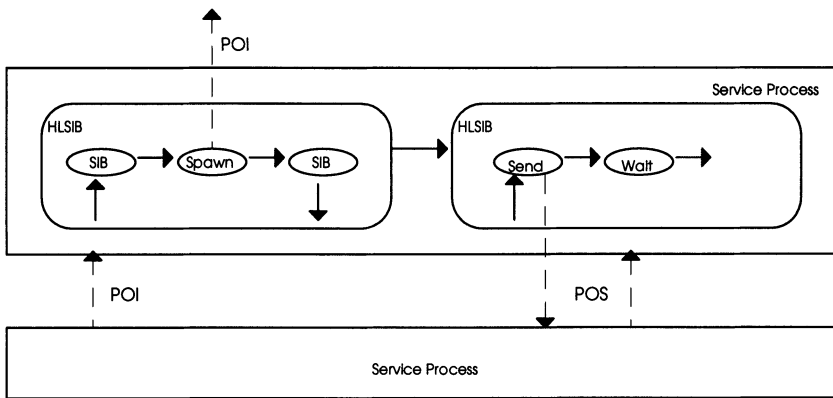


Figure 3. Communicating Sequential Service Processes.

A service process represents a chain (service logic) of SIBs or HLSIBs which are executed sequentially, but in parallel with other service processes. Service processes have the following characteristics:

- Service processes can be composed out of HLSIBs and SIBs, that are executed sequentially.
- One of the (HL)SIBs within a service process is the first to be executed; therefore, service processes have only one entry point initiated via a POI.
- Synchronisation between service processes can be performed by POSSs.
- Service processes need mechanisms to send, receive and process POIs and POSSs and to use the attached data.
- Call Instance Data (CID) is considered to be local to a service process, but global within that service process. Data exchange is performed explicitly via POIs and POSSs.

- The BCP can be regarded as a specialised service process.

Synchronisation between two service processes can be achieved via Points Of Synchronisation (POSSs). A POSS is a functional interface between service logic of two service processes over which asynchronous communication is initiated. This means that a particular SIB in the sending service process has capabilities to send a POSS to another service process that is executed in parallel. After the POSS has been sent this sending service process can continue its execution. The receiving service process, however, has to wait until the POSS has arrived. This means that the execution of a particular SIB in the receiving service process has to be suspended until the POSS has been received. If the receiving service process is not yet suspended at the time the POSS has been received, the service process must buffer the POSS. Full synchronous communication can be achieved by both SIBs of the two service processes that are performing a handshake with two POSSs.

Service processes consists of the actual service logic between Start and End SIBs, see Figure 4. But they also include a set of HLSIBs used within the service processes. In Figure 4 for example, HLSIBs x,y,z are included in service process A. And within a service process these HLSIBs can be reused several times, for instance HLSIB y is used within the service logic of the service process itself, within HLSIB x and z.

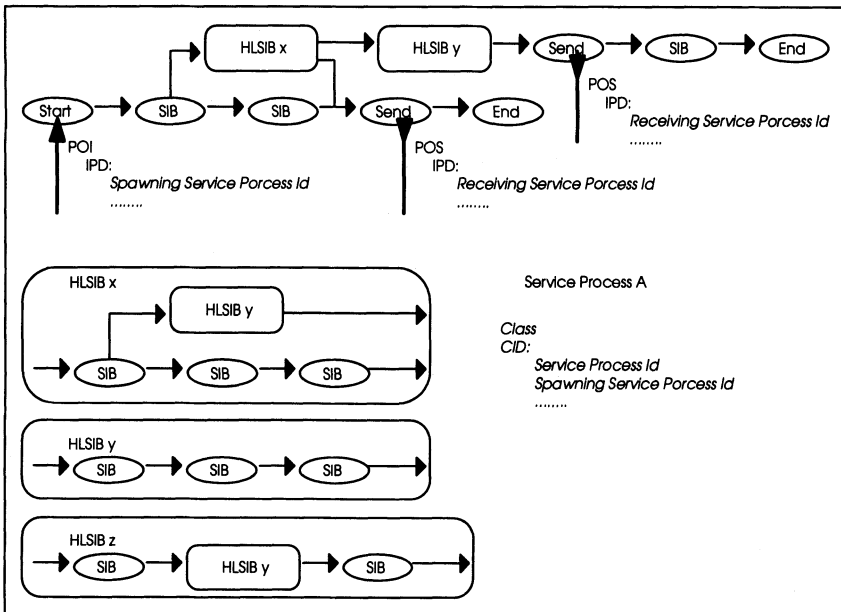


Figure 4. Modelling Service Processes.

Besides the asynchronous way of communication between service processes, it might be useful to have synchronous communication as well. Think of Remote Procedure Calls (RPCs) in which a remote procedure (different service process, or even different domain) is called and the calling side is waiting for the result. This could be modelled by HLSIBs. In Figure 4 for

instance, HLSIB z is not used by the service process itself, but it offers other service processes to invoke service logic of that service process via an RPC.

The service process of Figure 4 contains two Send SIBs that might be able to send two different POSs to another service process. For instance, this service process might send a POS with certain data attached in normal cases or a specific POS with error information in abnormal cases. But the receiving side does not know beforehand which synchronisation event to expect. Therefore it should be able to expect both. A way to handle this problem is by allowing for multiple threads. The next figure shows an example of *multiple threads* in which two different Wait SIB are interconnected via the continue outlet. Both Wait SIBs will wait for a certain POS at the same time, but only one outlet (one thread) will be executing when the expected POS arrives.

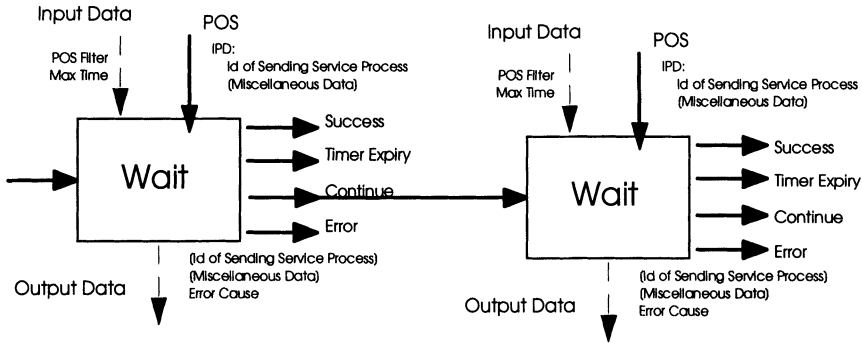


Figure 5. Multiple Threads within a Service Process.

3.3. Domains

Domains, which encapsulate predefined roles in the IN structured network are modelled by service processes that communicate over domain boundaries. Therefore, the boundaries between different domains identify logical interfaces. Within such a domain, the IN structured network is regarded as a single entity.

The domains are visible on the GFP and this visibility is illustrated in Figure 6. This figure shows as an example how a service process of the service provider (e.g. a UPT service provider) can be put on top of the service processes of the network provider (e.g. offering basic IN to the UPT service provider). POIs and POSs are used to communicate over domain boundaries and are also used explicitly to exchange data between service processes and therefore between domains. So, the set of POIs and POSs between two domains define exactly their logical interface. Furthermore, to have full control within a certain domain, service processes may not exceed domain boundaries.

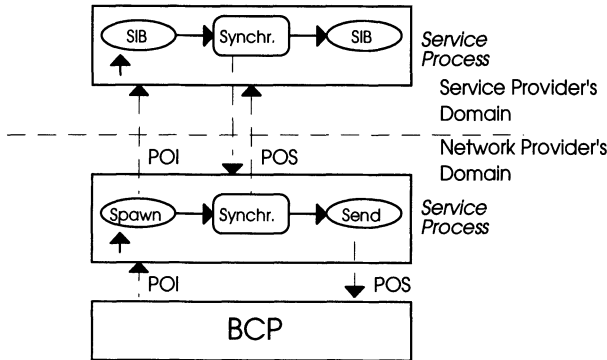


Figure 6. Domains on GFP.

What we have obtained now is not new, in fact we have modelled logical interfaces between domains which are called computational interfaces in ODP.

4. APPLICABILITY TO PERSONAL COMMUNICATIONS

In PSCS each stakeholder as shown in Figure 1 has his own domain in the GFP. Which means that also End-Users and Subscribers have their own domains. This is also recognised in services proposed for IN CS-2, such as Customized Call Routing (CCR) in which the subscriber's domain is queried during an incoming call to get further instructions (call processing and routing information) before attempting to complete the call. In PSCS such a service is called routing schemes [Mobilise D12].

In this section we will show how such a service can be implemented by using the presented concepts for modularity, communicating sequential processes, and domains. To do so we will use a scenario in which the subscriber's domain and end-user's domain is queried for further instructions. The result is shown in Figure 7.

For each of the involved stakeholders requirements concerning the processing of incoming PSCS calls are to be dealt with:

The PSCS service provider

The PSCS service provider handles all incoming PSCS calls but queries the subscriber's domain for further instructions. It provides the subscriber's domain with information such as A-number, B-number. As result a C-number to forward the call to is expected. If no response is received in time, the query fails and a notification message has to be send.

The subscriber

The subscriber's domain distributes incoming calls according to the time the call has arrived. Incoming calls will be forwarded to a mailbox, a secretary or to the end-user himself. When the end-user's domain is queried, it is provided with the A-number, B-number. As result a C-number to forward the call to is expected, this destination is checked by the subscriber as the

subscriber has to pay for the service. If no response is received in time, the query fails and the call is forwarded to the help desk.

The end-user

The end-user checks whether the caller is actually on his VIP list. If so, the call will be forwarded to his current location. Otherwise the call might be rerouted (time dependent distribution) to a mailbox.

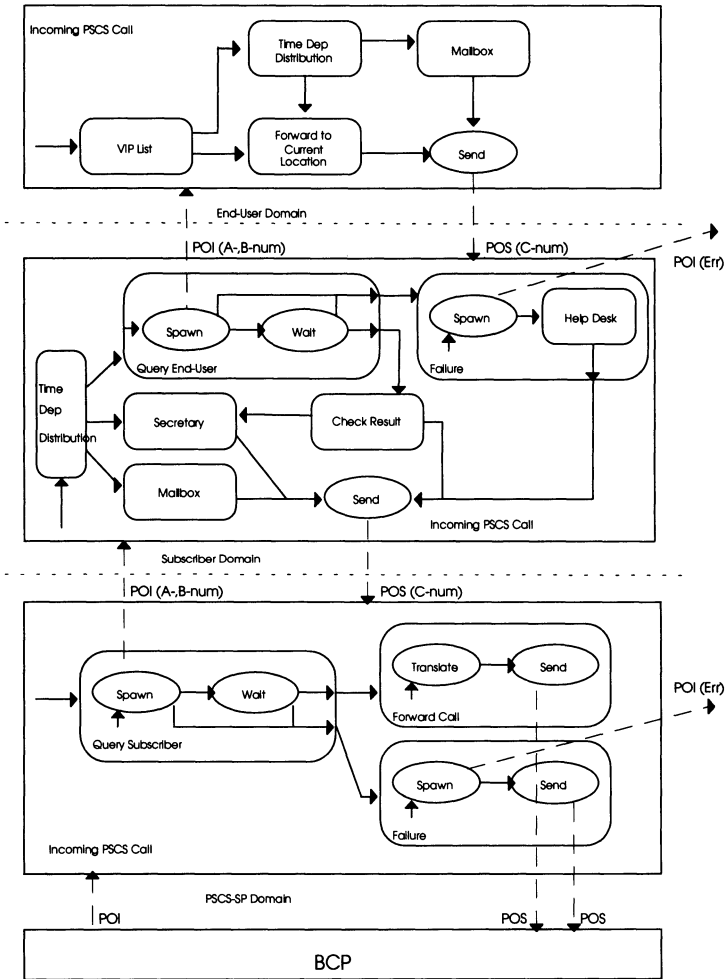


Figure 7. End-users and subscribers in control of their own domains.

This scenario is realised with IN service components such as service processes, high level SIBs and SIBs regarding the PSCS service provider's domain, the subscriber's domain, and the

end-user's domain. There are several ways in which this scenario can be mapped onto the DFP. For example, all domains could be mapped onto one SCF executing all the service logic. Or, the domains could be distributed over several SCFs and even a private SCF owned by a subscriber. And, as the logical interfaces between service processes and therefore between domains are well defined, parts of the service can easily be supported by non-IN technology such as intelligent terminals (e.g. PDAs). Such a device can simply be regarded as a black box, it needs certain input (e.g. A- and B-number) and generates certain output (e.g. a C-number).

5. CONCLUSION

This paper presents enhancements of the CS-1 concepts for Global Functional Plane to realise personalised service for subscribers and end-users. The two main enhancements are:

- The introduction of concepts for recursion (HLSIBs) and parallelism (service processes) to allow IN service engineering to be more object-oriented.
- The introduction of domains in the GFP.

SIBs are considered as the smallest service component, they are not further refined in the GFP. HLSIBs, however, are SIBs which are composed out of other (HL)SIBs. And service processes, executed in parallel, encapsulate a chain of (HL)SIBs that are executed sequentially. Furthermore, by using service processes, HLSIBs and SIBs for GFP modelling gives also good opportunities to further decompose/refine SIBs and service processes at a later stage.

The notion of domains is needed because in PSCS each stakeholder has in fact his own domain, including end-users and subscribers. This will enable personal communications adapted to personal needs in which IN and non-IN architectures work together.

6. REFERENCES

- [ITU-T Q.1200] International Telecommunication Union, Standardization Sector, "Q.1200: Q-Series Intelligent Network Recommendation", ITU-T, Study Group XI, March 1993.
- [ITU-T Q.1203] International Telecommunication Union, Standardization Sector, "Q.1203: Intelligent Network - Global Functional Plane Architecture" Q-Series Intelligent Network Recommendation", ITU-T, Study Group XI, October 1993.
- [ITU-T Q.1213] International Telecommunication Union, Standardization Sector, "Q.1213: Global Functional Plane for Intelligent Networks CS-1", ITU-T, Study Group XI, March 1993.
- [ITU-T Q.1290] International Telecommunication Union, Standardization Sector, "Q.1290: Glossary of Terms Used in the Definition of Intelligent Networks", ITU-T, Study Group XI, March 1993.

- [ETSI NA7] ETSI NA7, Technical Report ETR NA-70201, "Universal Personal Telecommunication: General Service Description", July 1992.
- [ISO ODP] ISO/IEC JTC 1/SC 21/N 7053, "Working Draft - Basic Reference Model of Open Distributed Processing - Part 1: Overview and Guide to Use".
- [Meyer OO] Bertrand Meyer, "Object-oriented Software Construction", Prentice Hall International, 1988.
- [Mobilise D4] RACE Mobilise (R2003), Fourth Deliverable of the Mobilise Consortium, "PSCS Concept: Definition and CFS, Draft Version", April 1993.
- [Mobilise D12] RACE Mobilise (R2003), Twelfth Deliverable of the Mobilise Consortium, "PSCS Specification and CFS: Architectural Framework - Draft Version", December 1993.