Service Creation From IN to Mobile and Broadband

Caroline Knight
Intelligent Networked Computing Laboratory
HP Laboratories Bristol

HPL-94-99
October, 1994

service creation, SDE, broadband services, mobile services

This paper describes the HP Service Creation Environment (SCE) and experiences with its use. The SCE is used as part of the complete process of specification, creation, test and deployment of services within an intelligent network, thus supporting the service creation life cycle. The SCE supports the creation of services for multiple network elements including an SCP/adjunct as well as a voice-based intelligent peripheral. Both low level programs and the use of Service Independent Building Blocks (SIBs) are supported. The interfaces are being extended to include broadband devices such as video servers and mobile requirements such as home location registers (HLRs). The paper includes descriptions of potential services in mobile and broadband.

Service Creation from IN to Mobile and Broadband

Caroline Knight
Hewlett-Packard Laboratories
Filton Road
Stoke Gifford
Bristol BS12 6QZ
UK
TEL:+44-272-228040
FAX:+44-272-228972
email:cdfk@hpl.hp.com

1. Service Creation Now and Future

This paper describes the functionality of the research version of HP's Service Creation Environment (SCE). Some of the design decisions and reasons behind them are given.

Section 2 describes the current HP lab's SCE[1] looking particularly at supporting the call model, providing user customisation, aiding programming abstraction and enabling the programming of multiple network elements.

Section 3 extends the model to include mobile services and section 4 discusses broadband services.

Section 5 goes beyond the programming of services in two areas; the user's interface to services and at a different type of service based on multi-media.

Section 6 concludes the paper and the references and glossary are in sections 7 and 8.

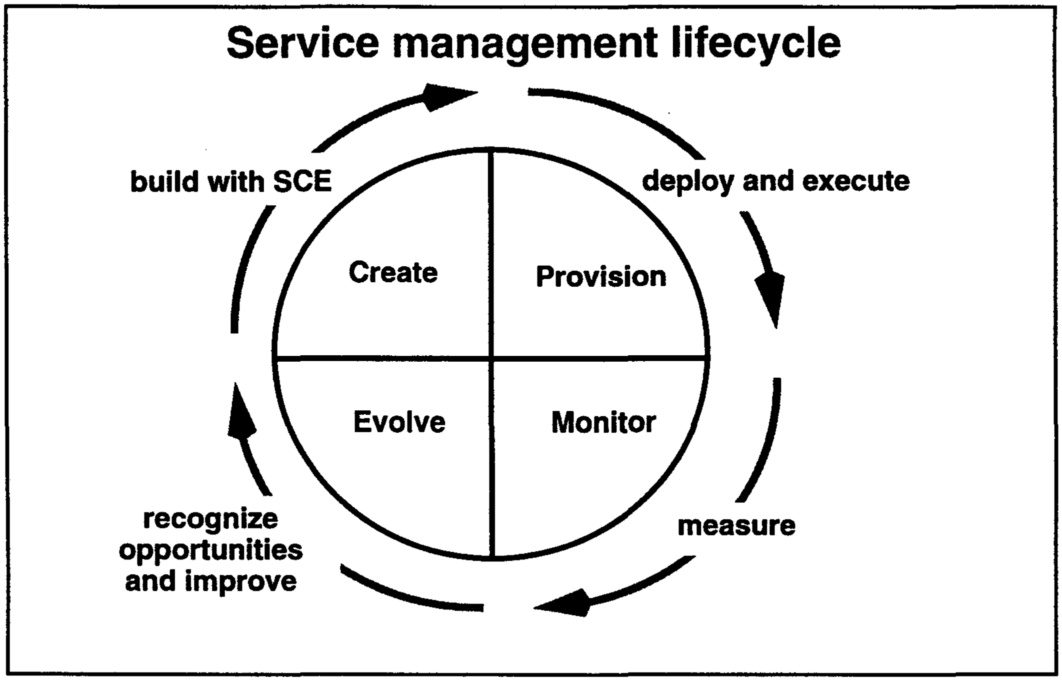
This paper does not go into the underlying transport technology although there is an assumption of an IN-like programmability, nor does it address billing or feature interaction. The architecture we are using is based on Bellcore's AIN model, not CS.1.

2. SCE Development

We have placed service creation within a lifecycle model. It is important to see service creation in context, it is not an end in itself but comes after a phase of definition, which might include marketing and major customers inputs, and is followed by the deployment, monitoring and use of the service. Throughout the lifetime of the service there will be modifications and then finally phasing out, possibly with something else as a replacement.

Our approach has been to build various tools to support services throughout the lifecycle to edit, test, monitor and improve services.

[1] All screen dumps in this paper come from working systems.

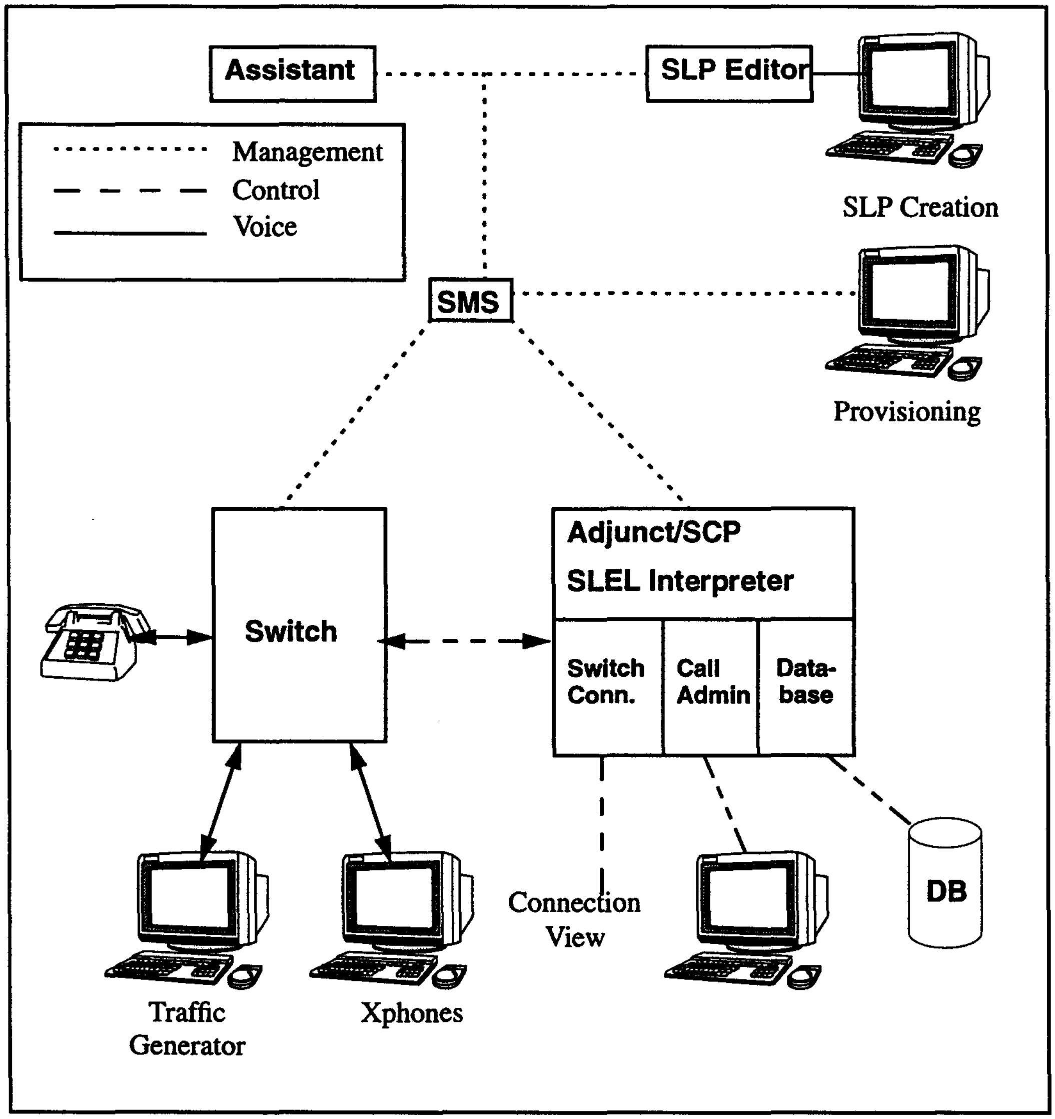


It is useful to note where the major constraints to service creation come from, namely:

- Interfaces with switches and other network elements
- Customer interfaces (telephones)
- Interfaces with billing
- Security restrictions
- Legal and regulatory restrictions

This paper concentrates on the first two of these, though one aspect of providing a restricted, more secure, interface for end-user customisation is addressed.

The work we have done has been based on a standard AIN architecture [Bellcore '90]. During the initial design and testing of a service, the switch is simulated and a cut-down version of the service management system (SMS) is used to provision the service on the service control point (SCP), software telephones (xphones on the diagram) provide number pads, on and off-hook etc. but there is no real voice path. This all runs on a single workstation. Once the service is ready a real switch can be connected and the service tried out and tested further. When we extended this architecture to include a voice-based intelligent peripheral (IP) we used real voice paths from the start as it is much more convincing to test such a service live, though again the logic of the program can be tested separately first.

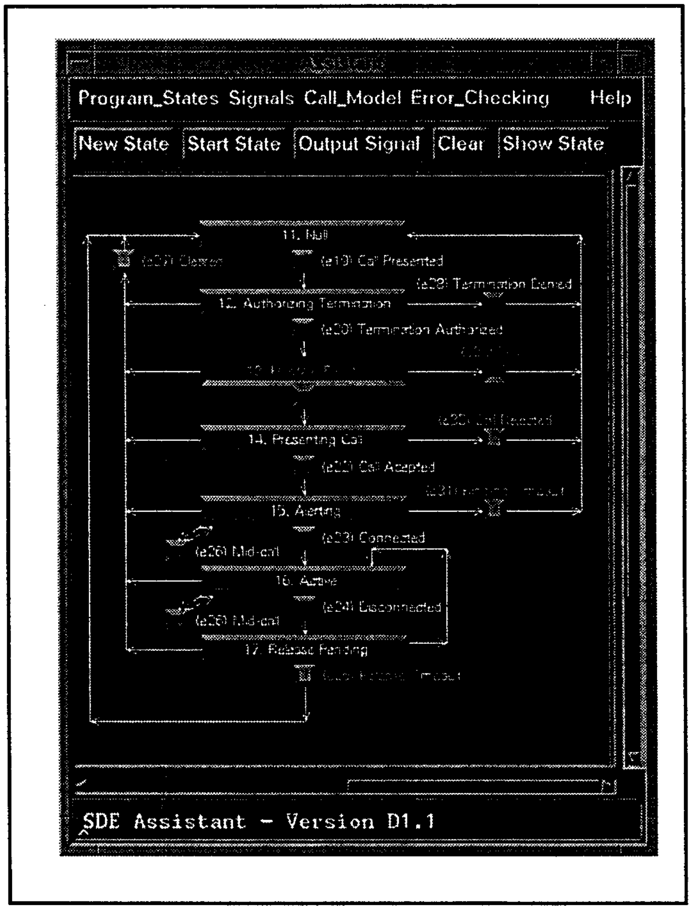


These are the components:

- The Assistant is described in the section on supporting the call model.
- The SLP editor provides three different levels of programming abstraction: textual programming language, a graphical SDL-based view and service independent building blocks (SIBs). See Wray 94.
- The SMS is the service management system, services are down loaded onto the SCP via this and the appropriate initial trigger checkpoints (TCPs) set on the switch enabling the service for a specified line.
- The switch can be any switch for which a call model is definable. Phones, either real or simulated are connected to the switch.
- The SCP executes the service logic language (SLEL), it also provides a connection view of the network which is used when testing a service to reflect the current state of the connection segments and legs.

2.1 Supporting the call model

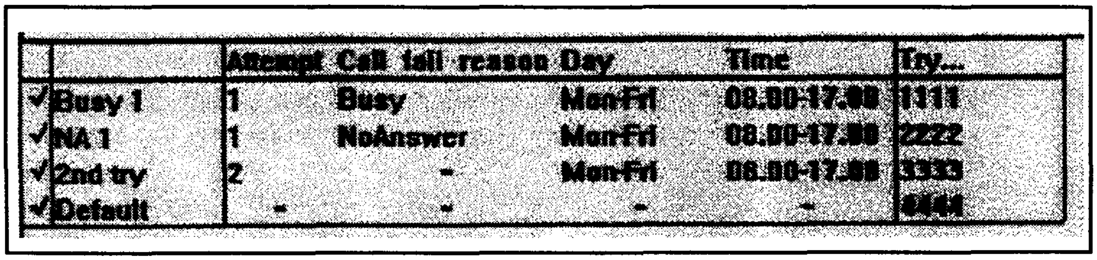
The key to IN is the call model. We have an explicit representation of the call model in a tool called the assistant. Using this tool the service logic programmer has a guide to which events and trigger checkpoints are available and which must be handled. The assistant can also translate a service between two different call models (if the new call model is rich enough to support the service). Even if the programmer does not use the assistant to guide or translate, it is useful as an active picture of the call model.



When the user clicks on an event of interest, the appropriate trigger checkpoints are highlighted.

2.2 Providing user customisation

An important concept in service creation is that of on-site, real time customisation, using customer premises equipment (CPE). This is something which can make a service more valuable to the user, but carrier and service providers will be keen to ensure that customisation is a secure activity. Therefore to simplify the task and to ensure the restrictions necessary we have a tool which allows simple default call processing rules to be created. In user studies these were found to be a more effective representation than the more common decision tree when modelling policy information. The rules are in fact compiled into decision trees but they provide a neatly condensed view of the information needed:



		Attempt	Call fail reason	Day	Time	Try...
✓	Busy 1	1	Busy	Mon-Fri	08.00-17.00	1111
✓	NA 1	1	NoAnswer	Mon-Fri	08.00-17.00	2222
✓	2nd try	2	-	Mon-Fri	08.00-17.00	3333
✓	Default	-	-	-	-	4444

This example shows a hunting policy where during office hours the first attempt at redirection depends on the reason for failure so far:

- If the line is busy then the call is forwarded to 1111
- If on the other hand the call hasn't been answered after a specified number of rings then it is forwarded to 2222.
- No matter which of these were used, if they fail then the call is forwarded to 3333.
- Out of hours and at weekends the last rule is used and the call forwarded to the office answer phone on 4444. This is also the last resort if 3333 fails to answer.
- Notes that the "-" is a wildcard and therefore the least specific value that can be given. The rules are applied so that the most specific, relevant rule is applied first.

It is simple for the user to update days, times, target numbers as well as adding new rules whilst all these choices are restricted to only these variables and to ranges of acceptable values for these variables. The user states the general rule then enumerates the exceptions separately.

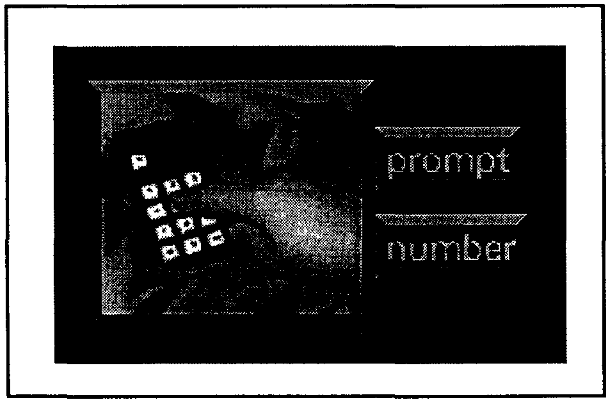
2.3 Aiding abstraction - SIBs

The graphical layout of SDL provides an extremely good way of describing and showing the control flow of a service. However, it has some major limitations:

- A real service is hard to see all at once on the screen. Good zooming in and out reduces this problem, but at some level the sheer size needs to be reduced to aid understanding.
- The only way to re-use bits of code is by copying and pasting. This can lead to errors and makes updating complicated; it also fails to provide a clear indication to the viewer that the same actions are being used in more than one place.
- Another consequence is that one is less likely to make generalisations that help in the understanding of the structure. Such generalisations require some form of input variables and some way of returning results.

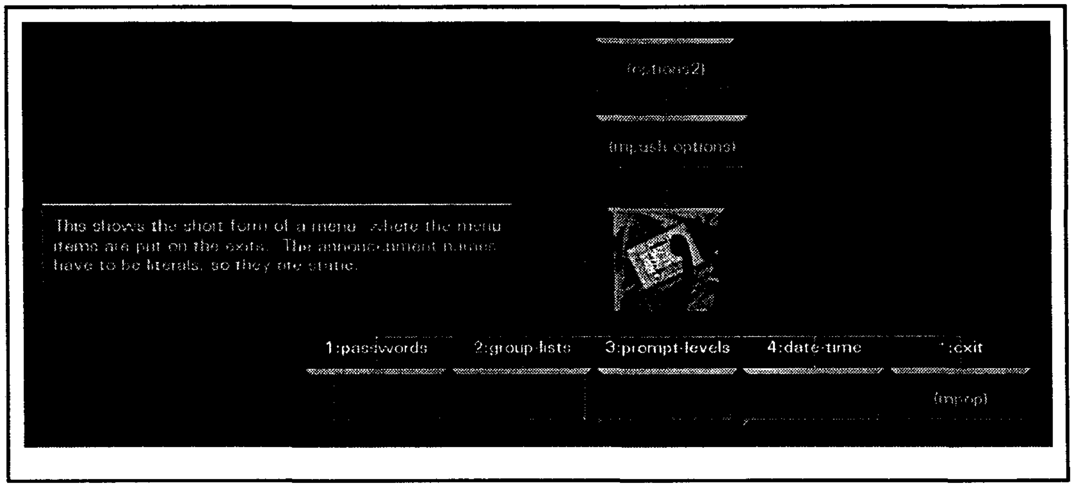
The textual solution to this is a subroutine or function, the graphical solution is a service independent building block (SIB)[1].

Therefore the SIB is an important aid to writing and reading services. Our SIBs take arguments, and return results.



This SIB is "Gather Digits". It takes two arguments: a *prompt* which is played, e.g. "Please type in your personal identification number now" and *number* which indicates the number of digits required. The *prompt* is optional and the programmer may remove it.

One difference from a standard programming subroutine is that when programming graphically it is natural to have multiple exit points.



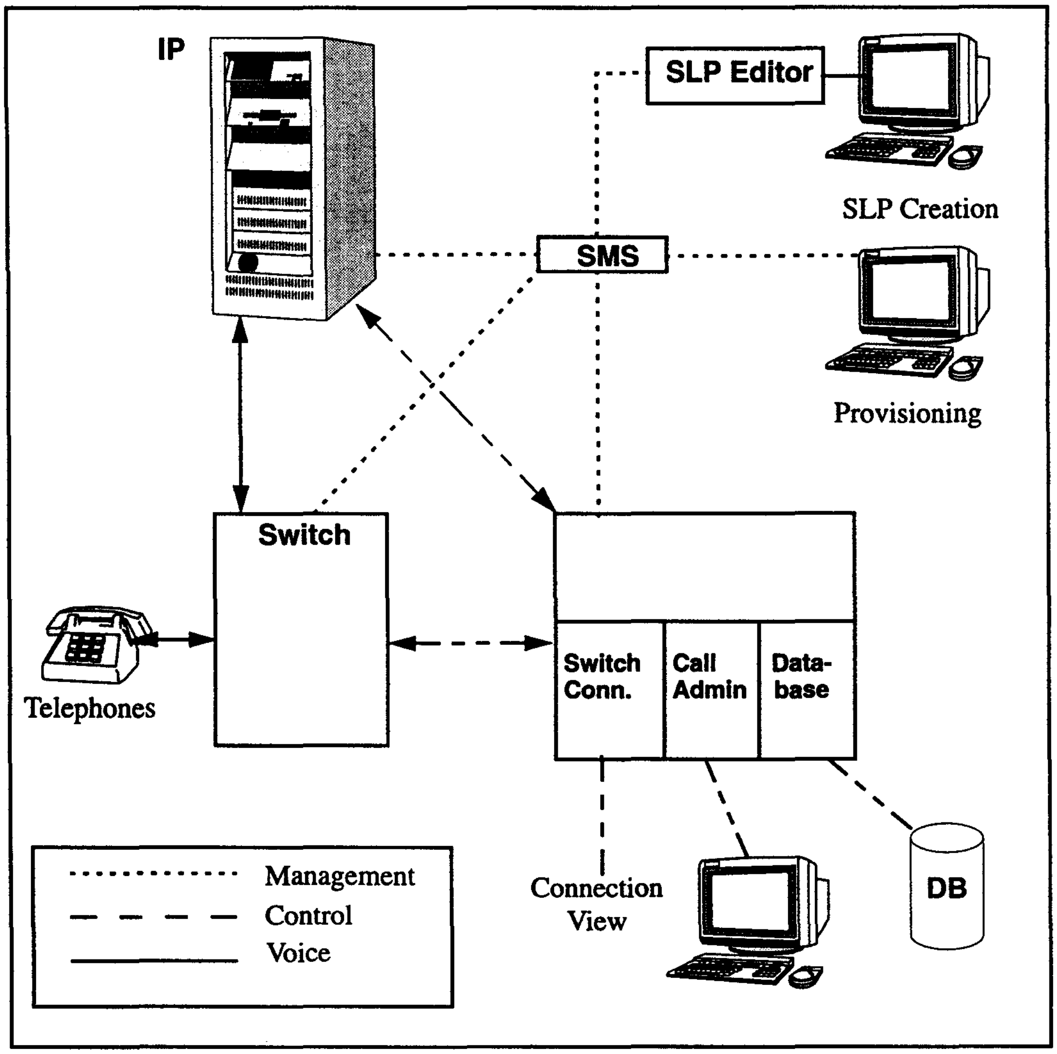
This shows the short form of a menu where the menu items are put on the exits. The announcement names have to be literals, so they are static.

This menu SIB has five normal exit points which correspond to the choices available to the caller with the associated results.

[1] CS.1 has a defined set of SIBs, our implementation is based on a generalisation of the idea not the CS.1 set.

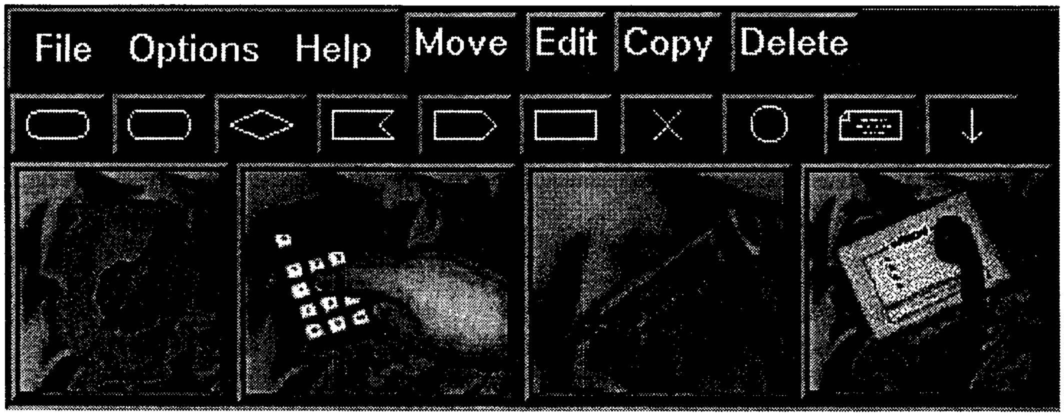
2.4 Multiple network elements

We have extended the IN architecture to add in a voice-based intelligent peripheral (IP). This is an example of having multiple programmable elements in the network which must be orchestrated to provide services. The IP is added to the architecture thus:

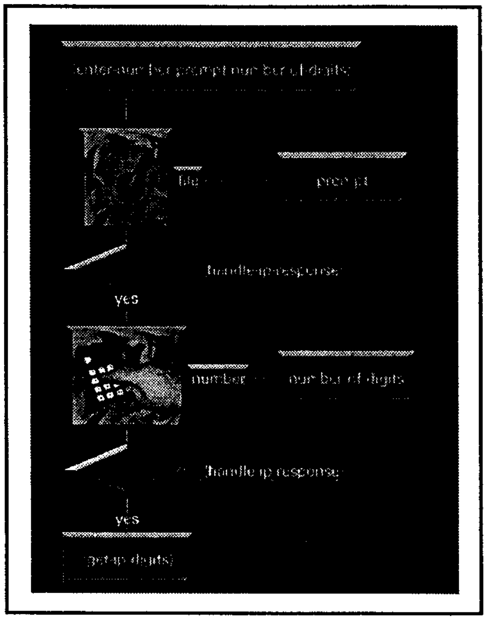


The SMS provides a management path to all components and the SCP has a control path to both the switch and the IP. The switch treats the IP as any other element connected via voice paths and it may be physically remote via trunk lines.

Using SIBs we packaged up the interface to the IP. Here are the SIBs for “Play Announcement”, “Gather Digits”, “Record voice message” and “Menu”.



Using these the service writer can make use of the voice-based IP without deeper knowledge of its functioning. One can use these SIBs to define further SIBs such as “Check authorisation” the IP part of which might look like:



To complete the SIB one would need to check the number with a secure database of PINs and might put in a loop to allow the user a couple of tries before failing. This and other enhancements such as producing an alert after a specified number of failed attempts from the same user could all be packaged inside the “Check Authentication” SIB. When further enhancements are made these can be automatically made available to all other services using this SIB.

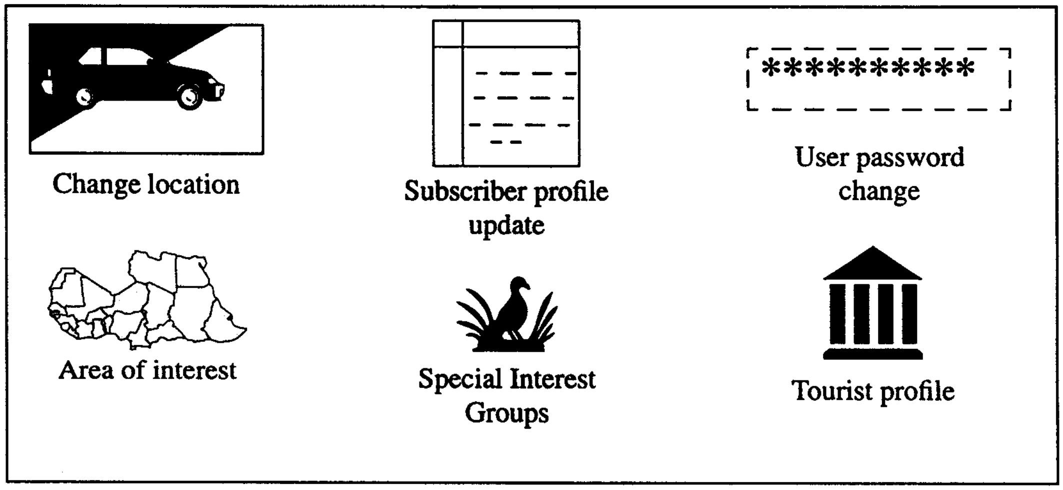
3. Extending to Mobile

In mobile the whereabouts of the subscriber is of key importance. Therefore location tracking and authentication for secure communication are needed. For example GSM uses Visitor Location Registers (VLRs), Home Location Registers (HLRs) and Authentication Centres (AuCs) to keep track of the dynamic user.

Consider a few potential mobile services:

- Roadworks information based on location
- Location of stolen car
- Distribution of tasks to mobile workers, e.g. taxis, roadside assistance
- Broadcast alerts relevant to area e.g. flood warning

To enable these services a call model is needed to package up the interactions with the mobile-specific components of the network thus enabling the service writer to concentrate on the logic of the service rather than the specifics of particular databases or protocols. For instance a key event is the change of location which could trigger various services. Other triggers which are common amongst mobile phones are on/off, subscribe/unsubscribe. Strictly speaking these are not elements of a *call* model but more of a general mobile service model, where calls are only one element of the service. Some parts will more naturally be packaged as SIBs, especially those which require inputs or return results.

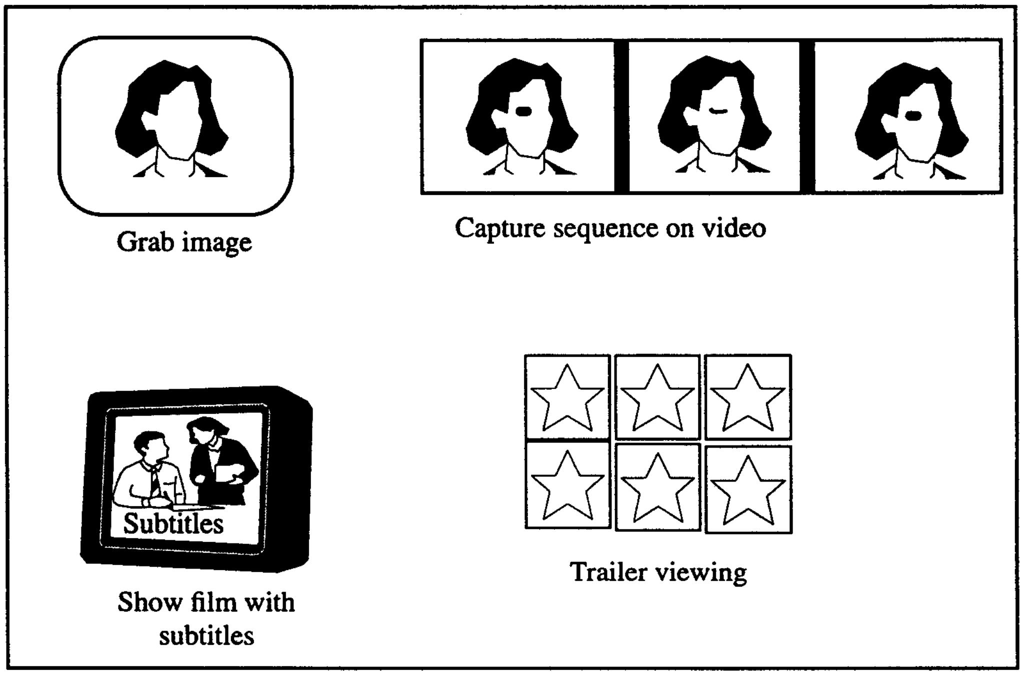


4. Broadening the Band

Another aspect of telephony that is changing the way people use and think about services is the greater bandwidth becoming available. This leads to potential video-based services. The standard ones which are quoted are:

- Tele-shopping
- Tele-banking
- Video on demand
- Video phones

Adding in video is a major change. Mobile is voice based, land-based telephony is voice based, the IP previously mentioned is voice based. Video requires changes in the user's equipment to display images. However from the service providers' point of view incorporating video requires many of the same elements as other new network elements; some form of call model defining the course of an interaction and the potential interrupt points and triggers plus the ability to package up reusable elements as SIBs.



Grab image

Capture sequence on video

Show film with subtitles

Trailer viewing

The user's interface to the service becomes both more flexible and more apparent. Many SIBs could package up frequently used user interactions such as:

- Trailer viewing - this is a way of quickly selecting a particular video, or position in a video. The video might be a film, a magazine, or a music video for instance. The trailer viewer shows stills from key points and when selected shows a burst of video or an edited trailer, the length of time will vary with the purpose.
- Recording and editing the welcome message on your video answer phone - or more gener-

ally recording and editing a short video. This needs to be an easy function to perform and not require the complexity of a full editing suite.

- Navigation menus, agents or services to aid the user in finding their way to fun, games, information and friendships.

Most of the current emphasis is on person-to-machine interaction rather than the traditional role of telephony which is person-to-person interaction. Both need to be addressed and they need to be combined to allow, for instance:

- Shopping with friends
- Family banking
- Film clubs
- Education, for instance a group tour of historic places where one also interacts with teachers and other students
- Distributed music making
- Multi-user games

Starting with this goal is important as there are consequences on the hardware which users will need; not only must the device be able to receive video, data and high quality sound but it must also be able transmit all three as well.

5. Beyond Programming

There is more to a service than the writing, debugging and provisioning on switches and other network elements. One of the main reasons for the lack of use of services currently available, especially on business site PBXs, is the user's interface to those services - the phone. Some phones have been created with buttons to support specific services but this is not a viable solution in a world of ever increasing new services.

5.1 Tomorrow's telephone

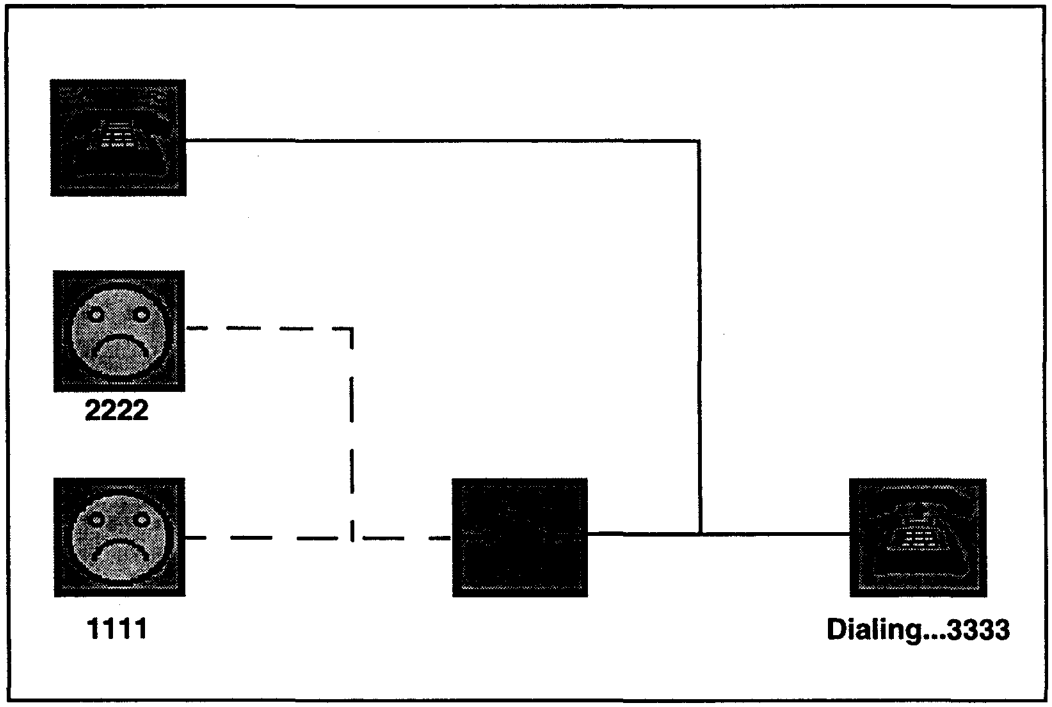
Smart phones of various descriptions are available, however there is a tendency towards unusable complexity. The difference between today's portable computers and tomorrow's telephones is small; it will be important to draw the line correctly between flexibility and the ease of use of a purpose-built solution. The user and designer will both be faced with choices. The shape of phones now and soon to be:

- A traditional handset, wired into a single location and supporting voice with only the barest level of support for services via the number pad and hook. An LCD number display if you are lucky.
- Next generation handsets with built in softkey screens, typically dependent on a particular switch or PBX e.g. ADSI
- A mobile phone, with limited battery power but flexible location. Supporting voice but likely to have better support for a small set of services through specific buttons. A small

LCD screen.

- A settop which connects your television to a video server. It may be smart enough for some shopping services to be implemented. It can use the TV screen for a much more flexible interface. This is not portable.
- A video phone to talk to those with whom voice contact is not enough. With a well designed screen you get the advantage of images and flexible control of services. Maybe it will even be touch sensitive.
- A palmtop computer phone that plugs into an ISDN line and provides access to new services.

One of the benefits of incorporating a screen for even a purely voice service is that many ordinary services become more attractive. Here is an example of a conference call:



The initiator has already connected with 1111 and 2222 who are now on hold whilst 3333 is being dialled. The current call models make it difficult to provide all participants with the same view but ideally everyone wants to be able to see who is in the conference and to be kept up-to-date as people leave or join.

5.2 Service development support for multi-media

There are several tools already on the market which support the creation of animation, video, with sound-tracks, interactive stories etc. The creator is more of a director or writer than a programmer though there are many elements in them which are akin to the programming of services. At the high-level where there are only a few well packaged interactions with the mechanics of telephony such tools could be adapted to service writing. To enable this sort of service creation we must look at what aspects of telephony will need to be supplied.

A call model that combines all the possibilities is required. Each network element will have its own subset of the call model. The potential states and triggers need to be defined. However this degree of complexity will need to be packaged up into worthwhile chunks that are embodied in SIBs. To clarify this let us take an example:

A Story At Bedtime

Imagine:

- Grandma writing a picture story for all her grandchildren
- She is effectively customising the service provided to her
- The service providers have a means of taking her graphics (stills, animation and video) and her sound (voice, special effects and music). They put this on a form of IP which her grandchildren are given the password to.
- The children telephone the IP, give the password and then play with Grandma's active story whilst she, several countries away is already asleep. They are themselves in two different towns and discuss the story whilst playing it together.

Maybe the simplest version of this is where the user, Grandma, simply customises a supplied story by including the names and pictures of the grandchildren and their pets. But suppose Grandma has drawn some cartoon characters and wants to dictate the plot and types of choices available. Now the tools need to include animators which take a few graphics and manipulate them to provide interesting continuous cartoons, plot editors which allow a flow of control to be mapped onto these characters actions and various types of events where choices can be made, the characters will also need voices and sound effects, all these need to be either under the control of Grandma or taken care of in an acceptable manner. Some of the aspects which require direct support during use are:

- Video display
- Synchronised soundtrack
- Additional voice connection between players
- Control of interaction
- Multiple views - players make different choices from each other so their views are all different - if the choices are mutually exclusive multiple worlds are needed
- Shared space - players might alter the environment as experienced by the other players within the same story world, other spaces might always be seen as new
- Representations of individual players

6. Conclusion

The scope of services is increasing rapidly; the requirements on service creation are growing equally. To enable networks to incorporate more than just switches the service writer needs either detailed models of each component or neatly packaged SIBs. An important element in service uptake will be the user's interfaces to those services - they must be flexible yet simple.

7. References

[Bellcore 90]: "Advanced Intelligent Network Release 1 Network and Operations Plan", June 1990

[Wray 94]: Mike Wray and Mark Syrett "Service Creation Using the Hewlett-Packard Service Creation Environment" Intelligent Network '94 Workshop, Heidelberg, Germany May 1994

8. Glossary of terms

These are mostly three letter acronyms:

ADSI: Analog Display Services Interface

AIN: Advanced Intelligent Network

AuC: Authentication Centre

Conn view: The connection view, synonymous with c-view

CPE: Customer Premises Equipment

DB: database

GSM: Global system for mobile communications

HLR: Home Location Register, see also VLR

HP: Hewlett-Packard

HPLabs: Hewlett-Packard Laboratories

IN: Intelligent Network

IP: Intelligent Peripheral, generally voice-based

ISDN: Integrated Services Digital Network

PBX: Private Branch Exchange

SCE: Service Creation Environment

SCP: Service Control Point

SDL: Service Description Language

SIB: Service Independent Building block, synonymous with SIBB

SLP: Service Logic Program

SMS: Service Management System

TCP: Trigger Check Point

VLR: Visitor Location Register