PERIPHERAL CONTROL IN AN INTELLIGENT NETWORK

Background To The Invention

The present invention relates to a telecommunications system using an intelligent network (IN) architecture, and in particular to the control of a peripheral in such a network.

In conventional public switched networks (PSTNs) the functions the network is able to perform depend on the design of the switch which handles basic call-routing. Introducing any new functionality requires the switch to be redesigned, with the provision of new software and possibly also new hardware. For a large network this process can take one or more years. As PSTNs are required to provided a rapidly growing range of services to users, this type of delay in service provision becomes increasingly unacceptable.

Intelligent networks (IN's) have been developed to address the problem of the relative inflexibility of conventional networks. As described in the paper by TW Abernethy and AC Munday ,"Intelligent networks, standards and services" BT Technol J Vol 13 No 2 April 1995, in the intelligent network the control of the basic call routing is carried out at a service switching point (SSP) and is kept functionally separate from the control of call progression by a service control point (SCP). The service control point is also able to access functions provided by an intelligent peripheral (IP). An intelligent peripheral might be equipped, for example, to play pre-recorded announcements to a subscriber. To ensure that the interworking of networks and components of different operators and suppliers is possible, the architecture and protocols of the intelligent network are required to conform to internationally agreed standards. The need to conform to these standards might however tend to negate the very flexibility which the IN architecture was intended to provide. For example, BT has developed and deployed in the UK PSTN a voice messaging service known as CALL MINDER (TM). It would be desirable to incorporate this service on an intelligent peripheral within an IN system. The Call Minder functions however do not correspond to either of the two classes of IP functions which the relevant standards provide for.

Summary Of The Invention

According to a first aspect of the present invention, there is provided a method of operating an intelligent peripheral (IP) under the control of a service control point (SCP) in a telecommunications network, the method comprising:

a) passing control messages conforming to a standard protocol from the service control point to the intelligent peripheral, some of the said control messages including an extension field identifying the messages as relating to an additional function not provided for in the standard protocol;

b) executing on the intelligent peripheral, in response to at least some of the messages, predetermined functions which are provided for in the said standard protocol; and

c) executing on the intelligent peripheral, in response to a message including the said extension field, a routine which implements the additional function.

The present invention provides for the first time a method of accessing any additional functions which may be provided on a given intelligent peripheral whilst maintaining conformity with standard protocols. This is achieved by effectively disguising calls to the additional functions as messages for invoking the standard functions, but with a field included in the message identifying the disguised nature of the message. The intelligent peripheral then responds by executing application code performing an appropriate one of the additional functions instead of the relevant standard function.

Preferably the said standard protocol is the Intelligent Network Application Protocol (INAP) and the said field identifying the message as relating to an additional function is the extension field of the INAP IP-to-SCP application context.

INAP - the Intelligent Network Application Protocol - is an internationally agreed standard for Intelligent Networks. The standard is set out in full in "Intelligent Network (IN); Intelligent Network Capability Set 1(CS1); Core Intelligent Network Application Protocol (INAP)", European Telecommunications Standards Institute, pr ETS 300 374-1. As described in further detail below in relation to the preferred embodiment, this protocol includes an unassigned field, the extension field, in messages passed between the IP and SCP. Conventionally this extension field would be used for passing additional data or parameters to one of the

standard functions of the IP. In the preferred implementation of the present invention, this field is used instead as the identifier for messages invoking one or more additional functions available on the IP the SCP.

Preferably the said additional function includes an in-band interaction with a user connected to the IP via a SSP. This function may include a voice recognition function, in which case preferably the IP returns to the SCP data obtained by the voice recognition function

According to a second aspect of the present invention, there is provided an intelligent peripheral (IP) for use in a telecommunications intelligent network under the control of a service control point (SCP), including means for reading control messages in a standard protocol received from the SCP and means for executing in response to the control messages predetermined functions provided for in the said standard protocol, characterised by means for recognising in the said messages an extension field identifying a message as relating to an additional function not provided for in the said standard protocol, and means responsive to the said means for recognising for executing the said additional function.

According to a further aspect of the present invention, there is provided a method of operating an intelligent peripheral (IP) under the control of a service control point (SCP) in a telecommunications intelligent network, including passing control messages conforming to a standard protocol from the service control point to the intelligent peripheral, and executing on the intelligent peripheral in response to at least some of the control messages predetermined functions provided for in the said standard protocol,

characterised in that others of the control messages conforming to the standard protocol include a field identifying the messages as relating to an additional function not provided for in the standard protocol , and by executing on the intelligent peripheral in response to a message including the said field a routine implementing the additional function.

The invention also encompasses a service control point which implements the invention of the first aspect, and a telecommunications network incorporating such a service control point.

Brief Description of the Drawings

Systems embodying the present invention will now be described in further detail, by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a diagram showing an intelligent network architecture.

Figure 2 is a schematic of a service switching point (SSP);

Figure 3 is a schematic of a service control point (SCP);

Figure 4 is a diagram showing one example of an intelligent peripheral;

Figure 5 is a diagram showing schematically the interaction between the SCP, SSP and IP;

Figure 6 is a diagram showing an overview of the functional architecture of an IP embodying the present invention;

Figure 7 is a schematic of an IP service platform;

Figure 8 is a schematic of an MTP shelf for use with the service platform of Figure 7;

Figure 9 is a schematic of an alternative implementation; and

Figure 10 is a schematic of a speech processor shelf; and

Figure 11 is a schematic showing the format of an extension field.

Description of Examples

A telecommunications network employing an intelligent network (IN) architecture includes a number of service switching points (SSP's) 1 which may be located, for example, at a transit exchange or local exchange. Subscriber terminals 2 are connected, directly or indirectly, to the SSP. The SSP carries out a call control function (CCF) which comprises the basic call processes associated with the setting up of a connection between different terminals on the network. In addition, as shown in Figure 2, the SSP includes a service switching function (SSF) which interfaces the call control function to other entities and functions on the network. In particular, the SSP is connected to an intelligent peripheral (IP) 3 which in conjunction with the SSP makes available so-called "specialised resource functions" (SRF's).

A service control point (SCP) 5 is connected to the service switching points 1 and the intelligent peripheral 3 via a signalling network 4. This network uses a digital message-based common channel signalling system known as Signalling System No. 7 (SS7) in an implementation developed by BT known as the

BT national user part or NUP. The SS7 signalling system is described in further detail in Fretten KG & Davies C G: "CCITT Signalling System No. 7; Overview", British Telecommunications Eng J (April 1988).

For completeness, the diagram of Figure 1 includes the service creation environment and service management system associated with the service control point.

In operation, enhanced core control functions run on the service control point, while the basic service switching is carried out by the service switching point. This allows the progression of a call to be suspended while further information regarding how the call is to be handled is obtained from a service data point (SDP) linked to the service control point. The decision to suspend call processing is based on the meeting of certain pre-specified criteria termed "triggers". These may be, for example, certain dialled digits or line conditions at points during the call. As seen in Figure 2, the SSP includes a trigger table linked to customer data.

Figure 3 shows in further detail the architecture of the service control point. The service control point is implemented on a general-purpose computing platform that hosts advanced service control software or service logic.

The structure and operation of the network, other than in the aspects described further below, are generally conventional in nature and are not further described here. For further information, reference is made to the above-cited paper by TW Abernethy and AC Munday and to the European Telecommunications Standards Institute final draft prETS300374-1 published July 1994.

The operation of a system embodying the present invention will now be described with reference to Figures 4 to 11.

As discussed in the introduction above, the invention is particularly concerned with the control of the intelligent peripheral IP by the service control point SCP. Conventionally, using the INAP protocol, the SCP has been able to access just two classes of operation in the IP. These are known respectively as (PromptAndCollectUserInformation (P&C)) and (PlayAnnouncement (PA)). These operations are defined in ASN.1 notation in Tables 4.1 and 4.2, and their arguments are defined in Tables 5.1 and 5.2. This notation is described in CCITT Recommendation X.208 (1988): "Specification of Abstract Syntax Notation 1

(ASN.1)". As will be understood by those skilled in the art, this notation serves not merely to define the relevant features, but also, following standard encoding rules, provides a basis for the generation of code to implement the features by decoding and encoding messages passed between the SCP and IP. The code may be generated, for example, using commercially available ASN.1 compliers.

The PA operation is used for in-band interaction with an analogue user or for interaction with an ISDN user. For example, in response to certain triggers during a call, such as a failure by a calling party to dial a recognised number, the SCP may establish a connection between the IP and the calling party, and instruct the IP to play a pre-recorded message: "the number has not been recognised". The PA operation does not return any data to the SCP. The parameters of the PA operation are listed in Table 6 below.

The P&C operation is used to interact with a call party to collect information. For example, in a recorded timetable service with a numbered menu of different timetables, the user may be asked to dial a digit indicating which timetable is to be played. The IP recognises the value dialled and returns this value to the SCP which can then instruct the IP to play the appropriate recording to the call party . The parameters of the P&C operation are listed in Table 7 below.

Figure 5 shows the sequence of operations and messages passing between the SCP, IP and SSP during an operation such as PA. To access an IP function, the SCP first sends an EstablishTemporaryConnection message (m1) to the SSP. This message includes the address of the IP, a correlation ID, and an identifier for the issuing SCP, the SCF ID. In response to this message the service switching point SSP initiates a call (m2) to the identified intelligent peripheral IP. The IP responds to the controlling SCP with an AssistRequestInstructions message (m3). The response from the SCP is a call (m4) to the relevant function provided by the IP. In the example of the intelligent peripheral shown in Figure 6 the call (m4) to one of the special resource functions is handled by the Application Manager 61 under the control of the SRF. The SRF is a module of the IP which interprets the call from the SCP and loads and using the Application Manager 61 runs on the computing platform 62 of the IP application code for implementing the relevant function. Message (m5) is the in-band interaction between the IP and the user via the SSP corresponding to the function called by message (m4).

In this example of an IP, additional special resources termed by the inventors "custom dialogues" D1 ... Dn are available. These functions are invoked by service requests from the SCP which retain the format of the conventional calls to the PA or P&C resources, but which include in addition the extension field provided for those operations in the INAP protocol. Thus, the sequence of messages m1 to m6 outlined above remains essentially unchanged, but message m4 includes the extension field. In systems embodying the present invention, the IP and SCP are programmed to treat this field as a flag identifying a message as relating to a custom dialogue rather than a standard resource. The IP then runs the dialogue indicated by the sub-field "dialogue identifier" contained within the extension field, instead of the standard dialogues the operation carrying this extension suggest (ie PA or P&C). The manner in which this field is used and interpreted is formally defined in Table 1 below using the ASN.1 formalism.

Any custom dialogue may include enhancements using data specific to the custom dialogue. By way of example, a first enhancement is described which provides support for voice recognition. As described above, of the two standard operations PA and P&C, only P&C has a result in which information gathered in an interactive dialogue can be returned. Since it is essential for the voice recognition operation to be able to return the results of the recognition, this enhanced dialogue, custom dialogue enhancement 1, is only defined for the P&C operation. This enhancement operates in addition to the main body of the P&C operation and therefore allows both digits and yes and no to be recognised at the same time. This enhancement is set out in ASN.1 notation in Table 2 and Table 8 lists the meaning of the parameters of this operation. Figure 11 shows the format of the signals making up the extension field passed from the SCP to the IP in this example. As shown in the Figure, the extension field , and each of the sub-fields, are identified by a tag having a predetermined value and each field or sub-field is of a predetermined length. The overall length of the extension field is 79 bytes, comprising a 2-byte tag and contents of length 77 bytes. The extension field has Type, Criticality and Value sub-fields, and the Value sub-field in turn has its own sub-fields, as shown in the Figure and defined in the ASN.1 listing.

In a second example of a system embodying the present invention three custom dialogues are provided to enable the IP to function as a messaging platform. The 3 dialogues are:-

1. Name = messaging, feature = record message, version = 0.1.
2. Name = messaging, feature = play back message, version = 0.1.
3. Name = messaging, feature = delete message, version = 0.1.

Each of the custom dialogues includes enhancements, contained in the dialogue data, which allows them to transport data specific to that dialogue. Table 3.1 below sets out the record message enhancement. This allows the SCP to specify the format of the message that is to be recorded by the IP. Two formats are currently possible, voice or fax.

Table 3.2 sets out the play back message enhancement. This enhancement specifies a message and the format (speech or fax) in which it is to be played back by the IP. If playback of messages were to be achieved using the conventional play announcement operation, then this would not allow fax messages to be retrieved. The playback message enhancement is therefore used to provide for the message being returned as a fax or in other formats in addition to the speech format of the standard PA operation.

Table 3.3 sets out the delete message enhancement. This enhancement allows the user to specify which of a number of stored messages is to be deleted from the intelligent peripheral storage.

One example of the hardware for a platform implementing the present invention will now be described in further detail, and the mapping of the different processes discussed above to the hardware outlined. Referring back to Figure 6, it can be seen that the highest level functions are the special resource function (SRF) and the associated application manager. The SRFs are linked to a transaction processing or TCAP - Transaction Capabilities Application Part layer. This in turn sits upon a SCCP, (Signalling Connection Control Part,) layer and an MTP (Message Transfer Part) layer. The circuit-related signalling is handled by a NUP layer. The peripheral also includes SRF management functions and C7 signalling management.

In the platform built by the present applicants and known as the "Interactive Speech Application Platform" (ISAP) there are four main sections, each implemented as a shelf containing a number of processor cards.

An IP service platform (Figure 7) implements the TCAP and SRF functions, and stores data for use by the IP. As shown in Figure 7, in this example the IP service platform comprises network processor, FDDI interface, TCAP processor and SRF processor cards mounted on an VME backplane and linked by a SCSI bus connection to 1 Gbyte hard discs. The network processor uses a Motorola 68040/4CPU, the TCAP processor uses a Motorola 68040/16CPU and the SRF processor uses a Digital SPARC 5VE processor with 16Mbytes of memory. The IP service platform shelf is linked by the FDDI local area network to SP, MTP and management shelves.

The SP (speech processor) shelf (100; Figure 10) includes a telephony processor configured to use the C7 protocol discussed above. The telephony processor (TP) supports NUP and also the MTPin MTPout processes necessary to communicate with the MTP (message transfer part) processes. The other major components of the SP shelf are application processors which support the application manager and the routines executing the SRF functions or "dialogues". The application manager uses a broadcast mechanism to select SRF instances. A digital line interface card (DLIC) terminates PCM signals received by the shelf. A number of speech cards SC each containing a DSP such as a Motorola 56000 series processor, implement speech recognition and related functions. Network processor (NP), resource management processor (MP), and telephony processor (TP) cards each use Motorola 68040/16 CPUs. The three application processor cards AP_{1-3} each contain a SPARC 5VE processor.

The MTP shelf supports the MTP and SCCP functions. The interface between MTP and SCCP is provided by MTPin and MTPout process pairs. There are two MTP processors per link set, each using one of two separately powered VME backplanes, with typically two links in each link set. The MTP processing for non-circuit related traffic is not mixed with circuit related traffic. There is therefore provided one set of MTP processor pairs for non-circuit related traffic and another for circuit-related traffic.

In the example illustrated in Figure 8, the MTP unit consists of an VME backplane with a network processor card carrying a Motorola 68040/4 processor and an FDDI interface card. There is then one MTP processor - a Motorola 68040/16 CPU with 68302 "Eagle - 03" module - for each signalling link set.

The MTP unit also includes an SCCP processor card including a Motorola 608040/16 SCCP processor. This processor implements SCCPin and SCCPout, NSPin and NSPout and SCCP manager functions. Two SCCP processors are provided for each installation for resilience.

A fourth shelf of the SAP contains a management processor (MP) which supports management functions for the SRF's and for C7 signalling and controls external interfaces to the ISAP.

Although in the examples so far described, the interface between the IP and the SCP defined at the INAP/TCAP protocol layers is supported by SCCP/MTP network protocols, the invention may also be implemented using other network-level protocols to support the INAP interactions. For example, Figure 9 is a schematic of a system in which the messages from the SCP to the IP are directed via the SSP using the facility field of the ISUP (integrated services user part) protocol. This is the protocol relating to the implementation of intelligent networks using ISDN technology. As a further alternative, the invention might be implemented using TCP/IP protocols over a link between the SCP and the IP using, for example, ethernet FDDI fibre LAN technology.

TABLE 1

```
DialogueIdExtension ExtensionField ::= {
    type            dialogueIdExtension,
    criticality     abort,
    value           [1] CustomDialogue
    }

dialogueIdExtension     INTEGER ::=13

CustomDialogue          ::= SEQUENCE {
        identifier      [0] DialogueIdentifier,
        data            [1] DialogueData        OPTIONAL,
        }

DialogueIdentifier      ::= SEQUENCE {
        name            [0] DialogueIdName,
        feature         [1] DialogueIdFeatureOPTIONAL,
        version         [2] DialogueIdVersionOPTIONAL
        }

DialogueData                            ::= SEQUENCE
        enhancementType                 [0] Integer,
        enhancement                     [1] ANY DEFINED BY enhancementType
```

TABLE 2

```
dialogueEnhancement    Integer ::= 1

DialogueEnhancement ::= SEQUENCE {
    voiceRecognition   [0] VoiceRecognition OPTIONAL
    }

VoiceRecognition ::= SEQUENCE {
    recogInfo          [0] RecogInfo,
    language           [1] Language DEFAULT english,
    errorTreatment     [2] ErrorTreatment
                           DEFAULT stdErrorAndInfo,
    voiceBack          [3] BOOLEAN DEFAULT TRUE
    }

Language ::= ENUMERATED {
    english(0),
    welsh(1),
    portugese (2)
    french(3),
    german(4),
    spanish(5),
    italian(6)
    }

RecogInfo ::= CHOICE {
    contextInfo        [0] contextInfo,
    vocabulary         [1] Vocabulary
    }

Vocabulary ::= VocabID
```

TABLE 2 cont'd

```
ContextInfo                    ::= SEQUENCE {
        resultMap                  [0] ResultMap,
        resultType                 [1] ResultType DEFAULT numeric
        context                    [2] Context
        }

VocabID                        :: Integer4

Context                        ::= CHOICE {
        yesNoContext               [0] YesNoContext,
        }

YesNoContext                   ::= NULL

ResultMap                      ::= VocabMapID

VocabMapID                     ::= Integer4

ResultType                     ::= ENUMERATED {
        numeric(0),
        text(1)
        }
```

TABLE 3.1

3.2.1. RecordMessage enhancement

```
DialogueData                ::= SEQUENCE {
        enhancementType                 [0] RecordMessageEnhancement,
        enhancement                     [1] RecordEnhancement
        }

RecordEnhancement           ::= SEQUENCE {
        recordMedia                     [0] RecordMedia
        }

RecordMedia                 ::= ENUMERATED {
        Speech (0),
        facsimile (1)
        }

RecordMessageEnhancement INTEGER ::=2
```

TABLE 3.2

3.2.2 PlaybackMessage enhancement

```
DialogueData            ::=SEQUENCE {
                enhancementType                 [0] PlaybackMessageEnhancement,
                enhancement                     [1] PlaybackEnhancement
                }

PlaybackEnhancement             ::= SEQUENCE {
                playbackMedia                   [0] PlaybackMedia,
                message                         [1] Message
                }

PlaybackMedia                   ::= ENUMERATED {
                speech (0),
                facsimile (1)
                }

PlaybackMessageEnhancement  INTEGER  ::= 3
```

TABLE 3.3

3.2.3 DeleteMessage enhancement

```
DialogueData            ::= SEQUENCE {
        enhancementType             [0] DeleteMessageEnhancement,
        enhancement                 [1] DeleteEnhancement
        }

DeleteEnhancement       ::= SEQUENCE {
        message         [0] Message
        }

DeleteMessageEnhancement INTEGER  ::= 4
```

3.2.3.1 Description

This enhancement specifies a message which is to be deleted from an MPs storage.

3.2.5 Common Data Types

Message::= INTEGER

3.2.5 Enhancement Constants

```
minNameLen          INTEGER  ::= 1
maxNameLen          INTEGER  ::= 200
minFeatureLen       INTEGER  ::= 1
maxFeatureLen       INTEGER  ::= 200
minVersionLen       INTEGER  ::= 1
maxVersionLen       INTEGER  ::= 200
```

Table 4.1

```
playAnnouncement OPERATION
   ARGUMENT
         SEQUENCE {
           informationToSend {0} CHOICE {
             inbandinfo [0]  IMPLICIT SEQUENCE {
               messageID  [0]  CHOICE {
                  elementaryMessageID  [0] IMPLICIT INTEGER (0..2147483647),
                  text          [1] IMPLICIT SEQUENCE {
                     messageContent [0] IMPLICIT IA5string  (SIZE (??..??),
                     attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL),
                  elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1...??) OF
                     INTEGER (0..2147483647),
                  variableMessage [30] IMPLICIT SEQUENCE {
                     elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
                     variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
                        CHOICE {
                           integer        [0]  IMPLICIT INTEGER (0..2147483647),
                           number         [1]  IMPLICIT OCTET STRING (SIZE (??..??)),
                           time           [2]  IMPLICIT OCTET STRING (SIZE) (2)),
                           date           [3]  IMPLICIT OCTET STRING (SIZE) (3)),
                           price          [4]  IMPLICIT OCTET STRING (SIZE) (4))}}},
               numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
```

TABLE 4.2

```
promptAndCollectUserInformation OPERATION
   ARGUMENT
         SEQUENCE {
            collectedInfo [0] CHOICE {
               collectedDigits [0] IMPLICIT SEQUENCE {
                  minimumNbOfDigits   [0] IMPLICIT INTEGER (1..127) DEFAULT 1,
                  maximumNbOfDigits [1] IMPLICIT INTEGER (1..127),
                  endOfReplyDigit [2] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
                  cancelDigit [3] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
                  startDigit [4] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
                  firstDigitTimeOut  [5] IMPLICIT INTEGER (1..127) OPTIONAL,
                  interDigitTimeOut [6] IMPLICIT INTEGER (1..127) OPTIONAL,
                  errorTreatment [7] IMPLICIT ENUMERATED {
                     stdErrorAndInfo (0),
                     help          (1),
                     repeatPrompt (2)} DEFAULT stdErrorAndInfo,
                  interruptableAnnInd [8] IMPLICIT BOOLEAN DEFAULT TRUE,
                  voiceInformation [9] IMPLICIT BOOLEAN DEFAULT FALSE,
                     voiceBack [10] IMPLICIT BOOLEAN DEFAULT FALSE)},
            disconnectFromIPForbidden [1] IMPLICIT BOOLEAN DEFAULT TRUE,
            informationToSend [2] CHOICE {
               inbandInfo [0] IMPLICIT SEQUENCE {
                  messageID [0] CHOICE {
                     elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
                     text              [1] IMPLICIT SEQUENCE {
                      messageContent [0] IMPLICIT IA5string  (SIZE (??..??)),
                      attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL),
                    elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..??) OF
                      INTEGER (0..2147483647),
                    variableMessage [30] IMPLICIT SEQUENCE {
                      elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
                      variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
                         CHOICE {
                            integer   [0]  IMPLICIT INTEGER (0..2147483647),
                            number    [1]  IMPLICIT OCTET STRING (SIZE (??..??),
                            time      [2]  IMPLICIT OCTET STRING (SIZE (2)),
                            date      [3]  IMPLICIT OCTET STRING (SIZE (3)),
                            price     [4]  IMPLICIT OCTET STRING (SIZE (4)))))},
               numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
               duration  [2] IMPLICIT INTEGER (0..32767) OPTIONAL,
               interval  [3] IMPLICIT INTEGER (0..32767) OPTIONAL),
            tone        [1] IMPLICIT SEQUENCE {
               toneID      [0] IMPLICIT INTEGER (0..2147483647),
               duration    [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL),
            displayInformation [2] IMPLICIT IA5string  (SIZE (??..??))} OPTIONAL,
         extensions [3] IMPLICIT SEQUENCE SIZE (1..??) OF
            SEQUENCE {
               type        INTEGER,
               criticality    ENUMERATED {
                  ignore         (0),
                  abort          (1)} DEFAULT ignore   ,
               value          [1] ANY DEFINED BY type       } OPTIONAL}
```

TABLE 5.1

```
PlayAnouncementArg                    ::= SEQUENCE {
    informationToSend                 [0] InformationToSend,
    disconnectFromIPForbidden         [1] BOOLEAM    DEFAULT TRUE,
    requestAnnouncementComplete       [2] BOOLEAM    DEFAULT TRUE,
    extensions                            [3] SEQUENCE SIZE(1..numOfExtensions) of Extension Field
                                                                                    OPTIONAL
```

TABLE 5.2

```
PromptAndCollectUserInformationArg  ::= SEQUENCE {
        collectedInfo                [0] CollectedInfo,
        disconnectFromIPForbidden    [1] BOOLEAN        DEFAULT TRUE,
        informationToSend            [2] InformationToSend        OPTIONAL,
        extensions                    [3] SEQUENCE SIZE(1..numOfExtensions) of ExtensionsField
                                                                   OPTIONAL
```

TABLE 6

Parameters

- informationToSend:
  This parameter indicates an announcement, a tone or display information to be sent to the end user by the SRF.
  - inbandInfo:
    This parameter specifies the inband information to be sent
    - messageID
      This parameter indicates the message(s) to be sent, this can be one of the following:
      - elementaryMessageID:
        This parameter indicates single announcement.
      - text:
        This parameter indicates a text to be sent. The text shall be transformed to inband information (speech) by the SRF. The attributes of text may consist of items such as language.
    - elementaryMessageIDs:
      this parameter specifies a sequence of announcements.
    - variableMessage:
      This specifies an announcement with one or more variable parts
  - numberOfRepetitions:
    This parameter indicates the maximum number of times the message shall be sent to the end-user.
  - duration :
    This parameter indicates the maximum time duration in seconds that the message shall be played/repeated. ZERO indicates endless repetition.
  - interval:
    This parameter indicates the time interval in seconds between repetitions, ie the time between the end of the announcement and the start of the next repetition. This parameter can only be used when the number of repetitions is greater than one.
- tone:
  This parameter specifies a tone to be sent to the end-user.
  - toneID:
    This parameter indicates the tone to be sent.
  - duration:
    This parameter indicates the time duration in seconds of the tone to be sent.
    ZERO indicates infinite duration.
- displayInformation:
  This parameter indicates a text string to be sent to the end-user. This information can not be received by a PSTN end-user.

TABLE 6 Cont'd

NOTE: As the current signalling systems (DSS1/ISUP) do not provide an indication whether or not information can be displayed by the user's terminal, in case of user interaction with an ISDN user two consecutive PlayAnnouncement operations are sent. The first contains the display information, the second contains the inband information to be sent to the user. Since the execution of he display information by the SRF should take a limited amount of time, the inband information will be immediately sent by the SRF to the user, in sequence with the display information.

- disconnectFromIPForbidden:
  This parameter indicates whether or not the SRF should be disconnected from the user when all information has been sent.

- requestAnnouncementComplete:
  This parameter indicated whether or not a SpecializedResourceReport shall be sent to the SCF when all information has been sent.

Table 7

Parameters

- collectdInfo:
  - collectedDigits:
    - minimumNbOfDigits:
      If this parameter is missing, the default value is defined to be 1. The "minimumNbOfDigits" specifies the minimum number of valid digits to be collected.
  - maximumNbOfDigits:
    This parameter should always be present and specifies the maximum number of valid digits to be cancelled. The following applies:
    "maximumNbOfDigits"≥"minimumNbOfDigits".
  - endOfReplyDigit:
    This parameter indicates the digit used to signal the end of input.
    In case the "maximumNbOfDigits" = "minimumNbOfDigits", the "endOfReply-Digit" (could be present but) has no further meaning. This parameter can be one or two digits.

    In case the "maximumNbOfDigits"."minimumNbOfDigits" the following applies:

    If "endOfReplyDigit" is not present, the end of input is indicated:
    - when the inter-digit timer expires; or
    - when the number of valid digits received equals the maximumNbOfDigits".

    If "endOfReplyDigit" is present, the end of input is indicated:
    - when the inter-digit timer expires; or
    - when the end of reply digit is received; or
    - when the number of valid digits received equals the "maximumNbOfDigits".
  - When the end of input is attained, the collected digits are send from SRF to the SCF. In the case the number of valid digits received is less than "minimumNbOfDigits" when the inter-digit timer expires or when the end of reply digit is received, the input is specified as being erroneous.
  - cancelDigit:
    If this parameter is present, the cancel digit can be entered by the user to request a possible retry. All digits already received by the SRF are discarded and the same PromptAndCollectUserInformation procedure is performed again, thus eg the same announcement to request user information is given to the user and information is collected. This parameter can be one or two digits. If this parameter is not present, the user is not able to request a possible retry.
  - startDigit:
    If this parameter is present, the start digit indicates the start of the valid digits to be collected. The digits that are received by the SRF before this start digit is received, are discarded and are not considered to be valid. This parameter can be one or two digits. If this parameter is not present, all received digits are considered to be valid.

TABLE 7 cont'd

- firstDigitTimeOut:
If this parameter is present, the first digit should be received by the SRF before the first-digit timer expiration. In case the first digit is not received before first-digit timer expiration, the input is regarded to be erroneous. After receipt of the first valid or invalid input digit, the corresponding first-digit timer is stopped.

If this parameter is not present, then the SRF uses a default value for the first-digit timer in which the first valid or invalid input digit is received.

If "startDigit" is present, the first-digit timer is stopped after the start digit is received.

- interDigitTimeOut:
If this parameter is present any subsequent valid or invalid digit, should be received by the SRF before the inter-digit timer expires. As result the inter-digit timer is reset and restarted.

In case a subsequent valid or invalid digit is not received before the inter-digit timer expires and the number of received valid digits is less than the "minimumNbOfDigits", the input is regarded to be unsuccessful.

In case a subsequent valid or invalid digit is not received before the inter-digit timer expires and the number of received valid digits is greater than the "minimumNbOfDigits", and less than or equal to the "maximumNbOfDigits", the input is regarded to be successful.

If the "interDigitTimeOut" is not present, then the SRF uses a default value for the inter-digit time.

- errorTreatment:
This optional parameter defines what specific action should be taken by the SRF in the event of error conditions occurring. The default value is stdErrorAndInfo.

- interruptableAnnInd:
this parameter is optional, where the default value is specified being TRUE.

If this parameter is TRUE, the announcement is interrupted after the first valid or invalid digit is received by the SRF. If the announcement is interrupted, a possible start-digit timer will not apply anymore. However, if the announcement has not been interrupted, a possible start-digit timer is started after the announcement has been finished.

If this parameter is present and explicitly set to FALSE, the announcement will not be interrupted after the first digit is received by the SRF. The received digits during the announcement are discarded and considered to be invalid. All other specified parameters ("minimumNbOfDigits", "maximumNbOfDigits", "endOf-ReplyDigit", etc.) do not apply before the announcement has been finished. The possible start-digit timer is started after the announcement has been finished.

- voiceInformation:
This parameter is optional, where the default value is specified being FALSE. If the "voiceInformation" parameter is FALSE, all valid or invalid digits are entered by DTMF.

If this parameter is present and explicitly set to TRUE, calling user is required to provide all valid or invalid information by speech. The SRF will perform voice recognition and translation of the provided information into digits. A possible end of reply digit will also have to be provided by speech.

TABLE 7 cont'd

- voiceBack:
  This parameter is optional, where the default value is specified being FALSE. If the "voiceBack" parameter is FALSE, no voice back information is given by the SRF.

  If this parameter is present and explicitly set to TRUE, the valid input digits received by the SRF will be announced back to the calling user immediately after the end of input is received. The invalid input digits will not be announced back to the calling user. A possible end of reply digit is not voiced back.

- disconnectFormIPForbidden:
  This parameter indicates whether the SRF should initiate disconnection to the SSF/CCF after the interaction has been completed. If the parameter is not present or set to TRUE, the SRF shall not initiate disconnection.

- informationToSend:
  This parameter indicates an announcement, a tone or display information to be sent to the end user by the SRF.

  - inbandfo:
    This parameter specifies the inband information to be sent.

    - messageID:
      This parameter indicates the message(s) to be sent, this can be one of the following:

      - elementaryMessageID:
        This parameter indicates a single announcement.

      - text:
        This parameter indicates a text to be sent. The text shall be transformed to inband information (speech) by the SRF. The attributes of text may consist of items such as language.

      - elementaryMessageIDs:
        This parameter specifies a sequence of announcements.

      - variableMessage:
        This parameter specifies an announcement with one or more variable parts.

    - numberOfRepetitions:
      This parameter indicates the maximum number of times the message shall be sent to the end-user.

    - duration:
      This parameter indicates the maximum time duration in seconds that the message shall be played/repeated. ZERO indicates endless repetition.

    - interval:
      This parameter indicates the time interval in seconds between repetitions, ie the time between the end of the announcement and the start of the next repetition. This parameter can only be used when the number of repetitions is greater than one.

  - tone:
    This parameter specifies a tone to be sent to the end-user.

    - toneID:
      This parameter indicates the tone to be sent.

TABLE 8

- voiceRecognition:
  This parameter specifies the voice recognition to be conducted.

  - recogInfo.
    This parameter specifies the information used to drive the recognition.

    - contextInfo:
      The parameter defines the recognition in terms of a context ie the type, presentation and format of the information to be recognised.

      - ResultMap
        This specifies the file which contains the yes and no vocabulary as well as their mapping to numbers. Even if the language is not English the words given in the vocabulary must be returned if a text yesNoResultType is specified.

        The result map can be used as a mask to the vocabulary indicated by the Context. For example the yesNoContext infers that a yes and no vocabulary should be used in the recognition. This may however contain additional words such as yes-please and no-thankyou. The result map explicitly enumerates the words that are acceptable ie:

        YES, 1
        NO, 2

        Hence if yes-please is recognised it would be treated as an error since it is not in the result map. Of course the full vocabulary can be put in the result map to allow such variants through.

      - ResultType
        This specifies the form in which the digits within the receivedInformationArg is formatted. If text then digits is set to contain the IA5 characters of the recognised word. If numeric then digits is set to the number given in the map. If this parameter is missing then the default is numeric.

      - context:
        This parameter specifies what recognition to conduct.

        - yesNoContext
          This provide details of how Yes and No should be collected. For yes and no there is no information currently identified hence the type is NULL

    - vocabulary:
      This parameter specifies a vocabulary file to be used in the recognition. The context (ie language etc) is derived from this vocabulary file. The receivedInformationArg will encode the recognised word as IA5.

- language:
  This is the natural language in which the recognition is be conducted. If the parameter is missing the default language is English.

TABLE 8 cont'd

- errorTreatment:
  This optional parameter defines the specific action to be taken by the SRF in the event of error conditions occurring. The default value is stdErrorAndInfo.

- voiceBack:
  This parameter is optional and the default is FALSE. If the voiceback parameter is set to false then no voice back information is given by the SRF.
  If the parameter is true then the valid recognised word is announced back to the calling user. Invalid words will not be announced back.

CLAIMS

1. A method of operating an intelligent peripheral (IP) under the control of a service control point (SCP) in a telecommunications network, the method comprising:

a) passing control messages conforming to a standard protocol from the service control point to the intelligent peripheral, some of the said control messages including an extension field identifying the messages as relating to an additional function not provided for in the standard protocol;

b) executing on the intelligent peripheral, in response to at least some of the messages, predetermined functions which are provided for in the said standard protocol; and

c) executing on the intelligent peripheral, in response to a message including the said extension field, a routine which implements the additional function.

2. A method according to claim 1, in which the standard protocol is the Intelligent Network Application Protocol (INAP) and the said field identifying the message as relating to an additional function is the extension field of the INAP intelligent peripheral-to-SCP application context.

3. A method according to claim 2, in which each of the said messages including the said extension field further comprises an identifier field for identifying the message as relating to one of a plurality of different additional functions.

4. A method according to claim 2 or 3, in which the said message including the extension field further comprises an additional function data field, data contained in the said additional function data field being passed to the additional function executed on the intelligent peripheral.

5. A method according to any one of the preceding claims, in which any data collected by an additional function is returned in the form of an argument of one of the predetermined functions provided for in the said standard protocol.

6. A method according to claim 5 when directly or indirectly dependent on claim 2, in which the collected data is returned as the argument of a PromptAndCollectUserInformation function.

7. A method according to any one of the preceding claims, in which the step of executing the additional function includes carrying out an in-band interaction with a user connected to the intelligent peripheral via an SSP.

8. A method according to claim 7, in which the additional function includes a voice recognition function.

9. A method according to claim 8, further comprising returning from the intelligent peripheral to the SCP data obtained by the voice recognition function.

10. An intelligent peripheral (IP) for use in a telecommunications network under the control of a service control point (SCP), the intelligent peripheral comprising:

a) means for reading messages in a standard protocol received from the SCP;

b) means for executing in response to the control messages predetermined functions provided for in the said standard protocol;

c) means for recognising in the said messages an extension field identifying a message as relating to an additional function not provided for in the said standard protocol; and

d) means responsive to the said means for recognising for executing the said additional function.

11. A service control point for use in a telecommunications network which includes an intelligent peripheral (IP) which is arranged to function under the control of the service control point, the service control point comprising:

a) a control processor which is arranged to execute a service control function;

b) a signalling interface which is connected to the control processor and which is arranged to generate messages in a standard protocol for transmission to an intelligent peripheral, the said signalling interface including means for writing in some only of the said messages an extension field which identifies a message as relating to an additional function which is not a function provided for in the said standard protocol.

12. A telecommunications network including an intelligent peripheral according to claim 10, and a service control point according to claim 11.

13. A method of operating an intelligent peripheral (IP) under the control of a service control point (SCP) in a telecommunications intelligent network, including passing control messages conforming to a standard protocol from the service control point to the intelligent peripheral, and executing on the intelligent peripheral in response to at least some of the control messages predetermined functions provided for in the said standard protocol,

characterised in that others of the control messages conforming to the standard protocol include a field identifying the messages as relating to an additional function not provided for in the standard protocol , and by executing on the intelligent peripheral in response to a message including the said field a routine implementing the additional function.