PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :		(11) International Publication Number:
H04Q 3/545, G06F 9/46	A1	WO 96/09730
		(43) International Publication Date: 28 March 1996 (28.03.96)

(21) International Application Number: PCT/SE95/01028

(22) International Filing Date: 12 September 1995 (12.09.95)

(30) Priority Data:
9403131-7 19 September 1994 (19.09.94) SE

(71) Applicant: TELEFONAKTIEBOLAGET LM ERICSSON [SE/SE]; S-126 25 Stockholm (SE).

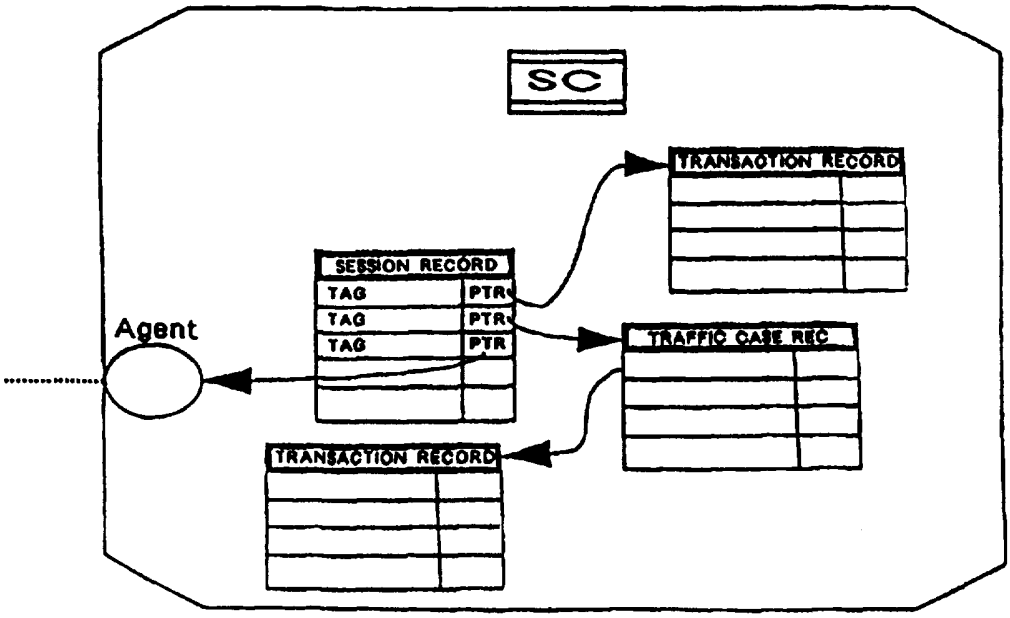
(72) Inventors: KILHAGE, Mikael, Per, Erik; Ulvsbyn 17, S-655 93 Karlstad (SE). STRAND, Jan, Erik; Värmlandsgatan 9, S-652 22 Karlstad (SE).

(74) Agents: BJELLMAN, Lennart et al.; Dr. Ludwig Brann Patentbyrå AB, P.O. Box 1344, S-751 43 Uppsala (SE).

(81) Designated States: AU, CA, CN, FI, JP, KR, MX, NO, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: A FLEXIBLE CALL RECORD MECHANISM



(57) Abstract

The present invention demonstrates a method and a system for a flexible record mechanism in a telecommunication system, preferably by way of software, which makes it possible to extend said system with new services and data without effecting an already existing main operating software of the system using the half-call principle, by storing the data in the executing session call processing by means of memory pointers (PTR) in records associated with each executed session, said pointers being further combined with a tag element (TAG) by means of which locally stored data will be uniquely identified and may during the duration of a session selectively be called and stored in an external data base for subsequent processing.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria
AU	Australia
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CS	Czechoslovakia
CZ	Czech Republic
DE	Germany
DK	Denmark
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgystan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakhstan
LI	Liechtenstein
LK	Sri Lanka
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SI	Slovenia
SK	Slovakia
SN	Senegal
TD	Chad
TG	Togo
TJ	Tajikistan
TT	Trinidad and Tobago
UA	Ukraine
US	United States of America
UZ	Uzbekistan
VN	Viet Nam

A FLEXIBLE CALL RECORD MECHANISM

TECHNICAL FIELD

The present invention refers to a record mechanism in a telecommunication system and particularly to a data structure enabling easy recording of necessary data in a traffic controlling process for a telecommunication application.

BACKGROUND OF THE INVENTION

During processing of a call in a telecommunication system a large amount of data needs to be handled or collected. Such call related data differs a lot between calls depending on what kind of services are utilized in the specific call, which protocols are used to communicate with the surrounding networks etc. Data contains information useful for different kinds of users of the telecommunication system. One network/service provider may want to create billing records while another wants to create statistics of different kinds. As the vendor wants to be independent of which data the user wants to use and still be able to add new data together with a new service without having to change already existing software, this call related data record have to be handled in a new efficient way.

There exist a number of possible solutions to handle the call related data. One obvious way is to use a conventional database to collect the information, which quickly renders into capacity problems. Another solution is choose a declarative solution where a declaration of the contents is made (e.g. compare a record in Pascal). The drawback of a declarative solution as a Pascal record is that it does not present the desired flexibility. Yet another approach is to send around the data between the objects whenever it is needed, which creates duplication of data.

In the state of the art there are found several concepts regarding object oriented software structures for processing in a modern telecommunication system. EP-0 524 089 A1 titled "Structure de logiciel pour système de traitement de données, notamment pour système de télécommunications" describes a logical

structure system for processing data, specifically for telecommunication systems. The structure particularly simplifies the real time communication between the objects according to the CCITT rules X 200. EP-0 524 077 A1 titled "Structure de logiciel pour système de traitement d'informations" describe a structure which hides the hardware and software system features to the application programs.

EP-0 470 415 A2 describes a method to supply a number of application processors in a telephony system access to call related information in a common database. The information is tagged and stored temporarily as a record in the database as long as the communication lasts. The information is particularly directed for being directly viewed on a display terminal for supervision in an operator controlled switching system.

SUMMARY OF THE INVENTION

Therefore there is a demand in a telecommunication system, to create call record mechanism, which makes it possible to extend said system with new services and data without effecting an already existing operating software of a system using the half-call principle.

A first object according to the present invention is, in a session of executing call processing, to perform the locally temporary storing of data by means of memory pointers in records associated with each executed session, the pointer being further combined with a tag element by means of which locally stored data will be uniquely identified and may during the duration of a session selectively be called by a record view object function and stored in a data base for subsequent processing.

A second object according to the present invention is that the specific session is using a session record and a transaction record for storing pointers and tags of objects and data, respectively, in the session and from which records it will be possible to locate all objects and data within the session, if

the tag element is known under which the desired data information is stored.

A third object according to the present invention is that in the call processing is defined a traffic case scope having a similar structure as a session scope and a traffic case record is created and referenced from said sesson to store executing objects of a call and additionally in the traffic case record there is a transaction record storing data belonging to said traffic case.

A fourth object according to the present invention is that services may be changed at any moment by simple modification of a tag-list stored in a local database without interfering with the existing operating overhead system.

A fifth object according to the present invention is that the tag element is being realized by an integer number, preferably as a binary word, uniquely assigned to each executing object or data object used in a session.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention, together with further objects and advantages thereof, may best be understood by making reference to the following description taken together with the accompanying drawings, in which:

Figure 1 is an illustration of a session having a session controller, SC, handling several traffic cases including for each traffic case a respective originating call, OC, in communication with other traffic cases including a respective terminating call, TC;

Figure 2 demonstrates a session controller, SC, using, according to the method and system of the present invention, a session record to store references to executing objects, and a transaction record to

store references to data objects;

Figure 3 demonstrates collections according to the method and system of the present invention to store a traffic case object within an originating call, OC;

Figure 4 is an demonstration of objects controlling the data flow in a session;

Figure 5 demonstrates an example when data for a charging basis is extracted from a session;

Figure 6 shows in a simple example the relation between created managed objects;

Figure 7 shows the complete static view according to the simple example of Fig. 6;

Figure 8 is a simple flow chart of call data collection in a Transaction Record during call processing; and

Figure 9 is a simple flow chart of specification of data to include in an output.

FUNDAMENTALS

To be able to handle the subject of the present application in an efficient way it will be practical to first define a number of technical terms which will be useful throughout the following description.

A common way to structure software in a call processing switching system for telephony is to divide the control of the call into two halves, a Half-Call A and a Half-Call B. The software which controls a Half-Call is executing in a process called a Session. A session can handle one or several Traffic Cases simultaneously (for example in a multi call situation). The Traffic Case defines the functionality and data that handles a call in a Session. Note

also that a three party call is handled by two Traffic Cases in a Session, one for each call leg.

For the sake of simplicity the session is structured in different scopes and therefore is introduced the Session Scope and the Traffic Case Scope. The Session Scope is controlled by the Baseflow Session Controller, SC. The main task for the session controller is to act as a command interpreter against the Access Protocol, ACP and make a service analysis on these commands (Messages). This includes then, for instance, initiating and terminating new Traffic Cases, distributing information from the Access Protocol to the correct Traffic Case, initiating new services, etc.

Every Traffic Case within the Session is controlled by one baseflow. Such a baseflow may be either an Originating Call, OC or a Terminating Call, TC. The main task for this baseflow is to take care of the basic call handling. This includes for example establishing/disconnecting a call (including handling of the Telecommunication Service Protocol, TSP, between the call halves), ordering establishment/disconnection of connections (for example a speech connection), and ordering address information analysis, etc.

To support the different scopes and the control logic operating within those, there is a need of a similar data structure. Thus the data must be structured in a certain way to make it possible to implement and maintain the applications. Correspondingly there exists two different types of objects, which in this description are denoted Executing Objects and Data Objects.

An Executing Object will execute in the session, e. g., control objects, protocol objects, resource objects etc. A pure Data Object will contain data received for example from a Teleservice Protocol Message. It shall also be possible to make an output of this type of data for charging or statistic purposes. The two types of Objects have different semantics and are stored in

different records in the Session. Such a record is referred to as a Session Record and is used to store pointers to protocol objects and resource objects instantiated by control and resource objects within the Session. The Objects stored in a Session Record are common for the whole Session. For storing references to pure Data Objects is used a Transaction Record. In a similar way as the Session Record stores pointers to objects the Transaction Record (also named call record) is used to store pointers to pure Data Objects instantiated by control, protocol, and resource objects within the Session or a Traffic Case executing in the Session.

A users view of a Session Record is referred to a Session Record View and gives the user an interface to the Session Record on a high abstraction level. Similarly a users view of a Transaction Record is referred to as the Transaction Record View and gives the user an interface to the Transaction Record on a high abstraction level.

Finally there is also found a Traffic Case Record which is a record where pointers to Objects belonging to a Traffic Case are stored. Only pointers to Protocol Objects and Resource Objects are stored in this record. For storing pure Data Objects a Transaction Record should be used. A users view of a Traffic Case Record is referred to a Traffic Case Record View and gives the user an interface to the Session Record on a high abstraction level.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

To support the different scopes and corresponding control logic for a call record mechanism in a telecommunication system we need a suitable data structure. Data must be structured to make it possible to implement and maintain the applications. We therefore introduce two different types of objects, the executing objects and data objects, respectively, to keep track in an session. These two terms, which were already defined above, do have different semantics and are stored in different records in the

created session. When storing an object in a collection it is only a question of storing a pointer to the object that is to be stored and consequently no duplication of the object itself is made in such a step. This also implies that for such a pointer storage there is actually no need to know the size of the particular object.

Figure 1 is a generalized view of a session scope, which is controlled by the session controller SC. The session controller is acting as a command interpreter against the access protocol ACP, which is the generic term used for the subscriber or network accesses. As is evident from figure 1, the session contains one or several traffic cases, and here the particular session contains two traffic cases which are both of the OC type (originating call). Each one of the two traffic cases of type OC is established by means of the respective traffic case to another traffic case of type TC (terminating call) through a handling telecommunication service protocol, TSP.

As indicated in figure 2 there is in the session scope a session record, SR, which shall be used for storing a pointer, PTR, to each executing object, for example to a so called session agent. The session record is by means of other pointers the root for the data structure in each session. The data objects of the whole session is found in the transaction record by means of their respective pointers, PTR. Each entry in the session record is having a particular name or key, TAG, which makes it possible to locate any object within the session scope if the particular system operator knows the particular name or TAG.

Figure 3 is a generalized view of a traffic case scope, here containing an originating call type, OC, but a terminating call type, TC, would have the corresponding structure. This scope has to be introduced if the application has a need to execute an arbitrary number of parallel traffic cases in the session. The structure of the traffic case scope is thus similar to that of the session scope. For each traffic case in a session there is

created a traffic case record to store executing objects. Like in the session record is used a name or TAG and a pointer PTR. The traffic case record is accordingly referenced from the session record. To store data objects belonging to the traffic case is consequently used a transaction record, TR, creating a table for the data objects at this traffic case level.

Every user of a session or traffic case record has an own view object through which the stored executing objects or data objects may be accessed.

Figure 4 demonstrates in greater detail the data flow through a session executing an originating call, OC. The data flow starts when some data is received by an access agent or the input agent. The received data is converted to an AXE internal representation. The converted data is then stored in the transaction record, TR. The data object is stored with a tag. The tag is an integer that is reserved for this particular data object. Other users, e.g. an Application Analysis, needing the data object can fetch it from the transaction record by means of the tag and by utilizing a transaction record view object, TR_View. The above example also illustrates when data is sent by the output agent to the other Half-Call via the telecommunication service protocol, TSP. Data is sent in a parameter which besides the data contains the tag which identifies it.

As stated above a data object is stored in the transaction record (a synonym for transaction record is also call record). The transaction record, TR, is as already stated always accessed via a view object. The view object gives the user a high level interface to TR, which will be further described below. Each data object that is stored in the transaction record is semantically identified by a name or key referred to as the TAG. The TAG is an integer, in an exemplifying embodiment a 16 bit word, which has been reserved for one particular data object. By using a dynamic storage such as the transaction record, where the data objects are stored with tags it will be possible to support a very

flexible output mechanism. In other words it will be extremely easy, without influencing the general operation of the telecommunication system, to at any particular time period extract any chosen data objects on demand of the user for a later analysis. A consequence of this is that it will be extremely easy to add additional services into a system operating according to such a structured way of operation.

Assume that the agent receive the parameter "calling party number" on the protocol, ACP. The data will be converted to an AXE internal representation and stored in the TR together with an dedicated tag, "AppCallingPartyNumberTag". Other users of TR that needs the calling party number can then turn to the TR and ask for the data object that is stored with the TAG "AppCalling-PartyNumberTag". An interface Application Platform Tags Interface, ATI, contains the number of tags used by the functions. ATI also contains the rules to follow when new tags are reserved.

As already mentioned the TR is always accessed via a view object. The view object has two main tasks. The first is to present a customized interface towards the TR. Each user of the TR should have a dedicated interface to the contents in the TR. The second task is to act as a handle object towards the TR, the handle ensures that TR is not removed until all handles are deleted.

View objects are also used to access the contents of the other two types of record that exist, the session record and the traffic case record. As mentioned above one task of the view object is to provide the user with a customized interface on a high abstraction level towards a record. The customizing means that the interface gives the users access only to the objects needed to be accessed, which may be only a part of the total contents in a record.

The second major task of the view objects towards the transaction record and the traffic case record is that they act a handles. As long as a record has a handle it can not be deleted. When the

last handle towards a record is removed the record and all its content is also removed from the local memory storage. It is apparent that this creates a very convenient local memory storage management.

The call record output mechanism already mentioned is used to output parts of the content of a transaction record for post-processing. It should be kept in mind that the contents of the session record and a traffic case record and a transaction record are existent only during the duration of that particular session and will disappear when the session is terminated. The output mechanism is built around a number of managed objects containing tag lists. In the operation of a telecommunication system there is for instance a need to collect charging data to be able to correctly bill the different subscribers. In Figure 5 is exemplified what may take place in a session. A control object "Charging" has opened an object Cro_Type. This particular Cro_Type object contains a Tag list, fetched from the data base, denoting the data objects to be extracted from the transaction record. Cro_Type is then ordered to compile a report consisting of the data objects identified by the tag list which is stored in the data base. The control object then uses the Cro_Type interface to order it to collect the data during the existence of the particular session. The data may be packed in a data area which then will be sent to a post-processing node. Consequently a charging basis due to increased services may be changed at any moment by simple modification of the tag list without interfering at all with the existing system having a structure according to the present invention.

The effective result of this is that even if the contents of the different sessions are defined as local data, it is possible to simultaneously make use of desired parts of the content as if it constitutes global data. A difference between local and global data is for example that the latter by necessity has normally to be allocated in predetermined memory locations to be able to be accessed by other users.

In the illustrative embodiment we use three types of managed objects to effectuate the flexible output mechanism described here. They are denoted as CroServiceTemplate, CroType and Cro-CustomerTemplate. The first managed object type, the CroService-Template is used for specification of what data objects are possible to extract for a specific basic or supplementary service. CroServiceTemplate contains one attribute, possible TAGs, denoting which data is possible to extract from the transaction record, TR, for a particular service, for example in this context a "Basic Call" or a "Three Party Call".

The second managed object type is CroType, which is used for specification of a certain output type. Every instance of CroType is connected to one or more instances of CroServiceTemplate. The union of data in these CroServiceTemplates determines what data is possible to output for a specific CroType.

The third and last management object type is CroCustomerTemplate, which is a managed object holding the information of which data to extract for a specific customer from in a specific output type, CroType.

Figure 6 demonstrates a small example having the conditions:

- There are two customers, A and B.
- There are two services, "Basic Call" and "Three Party Call".
- There are two CroTypes, CroType 1 and CroType 2.

Because there are two services we need two CroServiceTemplates:

- CroServiceTemplate Basic Call, containing the Tags 1, 2, 5 and 8.
- CroServiceTemplate Three Party Call, containing the Tags 1, 2, 6 and 9.

This means that for the "Basic Call" we can output the data stored in TR having the Tags 1, 2, 5, and 8, while for the service "Three Party Call" we may output the data stored under the Tags 1, 2, 6, and 9.

We then define two output types, CroType 1 designed such that it will be able to output data related to both services and CroType 2 designed such that it will be able to output data related to the Basic Call. In Figure 6 is visualized the basic structure and the relation between the created managed object.

One CroCustomerTemplate is required for each customer and CroType to make the output mechanism "Call Record Output", CRO, able to perform outputs of all CroTypes to all customers. This results in this example in a total of four CroCustomerTemplates. In Figure 7 is demonstrated the resulting structure. Customer A requires all possible Tags from CroType 1 and Tag no. 1 and 2 from CroType 2 and customer B requires all Tags with lower number than 8 from all CroTypes. We then have a final structure that the output mechanism CRO needs to make a proper distribution. We have specified which data fields all different customers need from all different CroTypes.

A final part of the data flow in Figure 4 describes when the data shall be sent to the other Half-Call. The Half-Calls communicate by means of the Telecommunication Service Protocol, TSP. The TSP carries self-identifying parameters. A parameter contains a data object and is identified by a Tag. The receiver can determine what data is received by looking at the Tag. The Tag which is used to identify a parameter on the TSP is the same Tag used to identify a data stored in TR.

In Figure 8 is summarized by a number of steps in a simple flow chart of a call data collection in a Transaction Record during call processing. Such a processing is started in a step 100. In the first real step 101 of the process a message is received over an external protocol. It is received in a protocol agent within a dynamic process within the system. Next in a step 102 the data is converted from an externa representation to an internal representation. A Data Object is created within the process. This Data Object then contains the internal representation of the received data.

In a third step 103 the Data Object is stored under a unique tag element in a transaction record. During call processing data is fetched in a fourth step 104 from the transaction record using a transaction record view object then utilizing the tag element to get the correct pointer PTR to retrieve the specified data.

When the call ends or when an output of call data is wanted for statistical or charging purposes the function call record output is called in a fifth step 105. This function accesses the database to find out which data to output. As a result the function gets a list of tag elements. The wanted data is collected from TR in step 104 and put into an output buffer. This buffer may then be output to an external media. The data may later be post-processed in order to for instance produce billing information etc.

Finally in Figure 9 is shown by means of three steps a simple flow chart a specification of data to be included in an output. The procedure starts in a step 200. In a step 201 the service provider or any other operator administrating the system decides which data to output for different call types. These different output types are specified in a second step 202 by filling in templates with lists of tags to output. In a final step 203 these templates are stored in the database by for example entering the list of tags by means of a separate terminal and/or a keyboard. These entered list of tags are later accessed during the call processing. Entering of the list of tags will not interfere with the general call processing in the telecommunication system for initiating and terminating traffic cases, distributing information from the access protocol to the correct traffic case, initiating new services, etc., but when entered it will decide which data to be stored in the database for post-processing.

It will be understood by those skilled in the art that various modifications and changes may be made to the present invention without departure from the spirit and scope thereof, which is defined by the appended claims.

CLAIMS

1. A method for a flexible call record mechanism in a telephony or telecommunication system, which makes it possible to extend said system with new services and data without effecting an already existing main operating software of the system using the half-call principle, **characterized in** that at the time of an execution of a session call processing the locally temporary storing of data is performed by means of memory pointers (PTR) in records (SR, TR) associated with each executed session, said pointer (PTR) being further combined with a tag element (TAG) by means of which particular desired locally stored data will be uniquely identified and may during the duration of a session selectively be called by a record view object function and subsequently stored in an external data base for subsequent processing.

2. The method according to claim 1, **characterized in** that the specific session uses a session record (SR) and a transaction record (TR) for storing pointers (PTR) and tags (TAG) of objects and data, respectively, in said session, and from which records it will be possible during the existence of the session to extract any objects or data within the session, if the tag element (TAG) is known under which the desired data information is being stored.

3. The method according to claim 2, **characterized in** that in the call processing is defined a traffic case scope having a similar structure as a session scope and a traffic case record is referenced from said session record (SR) and a traffic case record is created to store executing objects of a call.

4. The method according to claim 2, **characterized in** that in said traffic case record a transaction record (TR) stores data belonging to said traffic case.

5. The method according to claim 4, **characterized in** that services may be changed at any moment by simple modification of

a tag-list stored in a local database without interfering with the existing operating overhead system.

6. The method according to claim 5, **characterized in** that said tag element (TAG) is realized by an integer number, preferably as a binary word, uniquely assigned to each executing object or data object to be stored.

7. A call record mechanism for a telephony or telecommunication system, which makes it possible to extend said system with new services and data without effecting an already existing main operating software of the system using the half-call principle, **characterized in** that by means of a memory pointer (PTR) the locally temporary session objects or data are stored in records associated with each executed session in the executing call processing, said pointer (PTR) being further combined with a tag element (TAG) by means of which particular desired temporally locally stored data will be uniquely identified and may during the duration of a session selectively be called by a record view object function and subsequently stored in an external data base for subsequent processing.

8. The system according to claim 7, **characterized in** that the specific session uses a session record (SR) and a transaction record (TR) for storing pointers (PTR) and tags (TAG) of objects and data, respectively, in said session and from which records during the existence of the session it will be possible to extract any objects and data within said session, if the tag element (TAG) is known under which the desired data information is being stored.

9. The system according to claim 8, **characterized in** that a traffic case scope has a similar structure as a session scope and a traffic case record is referenced from said session record and said traffic case record is created to store executing objects of a call.

10. The system according to claim 9, **characterized in** that in said traffic case record a transaction record (TR) stores data belonging to said traffic case.

11. The system according to claim 10, **characterized in** that services may be changed at any moment by simple modification of a tag-list stored in a local database without interfering with the existing operating overhead system.

12. The system according to claim 11, **characterized in** that said tag element (TAG) is realized by an integer number, preferably as a binary word, uniquely assigned to each executing object or data object to be stored.

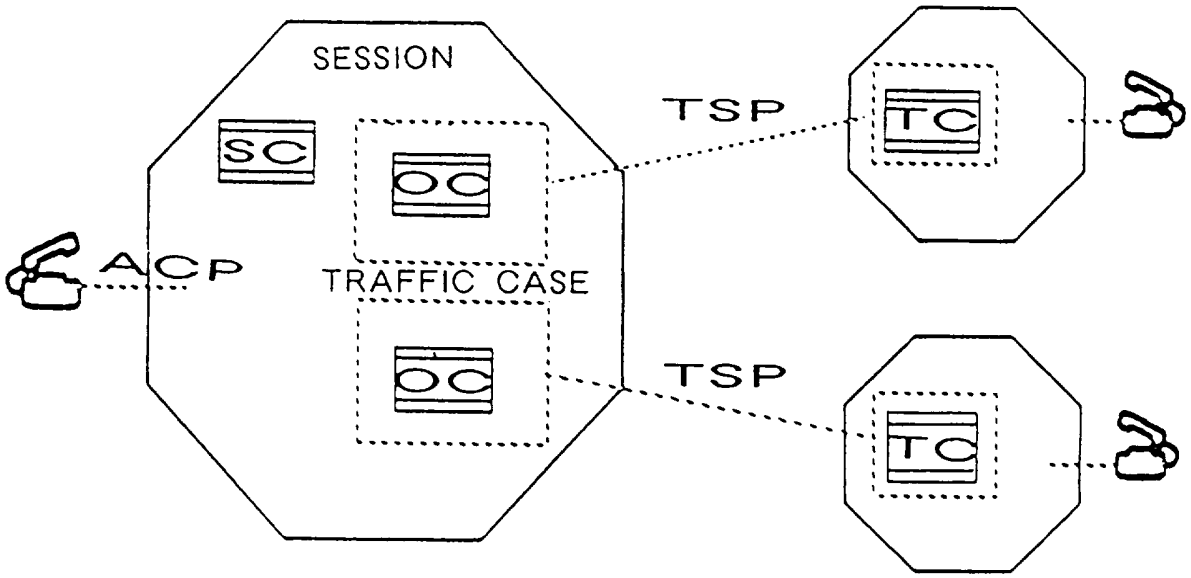

SESSION
SC
OC
TRAFFIC CASE
OC
ACP
TSP
TC
TSP
TC

Fig. 1

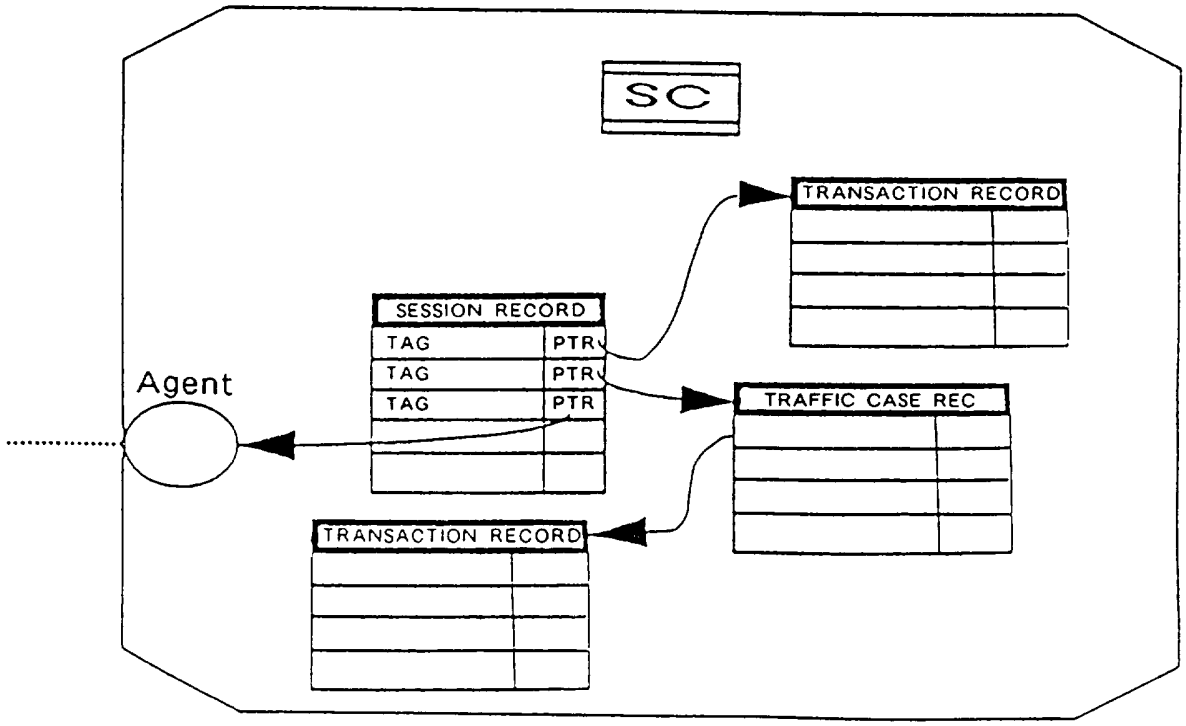

SC
TRANSACTION RECORD
SESSION RECORD
TAG
PTR
TAG
PTR
TAG
PTR
Agent
TRAFFIC CASE REC
TRANSACTION RECORD

Fig. 2

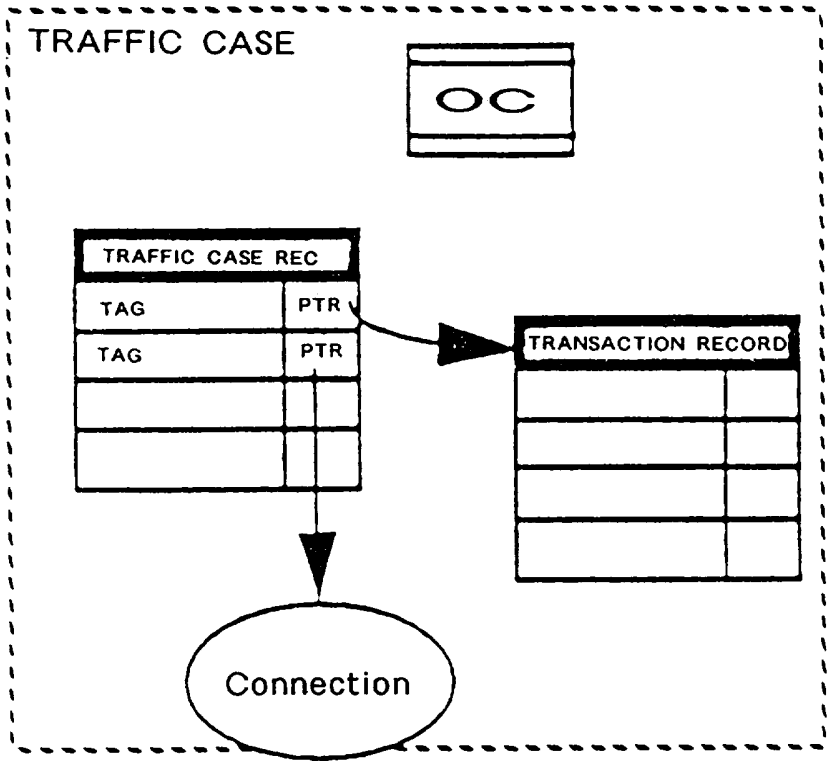

TRAFFIC CASE
OC
TRAFFIC CASE REC
TAG
PTR
TAG
PTR
TRANSACTION RECORD
Connection

Fig. 3

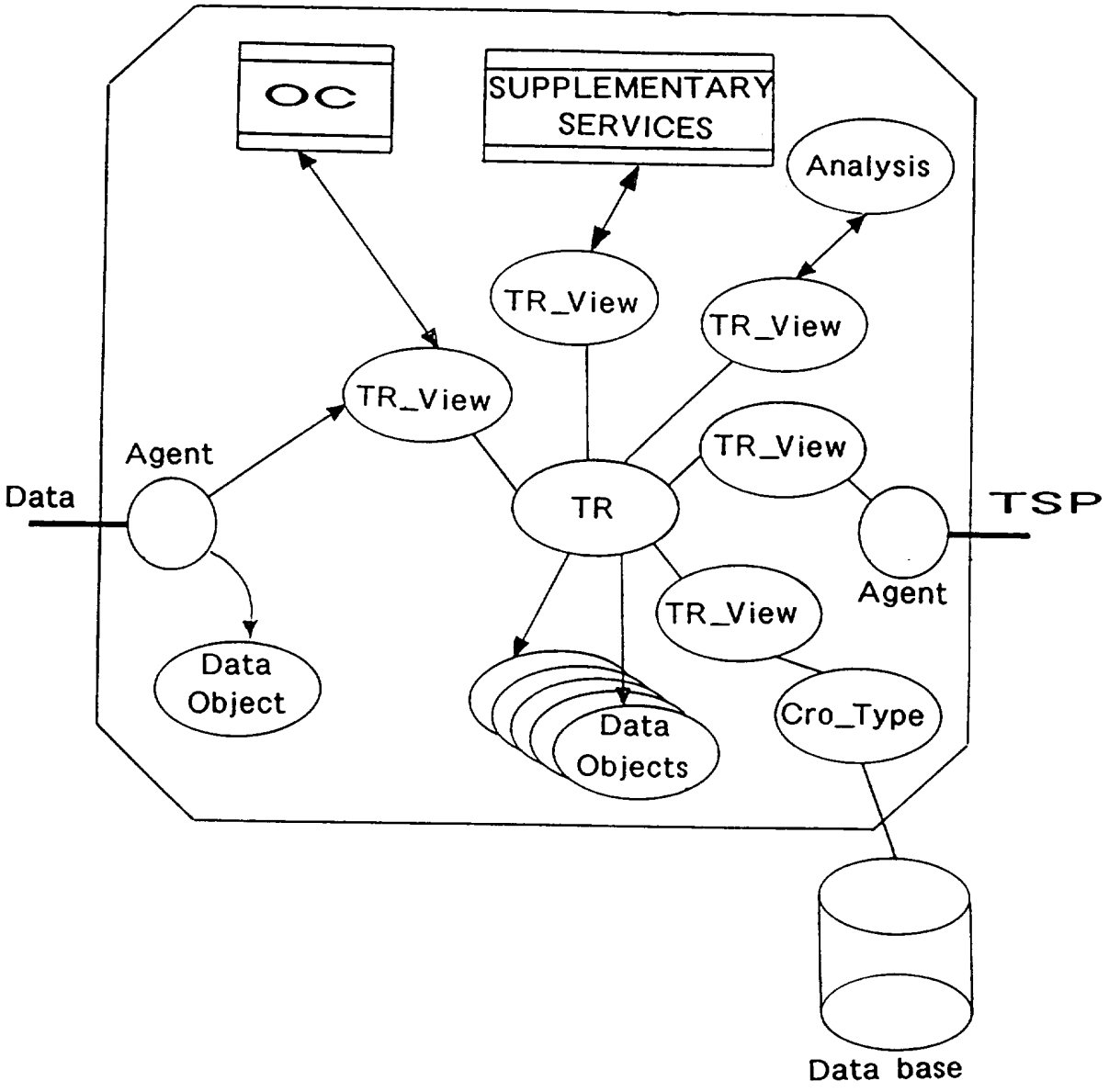

OC
SUPPLEMENTARY SERVICES
Analysis
TR_View
TR_View
TR_View
TR_View
TR_View
Agent
Data
TR
TSP
Agent
Data Object
Data Objects
Cro_Type
Data base

Fig. 4

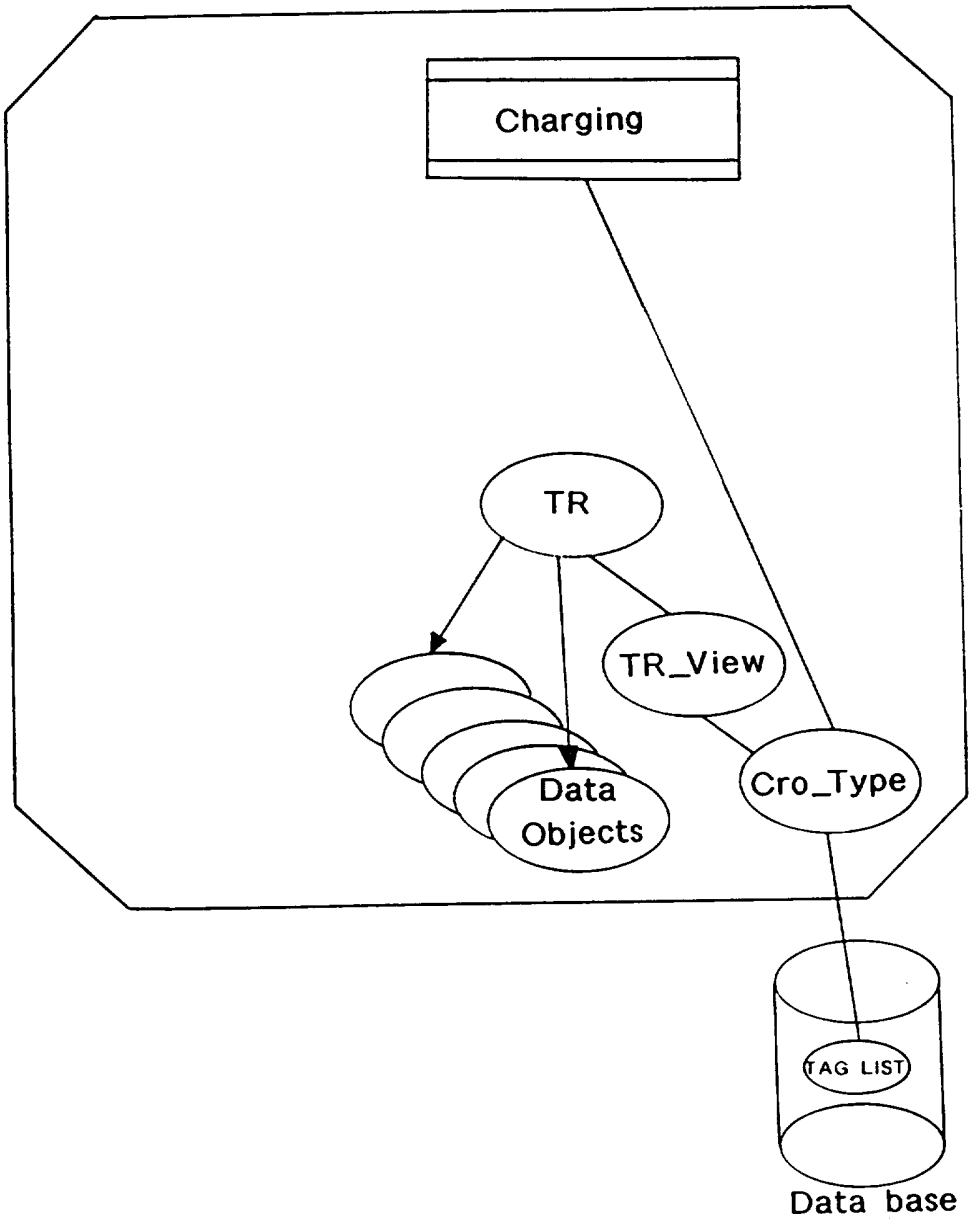


Fig. 5

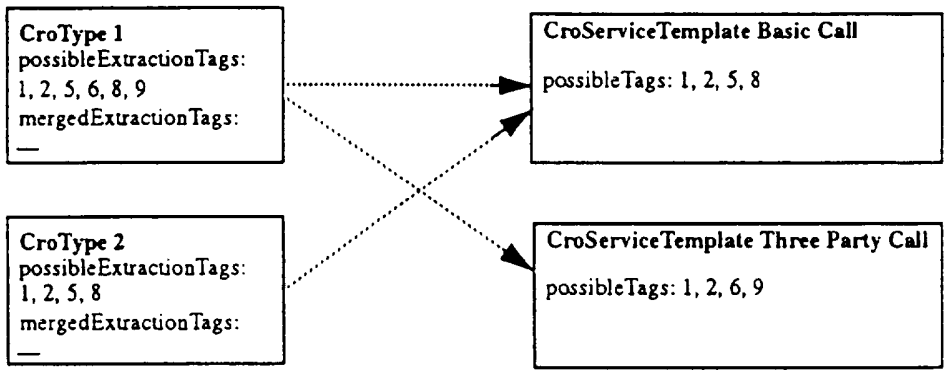

CroType 1
possibleExtractionTags:
1, 2, 5, 6, 8, 9
mergedExtractionTags:
—
CroType 2
possibleExtractionTags:
1, 2, 5, 8
mergedExtractionTags:
—
CroServiceTemplate Basic Call
possibleTags: 1, 2, 5, 8
CroServiceTemplate Three Party Call
possibleTags: 1, 2, 6, 9

Fig. 6

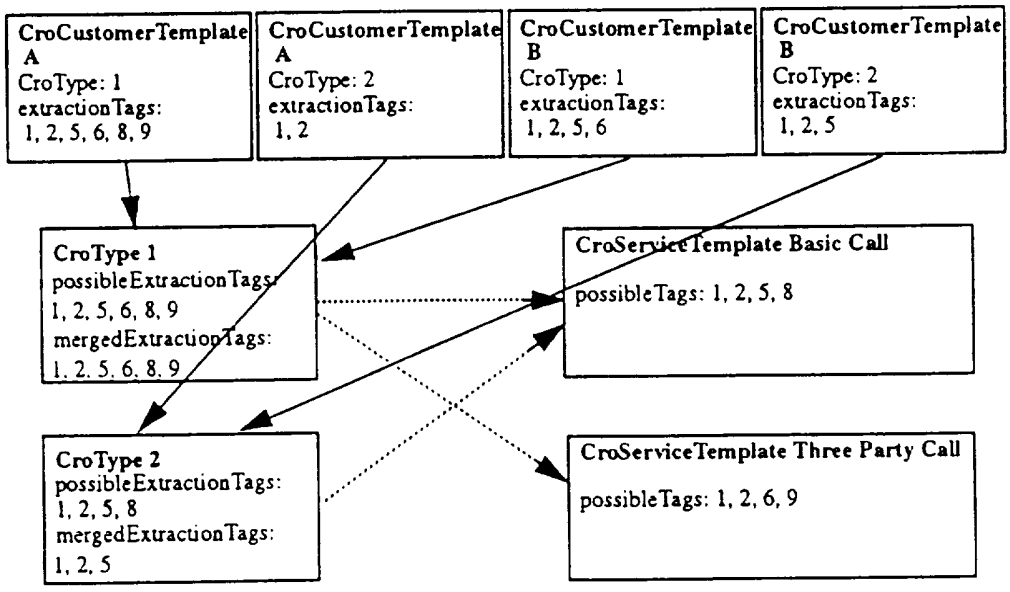

CroCustomerTemplate A
CroType: 1
extractionTags:
1, 2, 5, 6, 8, 9
CroCustomerTemplate A
CroType: 2
extractionTags:
1, 2
CroCustomerTemplate B
CroType: 1
extractionTags:
1, 2, 5, 6
CroCustomerTemplate B
CroType: 2
extractionTags:
1, 2, 5
CroType 1
possibleExtractionTags:
1, 2, 5, 6, 8, 9
mergedExtractionTags:
1, 2, 5, 6, 8, 9
CroType 2
possibleExtractionTags:
1, 2, 5, 8
mergedExtractionTags:
1, 2, 5
CroServiceTemplate Basic Call
possibleTags: 1, 2, 5, 8
CroServiceTemplate Three Party Call
possibleTags: 1, 2, 6, 9

Fig. 7

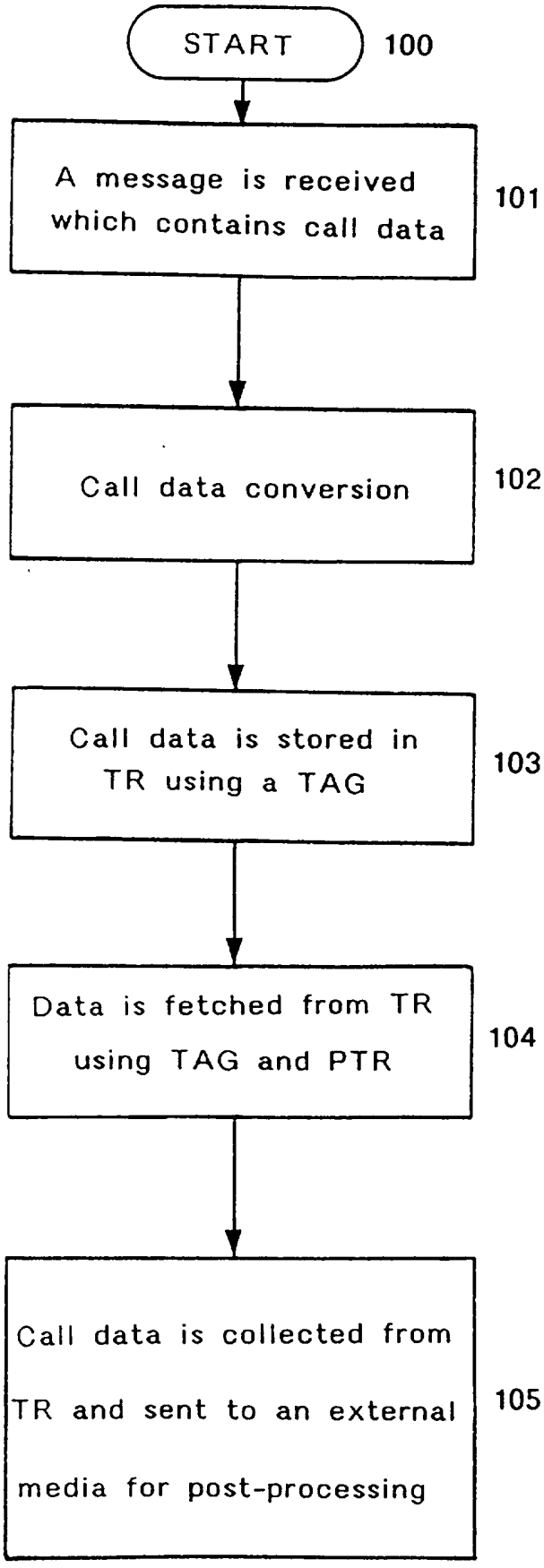

START
100
A message is received which contains call data
101
Call data conversion
102
Call data is stored in TR using a TAG
103
Data is fetched from TR using TAG and PTR
104
Call data is collected from TR and sent to an external media for post-processing
105

Fig. 8

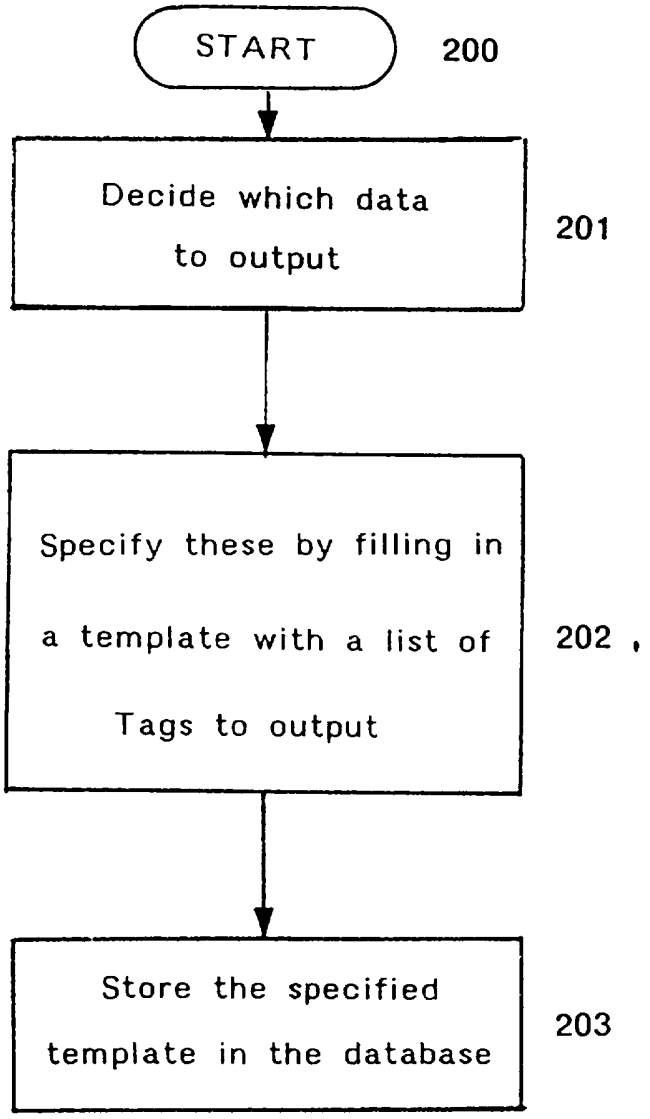

START
200
Decide which data to output
201
Specify these by filling in a template with a list of Tags to output
202
Store the specified template in the database
203

Fig. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 95/01028

A. CLASSIFICATION OF SUBJECT MATTER

IPC6: H04Q 3/545, G06F 9/46
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC6: G06F, H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

DIALOG: WPI, CLAIMS, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5218632 A (ANNA M. COOL), 8 June 1993 (08.06.93), column 2, line 41 - column 3, line 50; column 11, line 1 - line 60, figure 10 --	1-12
A	EP 0470415 A2 (ROLM SYSTEMS), 12 February 1992 (12.02.92), see whole document, cited in the application --	1-12
A	EP 0524089 A1 (ALCATEL N.V.), 20 January 1993 (20.01.93), see whloe document, cited in the application --	1-12

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" erlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 February 1996	22 -02- 1996
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Göran Magnusson Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 95/01028

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0520477 A1 (ALCATEL N.V.), 20 January 1993 (20.01.93), see whole document, cited in the application -- --------	1-12

INTERNATIONAL SEARCH REPORT

Information on patent family members

05/02/96

International application No.
PCT/SE 95/01028

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US-A- 5218632	08/06/93	AU-B- 658102	30/03/95
		AU-A- 2779392	21/05/93
		GB-A,B- 2274045	06/07/94
		GB-D- 9404130	00/00/00
		JP-T- 7500464	12/01/95
		SE-A,D- 9401229	09/05/94
		WO-A- 9308661	29/04/93
EP-A2- 0470415	12/02/92	US-A- 5181239	19/01/93
EP-A1- 0524089	20/01/93	CA-A- 2073902	17/01/93
		FR-A,B- 2679350	22/01/93
		JP-A- 5204853	13/08/93
EP-A1- 0520477	20/01/93	CA-A,C- 2072547	28/12/92
		JP-A- 5186478	27/07/93
		US-A- 5103032	07/04/92