INVOKING COMPUTATIONAL OBJECTS ON MOBILE DEVICES*

Axel Küpper[1], Claudia Linnhoff-Popien[2]

[1] Department of Computer Science, Informatik 4, RWTH Aachen
[2] Institute for Computer Science, Ludwig Maximilians University Munich

mail@axel-kuepper.de, linnhoff@informatik.uni-muenchen.de

Abstract: Future nomadic communication will be enabled by huge TINA-like systems with millions of computational objects in it interacting with each other for the purposes of service provision and mobility management. In a very high degree, the nomadic customer will stay within a mobile cellular network and thus, will be connected to the fixed network part over a wireless link. However, up to now distributed platforms do not provide for any way at all to perform a binding to computational objects arranged in mobile devices.

This paper presents an approach for realising CORBA-compliant wireless distributed systems that can be embedded in mobile cellular networks like today's GSM or the future UMTS. The aspects of terminal mobility are hidden from the objects, and the existing infrastructure of these systems can be integrated without making any modifications.

Keywords: CORBA, mobile cellular networks, nomadic communication, IN

1 INTRODUCTION

The technological progress in the telecommunication domain as well as deregulation efforts of the telecommunication markets currently taking place in many countries will lead to a new kind of user: the *nomadic customer*. According to [1], a nomadic customer is a user who may be mobile and who may, at different times, be communicating using wireless or wired media, with different characteristics, from different (local or remote) locations, using communication or computing devices of different types and capabilities. This implies that the nomadic customer wants to make requests according to his personal service profile and that he wants to be reachable

* This work has partially been funded by the Deutsche Forschungsgemeinschaft (grant no. Sp.230/12-2)

L. Kutvonen et al. (eds.), *Distributed Applications and Interoperable Systems II*

under the same number, independently from the terminal with which he is currently registered and to which provider he is currently connected.

Many research and standardisation groups are working on architectures for providing the required service capabilities and mobility management functions. The *Object Management Group* (OMG) has launched a dedicated telecommunication domain task force to adapt its middleware platform, the *Common Object Request Broker Architecture* (CORBA), to the needs of telecommunication networks [2,3]. These activities can be seen as a first step towards the introduction of the *Telecommunication Information Networking Architecture* (TINA), a framework for service provision and network management [4].

TINA-compliant telecommunication networks can be seen as a huge, global distributed system with millions of interacting *Computational Objects* (CO). COs are running on top of a middleware kernel, like CORBA's *Object Request Broker* (ORB), that provides different kinds of transparency, for example *location* and *migration transparency* [5]. A CO is a piece of object-oriented software. COs are used to provide an abstraction of resources, to realise services, or to represent the customer in the network domain. Their capability to migrate between different hosts makes them attractive for realising nomadic communication. The nomadic user will frequently be connected to the network over a wireless link of a mobile cellular system like today's *Global System for Mobile Communication* (GSM) or its successor, the *Universal Mobile Telecommunication System* (UMTS). Thus, the future range of mobile terminals (MT), be it a simple mobile phone or a very sophisticated mobile computing device, needs a runtime environment for COs. This raises the question how to appropriately address these COs, which are affected from terminal mobility. The OMG has recently published a Request for Information to prepare standardisation activities in this field [2]. The Eurescom project P608, a TINA auxiliary project aiming at investigating the deployment of TINA in mobile cellular systems, has also declared this matter as an unsolved question [6].

This paper presents an approach that enables the addressing and binding to COs over the radio interface of mobile cellular networks. This approach works in parallel with the existing infrastructure installed for mobility management and does not require any modifications of this infrastructure. Furthermore, aspects related to terminal mobility are hidden from COs, that is, it makes no difference for a CO whether it runs on an MT or on a wired device, or whether it invokes a CO located in an MT or in a wired terminal.

2 MOBILITY MANAGEMENT IN CELLULAR NETWORKS

The appearance and development of cellular mobile networks is mainly influenced by today's GSM system and its successor, UMTS. The term 'cellular' indicates that - due to frequency reuse - a provider's catchment area is subdivided into many cells each of which covered by a base station.

2.1 Mobility Management

In a considerable degree, the cell structure influences the measures which have to be taken for guaranteeing that users can be located within a cellular system. GSM as well as UMTS require a sophisticated mobility management architecture to be installed. Mobility management comprises all functions necessary to manage user reachability and to guarantee unlimited access to the mobile network. These procedures are depicted in figure 1.

To guarantee terminal mobility, a *handover* mechanism has to be installed. A handover is the process of changing a mobile user's radio channels and fixed network connections during an active call without any affect on the current state of the call. Typically, a handover occurs when a user with an active terminal, e.g. during a telephone call, crosses cell boundaries leaving the coverage area of a base station, and entering the coverage area of another. However, a handover can also be triggered by a radio related event, be required by a user service profile, or through capacity or network management issues.

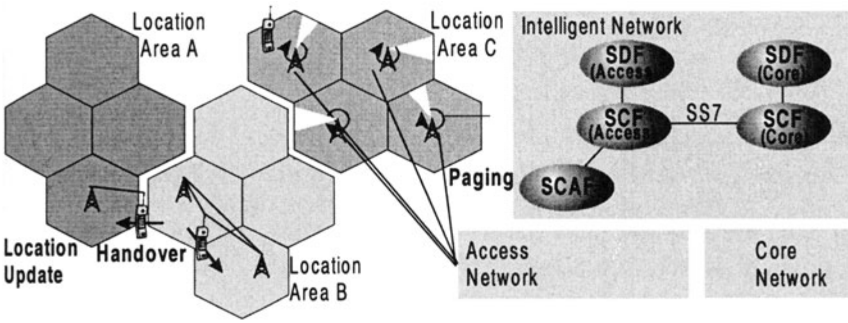

Figure 1. UMTS Mobility Management and Network Architecture.

To enable efficient location management, the coverage area of a cellular system is subdivided into *location areas* (LA). An LA represents the smallest unit for which the network maintains the current position of a user. Consequently, it consists of a number of cells greater than zero. Keeping the location database in the fixed network part up-to-date is done by performing a *location update* procedure each time the user crosses the boundaries of an LA. The location update is triggered when the terminal recognises that the *LA identifier* (LAI) issued by the operator on a broadcast channel is different from the previous one.

In case of an incoming call, the current location of the called user can be determined by a database request. Once the location area is determined, the user can be *paged* in all cells within this LA by broadcasting his identifier on a dedicated broadcast channel which is permanently sensed by all terminals. When a terminal recognises the identifier of a user that is currently registered with it, it sends a response message and, the user's current cell can then be determined. Subsequently, the connection can be established.

The LA structure represents a compromise between permanently keeping track of a user's position on cell basis, and paging him in the entire network.

2.2 Network architecture

Location management as well as service provision is realised and controlled by a sophisticated network architecture. Figure 1 depicts the UMTS network architecture which is subdivided into several *access networks*, a *core network*, and the *intelligent network* (IN). An access network is mainly responsible for radio-related aspects, such as handover, radio resource control, and local switching functionality. Usually, it controls and administers several LAs. The core network provides switching functionality for call and bearer control, and manages the information exchange between various access networks for mobility and service support.

Analogous to the separation of access and core network, the IN is subdivided into an access and a core network related part. It comprises of many *functional entities* (FE) distributed over the different sub-networks. The FEs considered here are the *Service Control Function* (SCF), the *Service Data Function* (SDF), and the *Service Control Agent Function* (SCAF). The SCF contains the overall service and mobility control logic and handles service related processing activities whereas the SDF can be seen as that component of a distributed database that handles storage and access to service and mobility related data. Finally, the SCAF is used for triggering and controlling the functions of mobility management, like handover or paging [7]. The FEs are interconnected via the *Signalling System No. 7* (SS7), a fast and reliable OSI-like protocol stack.

The concept of distributed databases in UMTS is a result of experiences made in today's GSM systems, where the partially centralised approach of one *home location register* (HLR) and many *visitor location registers* (VLR) has often led to a bottleneck.

2.3 Transport layer

Up to now, mobile cellular networks allow only circuit-switched communication. However, running CORBA-like systems in those networks would require the existence of a packet-oriented transport protocol. Various research activities have focussed on this issue in the recent years. These activities are summarised and discussed in [2].

According to these comments, existing packet-oriented transport protocols like the *Transmission Control Protocol* (TCP) are not suited for a deployment in cellular networks. One reason for that is the recovery mechanism of TCP which causes a decreasing send rate in case of insufficient capacity somewhere in the network. However, due to bursts of errors occurring on the radio link when the MT moves through an uncovered area, highly variable transmission delays are typical in cellular networks and consequently, TCP would behave incorrectly.

Instead, a transport protocol is needed that separates flow and congestion control functionality on the wireless link from that on the fixed network. Furthermore, the protocol has to cope with problems arising from terminal mobility. Therefore, a connection on the transport layer between an MT and a fixed host should be established as two separate connections: one between an MT and the access network and another one between the access network and the fixed host.

The approach presented assumes the existence of such a transport protocol. Between an MT and an access network a transport protocol with adequate flow and congestion control mechanisms is deployed, whereas between access network and fixed host any transport protocol is conceivable. In the following, TCP is assumed for the transmission of user data and the *Signalling Connection Control Part* (SCCP) of the SS7 stack for signalling purposes.

However, in contrast to the demands described above, mobility management related aspects are separated from the transport layer, since mobility of the nomadic user is realised by sophisticated naming services arranged on higher layers. Instead, we focus on a mobility model that utilises bridging as it is intended by the CORBA framework for interoperability between different domains. The bridging is described in the following section.

3 BRIDGING AND DOMAIN INTEROPERABILITY

To overcome the incompatibilities of different transport technologies, the OMG has adopted a common protocol specification named the *General Inter-Operation Protocol* (GIOP). GIOP has been designed to be applied on top of each connection-oriented transport protocol, or which at least guarantees that the order of messages sent is maintained. Thus, GIOP closes the gap between the requirements of the CORBA middleware kernel, the ORB, and the respective transport protocol. However, since each transport protocol has certain characteristics, i.e. the address format used, GIOP must be adapted further.

Up to now, the only existing GIOP specification is the *Internet Inter-Operation Protocol* (IIOP). As it has already been stated above, CORBA will also increasingly gain momentum in the telecommunication sector. However, the TCP-based IIOP is not able to meet the strong demands of reliability and fastness required in telecommunication systems. Therefore, a major activity of OMG's telecom task force concerns the definition of protocols for operating COs on top of the SS7 stack, strictly speaking on top of the SCCP. In this way, conventional entities of the IN, like the SCF, can interact with COs or can even be realised by them. Accordingly, [3] suggests to name this version *SCCP Inter-Operation Protocol* (SIOP). Since CORBA is seen as a main candidate to be deployed as TINA's middleware technology, this activity represents a first step in an evolutionary process focusing on replacing of INs by TINA.

Another activity in this field focuses on the adaptation of GIOP to a wireless transport protocol, which might be named *Wireless Inter-Operation Protocol* (WIOP) according to [2]. In contrast to transport protocols applied in wired networks, this kind of protocol has to cope with increased error rates and the immediate re-routing of connections due to handovers. This also includes the permanent change of the mobile terminal's network address when it is moved to another site.

To provide interoperability between domains with different transport networks, the CORBA framework introduces a mapping between domains, called *bridging*. As it is depicted in figure 2, a bridge resides at the boundary between the domains, transforming requests expressed in terms of one domain's model into the model of the destination domain. This transformation process is done by *proxy objects* (PO) arranged in the bridge. For example, requests addressed to COs in mobile devices are first sent to a PO in the bridge, mapped onto another GIOP version, and then forwarded to the target CO. Making requests from the wireless to a wired domain works vice versa. The cost of this mapping depends on the respective transport protocols involved and whether the mapping also includes other, for example administrative, tasks.

POs are realised as conventional COs. However, since it is desirable that POs work independently from the target CO's type, that is from the signature of the operation invoked, they must have a common skeleton to receive requests and a common stub to forward them. To this end, CORBA defines the *Dynamic Skeleton Interface* (DSI) and the *Dynamic Invocation Interface* (DII). The former allows to program a server with no need for knowledge of the interfaces the requests to this server are based on. This technique might be useful for applications that filter input streams for certain information, for example as it is done in the case of a PO, or when the type of object the server implements is not known at compilation. In this case, the interface required can be requested at run-time from a so-called *interface repository*. The DII is exactly the counterpart of the DSI. It allows the creation of common stub routines in a client application when the server's type is not yet known.

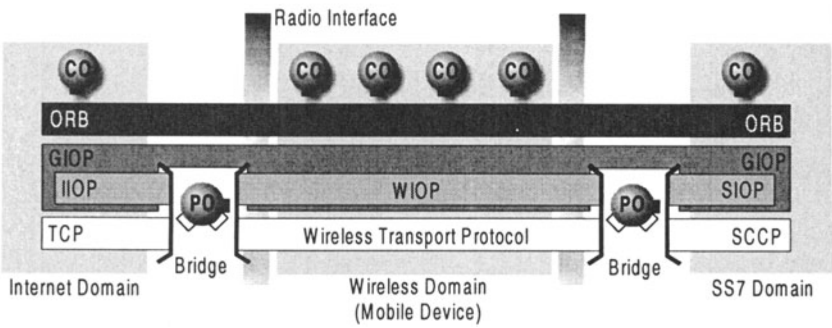

Figure 2. Bridging between different transport domains.

Up to now, neither the WIOP nor the POs in the bridge between the wired domain, be it an Internet domain or an SS7 domain, and the wireless part have been specified. In this context, addressing of COs is of special interest since the support of terminal mobility requires the consideration of location management issues here. The following sections provide a closer look at these matters.

4 ADDRESSING, REGISTRATION, AND LOCATION MANAGEMENT

In the following, the term *Mobile Computational Object* (MCO) denotes a CO that is arranged in an MT, whereas *Fixed Computational Object* (FCO) refers to an object running in a wired device. Furthermore, it is assumed that bridges between wired and wireless domains are arranged in the access networks of the cellular system. A bridge is running on a conventional host which is identified by its network address used in the wired domain. Due to scalability and load balancing reasons there might be several bridge hosts per access network.

Each bridge maintains a directory service to provide information about the MCOs it represents. The local bridge directory service is embedded in a global directory which may be distributed over different hosts all over the world. The directory service considered here is the OMG *trading service* [8] which is the best way to provide location transparency in CORBA-compliant distributed systems. However, other solutions might also be appropriate, like the *OMG naming service* or *X.500*.

Usually, a bridge comprises several POs, and each PO in turn represents the gateway for a certain number of MCOs. Thus, there is a 1:n relationship between POs and MCOs. The relation between an MCO and its corresponding PO is maintained while the terminal accommodating that MCO is located in the coverage area of the access network. Thus, when the MT is entering a new coverage area, the old bridge must delegate the responsibility for all MCOs located in this MT to the new bridge.

All in all, the bridge must support the registering and de-registering of MCOs, the connection set-up from the wired to the wireless domain and vice versa, as well as location update functionality. In the following, these procedures are described in detail.

4.1 Addressing

To send a request to a target CO, it must be addressed appropriately by the originating CO. This is done by means of an *object reference* that denotes a particular object. An object reference identifies the same CO each time the reference is used in a request, and an object may be denoted by multiple, distinct references. To hide special addressing characteristics of the transport protocol used, CORBA introduces the IDL datatype `object`. Before a request can be initialised, the originating CO must first obtain a reference to the target object, e.g. from a naming or trading service, which is then kept in a parameter of type `object`. To provide transparency concerning various address formats used in different transport domains, `object` is realised by a common form called *Interoperable Object Reference* (IOR). IOR is capable of referencing a certain object from the viewpoints of different transport domains. It contains several data entries of type `Profile_Body`, one for each transport domain supported. The internal data structure of this type depends on the GIOP-version used in the transport domain. For example, the profile body of IIOP contains a field for storing the IP-address of the Internet host and a port number listened by the process of the target CO for incoming requests. However, due to terminal mobility, addressing matters in wireless environments are quite more complex, as can be seen from the WIOP module which is proposed here and which is depicted in figure 3.

As prescribed for UMTS-compliant systems, each terminal connected to the network is assigned a *Temporary Mobile Terminal Identifier* (TMTI) which is valid for the duration a customer is connected to the network and which is used when the terminal is paged in case of an incoming call. Thus, the field `TMTI` is an important part of the WIOP profile body. `port` denotes a port in the mobile device the requested CO's process is running on. TMTI only provides a flat addressing scheme; the actual network address is contained in the field `reference`. Such a network address consists of two identifiers, one for referencing the cell the terminal is moving in and another for specifying a logical radio channel within this cell. Since macro diversity should be supported by the WIOP profile body, `reference` might contain any number of such pairs. `object_key` is used by a daemon process in the specified endpoint address to unambiguously identify the CO.

4.2 Registering and De-Registering

Generally, two registration methods are conceivable: *explicit* and *implicit registration*. The former is to be preferred when invocations are sent from the wired domain to MCOs in the wireless domain. Here, the respective target MCO must explicitly register with a directory service in the wired domain. The implicit registration is sufficient, whenever MCOs invoke FCOs, but is not absolutely suited for the reverse case. Both methods are explained here.

```
module WIOP {
      struct Location_Area {
            unsigned short              LAI;
            date                  time_stamp;
      };
      struct ReferencePoint {
            unsigned short              Cell_Id;
            unsigned short              Channel_Id;
      };
      struct Profile_Body_1_0 {
            Version                     WIOP_version;
            string                      TMTI;
            unsigned short              port;
            sequence <ReferencePoint>   reference;
            sequence <octet>            object_key;
      };
};
```

Figure 3. IDL specification of the WIOP address format.

4.2.1 Explicit registration. The explicit registration process works similar as for conventional FCOs and it is done by contacting a trader and establishing a service offer there. However, in contrast to FCOs, network addresses of MCOs change from time to time when the site of the terminal changes. To hide terminal mobility from MCOs, it is proposed to equip each mobile device with a local *Trading Proxy Service* (TPS). The TPS groups the service offers of all MCOs arranged in an MT and sends this grouped offer to the responsible bridge in the access network. Furthermore, the TPS receives normal import requests initialised by local MCOs. Since it contains a small database with all offers from local MCOs, it first checks whether or not one of these can fulfil a request. If none can, the request is forwarded to another trader in the fixed network part.

Using the TPS, the detailed registering procedure works according to figure 4. When an MCO comes into being, it must first determine the object reference of the local TPS which is done by invoking the operation

`Object resolve_initial_reference (TradingService)`

on the local ORB. According to [9], this operation is mandatory for each CORBA-compliant ORB. Subsequently, the MCOs send their service offers to the TPS via the `Register` interface (1). It must be stressed that this interface is exactly the same as the one used by the conventional trading service and that the TPS is seen accordingly by the respective objects. The TPS collects the service offers of all MCOs and groups them according to the interface `Group` which is provided by the PO in the bridge (2), and which is depicted in figure 5. Establishing a group is done by grouping all service offers in a parameter of type `OfferGroup` and sending it to the bridge using the operation `export_group`. The successful execution of this operation is acknowledged by returning an unambiguous identifier of type `GroupId` which is used as a reference for further operations, like the withdrawing of the group or the adding or removing of components. For the sake of simplicity, this group identifier may correspond to the *Temporary Mobile Terminal Identifier* (TMTI) which references the MT during its association with the network, and which is used for paging it.

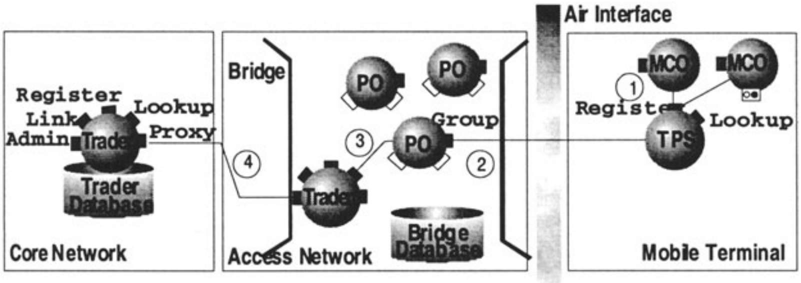

Figure 4. Registration Procedure.

The grouping mechanism has some advantages: first, it saves valuable resources at the radio interface since it avoids that each MCO registers separately with the bridge. Furthermore, it simplifies the set-up of connections (explained in section 4.3) and finally, it simplifies the changing of the bridge when the terminal moves to the coverage area of another access network (explained in section 4.4).

Figure 6 provides a closer look at events triggered when an `export_group` operation arrives in a PO. First, the PO establishes a new entry in the *grouping table* of the bridge (2a). The grouping table contains four fields for storing the *group identifier*, which is derived from the TMTI, the *group reference*, a *time stamp*, and a so-called *forwarder reference*, respectively. The group reference is derived from the field `reference` of the WIOP profile body. Since the MCOs of a certain group are always located in the same terminal, the common routing information, like cell and channel identifier, can be used together to identify these objects. The time stamp and the forward reference are used for location management, and are explained in the next sections.

In the next step, the PO registers each MCO with the *object table* of the bridge. Within this table, all references are organised according to the object key which identifies the respective MCO unambiguously. Furthermore, each entry contains the MT's port number the respective MCO is running on, and a pointer to the group identifier in the grouping table.

Having created the table entries, the PO splits the group into the single service offers again and prepares them for registering with the local trader. All object references contained in the service offers are replaced by the PO's reference which is available in the addressing format of the wired domain where the bridge resides, i.e. as a profile body of IIOP or SIOP. Furthermore, the object key is copied from the old reference to find the appropriate entry in the object table again, in case an invocation of an MCO passes the bridge.

```
interface Group {
  typedef string OfferId;
  struct Offer {
        Object reference; OfferId localId;
        ServiceTypeName type; PropertySeq properties;
  };
  typedef sequence<Offer> OfferGroup;
  GroupId export_group (
        in OfferGroup offers;
  ) ;
  void withdraw_group (
        in GroupId id
  ) ;
  void add_member (
        in GroupId id, in OfferGroup offers;
  );
  void remove_member (
        in GroupId id, OfferId localId;
  );
};
```

Figure 5. Group Interface.

Then, the offers are registered with the trader (3) which may then establish a proxy offer in a central trader somewhere in the core network (4). Subsequently, a confirmation is returned to the PO in the bridge. As a result, the PO, in turn, sends the group identifier as an acknowledgement to the TPS in the MT. This identifier may be used for further transactions on this group using the operations `add_member` and `remove_member` of the interface `Group`.

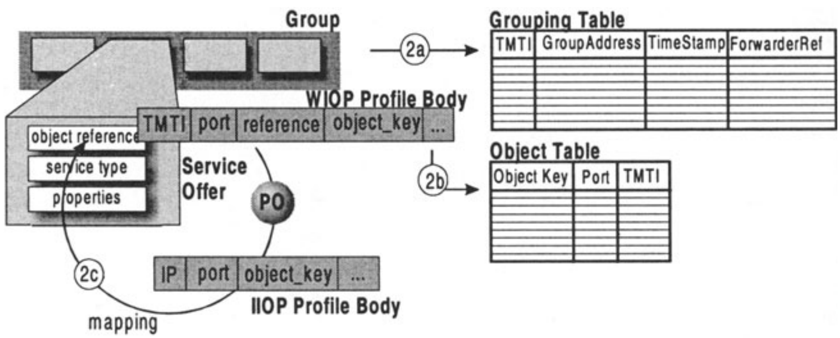

Figure 6. Reference Mapping inside a bridge.

De-registering of MCOs works similar. Either a complete group is removed from the bridge using the `withdraw_group` operation, or a single offer is deleted from the group via `remove_member`. In both cases, the bridge makes the trader to remove the corresponding service offers.

The approach presented here is not limited to trading. It can be extended to be used in conjunction with a naming service or any other directory service.

4.2.2. Implicit registration. The implicit registration is done by a bridge, when an MCO makes an invocation of an FCO without having explicitly registered before with the responsible bridge. The PO scans each invocation passing the bridge from the wireless towards the wired domain for the address of the invoking MCO in the sender field of the transport protocol used. On the basis of this address it then checks whether or not the grouping and object tables contain corresponding entries. If they do, the PO maps the invocation to the transport protocol of the wired domain, replaces the sender field by its own address, and forwards the invocation to the FCO. Otherwise, it must first establish a new group according to steps 2a-2c depicted in figure 6.

Due to the fact that de-registering is not possible here, the PO must remove the entry automatically. Two ways a conceivable here. The PO removes the entry after the response corresponding to the invocation has been passed or it removes the entries from the tables after a certain period of time, using e.g. the time stamp field in the grouping table. Both methods are appropriate and may be used depending on certain policies followed by the bridge.

4.3 Invoking MCOs

Figure 7 depicts all events related to the invocation of an MCO by an FCO. First, the FCO performs a lookup on the trader in the core network in order to get an object reference of the required MCO (1). Assuming the trader finds an appropriate proxy offer, it forwards the request to the trader of the bridge the MCO is attached to (2). The trader may also maintain further information concerning the availability of that MCO. It returns a reference which contains the network address of the responsible PO as well as the MCO's object key. Subsequently, the FCO performs the invocation, whereupon this invocation is first passed to the PO (3). The PO is then responsible for establishing a connection to the MT accommodating the requested MCO. This requires a number of single steps.

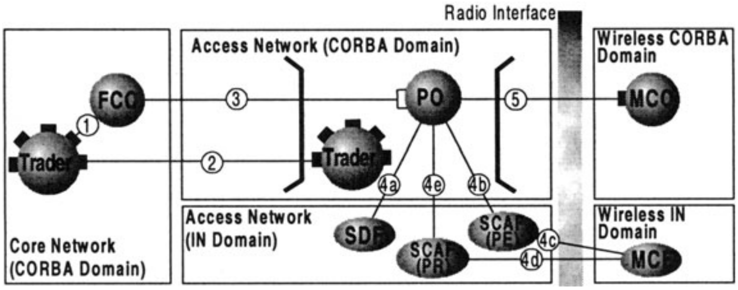

Figure 7. Process of invoking an MCO.

First, the PO locates the WIOP profile body and group identifier in the object table by means of the object key, which has been transmitted together with the invocation. With the group identifier, the PO searches the grouping table for the time stamp that specifies the last time of interaction between the PO and any MCO in the concerning MT. If the time seen since then exceeds a certain limit - which may depend on the time a radio channel remains reserved after a transmission - the PO has to initialise the paging of that MT (4a-4e). Otherwise an existing path to the MT is re-used and the invocation is forwarded to the MCO (5), using the reference contained in the field `GroupAddress` of the object table. This path is composed of the link from the access network accommodating the bridge to the respective base station, and from there over the radio connection to the terminal.

However, forwarding capabilities of a PO also requires a sophisticated connection management since MCOs may be disconnected either temporarily or even permanently due to interruptions on the radio interface. In this case, the PO has to inform the invoking FCO by sending an exception or location forward targeting at another PO the MCO in question is - due to macro diversity - also registered with. To this end, the GIOP reply message provides sufficient capabilities.

4.4 Location Management

As explained above, mobility management comprises paging and localisation of customers and terminals as well as the handover procedure. These procedures are realised by existing infrastructure, like SCFs and SDFs in the access networks, and thus the approach presented here aims to integrate the bridging functionality into this infrastructure without making any modifications on it. However, it is inevitable that the bridge co-operates with this infrastructure. This co-operation is described in this section. Since the interaction between COs and FEs is currently being standardised by the OMG [3], the communication between bridge and SCFs, SDFs, and other entities is ensured. In the long term, it might even be imaginable that FEs are realised as COs. Therefore, it is assumed in the following that a PO is seen as a conventional SCF by other IN entities. First, the paging procedure is described, see figure 7.

To page a certain MCO, the PO first has to request an SDF for the current LA where the MT resides. That is, the PO transmits the TMTI to the SDF (4a) (the PO can determine the corresponding TMTI by means of the object key and the object table). After receiving the LAI, the PO checks whether or not it is responsible for that LA. If it isn't, the MT has recently moved to the coverage area of another access network and thus the PO has to delegate the responsibility to another PO in the new access network (see below). Otherwise, the PO triggers the *Paging Entity* (PE), which is arranged in an SCAF of the corresponding LA, to page the terminal (4b-4c). If the *Mobile Control Function* (MCF) of that MT receives the paging request and identifies its TMTI in this request, it sends a reply, whereupon the network can determine the cell of the MT. This reply is received by a *Paging Receiver* (PR) (4d) which then performs the necessary actions for subsequent communication between MT and network, and sends the resulting cell and channel identifier to the PO (4e). Then, the request can be forwarded as described in the previous section. The working of PE and PR are described in detail in [7].

Considering terminal mobility, one has to consider the movement within one LA, between different LAs but within the coverage of the same access network, and between the coverage areas of different access networks. The first and the second case can be handled in an identical simple way. When a request that is addressed to an MT that has recently been moved to another cell (within the same LA or between different LAs but within the coverage area of that access network), arrives at a bridge, the exact position can be determined by performing the paging procedure described in steps 4a-4d of figure 7. However, if the request arrives within the time period of the reuse interval, an error occurs which has to be reported to the bridge. This error report will then lead to the execution of a new paging attempt.

The situation gets quite more complex when the MT moves to the coverage area of another bridge. Here, the old bridge has to delegate the responsibility for the affected MCOs to a bridge in the new access network. The movement is not reported to the new bridge since this would require a change of existing IN entities. Instead, the registration with the new bridge is done when the first interaction occurs. Here, two cases must be distinguished.

When the first invocation after the movement passes the bridge from the MT to the wired domain, it performs an implicit registration (see section 4.2.2.), even if the respective MCO has been registered explicitly with the previous bridge. The explicit registration is done when the first invocation passes from the wired to the wireless domain.

When this happens, the old bridge is contacted first since the trading service has so far not been informed about any movement. When the PO performs a request on the SDF according to step 4b in figure 7, an LAI would be returned that is related to an LA outside the coverage area of that bridge. In this case, the PO delegates the responsibility to a bridge associated with that LA. This is done like the initial explicit registration procedure: the PO groups the service offers of all affected MCOs and sends them to a PO in the bridge via the `Group` interface. Subsequently, the reference of the new PO is left in the old bridge and deposited in the forwarder field of the grouping table in order to guarantee that invocations from FCOs having the old references are forwarded appropriately. However, after a while, this forwarding is cancelled and the entries are removed from the grouping and object tables. Furthermore, the service offers are removed from the local trader, and the pointer in the proxy offers stored in the global trading system is changed appropriately.

5 CONCLUSIONS

Since future nomadic communication will be determined by mobile networks on the one hand and object oriented control mechanisms on the other, it will be unavoidable to break up the boundary of the air interface. The presented approach can be seen as a proposal aiming at the seamless integration of distributed systems into the existing (or planned) infrastructure of mobile networks. Furthermore, it provides full location transparency for both FCOs and MCOs.

To get an idea of performance matters, and to come to a stepwise improvement of the approach, mathematical and simulative analysis have to be carried out that consider the customer's movement and call behaviour appropriately. Similar analysis has already been done in the context of mobile user agents that follow their mobile customer's through the fixed network part, see [10]. The mobility models used here can be applied to evaluate the bridging approach.

References

[1] BELLCORE, *Middleware Issues and Requirements for Nomadic Access to CORBA-Based Enterprise Information Systems.* Response to OMG Telecom Domain Task Force RFI on Supporting Wireless Access and Mobility in CORBA, OMG DTC Document telecom/98-06-04.

[2] K. RAATIKAINEN et. al.: *White Paper on Wireless Access and Terminal Mobility in CORBA*. OMG DTC Document telecom/98-11-09, 1998.

[3] THE OBJECT MANAGEMENT GROUP, Telecom Domain Task Force: *Interworking between CORBA and IN Systems*, Response to Request for Proposal, OMG DTC Document telecom/97-12-06, 1997.

[4] TINA-C: *Overall Concepts and Principles of TINA*, Version 1.0, 1995 http://www.tinac.com/deliverable/deliverable.htm.

[5] LINNHOFF-POPIEN C., *CORBA - Communication and Management* (in german). Springer Publishing Company, Berlin Heidelberg New York, 1998.

[6] CLAPTION A. et. al.: *TINA Concepts for Third Generation Mobile Systems - TINA concepts applied to UMTS and concepts for mobility support in TINA*. EURESCOM Project P608, Deliverable 2, January 1998.
http://www.eurescom.de/~pub-deliverables/P600-series/P608/D2/Vol[1-4]/Vol[1-4].pdf

[7] KÜPPER A., SPANIOL O., JAKOBS K., Universal Mobile Telecommunications System (UMTS)-Mobile Communication in the 21st Century. *Encyclopaedia of Telecommunications*, pp. 151-179, Published by Marcel Dekker, Inc., Volume 17, 1998.

[8] THE OBJECT MANAGEMENT GROUP: *CORBA Service Specification*, OMG Document formal/98-07-05, 1998. ftp://ftp.omg.org/pub/docs/formal/98-07-05.pdf.

[9] THE OBJECT MANAGEMENT GROUP: *CORBA/IIOP Specification*, OMG Document formal/98-07-01, 1998. ftp://ftp.omg.org/pub/docs/formal/98-07-01.pdf.

[10] KÜPPER A., PARK A., Stationary vs. Mobile User Agents in Future Mobile Telecommunication Networks, *Proceeding of 2nd International Workshop of Mobile Agents '98*, pp. 112-123, Kurt Rothermel, Fritz Hohl (Eds), Springer Publishing Company, LNCS 1477.

Biographies

Axel Küpper works as a research assistant at the Department of Computer Science (Informatik 4) at Aachen University of Technology where he received his diploma in computer science in 1996. At the moment, he is involved in the programme Mobile Communication of the Deutsche Forschungsgemeinschaft. His forthcoming Ph.D. thesis is dedicated to the deployment of TINA-like approaches in future mobile cellular networks.

Claudia Linhoff-Popien is professor at the Institute of Computer Science at the Ludwig Maximilians University in Munich since 1998. Before that she was research assistant and senior assistant at the Aachen University of Technology where she received the Ph.D. in 1994 and the habilitation in 1998. Currently she is involved in several industrial projects concerning service mediation and distributed platforms.