Using WAP as the Enabling Technology for CORBA in Mobile and Wireless Environments

Rainer Ruggaber, Jochen Schiller, Jochen Seitz
Insitute of Telematics, University of Karlsruhe, Germany
E-Mail: [ruggaber|schiller|seitz]@telematik.informatik.uni-karlsruhe.de

Abstract

CORBA is currently the most popular middleware platform. Due to the progress and acceptance of mobile communications, it is inevitable to offer CORBA services for mobile terminals, especially over a wireless link. The Object Management Group, the standardization body for CORBA, already requested proposals for extending CORBA in the wireless environment, leading to several submissions. However, the standardization process is far from being completed. Therefore, we propose the adaptation of an outstanding approach to supply a range of services in the mobile environment, the Wireless Application Protocol (WAP) framework. In the paper, we show how the CORBA standard and the WAP framework can converge, leading to a comprehensive middleware platform for distributed applications in a wireless and mobile environment.

1 Introduction

Distributed applications and nomadic computing are tremendously growing areas in today's information age. Thus, it is only natural to combine both technologies to get the best of both worlds.

Due to the progress in mobile communications and the widespread use of ultra-mobile devices like PDAs or cellular phones it is necessary to offer CORBA for these devices. This extension of the middleware platform eliminates the needs for special purpose solutions to integrate mobile devices into an existing application environment. By using CORBA as a middleware platform the mobile device gains easy access to CORBA-Services like the Event-Service, the Naming-Service or the Trading-Service.

2 Related Work

Distributed applications are normally built using a middleware-platform providing several services and an easy-to-use communication infrastructure. However, common middleware platforms do not support mobile hosts. Therefore, either proprietary platforms evolved (e.g. MOBIWARE [Campbell 98] or MOWGLI [Alanko 96]), or existing platforms have to be extended. The latter point seems to be advantageous because existing distributed applications can easily be provided in a mobile environment.

Our project focused on the commonly applied middleware CORBA which is detailed in the next paragraph. The Object Management Group, the standardization consortium behind CORBA, has also recognized the need for integrating mobile environments. Therefore, they released a "Request for Information" (RFI) on supporting wireless access and mobility in CORBA. This RFI and the responses are presented in the second paragraph

2.1 CORBA

CORBA (Common Object Request Broker Architecture) is currently the most popular middleware platform, facilitating the implementation of object-oriented location-independent distributed client/server-applications. CORBA supports the interoperability between clients and servers written in different programming languages, running on different operating systems and connected by different network technologies, thus making CORBA the ideal basis for developing applications in a heterogeneous environment.

The central component in CORBA is the ORB (Object Request Broker), which is responsible for transparently forwarding requests and replies to the appropriate entity, thus allowing a client to invoke operations on a server without knowing anything about the location of the server or its implementation.

The CORBA specification comprises a set of interface definitions [CORBA 2.3]. However, the specification makes no assumptions about implementation issues.

2.2 Wireless Access and Mobility in CORBA

In 1998, the Object Management Group released a Request for Information on supporting wireless access and mobility in CORBA [OMG 98]. The RFI

0-7695-0468-X/99 $10.00 © 1999 IEEE

differentiates between wireless access, terminal mobility and personal mobility. The main question that arises from this problem field is whether the current Internet Inter-ORB Protocol (IIOP) is sufficient for wireless access and mobility in CORBA or whether environment-specific Inter-ORB Protocols (ESIOPS) are needed. A decision on this question will also influence the requirements for additional services to be incorporated into the CORBA platform.

Submissions to this RFI were due on August 21, 1998. Up to now, five responses were received. Two of them, one be the Deutsche Telekom and one by the DOLMEN project, are closely related to the Telecommunication Information Networking Architecture TINA [Gavras 98, Raatikainen 98]. The second one proposes a special ESIOP, the so-called Light-Weight Inter-ORB Protocol LW-IOP. The Ericsson input [Ismailov 98] also proposed a tailored ESIOP, without going into too much technical detail. However, there is already the hint to the WAP consortium. Another proposal, by Ericsson and Kent Ridge Digital Labs [Biswas 98], put CORBA into the wireless ATM environment. Finally, the submission by Bellcore [Jain 1998] pointed out that the usual mobility enhancements of the Internet like Mobile IP may not be sufficient in a mobile CORBA environment. Hence, the CORBA services should be investigated and re-designed.

To summarize the proposals, all are based on ESIOPs specially tailored for the wireless link. The traditional TCP/IP protocol suite is not suited for a mobile and wireless CORBA environment. However, the proposed approaches are either very specific to certain communication technologies (like wireless ATM) or not very detailed. Therefore, we incorporated the hint to the Wireless Application Protocol and developed a generally applicable architecture which is introduced in the next section.

3 Technical Aspects

CORBA in a wireless environment has to cope with the varying characteristics of a wireless link. Therefore, the CORBA communication infrastructure has to be investigated. This infrastructure is made up of different Object Request Brokers as detailed in the next paragraph.

3.1 The CORBA Communication Infrastructure

CORBA-ORBs use the General Inter-ORB Protocol (GIOP) to inter-operate. GIOP can be mapped onto connection-oriented protocols that meet a set of assumptions. GIOP defines the transport protocol independent properties, e.g. message formats. ORB-Interoperability issues that are transport protocol dependent are defined in the mapping of GIOP onto the specific transport protocol. Other protocols besides GIOP are possible, e.g. the DCE-ESIOP to integrate CORBA with DCE, but are not considered here.

The GIOP specification consists of the Common Data Representation (CDR), the GIOP message formats and a set of GIOP transport assumptions.

The CDR is the transfer syntax that maps IDL data types onto a low-level representation for transfer between ORBs and Inter-ORB bridges.

GIOP messages are exchanged between CORBA entities to invoke requests, locate object implementations and to manage communication channels. GIOP communication is not symmetric. This is due to an easier and deadlock-avoiding implementation of GIOP. Therefore to describe GIOP messages it is necessary to define client and server roles. In the GIOP context a client is an entity that opens a connection and may send *Request*, *LocateRequest* and *CancelRequest* messages. The server accepts connections and may send *Reply*, *LocateReply* and *CloseConnection* messages. Both are allowed to send *MessageError* messages. The format of GIOP messages is defined in CORBA-IDL.

Every GIOP message contains the address of the server object. The address of a server object consists of a transport dependent part that contains its network address and an opaque transport independent part that identifies the object inside the server. If a server object can be reached via different addresses, e.g. because it can be reached via different networks that use a different addressing scheme, multiple addresses can be added to the GIOP message.

The assumptions the GIOP definition makes about the transport behavior include that the transport protocol is connection-oriented because a connection defines the scope for identifying requests. The connection has to be reliable, which means that bytes are delivered in the same order they are sent, at most once, and that a positive acknowledgment of delivery is available. The transport provides notification of disorderly connection loss. A client may multiplex connections to multiple target objects onto one transport connection.

Transport-dependent specifications like the addressing of the host and the server-object are defined in the mapping.

The Internet Inter-ORB Protocol (IIOP) is a mapping from GIOP to TCP/IP and must be supported by every ORB implementation. Mappings to other transport protocols may be defined.

Altogether GIOP defines a kind of *lingua franca* for different middleware implementations.

3.2 TCP-Drawbacks

It is a well-known fact since many years that TCP does not provide a good performance in wireless and mobile environments [Balakrishnan95, Caceres95]. The main reason is the error semantic of TCP and the resulting behavior. If a packet is lost during transmission, TCP assumes a congestion within the network and slows down transmission using the slow start mechanism. While this behavior makes sense in fixed networks, it is in most cases wrong in wireless and mobile networks. Wireless connections typically have much higher error rates compared to fixed connections. Thus, packet loss is quite often due to these transmission errors. Furthermore, handover procedures in mobile networks can cause additional packet loss (packets in transit during the handover might be lost). Slowing down the performance is not at all useful in these cases, as the loss does not result from congestion. Several approaches have been proposed to solve this problem, either by splitting up the connection into a wireless and a wired part [Bakre95], by doing local retransmissions for lost packets [Brewer98], or by additional mechanisms controlling TCP directly [Brown97]. However, all these approaches represent add-ons but not really integrated solutions.

3.3 Wireless Application Protocol (WAP)

As the current standard TCP is inappropriate for wireless links, the protocol stack used in CORBA must be modified. The most sophisticated solution providing Internet services for mobile, wireless devices has been worked out by the Wireless Application Protocol Forum (WAP Forum), which was founded in June 1997 by Ericsson, Motorola, Nokia, and Unwired Planet [WAPForum99]. The basic objectives of the WAP Forum are to bring different Internet content (e.g., Web pages, push services) and other data services (e.g., stock quotes) to digital cellular phones and other wireless, mobile terminals (e.g., PDAs, laptops). Furthermore, a protocol suite should enable global wireless communication across different wireless network technologies, e.g., GSM, CDPD, UMTS etc. Therefore, the Forum embraces and extends existing standards and technologies of the Internet wherever possible and creates a framework for the development of contents and applications that scale across a very wide range of wireless bearer networks and wireless device types. Hence, the idea of using WAP for CORBA is evident.

Figure 1 gives an overview of the WAP architecture, its protocols and components. The basis for transmission of data are different *bearer services*. WAP does not specify bearer services, but uses existing data services and will integrate further services. Examples are message services, such as SMS (short message service) of GSM, circuit-switched data, such as HSCSD (high-speed circuit switched data) in GSM, or packet switched data, such as GPRS (general packet radio service) in GSM. Many other bearer services are supported, such as CDPD, IS-136, PHS. No special interface has been specified between the bearer service and the next higher layer, the transport layer with its *wireless datagram protocol* (WDP) and the additional *wireless control message protocol* (WCMP), because the adaptation of these protocols are bearer-specific. The transport layer offers a bearer independent, consistent datagram-oriented service to the higher layers of the WAP architecture. Communication is done transparently over one of the available bearer services. The transport layer service access point (T-SAP) is the common interface to be used by higher layers independent of the underlying network. If a bearer already offers IP service, then UDP is used as WDP and T-SAP is similar to the socket interface.

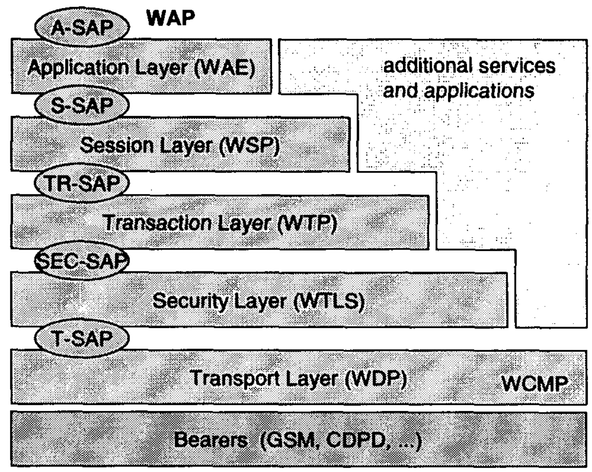

Figure 1. WAP architecture

The next higher layer, the security layer with its *wireless transport layer security protocol* WTLS offers its service at the security SAP (SEC-SAP). WTLS is based on the transport layer security (TLS, formerly SSL, secure sockets layer) already known from the WWW. But WTLS has been optimized for use in wireless networks with narrowband channels. WTLS can offer data integrity, privacy, authentication, and denial-of-service protection.

The WAP transaction layer with its *wireless transaction protocol* (WTP) offers a lightweight transaction service at the transaction SAP (TR-SAP). This service efficiently provides reliable or unreliable requests and asynchronous transactions. The next higher layer, the session layer with the *wireless session protocol* (WSP) currently offers two services at the session-SAP (S-SAP), one connection oriented and one connectionless. A special service for browsing the web (WSP/B) has been defined that offers HTTP/1.1 functionality, long-lived session state, session suspend and resume, session migration and other features needed for wireless and mobile access to the web.

Finally, on top of it all, the application layer with the *wireless application environment* (WAE) offers a framework for the integration of different WWW and mobile telephony applications. The main issues here are scripting languages, special markup languages, interfaces to telephony applications, and many content formats adapted to the special requirements of small, handheld, wireless devices.

WAP does not force all applications to always use the whole protocol architecture. Applications can use only a part of the architecture. If an application, for example, does not require security but the reliable transport of data, it can use a service of the transaction layer and does not need the security layer. Simple applications can directly use WDP. First WAP products already exist and several services are offered (stock quotes, TV program, ticket reservation, news etc.).

We identified the following components of the WAP architecture necessary in a CORBA environment:

- On the transport layer, WAP defines the *wireless datagram protocol* that offers a bearer independent, consistent datagram-oriented transport service. Thus, the shortcomings of TCP can be overcome. However, WDP is a connectionless and unreliable protocol and does not comply with the requirements of GIOP.
- Therefore, there is the need for a service providing reliability to improve WDP. WAP incorporates the *wireless transaction protocol* for reliable requests and asynchronous transactions.
- Furthermore, to handle sudden disconnections or link failures WAP introduced the *wireless session protocol.* Using this protocol in a CORBA environment makes the breakdown of the transport service transparent to the ORB and thus to the CORBA entities. This is even more than TCP can provide.

Hence, these three protocols of the WAP architecture provide a reliable transport service substituting TCP in the wireless domain. And if there is the need for securing the data transfer, one can easily integrate the *wireless transport layer security protocol*, another component of the WAP architecture.

4 Implementation Issues

We use ORBacus [ORBacus 3.1.3] by OOC as an ORB implementation. It is available with source code and is published under the ORBacus Royalty-Free Public License free of charge for non-commercial use. It includes an interface called Open Communication Interface (OCI). This is an interface below the GIOP-Layer where new mappings on transport protocols can be "plugged in" the ORB. The use of a PlugIn for a different transport protocol is transparent for the client- and the server-applications as long as both support this PlugIn.

The OCI is based on the connector and acceptor design patterns [Schmidt 96]. The intention of both design patterns is to decouple active (connector) and passive (acceptor) service initialization from the tasks performed once a service is initialized.

PlugIns for GFP and ISDN have been realized [Holz 97]. Currently we are working on a PlugIn for ATM AAL5. PlugIns for other transports can be easily realized as long as they meet the GIOP requirements. If the transport does not meet the GIOP requirements, additional functionality must be implemented in the PlugIn to meet the requirements. In the following we will show which functionality has to be integrated in a WIOP PlugIn.

```
#include <IOP.idl>
#pragma prefix "omg.org"

module IOP
{
  const IOP::ProfileId TAG_WIOP        = 4;
};

module WIOP
{
  struct Version
  {
    octet major;
    octet minor;
  };

  typedef unsigned long AddrType;
  const AddrType WAP_IP                = 1;
  const AddrType WAP_MSISDN            = 2;
  const AddrType WAP_X.25              = 3;

  struct ProfileBody
  {
    Version          wapiop_version;
    AddrType         address_type;
    string           host;
    unsigned short   port;
    sequence<octet> object_key;
  };

}; //End module WIOP
```

Figure 2. WIOP IDL-Definition

In order to include a new PlugIn it is necessary to describe the address format in IDL. Figure 2 shows the necessary definitions.

At first it is necessary to define a tag that identifies the WAP address of an object. Right now we have chosen the four because it is not used otherwise. For official use it is necessary to get a tag assigned from the OMG.

WAP allows the use of different bearers. We have to consider this by defining different types of bearers because they use different addressing schemes

(AddrType). Right now three bearers are considered (IP, MSISDN, X.25). Other addressing schemes might be easily added.

Finally, the ProfileBody contains all the information necessary to select a server with a given transport protocol. The version is necessary to detect incompatible changes in the WIOP protocol. The address_type selects the bearer that is used to establish the connection. The host contains the bearer dependent addressing information. The port selects the application address associated with the destination address. The object key is an opaque value that identifies the object. It is generated by the server and is not used for any client processing.

Not all of the WAP components are necessary to support CORBA applications. The components that are useful are shown in Figure 3 and are WDP, WTP and WSP.

For synchronous CORBA invocations it is most useful to select class 2 transactions in WTP. It is well suited for a request/response type of communication. In the invocation message from the client a transaction identifier (TID) is selected. It is used for further identification of the transaction. The server implicitly acknowledges the invocation message with a response message or a hold on message if costly processing is necessary and a timeout on client side should be prohibited. The response message is answered by an acknowledgment. WTP supports segmentation and reassembly of large messages and selective retransmission. WTP with class 2 transactions is the basis for reliable message transfer and fulfills the assumptions of GIOP.

For asynchronous CORBA invocations class 0 or 1 transactions may be selected. Class 0 transactions provide an unreliable datagram service like it is specified in the CORBA specification. If a reliable transport of oneway invocations is necessary a class 1 transaction may be selected.

The Wireless Session Protocol (WSP) provides the possibility to set up sessions that are independent of the underlying transport connection. If WSP is used the sudden and frequent transport connection losses in the wireless and mobile environment will not have implications on the distributed application because it will not recognize the transport connection loss.

Sessions in WSP are not tied to the underlying transport. The client, server or the bearer can suspend a session. The session then changes into the state suspended. The client or the server can resume a session on the same or a different bearer.

Multiple transactions can be invoked in a single session. The transactions are identified by their transaction ID, so requests can be asynchronously invoked. To use the method invocation facility in WSP transaction class 2 of WTP must be used.

It is most useful to hide the session management from the application and integrate it in the WIOP PlugIn. Automatic suspending and resuming of sessions can be done there, based on method invocations of the client.

5 Architecture Integration

There are two possible solutions to integrate WAP into a CORBA environment. One is that the CORBA server itself is able to "speak" WAP. The other one is, that a gateway is used to translate IIOP requests into WIOP requests.

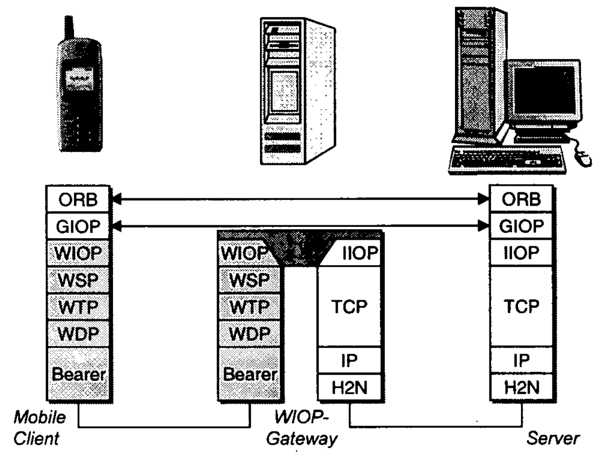

Figure 3. Gateway architecture

If a server operator wants to directly support WAP end systems he could add WAP connectivity and a WIOP-PlugIn to his server. Every reference to a object in this server would then contain the WAP based address of the server. If the server also provides IP connectivity TCP/IP addressing would also be contained. This allows universal accessibility of CORBA objects in this server. If a mobile device like a cellular phone does not support IP connectivity it uses the WAP address in the object reference for connecting. A wired client might want to use the IP address so it uses only this part of the address.

In order to access legacy software using IIOP a new system component is required. In our architecture this component is called *WIOP-Gateway* and is responsible for e.g. address translation. Additional mechanisms, such as caching and filtering, can be easily integrated into this gateway. The natural point of integration for the gateway is the WAP gateway that is the endpoint of every WAP connection at the service provider.

6 Conclusion and Outlook

CORBA as the most popular middleware for distributed applications is per se not well suited for the wireless and mobile environment. Problems of the protocol stack (IIOP over TCP/IP) are the reason for introducing the Wireless Application Protocol WAP. This innovative protocol stack is specially tailored for the wireless link and therefore overcomes the drawbacks of the traditional Internet protocols. In order to make our approach consistent to existing CORBA-based applications, a special gateway was conceived bridging the proposed Wireless Inter-ORB (WIOP) protocol stack and the standardized IIOP.

The proposed architecture is a very promising approach to introduce CORBA functionality in the wireless environment. There are still some questions to be answered in future work. Currently, we are working on a proxy approach to deal with naming and addressing issues. This is especially interesting in the area of context- and location-based applications. Furthermore, the implementation of services that migrate with the mobile users may also offer new possibilities in the mobile environment. As a test-bed, we are currently establishing the UNIQuE environment [Bechler 99], in which different installations of wireless techniques and different protocol stacks can be tested.

References

[Alanko 96] T. Alanko, M. Kojo, H. Laamanen, K. Raatikainen, and M. Tienari, „Mobile Computing Based on GSM: The Mowgli Approach", J.E. Encarnação and J.M. Rabaey (Eds.): *Mobile Communications – Technology, Tools, Applications, Authentication and Security*, Chapman & Hall, 1996, p. 151–158.

[Bakre95] A. Bakre, and B. Badrinath, "I-TCP: Indirect TCP for mobile hosts", *Proc. 15th Intl. Conference on Distributed Computing Systems (ICDCS)*, Vancouver, Canada, 1995.

[Balakrishnan95] H. Balakrishnan, S. Seshan, and R. Katz, "Improving reliable transport and handoff performance in cellular wireless networks", *Wireless Networks*, J.C. Baltzer, vol. 1, 1995.

[Bechler 99] M. Bechler, H. Ritter, J. Schiller, and J. Seitz, Research in the Unique Environment: Mobile Multimedia Services, *Third IASTED International Conference on Internet and Multimedia Systems and Applications*, Nassau, Grand Bahamas, October 18-21, 1999.

[Biswas 98] J. Biswas and M. Larsson, "Response to Request for Information telecom/98-08-06", *telecom/98-08-06*, August 1998.

[Brewer98] E. Brewer, R. Katz, Y. Chawathe, S. Gribble, T. Hodes, G. Nguyen, M. Stemm, T. Henderson, E. Amit, H. Balakrishnan, A. Fox, V. Padmanabhan, and S. Seshan, "A network architecture for heterogeneous mobile computing", *IEEE Personal Communications*, 5(5), 1998.

[Brown97] K. Brown, and S. Singh, "M-TCP: TCP for mobile cellular networks", *ACM Computer Communications Review*, 27(5), 1998.

[Caceres95] R. Caceres, and L. Iftode , "Improving the performance of reliable transport protocols in mobile computing environments", *IEEE Journal on Selected Areas in Communication*, 13(5), 1995.

[Campbell 98] A.T. Campbell "The Mobiware Toolkit", *Fourth International Workshop on High Performance Protocol Architectures (HIPPARCH '98)*, London, June 15–16, 1998.

[CORBA 2.3] Object Management Group (OMG), CORBA/IIOP 2.3 Specification, *ftp://ftp.omg.org/pub/docs/formal/98-12-01.ps.gz*, Feb 1998.

[Gavras 98] A. Gavras, T. Eckardt, T. Magedanz: "PCS in TINA – Concept, Components and Applications", *telecom/98-08-17*, August 1998.

[Holz 97] E. Holz, O. Kath, M. Geipl, G. Lin, and V. Vogel, "The Camouflage Project – Introduction of TINA into Telecommunication Legacy Systems", *Proc. of TINA'97*, 1997.

[Ismailov 98] Y. Ismailov, "Response to OMG Telecom RFI on Wireless Access and Mobility", *telecom/98-08-16*, August 1998.

[Jain 98] R. Jain and A. Capitanio, "Middleware Issues and Requirements for Nomadic Access to CORBA-Based Enterprise Information Systems", *telecom/98-08-12*, August 1998.

[OMG 98] Object Management Group, Telecom Domain Task Force, "Request for Information: Supporting Wireless Access and Mobility in CORBA", *telecom/98-06-04*, June 1998.

[ORBacus 3.1.3] Object Oriented Concepts (OOC), "ORBacus-3.1.3", *ftp://ftp.ooc.com/pub/OB/3.1/OB-3.1.3.tar.gz*, 1999.

[Raatikainen 98] K. Raatikainen (Ed.), "Dolmen – Service Machine Development for an Open Long-term Mobile and Fixed Network Environment", *telecom/98-08-08*, August 1998.

[Schmidt 96] D. C. Schmidt, "Using Design Patterns to Develop High-Performance Object-Oriented Communication Software Frameworks ", *Proc.8th Annual Software Technology Conference,* Salt Lake City, Utah, April 1996.

[WAPForum99] Wireless Application Protocol Forum, *http://www.wapforum.org/*, 1999.