

A UML 2.0 Profile for Architecting B3G Applications

Mauro Caporuscio, Valérie Issarny

▶ **To cite this version:**

Mauro Caporuscio, Valérie Issarny. A UML 2.0 Profile for Architecting B3G Applications. 3rd International Workshop on Rapid Integration of Software Engineering techniques : RISE 2006, 2006, Geneva, Switzerland. pp.18-34. inria-00415917

HAL Id: inria-00415917
https://inria.hal.science/inria-00415917

Submitted on 11 Sep 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A UML 2.0 Profile for Architecting B3G Applications

Mauro Caporuscio and Valerie Issarny

INRIA-Rocquencourt
Domaine de Voluceau
78153 Le Chesnay, France
{mauro.caporuscio, valerie.issarny}@inria.fr

Abstract. B3G is an emerging network technology which conceives the convergence of telecommunication and IP-based networks for providing enhanced services able to transfer both voice and non-voice data through wired and wireless networks. Moreover, B3G networks can be no longer considered as "*passive*" entities which only transport data between end-points, but they must be considered as "*active*" parties that have their own behavior and provide services. This creates a completely new application domain where applying current software engineering design tools, such as software architectures, fails. In fact, dealing with B3G networks requires to explicit low-level details usually abstracted by the architectural descriptions.

To this extent, we present an ongoing work on investigating B3G-oriented application modeling. In particular, we propose an enhanced UML profile to define and analyze software architectures that explicitly exploit the B3G domain properties.

1 Introduction

Beyond Third-Generation (B3G) [30] network is an emerging technology which conceives the convergence of telecommunication networks with IP based networks. Services associated with B3G provide the ability to transfer both voice data (i.e., a telephone call) and non-voice data (i.e., downloading information, exchanging email, and instant messaging) through different types of network, either wired or wireless. This opens to a new world where distributed applications can not only interact with each other over plenty of different networks, but they can also interact with non-software entities (i.e., humans), by exploiting services offered by telecommunication networks.

Such a vision breaks some assumptions posed so far by the software engineering community and requires for adapting/revising the software life-cycle, and the related tools, since the early activities. In particular, the *software design* is the first process activity to be affected by the B3G application domain. The *software design* aims to build the first description, both behavioral and structural, of the entire system under development. Consequently, since the *architectural design* is described by means of Software Architecture (SA) modeling [5][25], also this one should be adapted/revised accordingly.

N. Guelfi and D. Buchs (Eds.): RISE 2006, LNCS 4401, pp. 18–34, 2007.
© Springer-Verlag Berlin Heidelberg 2007

SA has been largely accepted as a well suited tool to achieve a better software quality while reducing the time-to-market. This results in a proliferation of different Architectural Description Languages (ADLs), defined by both the academia and the industry [20].

The use of SA descriptions allows software engineers to model large, complex applications by using suitable abstractions of the system components and their interactions. This forces the separation of architectural concerns from other design ones, thus abstracting away many details. On the other hand, SA descriptions allow for analysis and verification of architectural choices, both qualitative and quantitative, since the early stages of the software life-cycle [6].

However, the use of SA fails when specifying B3G-oriented applications. In fact, dealing with B3G networks requires to explicit those low-level details usually abstracted by SA specifications. To this extent, we argue networks involved in B3G scenarios can be no longer considered as "*passive*" entities which only transport data between end-points, but they must be considered as "*active*" parties that have their own behavior and provide services. Hence, since they actively affect the overall system's performance and behavior, they cannot be taken apart but must be considered, along with all their characteristics, within the design process. According to this and due to the high degree of network heterogeneity inherent to the B3G application domain, it is important to raise up some characteristics such as *network type*, *quality of service*, *security*, *network services*, etc.

This paper presents an ongoing work on investigating B3G-oriented application modeling. In particular, we show how, starting from a small survey of the networks involved in B3G scenarios, it has been elicited a set of properties that characterizes B3G-oriented applications and, then we present an enhanced UML profile [24] for explicitly defining and analyzing SAs that take into account these properties of interest.

The paper is organized as follows: Section 2 first describes the different networks available in the B3G application domain and then presents the sets of properties elicited. Section 3 presents the B3G-oriented UML profile we designed while Section 4 discusses the analysis aspects related to the B3G application domain. Finally, Section 5 concludes and discuss future work.

2 B3G Network Properties Elicitation

B3G network technology indicates the convergence of telecommunication networks with IP based networks. Services associated with B3G provide the ability to transfer both voice data and non-voice data through different types of network, either wired or wireless. In this setting, wireless networks are categorized into three groups based on their coverage range: (*i*) Wireless Wide Area Networks (WWAN) that includes wide coverage area technologies such as 3G cellular (UMTS), Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS), i-Mode, etc. . . , (*ii*) Wireless Local Area Networks (WLAN) that includes 802.11, HiperLAN, and several others, and (*iii*) Wireless

Personal Area Networks (WPAN) that represents wireless personal area network technologies such as Bluetooth and IrDA.

In this section we discuss the networks state-of-the-art by eliciting the properties of interest needed to architect B3G applications.

2.1 Networks in B3G

The widely accepted way to achieve interworking between heterogeneous network is the use of TCP/IP protocol suite. Figure 1 shows how the protocol stacks, implemented by (some) networks involved in B3G scenarios, relate to TCP/IP standard. In the following, we give a high-level description of such networks and their properties[1], namely Quality of Service (QoS), security, etc. . .

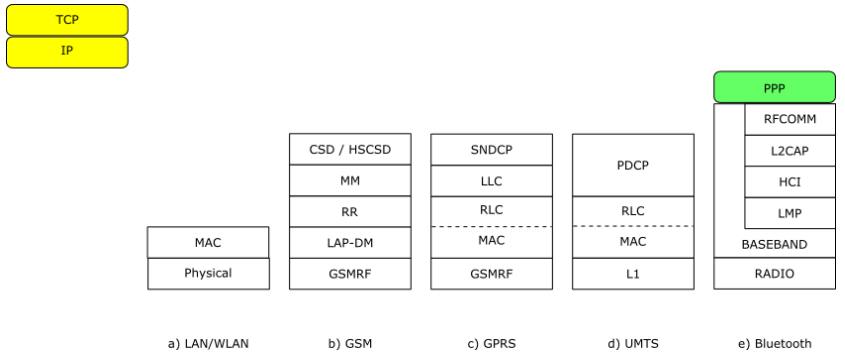


Fig. 1. Network protocol stacks underlying TCP/IP

Ethernet (IEEE 802.3) – Nowadays, the prominent technology for providing wired LANs is Ethernet [16]. Based on the shared-bus concept, it operates different data transfer rates (ranging from 10Mbps to 1Gbps) but it does not provide any support for QoS management neither any other type of service. As shown in Figure 1.a, using TCP/IP over this network does not require any additional effort.

WiFi (IEEE 802.11) – The WiFi 802.11 standard [15] (afterward enhanced by other task groups) defines the MAC and Physical levels in order to create Wireless LANs. It is worth noticing that WiFi network may operate in two different modes: (*i*) *infrastructure mode* and (*ii*) *ad-hoc*. While *infrastructure mode* defines a controlled network composed of a number of managed fixed nodes that provides access points to wireless clients, *ad-hoc mode* defines a self-managed network where wireless clients directly communicate each other without any control. Also in this case, using TCP/IP over this network does not require any additional effort (see Figure 1.a).

[1] For any further detail, please refer to the relative citations.

Global System for Mobile Communications (GSM) – The Global System for Mobile Communications standard [22] specifies infrastructures and services for digital cellular networks. GSM is based on circuit-switched protocol, where a traffic channel is allocated to a user for the entire duration of a call and it is simply unused if there is no data to be transmitted during the call. This limits both the data rates and the number of users that can be supported. Moreover, GSM specifies two circuit-switched protocols to tunnel and convoy data throughout the network. Thus, GSM clients can use TCP/IP directly on top of them (see Figure 1.b).

General Packet Radio Service (GPRS) – General Packet Radio Service (GPRS) [1] is a mobile communication standard based on packet-switched radio transmission. The main advantage over circuit-switched radio technologies is its handling of the radio resources. GPRS is structured as a GSM overlay, although it does require some changes to a few of the basic GSM network elements. GPRS provides an architecture for integrating external packet-data networks (e.g., the Internet backbone wired network) and mobile stations (i.e., cell phones, PDAs, and other such mobile devices). Thus, mobile stations are enabled to use TCP/IP based protocols directly on top of GPRS (see Figure 1.c).

Universal Mobile Telecommunication System (UMTS) – The Universal Mobile Telecommunication System (UMTS) [2] is part of the 3G Network family which defines both the circuit-switched and the packet-switched transmission for voice and data communications relatively. Also in this case, clients can use TCP/IP based protocols directly on top of UMTS (see Figure 1.d). Moreover, important characteristics of UMTS are (*i*) the high data transfer rate (up to 2Mbps) and (*ii*) the ability to negotiate/manage different fixed classes of QoS [3].

Bluetooth – The Bluetooth [8] technology is the de-facto standard for achieving short-range WPANs. In particular, Bluetooth allows users to set up wireless ad-hoc networks that achieve real-time voice and data exchange between limited-resources devices. However, Bluetooth does not directly support TCP/IP protocols and then the use of PPP protocol [23] is required (as shown in Figure 1.e).

2.2 QoS–Related Properties of Interest

This section describes the characteristics of the networks introduced above by summarizing their properties of interest. In particular, we firstly presents common QoS attributes that apply to all the B3G networks and, then we describe a set of enhanced QoS attributes that are specific of telecommunication networks such as GSM, GPRS, UMTS.

General QoS Properties – The set of general QoS properties is composed of few attributes that identify the basic metrics needed for describing the performance of a network. In particular, they relates to the *bandwidth*, *transmission time*, *packet ordering* and *error ratio* provided by the network under analysis:

- `Max Bitrate`: identifies the maximal data transfer rate that can be sustained between two end-points.
- `Transfer delay`: measures the time elapsed for delivering a packet from a sender, trough the network, to a receiver.
- `Jitter`: is calculated as the variation of the `transfer delay` attribute from one packet to the next packet within the same packet stream/flow.
- `Delivery order`: indicates whether the network shall provide in–sequence packets delivery or not.
- `Packet Error Ratio`: is the fraction of packets lost or detected as corrupted or erroneous.

Advanced QoS Properties – In order to fully describe B3G-related QoS properties we need to extend the set described above by adding some attributes specifically related to the GSM, GPRS, UMTS telecommunication networks. In fact, due to the inherent complexity of such networks, in terms of infrastructures and provided services, the general QoS properties set is not sufficient to describe all possible QoS purposes.

To this extent, UMTS QoS specification [3] defines four different classes of QoS related to the services provided by the network: (*i*) *Conversational* defines the QoS needed for real-time video/audio conversations (bidirectional), (*ii*) *Streaming* defines the QoS needed for real-time video/audio streaming (unidirectional), (*iii*) *Interactive* defines the QoS needed for achieving data communication used for request/response patterns (i.e., HTTP interactions) and, (*iv*) *Background* defines the QoS needed for providing background traffic (i.e., email, SMS/MMS messaging). Such classes are defined basing on both the general attributes detailed above and the following list of advanced attributes[2]:

- `Traffic Class`: is a string which identifies the class membership (i.e., "Conversational", "Streaming", "Interactive", "Background").
- `Guaranteed Bitrate`: defines the guaranteed number of bits delivered within a period of time, divided by the duration of the period.
- `Max. SDU size`: is the maximum size (number of octets) for which the network shall satisfy the QoS negotiated.
- `SDU format info`: is a list of possible exact size of SDU. This allows the up-layer application to specify the SDU size in order to let the bearer to be less expensive.
- `Residual bit error ratio`: is the undetected bit error ratio in the delivered SDU.
- `Delivery of erroneous SDUs`: specifies whether SDUs detected as erroneous shall be delivered or discarded.
- `Traffic handling priority`: represents the relative importance for handling of all SDUs belonging to the UMTS bearer.

[2] The following QoS attributes defined by UMTS cannot be directly applied to GSM and GPRS networks. However, the UMTS QoS Specification [3] defines how to map QoS attributes from UMTS to GSM/GPRS in order to achieve interworking.

- `Allocation/Retention priority`: is the relative importance for allocation/retention of UMTS bearer.
- `Source Statistic descriptor`: may be either "speech" or "unknown" and defines the characteristics of the of SDU's source.
- `Signaling Indication`: indicates if the SDU is for signaling or not.

2.3 Network Services

As introduced above, in our vision B3G networks are no longer "*passive*" entities, but they are "*active*" parties that have their own behavior and provide services to the end-users. Hence, in this section, we briefly describe some of these services.

IEEE 802.3 and 802.11 – While Ethernet does not provide any service, WiFi offers some basic security services such as identification, encryption and authentication.

- Service Set Identifier (SSID) is a 32 alphanumeric characters code attached to all packets transmitted on a WiFi network to identify each packet as part of that network. In order to communicate each other, all the clients accessing the network must share the same SSID.
- Wi-Fi Protected Access (WPA and WPA2) is a encryption protocol designed to secure wireless networks. In order to access the network every clients must be provided with a pre-shared key that is used to encrypt/decrypt network packets.

Global System for Mobile Communications (GSM) – Services provided by the GSM network are:

- Basic voice call services (i.e., incoming and outgoing call management) and optional enhanced voice call services (i.e., Call Forwarding, Call Hold, Call Waiting, Barring of Outgoing/Incoming Calls, ...)
- A circuit-switched data transfer service which allows users to send/receive data streams. In particular, GSM defines two distinct circuit-switched data protocols: (*i*) the Circuit Switched Data (CSD) and the (*ii*) High-Speed Circuit-Switched Data (HSCSD). Due to a fixed amount of bandwidth is dedicated to connections over CSD and HSCSD (9.6Kbps and 14.4Kbps respectively), they are charged on a per-second basis, regardless of the amount of data sent over the link.
- Short Message Service (SMS) which allows users to send/receive textual messages.
- mobility management service which manage the client mobility by supporting handover and network interoperability (aka global roaming).
- the GSM network provides a secure access mechanism based on personal information stored on the Subscriber Identity Module (SIM) card. In particular, the communications between the subscriber and the base station is encrypted by using a crypt variable stored on the SIM.

General Packet Radio Service (GPRS) – As introduced above, GPRS is an extension of GSM which aims to upgrade the GSM data transfer services. In particular, services provided by GPRS are:

- Packet-switched data transmission protocol where bandwidth is used only when there is actually data to transmit. Hence, the billing policy for this type of connection is by the kilobyte instead of by the second.
- Short Message Service (SMS) which allows users to send/receive textual messages.
- Since GPRS is based on existing GSM networks, they share the same secure access mechanism and encryption facility.

Universal Mobile Telecommunication System (UMTS) – UMTS services are grouped into three classes: (*i*) *teleservices* which support the same teleservices handled by GSM (i.e., speech, SMS), (*ii*) *facsimile services* which provide the ability to send and receive fax, and (*iii*) *bearer services* which provide the capability for data transfer between end-points. In particular, the *bearer services* are:

- *Circuit switched data services* and *real-time data services* which allow for interworking with the GSM network.
- *Packet switched data services* which allow for interworking with packet networks such as IP-networks and LANs. This service provide also mechanisms for ensuring packet based handover between GSM and UMTS.

Bluetooth – Bluetooth provides services for transmitting both voice and data, and for service discovery:

- Asynchronous Connectionless (ACL) provides a packet switched service for data transmission.
- Synchronous Connection Oriented (SCO) service provides a circuit switched data service for audio/voice transmission.
- Service Discovery Protocol (SDP) allows for discovery of services on Bluetooth enabled devices.

3 An UML Profile for B3G Software Architectures

In previous section, we presented the common networks involved in B3G scenarios by eliciting their main properties of interest. In particular, we discussed their QoS properties and the services they offer to the end-user. As previously remarked, while designing B3G-oriented application these characteristics should not be abstracted away by SA descriptions, but rather they should be considered and analyzed since the early stages of the software life-cycle.

To this extent, in this section, we describe a UML 2.0 Profile for specifying B3G-oriented SAs which take into account all the discussed properties. In particular, we exploit the Dually approach [17][13] by extending it with the features needed to describe B3G SAs.

3.1 Background: The DUALLY Profile for Software Architectures

Even though UML is considered a well known standard-de-facto notation for specifying and modeling software systems, it does not obey the Architectural Description Languages (ADLs) [20] peculiarities and lacks some relevant aspects. To this extent, many approaches have been proposed to fill this gap and allow for SA specification in UML [19][27][18].

Among them, a relevant contribution is provided by DUALLY. DUALLY is a UML profile which provides a minimal core set of architectural concepts along with the ability to extend it in order to fit any specific need. Furthermore, the use of DUALLY allows engineers to carry on different types of analysis (i.e., Model checking, Testing, Performance, ...) on the defined SA. Figure 2 presents the DUALLY profile and describes how the common UML elements [24] have been extended in order to meet the requirements posed by SA specifications.

SA-Component – UML 2.0 defines a component as "*a subtype of Class which provides for a Component having attributes and operations, and being able to participate in Associations and Generalizations*". Moreover, "*Component may optionally have an internal structure and own a set of Ports that formalize its interaction points*". Hence, DUALLY simply maps SA-Components directly into UML components.

SA-Connectors – UML 2.0 introduces, for the first time, the concept of connector in UML. In particular, it defines two types of connector: (*i*) *assembly*, which represents a binding between an output port and an input port, and (*ii*) *delegation*, which binds an external component port with its internal implementation. However, both the *assembly* and the *delegation* connector definitions lack expressiveness and do not fulfill the requirements needed to specify an SA according to [25][14]. Different solutions have been proposed to fill this gap [18]. DUALLY represents an SA-Connector by means of a stereotyped UML component which embodies both functional and non-functional properties of the connectors.

SA-Channels – In UML 2.0: "*An assembly connector is a connector between two components that defines that one component provides the services that another component requires. An assembly connector is a connector that is defined from a required interface or port to a provided interface or port*". In DUALLY, the SA-Channel stereotype is mapped into the UML assembly connector (see Figure 2).

SA-Relationships – While modeling SAs, a key issue is to analyze and keep trace of the relationships that occur among components [28][29]. In UML, "*a dependency is a relationship that signifies that a single or a set of model elements requires other model elements for their specification or implementation. This means that the complete semantics of the depending elements is either semantically or structurally dependent on the definition of the supplier element(s)*". Hence, DUALLY implements the SA-Relationship by using the UML Dependency core element.

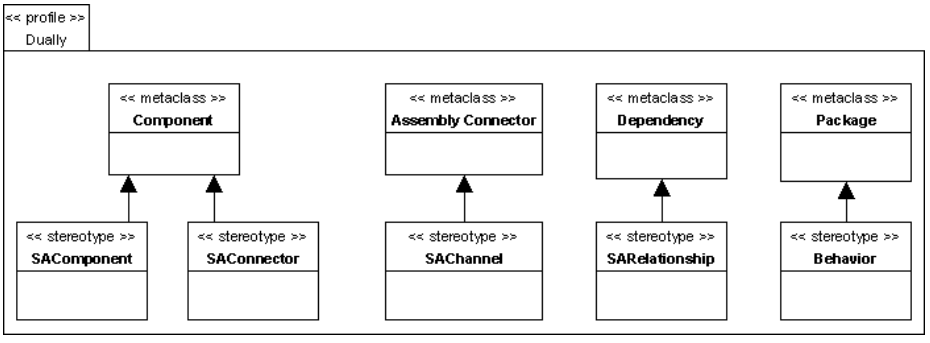


Fig. 2. DUALLY UML Profile

SA-Behavior – In order to carry on SA analysis DUALLY needs some additional information about the system dynamics by means of components internal behavior and components interactions. To this extent, DUALLY makes use of the UML State Machine for modeling components behavior and Sequence Diagrams for specifying components interactions.

3.2 Extending SA-Channels for B3G

The DUALLY profile described above fulfills the gap between ADLs and UML 2.0 in specifying SAs. However, it does not cope the needs posed by B3G-oriented SA specification. In fact, notwithstanding the SA-Connector is able to specify complex relationships between SA-Components, it cannot be used to describe networks properties (i.e. network type, QoS, security, network services, etc) which characterize the communication channels in B3G scenarios.

Hence, starting from the SA-Channel defined above, we extend it by creating a stereotyped class hierarchy, which exploits the characteristics of the different networks involved in B3G scenarios.

Referring to Figure 3, an SA-Channel can be specialized into a *Local Channel* (`CLocal`) which refers to local component interactions, or into a *Networked Channel* (`CNet`) which provides remote networked communications. Furthermore, `CNet` is specialized into the `CN_Wired` and `CN_Wireless` subtypes which provide *Wired Network Channel* and *Wireless Network Channel* respectively. While `CN_Wired` is not further specialized, `CN_Wireless` has `CNW_WWAN`, `CNW_WLAN` and `CNW_WPAN` as subtypes. `CNW_WWAN` defines *Wireless Wide Area Networks Channel*, `CNW_WLAN` represent *Wireless Local Area Networks Channel* and `CNW_WPAN` is *Wireless Personal Area Networks Channel*. Finally, while `CNW_WWAN` is specialized into `WWAN_GSM`, `WWAN_GPRS` and `WWAN_UMTS` which represent a possible set of formal telecommunication networks, `CNW_WLAN` is specialized into `WiFi_Structured` and `WiFi_Ad-Hoc` which represent two different types of wireless IP network.

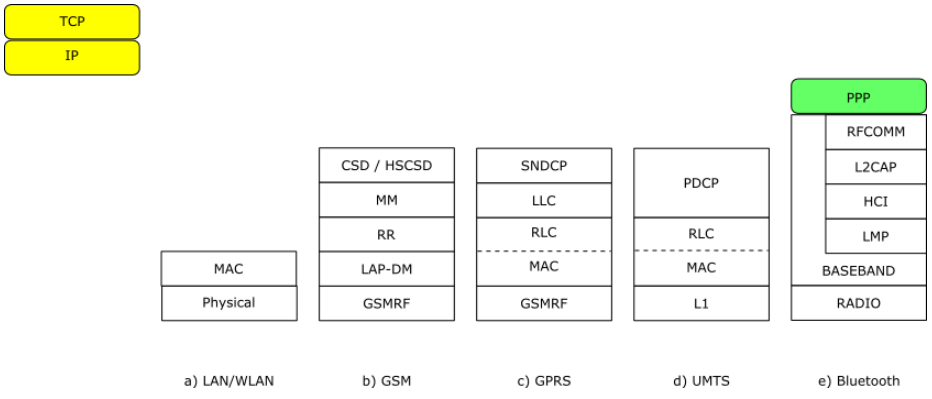


Fig. 3. SAChannel Hierarchy Tree

SAChannel – It represents the root of our hierarchy and provides the `Name` attribute which allows us to uniquely identify the instantiated channel within the SA under definition. It also defines the following naive services :

1. `void` **send**`(`*`pkt, dest`*`)`: provides the service for sending a data packet *`pkt`* to a given destination *`dest`*.
2. `pkt` **receive**`()`: provides the service for receiving a data packet *`pkt`* from the network.

It is worth noticing that these services might be overridden/overloaded by the subclasses in order to meet the requirements specified by the different network they implement. For example, given a network specification which defines a packet-switched data transfer, both **send** and **receive** services must be implemented accordingly.

CLocal – `CLocal` refers to local communication channels used by processes, tasks and components residing on the same machine. CLocal overrides the services defined in `SAChannel` and defines an additional attribute, named `Type`, which indicates the type of communication implemented by the channel: i.e. pipe, shared memory, etc. . .

CNet – On the other side, `CNet` refers to remote communication channels used by components to interact with each other across a network. It extends `SAChannel` by defining a minimal set of QoS attributes according to the General QoS Properties discussed in Section 2.2.

CN_Wired – This class concerns the communication achieved by using a wired network (i.e. Ethernet, ATM, USB, . . .). It overrides the services defined by its ancestors and instantiates the QoS attributes.

CN_Wireless – This class identifies a communication channel implemented by means of a wireless link. `CN_Wireless` defines some additional services:

1. `ack` **connect**(*ID*): provides the service for joining the wireless network. It requires to provide an *ID* which uniquely identifies the client within the network.
2. `void` **disconnect**(*ID*): provides the service for disconnecting from the network. The *ID* is used to clean up the network clients table.

CNW_WLAN – This class defines a communication channel implemented by means of a WLAN. It instantiates the inherited QoS attributes and defines some additional services:

1. `void` **enableWPA**(*pwd*): enables the WPA service by providing the required password.

WiFi_Structured – Structured WiFi channel instantiates the inherited attributes and define the following service:

- `[ID]` **getSSID**(): provides the service for retrieving the SSIDs of the networks in range. `[ID]` represents an array containing all the SSIDs retrieved.

WiFi_Ad-Hoc – Operating in ad-hoc mode requires devices within range of each other to discover and communicate in peer-to-peer fashion without involving central access points. Hence, the Ad-Hoc WiFi channel instantiates the inherited attributes and define the following service:

- `[ID]` **peerDiscovery**(): provides the service for retrieving the ID of the peers in range. `[ID]` represents an array containing all the IDs retrieved.

CNW_WWAN – This class identifies a communication channel implemented by means of a wireless telecommunication network. In particular, it extends the set of attributes according to the advanced QoS properties discussed in Section 2.2, appropriately overrides inherited methods, and provide the following common services:

- `void` **mobilityManagement**(*ID*): provides the service for managing mobility issues, such as handover and roaming.
- `void` **billing**(): provides the service for managing services billing.
- `void` **sendSMS**(*MSG*): provides the service for sending Short Messages.
- `MSG` **getSMS**(): provides the service for receiving Short Messages.

WWAN_GSM – The GSM Class instantiates all the attributes defined earlier, overrides inherited methods (i.e., connect, send, ...), and provides the following services:

- `Call` **getIncomingCall**(*ID*): provides the service for receiving incoming voice calls.
- `Call` **setOutgoingCall**(*NUM*): provides the service for setting an outgoing call.

WWAN_GPRS – The GPRS Class does not provide new services but overrides inherited methods (i.e., connect, send, ...) and instantiates all the attributes defined earlier according to its specification.

WWAN_UMTS – The UMTS Class instantiates all the attributes defined earlier, overrides inherited methods (i.e., connect, send, ...), and provides the following services:

- `Call` **getIncomingCall**(*ID*): provides the service for receiving incoming voice calls.
- `Call` **setOutgoingCall**(*NUM*): provides the service for setting an outgoing voice call.
- `void` **sendFAX**(*MSG*): provides the service for sending a Fax.
- `MSG` **getFAX**(): provides the service for receiving a Fax.

WPAN – WPAN Class identifies a channel implemented by means of a personal area network. It defines some common services and can be further derived into more specific classes:

- `[ID]` **peerDiscovery**(): provides the service for retrieving the ID of the peers in range. `[ID]` represents an array containing all the IDs retrieved.
- `Call` **setVoiceTrasmission**(*ID*): provides the service for managing audio/voice transmission to the device identified by `ID`.

3.3 Early Example

In this section, We present a simple example which aims to summarize the characteristics introduced above. Figure 4 shows an SA composed of a number of components: an `Application` is connected through a generic network to the `Web Server` that, by using a `XParlay` connector, is able to interact to different mobile clients (namely `GSM Phone`, `Laptop` and `UMTS PDA`) through WAN networks. Moreover, while the GPRS Mobile Termination (`GPRS MT`) acts as network access point for the `Laptop`, the `UMTS PDA` is directly connected to `Application` by means of a PAN channel and to `Web Server` trough the UMTS network. Note that, while `Web Server` accesses the `XParlay` connector services by using a `CN_Wired` channel, the wireless clients are linked to it by using different `CNW_WWAN` channels. Here, the role of the `XParlay` connector is to make the set of services implemented within WAN networks (i.e., messaging, presence, localization, phone calls) available to the web server [21].

In this setting, let the `UMTS PDA` holder want to send an SMS to both the `GSM Phone` and the `Laptop` clients. She can choose if sending the SMS by using the WWAN facilities (then paying the cost of two SMS) or using her web account that allows her to send free SMS. In the latter case, she has a further choice: she can browse the web site by using either the UMTS data transfer service (then paying the service cost) or a PAN network in order to use the `Application` component as router to access Internet and interact to `Web Server` for free.

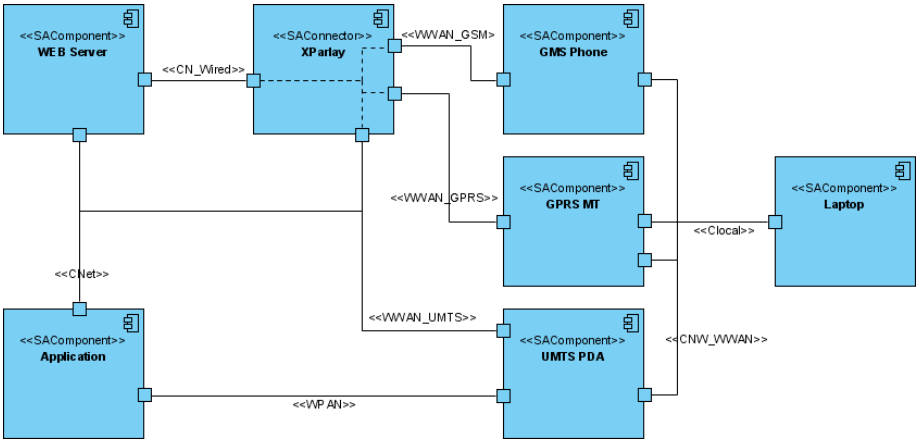


Fig. 4. B3G early example

Even though this example is really simple and the SA is composed of few components, it shows the overall complexity inherent to B3G scenarios. Modeling the same SA by using either ADLs or SA-oriented UML profiles would hide the main characteristics of this type of networks (i.e., QoS properties) and the services they provide (e.g, SMS service).

4 B3G Software Architecture Analysis

As remarked in previous section, SA descriptions allow for analysis and validation of architectural choices obtaining better software quality and shorter time-to-market development [6].

Since SA represents the first, in the development life-cycle, complete description of the system under development, carrying on accurate analysis at this stage would speedup and improve the next life-cycle phases. This is even more important while developing B3G Applications. In fact, since the complexity of these systems comes from both the application itself and the underlay networks, having qualitative and quantitative analysis results allows software engineers to understand in advance the behavior/misbehavior of the system.

To this extent, the B3G UML profile presented in previous section provides the ability to perform the following analysis by using UML-based automated tools: (*i*) *SA Model Checking* which allows for checking architectural model consistency, (*ii*) *Performance Analysis of SA* which allows for making quantitative analysis on SAs, and (*iii*) *SA-driven Testing* which allows for checking whether the implementation under test fulfills the architectural specification, and (*iv*) *Runtime SA-based System Reconfiguration* which allows for monitoring and reconfiguring the system at runtime.

Model Checking Software Architectures – Model-checking is a well known verification technique which aims to validate system specifications against functional properties of interest expressed by means of formal languages [11]. The use of such a technique at the architectural level allows architects to design correct SAs (i.e., system specifications) that satisfy the functional requirements (i.e., properties of interests).

The B3G profile presented in Section 3 exploits the functionalities provided by DUALLY which in turn is fully integrated into the CHARMY tool [10], a framework for specifying and model-checking SAs. This allows us to model-check B3G applications by taking into account the networks low-layer properties usually abstracted by ADLs.

Performance Analysis of SA – *Software performance* is the process of *predicting* and *evaluating* whether a system meet the user performance requirements [4]. Performing such an analysis requires to use modeling languages specifically designed for this purpose - e.g., Queuing Network (QN), Petri Nets (PN), etc.

However, given a UML-based SA specification, it is possible to automatically generate the relative QN model and carry on the analysis [12]. This, allows us to automatically transform B3G-oriented SAs into QNs and then carry on a performance analysis that takes into account also the network QoS properties discussed in Section 2.

SA-based Testing – While SA model-checking allows for an exhaustive and automatic analysis of the system model, SA-based testing aims to validate the implementation conformance to the SA model [7]. In UML-based SA specification, SA-based testing uses UML notations such as *State Diagrams* and *Sequence Diagrams* for describing component/connector behavior and test cases respectively [12].

Hence, casting this to the B3G-oriented SA presented above, allows us to automatically generate test cases that consider also the characteristics of the involved networks.

Runtime SA-based System Reconfiguration – In deploying complex distributed software systems, runtime QoS management is an important issue to address. Self-adaptation, based on on-line monitoring and dynamic reconfiguration, is considered a useful technique to solve it. To this extent, the use of SA performance analysis, for evaluating the actual system status and choosing the next system configuration, achieves fine tuned reconfiguration process that overcomes the observed QoS problems [9].

Thus, using the B3G-oriented UML profile for specifying the SA allows for evaluating the actual performance and for reconfiguring the system also with respect to the QoS properties of the underlay networks. Moreover, this also allows for designing applications able to dynamically negotiate Service Level Agreements (SLA) according to the connectivity and QoS they perceive, as long as properties match the requirements.

5 Conclusions and Future Work

B3G technology conceives the convergence of telecommunication networks with IP based networks for providing enhanced services. This new class of services exploits the ability to transfer both voice data and non-voice data through different kinds of networks, either wired or wireless. This define a new application domain where applications can not only interact with each other, but they can also interact with non-software entities (i.e., humans). This new application domain requires for an adaptation/revision of the software life-cycle and the related tools.

To this extent, we presented our ongoing work on investigating how to address the designing and developing of B3G-oriented applications. In particular, we first described the different network involved in B3G scenarios and their main characteristics. Then, we proposed an extended UML Profile which allows for defining B3G-oriented SA and for performing different kind of analysis which aim to validate B3G-oriented applications with respect to both qualitative and quantitative requirements.

The work discussed herein represents the first step of a long term research. In fact, we plan to use the proposed B3G-oriented UML profile for designing and evaluating the PLASTIC middleware. PLASTIC is a IST Project [26], we are involved in, which aims to construct a platform for delivering software services deployed over B3G networks. Main characteristics of the PLASTIC Services is that they will be adaptive to the environment with respect to resource availability and delivered Quality of Service (QoS), via a development paradigm based on Service Level Agreements (SLAs) and resource-aware programming.

References

1. 3rd Generation Partnership Project Technical Specification Group. Digital Cellular Telecommunications System (Phase 2+) - General Packet Radio Service (GPRS) Service Description - Services and System Aspects - Stage 2, Jan. 2002.
2. 3rd Generation Partnership Project Technical Specification Group. Universal Mobile Telecommunications System (UMTS) - Services and service capabilities, Sept. 2005.
3. 3rd Generation Partnership Project Technical Specification Group. Universal Mobile Telecommunications System (UMTS) - Universal Mobile Telecommunications System (UMTS) - Quality of Service (QoS) concept and architecture, June 2005.
4. S. Balsamo, A. Di Marco, P. Inverardi, and M. Simeoni. Model-based performance prediction in software development: a survey. *IEEE Transactions on Software Engineering*, 30(5):295–310, May 2004.
5. L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice.* Addison-Wesley, Massachusetts, 1998.
6. M. Bernardo and P. Inverardi. *Formal Methods for Software Architectures, Tutorial book on Software Architectures and Formal Methods.* SFM-03:SA Lectures, LNCS 2804, 2003.

7. A. Bertolino and P. Inverardi. Architecture-based software testing. In *Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops*, pages 62–64, New York, NY, USA, 1996. ACM Press.
8. Bluetooth. Bluetooth core specification v2.0. http://www.bluetooth.com.
9. M. Caporuscio, A. Di Marco, and P. Inverardi. Model-based system reconfiguration for dynamic performance management. *Journal of Systems and Software*, 2006. To appear.
10. Charmy Project. Charmy web site. http://www.di.univaq.it/charmy, February 2004.
11. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press, 2001.
12. V. Cortellessa, A. D. Marco, P. Inverardi, H. Muccini, and P. Pelliccione. Using uml for sa-based modeling and analysis. In *Proceedings of Int. Workshop on Software Architecture Description & UML*, Lisbon, Portugal, Oct 2004.
13. D. Di Ruscio, H. Muccini, P. Pelliccione, and A. Pierantonio. Towards weaving software architecture models. In *Proocedings of the MBD-MOMPES 2006 Workshop*, Potsdam, Germany, March 2006.
14. D. Garlan and M. Shaw. An introduction to software architecture. In V. Ambriola and G. Tortora, editors, *Advances in Software Engineering and Knowledge Engineering*, pages 1–39, Singapore, 1993. World Scientific Publishing Company.
15. IEEE 802.11 Working Group. IEEE 802.11 Wireless Local Area Networks (WiFi). http://grouper.ieee.org/groups/802/11/.
16. IEEE 802.3 Working Group. IEEE 802.3 CSMA/CD (ETHERNET) based Local Area Networks. http://grouper.ieee.org/groups/802/3/.
17. P. Inverardi, H. Muccini, and P. Pelliccione. DUALLY: Putting in synergy UML 2.0 and ADLs. In *Proocedings of Int. 5th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, Pittsburgh, November 2005.
18. J. Ivers, P. Clements, D. Garlan, R. N. B. Schmerl, and J. R. O. Silva. Documenting component and connector views with uml 2.0. Technical report, Carnegie Mellon, Software Engineering Institute, Pittsburgh, USA, April 2004.
19. N. Medvidovic, D. S. Rosenblum, D. F. Redmiles, and J. E. Robbins. Modeling software architectures in the unified modeling language. *ACM Trans. Softw. Eng. Methodol.*, 11(1):2–57, 2002.
20. N. Medvidovic and R. N. Taylor. A classification and comparison framework for software architecture description languages. *IEEE Transaction on Software Engineering*, 26(1):70–93, 2000.
21. A.-J. Moerdijk and L. Klosterman. Opening the Networks with Parlay/OSA: Standards and Aspects Behind the APIs. *IEEE Network*, 17(3):58–64, May-Jun 2003.
22. M. Mouly and M. Pautet. Current evolution of the GSM systems. Technical report, IEEE Pers. Commun., 1995.
23. Network Working Group. The Point-to-Point Protocol (PPP). RFC 1661, July 1994.
24. Object Management Group OMG. Unified Modeling Lanauge Specification - UML 2.0 Superstructure Specification. OMG Document: formal/05-07-04, Aug 2005.
25. D. E. Perry and A. L. Wolf. Foundations for the study of software architecture. *SIGSOFT Software Engineering Notes*, 17(4):40–52, Oct 1992.
26. Plastic Consortium. Providing Lightweight and Adaptable Service Technology for pervasive Information and Communication. http://www.ist-plastic.org/. IST STREP Project.

27. S. Roh, K. Kim, and T. Jeon. Architecture modeling language based on uml2.0. In *Proocedings of 11th Asia-Pacific Software Engineering Conference (APSEC'04)*, pages 663–669.
28. M. Vieira and D. Richardson. Analyzing dependencies in large component-based systems. In *Proceedings of the 17th IEEE International Conference on Automated Software Engineering*, pages 241– 244, Edinburgh, UK, September 2002.
29. M. Vieira and D. Richardson. The role of dependencies in component-based systems evolution. In *IWPSE '02: Proceedings of the International Workshop on Principles of Software Evolution*, pages 62–65. ACM Press, 2002.
30. T. Zahariadis and B. Doshi. Applications and services for the B3G/4G era. *Wireless Communications, IEEE*, 11(5), Oct 2004.