MDA Approach for the Development of Embeddable Applications on Communicating Devices

Eyob Alemu[1], Dawit Bekele[2], Jean-Philippe Babau[3]

[1]MicroLink Information Technology College, P.O.Box 5/1030, Addis Ababa, Ethiopia

[2]Department of Computer Science, Addis Ababa University,
P.O.Box 3479, Addis Ababa, Ethiopia

[3]CITI Laboratory INSA, LION, France. 20, avenue Albert Einstein, 69621 Villeurbanne cedex

Abstract. Focusing on the communications subsystem of embedded platforms, this paper introduces an MDA based approach for the development of embeddable communicable applications. A QoS aware and resource oriented approach, which exhibits the runtime interaction between applications and platforms, is proposed. Reservation based (typically connection oriented) networks are specifically considered.

1 Introduction

Recent technological advances are making possible the embedding of both processing and communication functions in highly integrated, low-cost devices such as PDA's and mobile phones. This is promoting the use of a distributed approach in many application fields including embedded systems, which is now leading to the current and future realm of pervasive computing [1]. As communication is extensively used as an interaction medium for such devices, it makes up the most important platform service in such distributed systems. Today, a large variety of networks are currently available to build distributed embedded systems. Moreover, most of them are competing in the same domain of application. For example, CAN [10] and I2C [11] are used in automotive and industrial systems, whereas Bluetooth [8] and IrDA [9] are used for interconnecting peripherals and portable devices. The middleware platforms considered so far in the MDA such as CORBA are heavyweight and do not generally fit the domain of embedded systems. Moreover, resource limitation is a typical characteristic of this domain, which makes the issue of Quality of Service (QoS) a major concern. In this paper, we propose a QoS aware MDA approach for the development of embeddable communicable applications focusing on the communication subsystem. The approach shows an adaptation of the enterprise MDA towards addressing platform variability in the development of applications for embedded devices.

Alemu E., Bekele D. and Babau J. (2006).
MDA Approach for the Development of Embeddable Applications on Communicating Devices.
In *Proceedings of the 2nd International Workshop on Model-Driven Enterprise Information Systems*, pages 88-97
DOI: 10.5220/0002478300880097

2 MDA and Embedded Systems

With the general MDA specification, systems are first modeled using Platform Independent Models (PIMs). The next step transforms the PIMs to Platform Specific Models (PSMs) through a systematic transformation process. In recent years, the capability of the hardware devices is enhanced to provide extensive interfaces and the possibility of hosting applications of different types. Programmable interfaces and software abstraction layers are becoming possible to support flexible system developments [5]. This evolutionary enhancement of embedded systems from their specific purpose functionality to a more general, multipurpose and more intelligent capability is making the devices not only capable of hosting embedded applications but also communicate with each other to share resources and to transfer information. The current and future vision of pervasive computing can benefit from this advancement since it makes extensive use of embedded devices. Besides the software development complexity of this domain, platform variation is a very critical problem. Moreover, resource limitation has made the development to focus on QoS and platform level issues.

2.1 The Embedded System Platforms

In [4], a definition for an embedded platform is presented as a "family of Micro-Architectures possibly oriented towards a particular class of problems". A recent initiative in this domain is the platform-based approach proposed in [5] and further improved in [4]. Using Platform Based Design approach, the platforms for embedded systems are modeled at different abstraction levels so that developers could choose the appropriate abstraction level that can avoid their concern about the details of the platforms. A typical layered architecture of an embedded platform is shown below (Fig 1) [4].

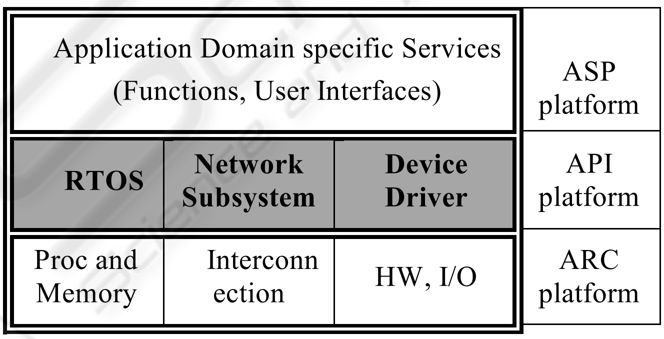

Fig. 1. Platform descriptions at different levels.

As shown in Fig 1, the ARC Layer includes a specific family of micro-architectures (physical hardware). The API Layer is a software abstraction layer wrapping ARC implementation details. API presents what kinds of logical services are provided and how they are grouped together and represented as interfaces. ASP (Application Specific Programmable) provides a group of application domain-specific services directly available to users. The API layer is the most useful layer among the three levels providing programmable and interactive interface for upper layer clients and applications [5][6].

2.2 QoS Offered By Embedded Platforms and Networks

QoS requirements specify not what the system does (provides services), but how the system satisfies its client requests while doing what it does [20]. The QoS relationship between the requester and the provider can be viewed from two aspects [3]:

- From Client/Server (horizontal) relationship: in which case a client specifies the required QoS and the server specifies the offered QoS for a negotiated contract.
- From an abstract/concrete (vertical) relationship: in which case the relationship is seen in a layered architecture . The MDA approach that we propose is related with this second aspect.

Considering the embedded networks, the two major categories of QoS mechanisms in Link Layer networks are **Reservation** and **Priority**. In reservation, network resources are allocated based on signaled requests originating from applications. Several parameters are used to define the reservation requirement and provision. Signaling messages are used to exchange such parameters. In prioritization (CAN, I2C), exchanged packets or frames are usually associated with a priority value that defines the handling in relation to other priorities. Several mechanisms for providing QoS exist in both categories. For example Bluetooth and IrDA use different **reservation** mechanisms. This work specifically focuses on the reservation and connection oriented category of the networks.

3 The Proposed MDA Approach

In enterprise MDA, the major focus is on modeling and transformation of functional elements and interfaces of applications from a more abstract to a more refined form, which does not consider the QoS aspects. We argue that such an MDA process is not generally suitable in the current and future embeddable communicable applications. Most of the application models must identify the behavior of their execution environment specially concerning QoS. More specifically, platform models in the embedded system development methodology greatly influence application models. The major concern is how to model the applications in order to use specific environments efficiently. Therefore, the model of the applications usually follows the model of the execution environment or is made along with the design of that specific environment (Co-design). Hence, unlike enterprise systems, the MDA approach for embedded systems in general should be based on the models of the platforms and their abstraction instead of application models and their refinement. The notion of “Abstract platform” [19], tailored with the MDA methodology will leverage the current challenges and visions in the embeddable communicable applications development.

Therefore what we propose is a model driven platform based (resource oriented) and QoS aware approach for embeddable communicable applications. This way the PIM of the platforms will be an abstract model that can be used within the model of the applications. Upon implementation the abstract platform will be mapped with a specific platform through a mapping layer. The mapping layer can target a number of different concrete platforms as shown in Fig 2.

3.1 Analysis of the Embedded Networks

Based on the analysis we have made on the reservation-based networks, the modeling elements are identified to be similar except the QoS expression and mechanism. Therefore, we present here the general model elements.
The objects (entities) identified are:

Channel/Connection: this refers to the channel identified with two endpoints on peers.

Event: every message exchange is produced as an event, which invokes a corresponding operation. It has four types: **Request**, **Indication**, **Response** and **Confirmation**,

QoS_Spec: this represents the QoS constraint (**Offered QoS**) of the link layers.

Service Interface: represents the service entity through which clients interact with the layer.

Classes will be used to represent these four entities. Using the terminology and modeling artifacts in the UML profiles defined in [2] and [3], Channel/Connection is considered as a Resource and QoS_Spec is a QoSConstranit. The other elements are modeled using the standard UML concepts.

3.2 The Platform Independent Model (PIM)

With the MDA standard, the PIM should be semantically similar to the platform models [7]. Hence, it has to reflect the connection oriented and the reservation based nature of the networks. The applications implemented in this network domain are aware of the reservation based and connection oriented nature of the networks. But their design and implementation will be independent of a specific network interface. Therefore the PIM concepts are based on abstract representations of platform specific characteristics. The essential model elements that the PIM must include in an abstract manner are the same as those of the specific platforms, except the QoS expression for which we propose a generic expression named **Flow_Spec** that can represent a reservation request. The **Flow_Spec** is an entity for the QoS as a generic reservation request specification taken from the flow specification proposed in [14] which is a **Token Bucket** based specification. It has been enhanced in [22] and further by Internet Engineering Task Force (IETF) for use in Internet reservation services [22][23]. For the QoS specification at the PIM level, the flow-based approach is selected for a number of reasons: First it is a closer approach to networks and in particular it is more appropriate for the connection oriented and reservation based networks considered in this work. Initially it was proposed for the Internet community. It is also a widely used model to quantitatively specify application requirements on a network. Second, it is more declarative than showing more technical details, which makes it appropriate for a PIM level specification according to the MDA standard . Third, its specification does not target a specific network protocol and reservation mechanism [12]21. This opens the possibility of mapping to many different specific implementations. Fourth, we believe that it is the most appropriate specification that can satisfy both requirements of a PIM stated in [15], i.e. platform independence and mappablity towards concrete platforms. We argue that this type of PIM specification can be transformed to Bluetooth, IrDA and other reservation based networks.

3.3 The Mapping Model

This section presents the detailed version of the proposed MDA approach as shown in Fig 2. The transformation between the PIM and the target network model (PM) is made through the intermediate mapping layer forming a PSM. This will meet the objective of MDA in that communicable embedded applications can be designed and implemented without the concern of the peculiar characteristics of the used network. For simplifying the model, the two concepts, i.e., the Functional Service and the QoS are separated into two groups (packages) as shown in Fig 2.

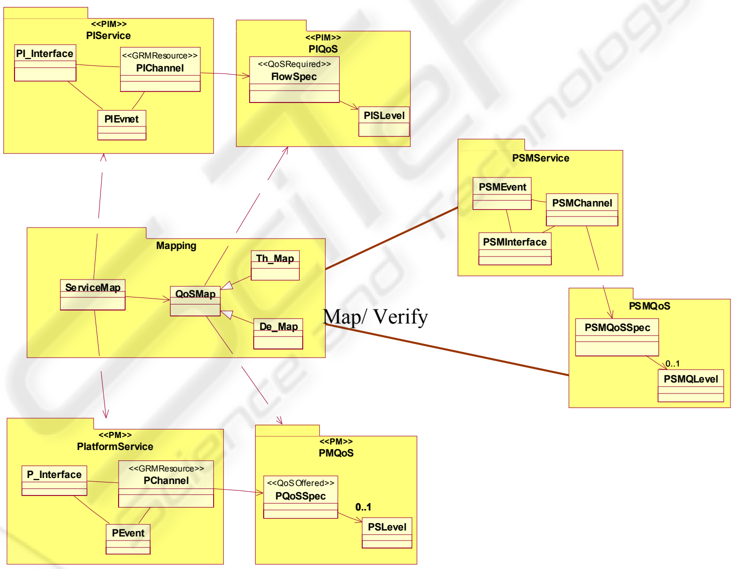

Fig. 2. The proposed MDA approach in UML.

3.5 The Mapping Strategy

The mapping layer will have two parts namely **ServiceMap,** responsible for mapping the functional service interface of the PIM with the PM and **QoSMap**, responsible for transforming the QoS modeling elements. It has two subtypes related to Throughput mappings (Th_Map) and latency mappings (De_Map).

Since the functional service mapping is relatively easier and is not our major focus area, we present here the most important part of the mapping layer, i.e., the QoS mapping. For the QoS mapping, the predictability nature of the specifications is considered. We have divided the QoS mapping strategy into three categories:

1. **Service Level Mapping**: This is done by using the semantics of the three levels at the PIM QoS. (**Guaranteed**, **Controlled Load** and **Best Effort**). Appropriate interpretations of the meanings in each specific network will be identified.
2. **Service Level Mapping**: This is done by using the semantics of the three levels at the PIM QoS. (**Guaranteed**, **Controlled Load** and **Best Effort**). Appropriate interpretations of the meanings in each specific network will be identified.
3. **Throughput Related Mapping:** for this case, we used a concept of Maximum Transmission Boundary (**MTB**) to determine the maximum amount of data bytes transferred within a period of time, based on the parameters for both PIM and PM. Hence, we must have:

$$\mathbf{MTB_{PIM} <= MTB_{PM}}\text{, where } MTB_{PIM} = \min(B + r*T, M + P*T) \tag{1}$$

where B=Bucket size, r = Token rate, T = a time interval for the flow, p = peak rate, and M = Maximum Transmission Unit [16][25]. Similarly, the corresponding value for the PM can be calculated from appropriate parameters.

A. **Latency Related Mapping:** this is done using the explicit specification at the PIM level and estimated from appropriate parameters at the PM level with the following relationship:

$$Latency_{PIM} >= Latency_{PM} \tag{2}$$

3.6 Procedures Used by the Mapping Layer

The mapping layer uses two procedures to link application requests with the underlying network level provisions. The **Map/Transform** procedure transforms and maps parameters from PIM to PM or vise versa. The **Verification** procedure verifies the Required/Offered relationship holds. Based on the requested service level appropriate action will be taken. If Guaranteed, requirements must be satisfied. If Controlled Load, requirements are flexible (negotiable), and if Best Effort, any value offered is accepted.

4 Applicability of the Approach

In this section, we present the applicability of our approach for the IrDA specific platform. We are forced to limit to the case of IrDA only due to page restrictions.

IrDA Platform Model: the IrDA link layer services are presented through the Link Management Protocol (IrLMP) layer which has a relatively similar purpose but slightly different functional services as the Bluetooth L2CAP layer. It also provides a connection-oriented service with a set of parameters for the level of QoS it provides to its clients.

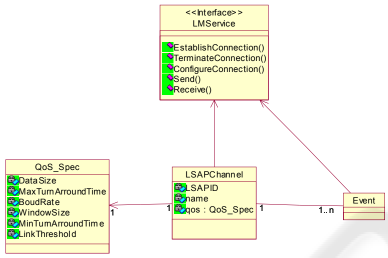

Fig. 3. The IrDA link layer Platform Model.

Since there is a difference in the parameters used to define the QoS provision, an indirect procedure is performed for mapping the PIM with the IrDA PM. The IrDA QoS parameters are defined in [24]. The mapping relationship is shown in Fig 4.

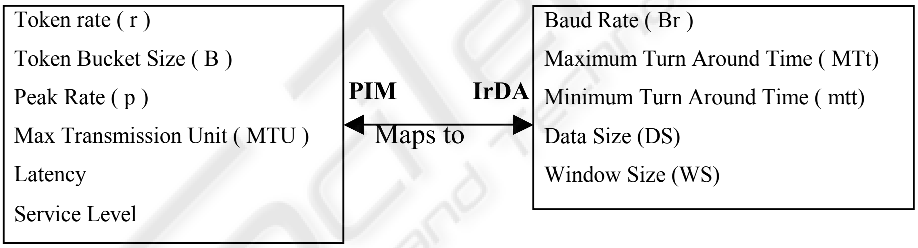

Fig. 4. Mapping the QoS parameters of the PIM and IrDA.

4.1 Mapping the Service Level

In the IrDA specification, there is no distinct expression for service levels. The most common scenario is that for the communication to begin between two application ends, both must agree on the exchanged QoS parameters. The negotiation values are distinct enumerated values. Furthermore, the two ends must honor the agreed upon values throughout the lifetime of the link. However, the PIM level specification takes the three service levels. Hence, the most appropriate accommodation in the link layer would be as follows.

If Guaranteed service level is requested, then strict parameters must be calculated and negotiated with the target IrDA link. The mapping layer then verifies the two values and decides on the success or failure. If “Best Effort” or “Controlled Load” service levels are requested, there will be a possibility that the Offered values can override the required values if they do not agree.

4.2 Throughput Related Mapping

As an example we show Throughput related mapping. The major parameters that determine the MTB limit for IrDA are the **Br** and **MTt**. However, the actual limit is set from DS and WS within the MTB limit. Hence, we take: $\mathbf{MTB_{IrDA} <= Br * MTt}$ Since $MTB_{PIM} <= MTB_{IrDA}$, must hold and MTB_{PIM} = Min (B + r* MTt, MTU + p* MTt) , (Taking MTB_{PIM} = B + MTt * R and MTB_{IrDA} = MTt * Br), we have:

B+ MTt * R <= MTt * Br, then

$$Br \geq \frac{B + MTt * r}{MTt} \quad (3)$$

We used the MTt (Maximum Turn Around Time) of IrDA as an interval, because its value determines the brief break intervals the sender makes between each continuous burst of data flow, handing the link to the other device.
If Guaranteed level is requested by application (PIM), then the expression **Br>= r**, should be verified.
If Best Eeffort or Controlled Load, then the provided (Br) can override the required (r or p) and used for reverse calculation, and we have:

$$r \leq \frac{B - MTt * Br}{MTt} \quad \textbf{Or} \quad p \leq \frac{M - MTt * Br}{MTt}, \text{ and } \mathbf{p >= r} \quad (4)$$

Similarly, the mapping and the verification mechanism can be made for the **latency** related mapping.

5 Related Works

5.1 MDA for SoC

An MDA approach for System on Chip Design (SoC) methodology of embedded systems development has been addressed by the work in [15]. This approach is more appropriate for systems dedicated to specific tasks such as signal processing so that the functionality can be modeled for both the hardware and the software with the co-design methodology. Moreover, it does not consider interconnection between remote entities and how communication protocols and their variability are handled. In addition, it takes the hardware architecture as only the machine elements such as buses and chips and not the API level abstractions.

5.2 Network Protocol Modeling with UML

A first attempt has been made by Sekaran [16] for modeling a data link layer protocol specifically the L2CAP layer of Bluetooth. However, the major drawback of this work is that it does not consider the QoS provided at the link layer. It has only considered the functional services of the layer. Another similar work is that made by Thramboulidis and Mikiroyannidis [17] for modeling the TCP. However, the QoS issues have not been included in the model. These two works have shown that object oriented modeling and implementation of communication protocols is possible with its inherent benefits although slight performance penalty is expected.

5.3 Quality of Service Modeling Approaches

Quality of Service modeling is among the important issues addressed by different researches works recently. Several approaches for modeling QoS have been proposed [18],[19]. However, they do not address specific domains that are closely associated with QoS such as networks and embedded systems. Moreover, most of the concepts they have introduced are incorporated with the UML profiles discussed previously. In [20], Aagedal presents the concept of orthogonal separation between the QoS specification and the functionality specification of a system. Furthermore, it has shown how to link QoS aspect models with functional elements of models called computational elements such as Actor, Component, Interface, Node, Object, Subsystem, Use case, and Use case instance. But it does not use the MDA concepts such as Platform Independent Modeling, Platform Specific Modeling and Transformations.

6 Conclusion

In this paper, we have shown a possible adaptation of the MDA towards a QoS aware and resource oriented application development for the embedded systems domain. The major focus has been the communication subsystem of embedded platforms where the variability in the reservation based networks can be handled in a formal model based process. The Required/Provided relationship between applications and networks has been represented with the PIM and PM perspectives of the MDA and a possible mapping layer that can transform the application level requests to network level provisions. The applicability case study for IrDA has also shown that the PIM QoS can be transformed and also verified with specific network QoS. We believe that this way the concerns of application level modeling and implementation could be separated from the platform level service specification as two different concerns of development in this domain. In addition, we believe that the applicability of the mapping can work for other reservation based networks such as HiPERLAN2 even if it is not initially intended for embedded systems. In our work, we used only parameters that are used to define the performance requirements and provisions. In the real case, other factors such as the overhead imposed by the mapping layer should be considered.

References

1. ARTIST- Adaptive real-time systems for quality of service management- Roadmaps for Research (Draft) IST-2001-34820, 2003, pp 250-263, official site: http://www.artist-embedded.org/, visited on Jan 20, 2005.
2. OMG, UML Profile for Modeling Quality of Service and Fault Tolerance
3. OMG, UML Profile for Schedulability,Performance, and Time Specification:
4. Chen, R., Sgroi, M., Lavagno, L., Martin, G., Sangiovanni-Vincentelli, A., Rabaey, J., "Embedded Systems Design Using UML and Platforms", System Specification and Design Languages (Forum Design Languages 2002), CHDL Series, Kluwer, 2003.
5. Alberto Sangiovanni Vincentelli. Defining Platform-based Design. EEDesign of EETimes, February 2002.
6. Grant Martin, UML for Embedded systems specification and Design (IEEE document): http://ieeexplore.ieee.org/iel5/7834/21541/00998386.pdf, visited on Feb 15, 2005
7. Model-driven architecture - a technical perspective. Technical Report ORMSC/2001-07-01, Object Management Group, 2001. Online: http://www.omg.org/cgi-bin/doc?ormsc/2001-07-01, visited on Feb 15, 2005
8. Bluetooth SIG, The Bluetooth Specification Document, available at www.bluetooth.com/pdf/Bluetooth_11_Specifications_Book.pdf
9. Patrick J. Megowan, David W. Suvak, Charles D. Knutson IrDA Infrared Communications, available at: An Overview: http://www.web-ee.com/primers/files/irda.pdf.
10. Robert Bosch, CAN Specification, Version 2.0, 1991, www.semiconductors.bosch.de/pdf/can2spec.pdf, visited on. Feb 25, 2005.
11. The I2C Specification, Version 2.1, January 2000: www.semiconductors.philips.com/acrobat/literature/9398/39340011.pdf, visited on May 2, 2005
12. Partridge C., “A Proposed Flow Specification”, Internet Engineering Task Force, Request for Comments: 1363, available at: http://www.ietf.org/rfc/rfc1363.txt, September 1992,
13. Javier Muñoz, Model Driven Development of Pervasive Systems, http://www.di.uminho.pt/~mompes04/Papers/Munoz.pdf, visited on Jan 15, 2005.
14. P.Boulet, J.L.Dekeyser, C.Dumoulin, and P.Marquet. MDA for SoC embedded systems design, intensive signal processing experiment. FDL03, 2003.
15. Jo˜ao Paulo Almeida et. al., Handling QoS in MDA: A Discussion on Availability and Dynamic Reconfiguration, CTIT Technical Report TR–CTIT–03–27, Univ. of Twente.
16. Sekaran K.C., Development of a Link layer protocol using UML, Proceedings of IEEE international Conference on Computer Networks and Mobile Computing, October 2001.
17. K. Thramboulidis, A. and A. Mikiroyannidis, Using UML for the Design of Communication protocols: The TCP Case study, 11th International Conference on Software Telecommunicationand Computer Networks, October 7-10, 2003.
18. Frolund, S. ; Koistinen, J.: QML: A Language for Quality of Service Specification / HP Labs. 1998 (TR-98-10). Technical report.
19. W. Torben, Model-Driven Development of QoS Enabled Distributed Applications, University of Electronics and Information Technology, Berlin, PhD thesis 2004.
20. Jan Øyvind Aagedal and Earl F. Ecklund, Jr., Modelling QoS: Towards a UML Profile: Presented at Fifth International Conference on the Unified Modeling Language -the Language and its applications (October 2002).
21. S. Shenker, J. Wroclawski, Network Element Service specification Template, RFC 2216, 1997: www.rfc-archive.org/getrfc.php?rfc=2216, visited on Jun 20, 2005.
22. Specification of Guaranteed Quality of Service (RFC 2212), 1997.
23. Specification of the Controlled Load Service (RFC 2211) IETF.
24. Andy Seaborne et.al., Infrared Data Association Link Management Protocol Specification:, Version 1.2, Jan 23, 1996