Modeling Parallel Applications on Mobile Devices

Daniel Giulianelli[1], Claudia Pons[2], Carina González[3], Pablo Vera[1], Rocío Rodríguez[1], and Víctor Fernández[1]

[1] National University of La Matanza
Department of Engineering and Technological Researches
Florencio Varela 1903, San Justo, Buenos Aires, Argentina
[2] School of Computer Science
Street 50 and 150 La Plata, Buenos Aires, Argentina
[3] La Laguna University
Technical High School of Computer Engineering
Pabellón de Gobierno, c/ Molinos de Agua s/n – 38200 La Laguna, Spain
dgiulian@unlam.edu.ar, cpons@lifia.info.unlp.edu.ar, cjgonza@ull.es, pablovera@unlam.edu.ar, {rrodri,vfernandez}@unlam.edu.ar

Abstract. Nowadays more and more users have mobile devices with high computing power. This fact allows taking advantage of that processing power to design applications that allow users to interact remotely with the system simply using, for example, their cell phones. This requires having a modeling tool that allows incorporating the concepts of mobile computing and their particular domain characteristics. This work shows an extension to the UML profile called PROCODI (Concurrent and Distributed Processes), which adds the possibility of incorporating mobile devices as nodes within the model. This fact allows a quick visualization using a single diagram, of structural and behavioral aspects of the system, also including, now, mobile devices with their peculiarities.

Keywords: Mobile Computing, UML, Profile, Distributed Computing.

1 Introduction

The use of mobile devices in the field of Enterprise applications began several years ago. Initially it began by taking advantage of the capabilities of PDA (Personal Data Assistant) type devices like Palm, Symbian or Pocket PC. These devices had limited processing power, limited memory and their connectivity capabilities were not permanent. A little bit later other devices emerged that were able to combine the connectivity features of cell phones with multipurpose operating systems like, for example, windows mobile, creating the firsts smart phones. With these devices, despite of still lacking of computing power and memory, the users had a permanent connection, because of the capability of connecting to the cell phone network, opening the possibilities of making connected interactive applications, where the user

F.V. Cipolla-Ficarra et al. (Eds.): CCGIDIS 2011, LNCS 7545, pp. 136–143, 2012.

is capable of interacting with the system without a fixed location. However these devices where not massively used, and they were limited to some particular sectors.

Nowadays mobile technology has progressed considerably and continues with a fast growth every day. There are in the market a great diversity of cell phones with a processing power that far exceeds the power of Palm or Pocket PC devices, and that is close to a desktop computer. These new mobile devices are not limited to cell phone because Tablet PC devices are becoming popular with devices like Apple IPAD, Motorolla Xoom, Blackberry Playbook, etc becoming a continuous growing's field. Most of these devices, cell phones and tables, have several ways of internet connectivity, like WIFI, GPRS, EDGE and 3G. The hardware's advance makes that, when designing applications, designers start thinking that it might be running on both a desktop and a mobile device, where, although it will require some adaptation for example for displaying on small screens, its processing capabilities allow considering more complex applications.

One of the main subjects that must be considered when designing an application is to take advantage of the multiprocessing capabilities of the new mobile operating systems, where, with or without the presence of multiple cores, it is possible to run several threads simultaneously, speeding heavy processing tasks in the applications. Therefore, when modeling a distributed and/ or parallel application, it should be considered that the application could also be used in a mobile device because it could be seen as one more processing node in a particular application with similar features of a fixed node.

2 Extension of UML Profile PROCODI

In 2010 the research team finished the design and building of a profile based on UML [2] which main goal was to model parallel and concurrent applications in an easy way by using a single diagram that allows identifying:

- The different nodes involved in the application
- The communication channels between the nodes
- Activities that run simultaneously by the specification of the tasks of each thread
- Generic parallelizable activities which are distributed automatically on different threads.
- Access to shared resources and signaling

This work shows and extension to PROCODI to allow modeling parallel and distributed applications that include mobile devices as some of the nodes. For that purpose the following characteristics have been added to the profile:

- Mobile device: represents the nodes that can change their location.

These devices are:

- o cell phones,
- o tablets,
- o PDAs,
- o etc.

- Location: the fixed place where a mobile device can interact with the system. For example a Bluetooth connection in the station A of the subway.
- Mobile Location: if the mobile device is being moved within a transport with connectivity. For example a plane or a train with WIFI connection.
- Text messages: represents the SMS messages to and from the mobile device.
- Bluetooth messages: represents messages from and to the mobile device that are made with a Bluetooth connection so the distance between the sender and the receiver is limited.
- GPS: Represents the use of the location capabilities of the mobile device using the satellite positioning system.

By including these characteristic to the existing profile, the particularities of the mobile devices will be shown in the model allowing:

1. Differentiating the mobile devices of the system and their location allowing switching between fixed and mobiles scenarios.
2. Modeling the necessary actions that must be performed when the mobile device is moving from one scenario to other.
3. Modeling cell phones' own characteristics like, for example, SMS text messages, or the use of the GPS.

Figure 1 shows the full class diagram of PROCODI profile with the extensions for mobile devices (in green).

Table 1 shows the association among the elements of the extension and the UML metaclasses in which the stereotypes are defined.

The stereotypes generate new metaclasses that can be also extended.

Table 1. Association between elements of the extension and UML metaclasses

Profile Element	Stereotypes	UML Metaclass
Mobile Device	ProMobileDevice	Node
Location	ProLocation	Class
Mobile Location	ProMobileLocation	ProLocation
Text Message	ProSMSMessage	ProExtendedMessage
BlueTooth Message	ProBluetoothMessage	ProExtendedMessage
GPS Location	ProGPSLocation	Class

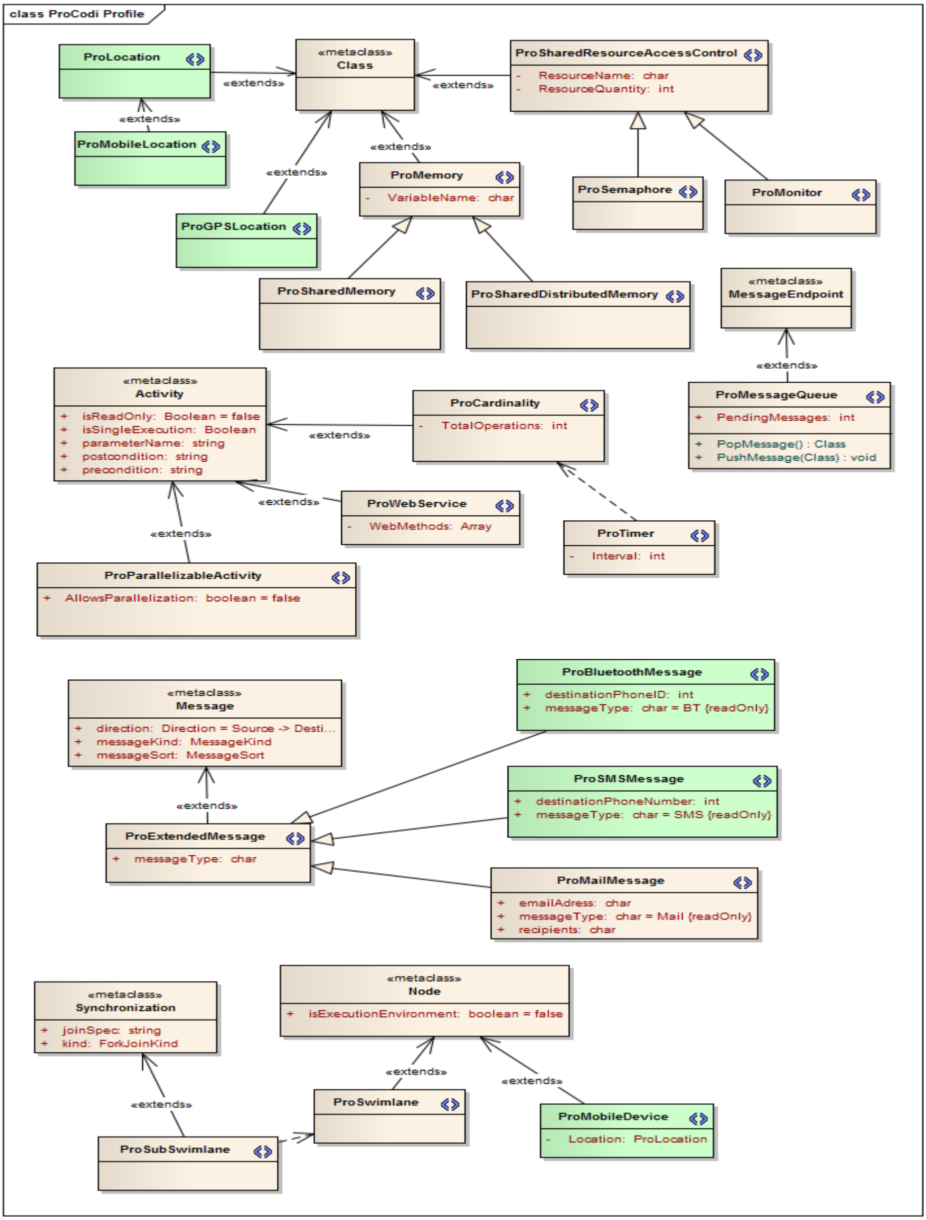

Fig. 1. Class diagram of PROCODI profile with the extension for mobile devices

3 Restrictions

Restrictions are defined for PROCODI and for the domain of mobile devices. Although restrictions could have been created using natural language, OCL [8]

(Object Constrain Language) was choose. OCL is adopted as a standard language by OMG [4] in order to describe restrictions in UML models. These restrictions can be applied both to the business object level of a system and to a metaclass or objects of a metamodel. The advantage of using a formal language like OCL is the easiness of using a tool for checking the fulfillment of the defined restrictions.

These OCL expressions are executed in a context; this is the link between one entity of the UML diagram and de OCL expression. The definition of this context specifies the entity of the model for which the expression is defined. The context is defined using the keyword "Context" and when referring the context within the OCL expression the keyword "self" is used. For more details of OCL restrictions it is advisable to read paper [5].

It was necessary to define different types of OCL restrictions for the PROCODI profile that is developed in this paper:

- Invariants: this declaration must be true for all instance of a specific type at every moment.
- Preconditions and Postconditions: They represent a condition that must be true before or after executing an operation. Keywords "pre" and "post" are used. In postconditions the value at the beginning of the operation can be referred using the keyword "@pre"
- Initial values and derivatives: Initial values of the model elements are set or their derivatives values (calculated based on other values) are taken

3.1 Invariant Restrictions

The following are restrictions of the profile of invariant type expressed in OCL The first restriction is for the `ProCardinality` stereotype which represents a group of activities that will be replicated in several threads in parallel. This group of activities is called operation, so the final quantity of operations will be greater than 1 because it will execute several threads.

```
Contex ProCardinality
inv : self.TotalOperations>1
```

When defining a frequency type with which a set of operations will execute periodically the `ProTimer` is used. For this stereotype, a time restriction is defined to set the range greater than zero.

```
Contex ProTimer
inv : self.Interval>0
```

Every defined Webservice must have at least one method. The OCL expression to represent this is:

```
Contex ProWebService
inv : self.WebMethods -> size()>0
```

Monitors and semaphores must have a quantity of resource to control at least of 1

```
Contex ProSharedResourceAccessControl
inv : self.ResourceQuantity >0
```

In the three classes that derive from ProExtendedMessage the message type must be declared. The message could not be change in any class instance.

```
Contex ProMailMessage
inv : messageType="Mail"
Contex ProSMSMessage
inv : messageType="SMS"
Contex ProBluetoothMessage
inv : messageType="BT"
```

3.2 Pre/Post Condition Restrictions

For the message´s queue the OCL restrictions express that, when the message counter is greater than zero and then a pending message is took in order to be process, the message counter value will be decrease in one, which is the message that has been dequeue.

```
Contex ProMessageQueue::PopMessage():Class
pre: self.PendingMessages>0
post: self.PendingMessages=self.PendingMessages@pre -1
```

A similar restriction is used to queue messages

```
Contex ProMessageQueue::PushMessage(message:Class):Void
pre: messages->notEmpty()
post:self.PendingMessages=self.PendingMessages@pre+1
```

3.3 Initial Values and Derivatives Restrictions

It was necessary to write a restriction to set the initial default value for the attribute "`AllowsParallelization`" of the class `ProActivity`. So, by default, no activity is parallelizable unless stated.

```
Contex ProActivity::AllowsParallelization::Boolean
init:  -> false
```

4 Related Work

There are several works related with representing mobility in UML. Some of them are:

- "UML Profile to Model Mobile Systems" [3]: This work shows the characteristic adaptations in the design of applications that operate in the mobile computing area. It defines artifacts to identify several aspects of mobile applications like for example: Place, NodeLocation, MobileElement, MobileCode, CurrentDeployment, AllowDeployment, MobilityManager, MoveActivities. It also includes the mobile code paradigm that allows modifying the load of the objects and the traffic between them to adapt to the current node condition and the network load.
- Another interesting work is from the University of München in Germany [6]. It shows an extension of the class and activity diagrams of UML to model mobile systems. The extensions that apply are based in the use of stereotypes and are useful for identifying mobile objects, locations and activities like, for example, mobility or cloning.
- In the work of Flabio Oquendo [1] the architecture description language (ADL) is applied to design mobile agents. ADL is presented as a UML profile that includes a graphical representation using implementation diagrams and components.
- The work shown in [7] extends UML for allowing the real time representation, tracking and management of mobile objects.

5 Conclusions

Modeling a parallel and distributed system requires the use of several UML diagrams for representing each one of its characteristics. With the extensions made to PROCODI it´s possible by means of a single extended activity diagram, to visualize both structural and behavioral aspects simultaneously.

This is especially profitable when modeling distributed applications because it allows seeing the nodes and their interconnection, faster.

By extending this model adding the mobile devices, all benefits of PROCODI are taken and it is powered by making seamless the modeling of mobile applications by treating the mobile devices as another node of the model, but including their particular communication characteristics. It also allows modeling different scenarios taking in consideration the geographical position of the device.

As future work we propose to model server applications with this extension of PROCODI.

References

1. Oquendo, F.: An Architecture Description Language based on the Higher-Order Typed π-Calculus for Specifying Dynamic and Mobile Software Architectures. University of Savoie at Annecy ESIA – LISTIC Lab – Formal Software Architecture and Process Research Group, France (2004), `http://portal.acm.org/citation.cfm?id=986728`
2. Giulianelli, D.A., Pons, C., Rodríguez, R.A., Vera, P.M., Fernandez, V.M.: A Profile´s Design for Parallel Applications Modelling. Revista Colombiana de Computación (Colombian Journal of Computing) 11(1), 56–68 (2010), ISBN: 1657-2831; Universidad Autónoma de Bucaramanga. Editores: Alina Fedossova, Eduardo Carrillo Zambrano. Formato: 137 páginas

3. Grassi, V., Mirandola, R., Sabetta, A.: A UML Profile to Model Mobile Systems. Universit‘a di Roma “Tor Vergata”, Italy (2004)
4. OMG (Object Management Group). “Catalog of UML Profile of Specification” (2009), http://www.omg.org/technology/documents/profile_catalog.html
5. Becker, V., Pons, C.: Definición Formal de la Semántica de UML-OCL a través de su traducción a OBJECT-Z, Universidad Nacional de La Plata, Facultad de Informática, pp. 2–6 (2003)
6. Baumeister, H., Koch, N., Kosiuczenko, P., Wirsing, M.: Extending Activity Diagrams to Model Mobile Systems. Institut fur Informatik Ludwig-Maximilians-Universiat Munchen, Germany (2003), http://www.springerlink.com/content/fec97clg6c2aw6qj/
7. El Bouziri, A., Boulmakoul, A., Mouncif, H., Mohamed: Modeling Mobile Object for Transportation Information System and Mobile GIS., LIST Lab., Computer Sciences Department Mohammedia Faculty of Sciences and Technology (FSTM), Moroco (2005)
8. OMG. Object Constraint Language (2011), http://www.omg.org/spec/OCL/2.3/Beta2