NOMAD GOES MOBILE

J. Locatelli*, F. Cecillon, C. Cocho, A. Elaazzouzi, Y. Le Goc, P. Mutti, H. Ortiz, J. Ratel
Institut Laue-Langevin, Grenoble, France
A. Perez-Subtil, Université Claude Bernard, Lyon, France

Abstract

The commissioning of the new instruments at the Institut Laue-Langevin (ILL) has shown the need to extend instrument control outside the classical desktop computer location. This, together with the availability of reliable and powerful mobile devices such as smartphones and tablets has triggered a new branch of development for NOMAD, the instrument control software in use at the ILL. Those devices, often considered only as recreational toys, can play an important role in simplifying the life of instrument scientists and technicians. Performing an experiment not only happens in the instrument cabin but also from the office, from another instrument, from the lab and from home. The present paper describes the development of a remote interface, based on Java and Android Eclipse SDK, communicating with the NOMAD server using CORBA via wireless network. Moreover, the application is distributed on Google Play to minimise the installation and the update procedures.

INTRODUCTION

Performing an experiment not only happens in the instrument cabin but also from the office, from another instrument, from the lab and from home. NOMAD architecture is heavily based on network technologies and the application can run without graphical interface or be controlled by several different interfaces. All this is made possible by the use of CORBA [1], acronym for Common Object Request Broker Architecture. CORBA makes the connection between two processes totally transparent whether they run on the same machine or on different ones. This technology together with the availability of powerful mobile devices pushed the development towards an increase remote connectivity. The main goal of this new development was to offer an alternative graphic user interface (GUI) for NOMAD running on a tablet support.

Most of the new instruments have considerably increased their dimensions and the control room is today geographically distant from the sample area. All classical operations performed especially at the setup of a new experiment like manual sample and beam alignment or the configuration of the sample environment, becomes then tedious and extremely time consuming. After a series of benchmark tests performed on a number of different tablets, we have selected the Android based ARCHOS 80 G9, offering an overall good price-to-performance ratio. The compact dimension and the light weight of the tablet, only 8 screen diagonal, offers to the user the possibility to access all functionalities using a single hand, transforming the tablet in a powerful remote control for the most common instruments operations. The choice of Android as operating system was driven by the possibility to produce a Java application using most of the development already performed for the main NOMAD GUI. The interaction between the new remote interface and the server running on the instrument control computer takes place via a specific version of the CORBA protocol (JACORB). The porting of JACORB on Android has been achieved thanks to a collaboration with the official JACORB developers team.

* locatelli@ill.eu

CONNECTION WITH NOMAD CORE

NOMAD Overview

The NOMAD application contains a variety of different components as illustrated in Fig. 1.

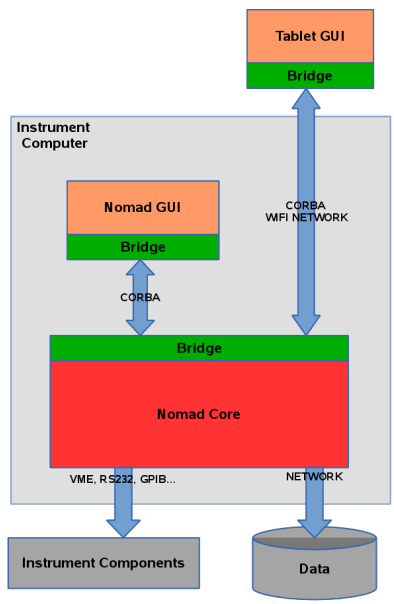

Figure 1: Layout of the various NOMAD components.

A typical NOMAD deployment consist of:

- A dedicated computer per instrument. In the typical implementation, both applications (Core and GUI) are

ISBN 978-3-95450-139-7

hosted on the same machine even if this is not mandatory. The GUI can run on a different computer as it is the case with the tablet application.

- The NOMAD Core, written in C++ language, is in charge of all instrument operations including the storage of the acquisition data, the logging and the on-line parameters survey. It interacts with the instrument's hardware using a variety of communication buses (VME, GPIB, RS232/485, network socket, etc...).
- The NOMAD GUI, based on Java SWT, does not contain any functional intelligence. It allows the user to request actions to the hardware and to control the status of the experiment. A dedicated effort has been made to provide an easy-to-use graphical programming of complex sequences of instructions. A dedicated real-time plotting library based on OpenGL is responsible for all kind of data visualisation.
- The communication layer. The Core and GUI interact by means of a network layer using the CORBA protocol. All CORBA components are integrated in a bridge library which is compiled together with the client source. Our CORBA implementation for the Core application is based on OmniOrb (CORBA ORB for C++ and Python). For the client side we adopted JacORB (Java implementation of the OMG's CORBA standard).

Tablet Connection

The tablet application is another client for the NOMAD Core, as described in Fig. 1. This client is also implemented in a bridge for managing CORBA packages which is the same as the one used for the NOMAD GUI. The problem we encountered was to run JacORB on an android platform. JacORB implements Java Remote Method Invocation (RMI). Android Java doesn't use the RMI classes. Fortunately, the JacORB version 2.3.1 has a compilation option which allows to remove RMI classes, but in the newest releases this option has disappeared leaving the problem unsolved.

TABLET CLIENT

Development Tools

The Android SDK and ADT [2] provide the API libraries and development tools necessary to build, test, and debug Android applications.

On IDE platform, it is possible to build rich Android UI with drag and drop or editing XML file as shown in Fig. 2. The application debugging could be done directly on the device over USB port or using a virtual machine.

NOMAD Bridge

This bridge is a library included to our Android client project. It contains all generated CORBA classes and all the methods for accessing NOMAD core data like properties, command and events.

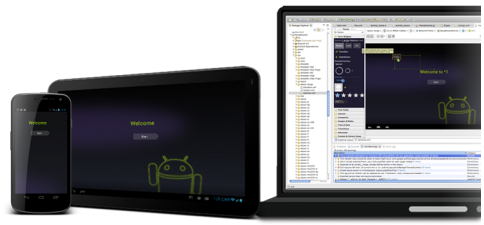

Figure 2: Android IDE.

GUI Design

There is a fundamental consideration to be taken into account when designing a tablet based application, everything must be larger than normal. This includes things such as icons, buttons, text boxes, labels, etc. The first reason is that a tablet application is meant to be used on the go and every element should be easily reachable. Second, the pointing device is often your finger and not a mouse. In order to enhance the user experience and to increase the functionalities of the new GUI, a set of dedicated screens has been developed. Those are not only better adapted to the reduced screen size of the tablet, but they also give access to options specifically developed for the mobile environment.

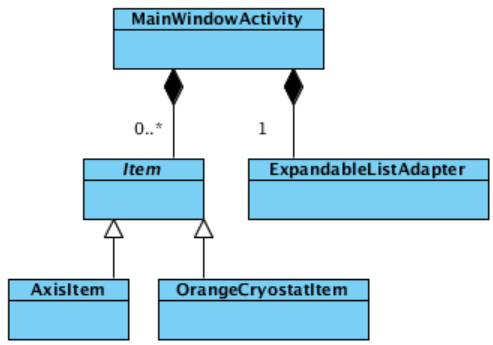

Figure 3: Plugins view model.

The tablet GUI consists of different activity classes. Some are used for managing the connection with NOMAD core:

- Start screen: it affects the tablet to a specific instrument based on the name which depends on the tablet MAC address. In this way we prevent a tablet to connect to a different instrument by mistake and we avoid unwanted connections.
- Instrument choice screen: this activity is specific to the administrator tablets. For maintenance purposes or technical interventions some tablets have the possibility to connect to any NOMAD core.

ISBN 978-3-95450-139-7

- Loading screen: ensures the connection to the NOMAD core and it loads everything it is needed from it.

Other screens are dedicated to the interaction of the user with the instrument as depicted in Fig. 3. An item is a representation of what we call a controller in NOMAD. This is an object which contains properties and can execute commands. It can be a basic component of an instrument (e.g. an axis, a temperature device or a power supply) or a more complex object consisting of several basic ones like a cryostat or a scientific setting. The list of available controllers, properties and commands is delivered by the NOMAD core.

- Main window: this is the main view of the application. It contains a list of all available items for a given instrument. When the user selects a specific item of this list, the associate view is shown.

- ExpandableListAdaptater: it manages and generates the view of the controller list which appears on the left hand side of the main view.

- Item: this is the abstract class of items implementation. It contains all common specific codes. As an example, the AxisItem and CryostatOrangeItem are specific item implementations. They contain the association between the GUI components control and the NOMAD core properties and commands.

The generation of a complex sequence of instrument operations, available in the main NOMAD GUI will not be included in the mobile interface since the tablet is not meant to replace the desktop GUI to run the experiment.

SCREEN EXAMPLES

Figure 4 shows the new axis view in which the standard tree on the left side, allowing the choice of the desired motor, is seconded by a panel dedicated to the selected movement.

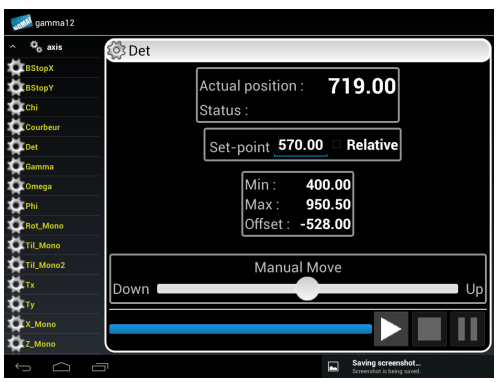

Figure 4: A tree view allows to select the desired axis and to access the new functionalities to manually control motion.

Classical options like the positioning to a desired value are complemented by new possibilities like a fixed increment starting from the present position or the SeekBar. This last one allows to control with a single finger the direction of the movement as well as the motor speed by simply moving the slide on the screen left or right. For security reasons, releasing the pressure on the slider will automatically stop the motion.

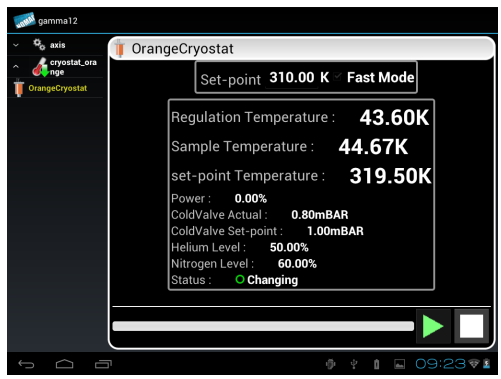

Figure 5: Example of the Orange Cryostat control screen.

The second implementation of the new GUI concerns the sample environment. Figure 5 depicts the screen available to control the orange cryostat, in use at the ILL to hold samples at temperatures as low as 1.5 K. All the basic function-alities relevant for the control of such a device have been included in the interface allowing the user to select the proper PID values, temperature limits, cold valve parameters and other options.

PERSPECTIVES

One of the objectives of the ongoing NOMAD development is the replacement of the ageing CORBA technology. We have already performed a series of tests and benchmarks using ZeroMQ [3] for the transport layer and Protocol Buffer [4] for the encoding of structured data. Since ZeroMQ does not have a pure java implementation, we have adopted JeroMQ [5] (pure Java implementation of libzmq) in our tests on Android platform. Moreover, we are trying to optimise the middle-ware library responsible for sending and receiving requests to NOMAD core in order to accelerate the application loading time.

All the screens in use by the NOMAD main GUI are created by a plug-in generator which is building the views from a description contained in an XML files. We are currently exploring the possibility to generate Android XML and Java view files from the same XML description. This will be even more needed once we will open the possibility to use NOMAD clients on more Android targets, like different tablet models or smart-phones. The screens will need to be ported on different screen size, but the automatic generation should take care of it. Last but not least, we are considering the integration of a light plotting library

ISBN 978-3-95450-139-7

in order to visualise on the mobile platform detector data or parameters survey curves.

On the user experience side, the set of tablet oriented screens and functionalities will be further enhanced. A portable rate meter including automatic peak detection is already in preparation as well as a new configuration manager for the sample environment.

CONCLUSION

So far the ANDROID client for NOMAD is operational on five different instruments. Despite the fact that the overall performances and the user feedback are more than satisfactory one technical aspect needs a deeper investigation. The loss of the WiFi connection during the execution of a command, always possible in a complex environment like the experimental areas of the ILL, should be further secured to avoid dangerous situations. On the development side, the set of tablet oriented screens and functionalities will be further enhanced. The Android NOMAD application is available for download on Google Play mainly to profit of the automatic update feature but a stand-alone version is in preparation.

REFERENCES

[1] OMG CORBA, http://www.corba.org/

[2] The Android developer tools, http://developer.android.com/tools/index.html

[3] ZeroMQ, http://www.zeromq.org/

[4] Google Protocol Buffers http://code.google.com/p/protobuf/

[5] JeroMQ, https://github.com/zeromq/jeromq/

ISBN 978-3-95450-139-7