

Dr. Mostafa El-Said, Editor-in-Chief
Dr. Dave Cavalcanti, Chair of the TCSIM

Chair's Message

By Dr. Dave Cavalcanti, Chair of TCSIM

It is with great pleasure that I welcome you to the first issue of the *IEEE TCSIM Newsletter* brought to you by the IEEE Computer Society Technical Committee on Simulation (TCSIM). This newsletter has been designed to share information about the important events, news and recent advances on simulation related topics. The *IEEE TCSIM Newsletter* is also a forum for publication of high-quality technical work, in the form of short, critically referred and invited articles.

This first issue features a list of interesting events sponsored by TCSIM happening in 2008 and 2009 and an invited article on the Android platform and its advanced emulation features. This issue also includes a call for contributions with the guidelines for articles to be considered for publication in this forum. This is a great opportunity to increase interaction and collaboration amongst members of the technical community interested on simulation research topics, tools and platforms. Therefore, I encourage you to spread the news about this initiative.

I hope you enjoy reading it and I look forward to receiving your suggestions and contributions for the following issues of the *IEEE TCSIM Newsletter*.

Benefits of Joining the TCSIM

TCSIM is an IEEE technical committee that promotes all simulation-related activities including research, design, development, methodologies, and applications of mathematical modeling. It organizes technical sessions at Computer Society conferences and cosponsors international conferences and workshops with focus on simulation. TCSIM members can easily meet and collaborate with other colleagues, learn about conferences, join the TCSIM leadership team, participate in TCSIM seminars, and collaborate to start new programs and events.

TCSIM especially encourages student members to participate and contribute to TCSIM activities. TCSIM student members are eligible for nomination of best simulation paper award and can apply for TCSIM sponsored travel grants. You can explore the TCSIM website at <http://tab.computer.org/tcsim/> and watch out for announcements on upcoming sponsored events and awards.

To join the TCSIM or update your membership, you must use your IEEE Web Account and login to the IEEE Technical Committee Archives (TECA) at https://cima.computer.org/TECA_Login.htm, where you can explore some of the TCSIM resources.

Members Corner

If you are interested in receiving this newsletter in the future, please send your e-mail address to elsaidm@gvsu.edu

Also, if you have some news you would like to share with other members or want to contribute to the TCSIM activities, please send your update to elsaidm@gvsu.edu with "Member Update" in the subject line.

Important Events

The TCSIM committee sponsors various events including:

IEEE MASS 2008 (Fifth IEEE International Conference on Mobile Ad-hoc and Sensor Systems)

September 29 - October 2, 2008 Atlanta, Georgia

<http://www.cse.psu.edu/IEEEMASS08/>

IEEE DS-RT 2008 (11th IEEE International Symposium on Distributed Simulation and Real Time Applications)

October 27-29, 2008 Vancouver, Canada

www.cs.unibo.it/ds-rt2008

PADS 2009 - Principles of Advance and Distributed Simulation

June 22-25 2009, Lake Placid, NY

www.pads-workshop.org

June 22-25 2009, Lake Placid, NY

Android: An Open Platform for Mobile Devices

Todd Boss

Application Developer, Bradford Company

Today, most mobile phones run one of a handful of proprietary operating systems. To reach the widest possible audience, developers must invest time and effort porting their software to each of the most popular devices. In addition, mobile carriers are often motivated to lock their customers down to their own proprietary set of applications and protocols in an effort to maximize their revenue. This stifles innovation in mobile software development, and negatively impacts the amount of value that a customer can get out of their investment in their device and monthly carrier fees.

In response to this situation, the Open Handset Alliance (<http://www.openhandsetalliance.com/>) announced the Android mobile platform in 2007. The Open Handset Alliance is a group of over 30 companies, led by Google, with a common goal of developing “the first open, complete, and free platform created specifically for mobile devices.” The companies involved include mobile operators, software companies, handset manufacturers, and semiconductor companies. Recently, other groups have emerged with similar goals, such as the LiMo Foundation (<http://www.limofoundation.org/>), Open Mobile Alliance (<http://www.openmobilealliance.org/>), and OpenMoko (<http://www.openmoko.com/>). Many of the Open Handset Alliance's members are also members of one or more of these groups.

This article starts with an overview of how the Android platform builds a multitasking mobile environment on the Linux kernel. After a brief mention of the Android SDK and the included development plugin for the Eclipse IDE, we go into detail regarding the Android Emulator and some of its advanced features. These features include simulation of multiple wireless networks, telephony events, and SD memory cards.

The Android Software Stack

Android is a complete software stack based on Linux that allows developers to write Internet-aware applications that run on devices from any manufacturer that supports the Android platform such as given in figure 1. The architecture of Android abstracts the underlying hardware. This allows the developer to access multimedia, telephony, network connectivity, and graphics capabilities without concern for the specific hardware on which the application may run. The hardware abstraction is implemented in the kernel (currently Linux 2.6).

Above the kernel is the Android runtime. This runtime provides a large subset of the features available in the Java core libraries. The runtime utilizes the Dalvik virtual

machine, which is a Java VM that is optimized for running multiple instances of itself on a mobile device with a small memory footprint. This optimization is necessary because each application process in Android runs on its own instance of the Dalvik VM.

The Android Application Framework builds on these technologies and utilizes the same system components that are used in Android's core applications. These components include UI widgets such as text boxes, drop-down lists, grids, buttons, and a web browser control. The Android Notification Manager is a class that is used to send notifications to the user in response to system or application events. The available methods of notification include status bar icons, hardware LEDs, ringing, and vibration.

The Activity Manager is responsible for window creation and display for each screen in an application and also provides application lifecycle management. It is challenging to provide a multitasking environment on a memory-constrained mobile device. Android addresses this problem by requiring all applications to implement handlers for lifecycle events that they may encounter during their execution. Android applications may be paused or terminated at any time due to memory constraints, user navigation, or telephony events such as receiving a phone call. When these events occur, an application must clean up system resources such as network connections and store its state in such a way that it can be restarted in the same condition in which it was stopped. Android ships with a few core applications built in. Among them are an email client, calendar and contact managers, and a web browser.

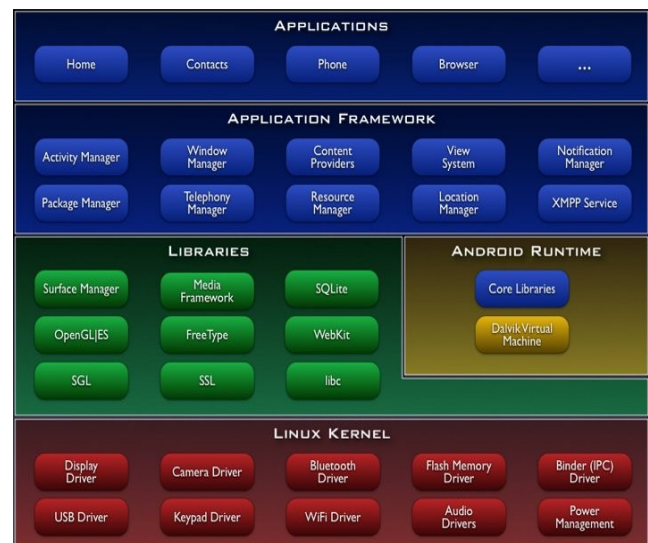


Fig 1. Architecture of the Android Platform

The Android SDK

The Android SDK is a set of tools provided to facilitate development of Android applications using Java. The most important of these are the Eclipse Android Development Tools plugin and the Android emulator. The SDK also includes command-line applications that allow the user to attach debuggers, create Android application packages, and compile to Android bytecode.

It is possible to develop Android applications using Java development environment, but the Android Development Tools Plugin for the Eclipse IDE makes the process much simpler. The plugin automates the project creation process by creating necessary project framework files and populates them with enough content to start a simple application. Configuration dialogs simplify the process of setting up an Android project for execution or debugging in the emulator.

Currently, the Android emulator shown in figure 2 is especially important for development because the platform is in pre-release and there are no commercial devices available for testing. A developer can interact with the emulated mobile device with a mouse and keyboard as if it were an actual physical device. Audio, video, and network functions are all available, and the user interface can be used to launch and navigate between applications.



Fig 2. The Android Emulator

Advanced Emulation Options

In addition to the GUI interface mentioned above, the Android emulator communicates on two telnet ports: a console port and a debug port. The console can be used to fine-tune the emulation environment to simulate various wireless networks, telephony events, and memory cards. After connecting to the console, typing 'help' will retrieve a list of possible commands. Further information about each command can be found by entering 'help <command name>'.

By using the console to adjust the network speed and latency, the emulator can simulate the user experience provided by various wireless networks. The 'network delay' command sets the latency of the emulator's network connection. The latency value can be turned off or set in milliseconds to an exact value or allowable range. The command also includes three presets that simulate GPRS, EDGE/EGPRS, and UMTS/3G respectively. The 'network speed' command works in a similar manner. The maximum upload and download speeds can be set together or separately. The network speed

setting also includes presets for GSM/CSD, HSCSD, GPRS, EDGE/EGPRS, UMTS/3G, and HSDPA.

The emulator includes a GSM modem which allows for simulation of telephony events such as incoming calls and switching between home and roaming networks. The 'gsm voice' and 'gsm data' commands are used to simulate registration, connectivity, and roaming states. These can be used to ensure that a mobile application gracefully handles disconnection if the user loses wireless signal. The 'gsm call' command is used to simulate an incoming call from the given phone number. As expected, the emulator plays the ringtone through the host computer's speakers and displays the caller ID information such as described in figure 3. At this point, the user can accept the call and use Android's in-call options menu. While the in-call functions are useful for application testing, it should be noted that the emulator does not support placing actual phone calls or simulating call audio in the current release. The emulator also supports simulating incoming text messages. The 'sms send' command will forward an SMS message to the Android framework. It will then be passed on to an application that is able to handle SMS messages.

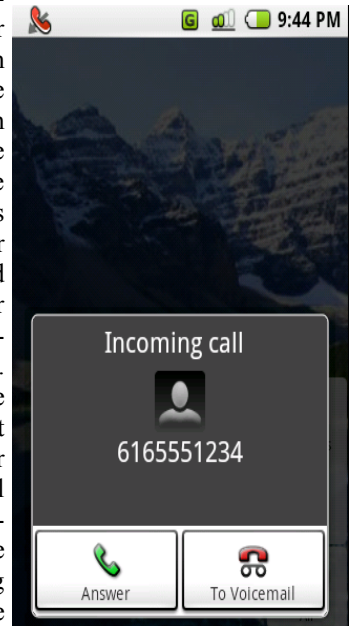


Figure 3. Simulating an Incoming Call

Even though the emulator does not yet support insert or eject events for SD memory cards, it is possible to simulate their presence by preparing a disk image and loading it at startup. The Android SDK includes the 'mksdcard' utility, which creates a FAT32 image of a specified size. This image can then be mounted on the host machine and populated with files as necessary. Because the emulator does not yet support insert or eject events, the image must be loaded at emulator start time using the '-sdcard' switch. Once the image is loaded, however, the emulator can read and write to it normally.

The Future

Though it is still in its infancy, the Android mobile platform has a good foundation of useful development tools. As the product matures, a large and vibrant development community should spring up and grow around it. Google is encouraging this growth by holding a developer challenge with \$10 million in posted awards. A slideshow of the finalists' applications is available at http://code.google.com/android/images/adc1r1_deck.pdf. A new breed of devices based on open standards should prove to be fertile ground for significant innovation in the wireless industry. Combined with open networks, the applications that they will run have the potential to transform the way people communicate.

For more information and to download the SDK, see <http://code.google.com/android/>.

Call for Papers – IEEE Simulation Newsletters

The IEEE TCSIM Newsletters will publish original and high-quality ideas for publication in the form of short technical papers. The submissions should emphasize modeling, design, and analysis of computational methods for simulations and its applications in various areas, including, but not limited to, computer science, engineering, communications, and simulation applications. The submissions are invited covering, but not limited to, the following topics:

1. Simulation architecture modeling and prototyping
2. Simulation algorithm design, implementation, and analysis
3. Simulation complexity in computing
4. Parallel and distributed simulation
5. Design and usage of simulation tools
6. Real-time simulation monitoring
7. Simulation tools for communications and networks
8. Simulation of computer systems and applications
9. Agent-based simulation tools focus on the use of agents in engineering, human and social dynamics, military applications
10. Systems and process simulation
11. Simulation of ubiquitous networking and computing
12. Simulation of transportation systems
13. Automotive simulation applications
14. Building and energy management simulations
15. Machine learning
16. Virtual reality systems
17. Knowledge and data systems
18. Systems optimization
19. Web-based simulation and applications
20. Department of Defense Architecture Framework (DoDAF)-based network simulations
21. DoDAF-based vulnerability assessment

Submission

All papers must be submitted to elsaidm@gvsu.edu in four pages or fewer, including all figures, tables, and references. A manuscript submitted for publication should be original work that should not have been previously published and should not be under consideration for publication elsewhere. If an author uses charts, photographs, or other graphics from previously printed material, he/she is responsible for obtaining written permission from the publisher to use the material in his/her manuscript. The maximal number of figures and tables are five, and the number of reference is limited to ten. Submissions exceeding this length will be returned without review. Papers should use 7.875in x 10.75 in (20cm x 27.30cm) trim size and the IEEE transactions two-column format in 10-pt. font. Please submit electronically in PDF file, and ensure that the submitted file can be viewed in Acrobat Reader 8.0. No hard copy is necessary. A standard IEEE copyright release will also be required before fully acceptance.

All papers must include the authors' affiliation and e-mail addresses of all authors. All papers will be fully refereed for accuracy, technical content, and relevance. Contact Dr. El-Said at elsaidm@gvsu.edu with any questions concerning the paper submission and review process, or questions regarding the relevance of a paper to the IEEE TCSIM Newsletters.

Special Issues

IEEE TCSIM Newsletter is seeking for special issue proposals. Each special issue typically includes 3-5 peer-reviewed articles and a Guest Editors' Introduction. To propose a theme to guest edit, please contact the editor in chief or TCSIM chair for details.

Editor-in-Chief

Dr. Mostafa El-Said
 Director of Data Communication Center (DCC)
 School of Computing and Information Systems
 Grand Valley State University
 Email: elsaidm@gvsu.edu

TCSIM Chair

Dr. Dave Cavalcanti,
 Senior Member Research Staff
 WiCAN - Wireless Communications and Networking Department
 Philips Research North America
 E-mail: dave.cavalcanti@philips.com