TmL - Tools for mobile Linux

TmL Creation Review

First Proposal

Christian Kurzke, Motorola
(Proposed Lead)

Mauren Brenner, Motorola

Agenda

- Overview Mobile Linux Devices
- Project Goals and Scope
  - Multi-target C/C++ Projects with Toolchain and Device-Specific Library Management
  - Generic Emulator Framework
  - Mobile Infrastructure Simulation
- How TmL relates to existing Eclipse Projects
- Participation
- Initial Roadmap
- Current Status

Overview Mobile Linux Devices

Linux is gaining strong acceptance as a mobile device operating system by manufacturers. As a mature platform, it provides a large open source code base and developer community.

In order to harness the full potential of Linux-based Mobile Devices, powerful integrated development environments (IDE) and tools are needed to create sophisticated applications.

Eclipse has a great popularity as IDE for desktop Java applications. CDT is expanding Eclipse's capabilities into C and C++ development, and other projects in DSDP (e.g. TM, DD) are contributing basic functionality needed for mobile Linux application developers.

What is missing is a way to provide a rich environment for developers of applications for mobile Linux developers. End-to-end services for Linux-based mobile phones span a wide range of applications, including enterprise web services, location-based-services, multiplayer gaming, and mobilized corporate data.

The developers of such applications often need supporting development tools and infrastructure like:

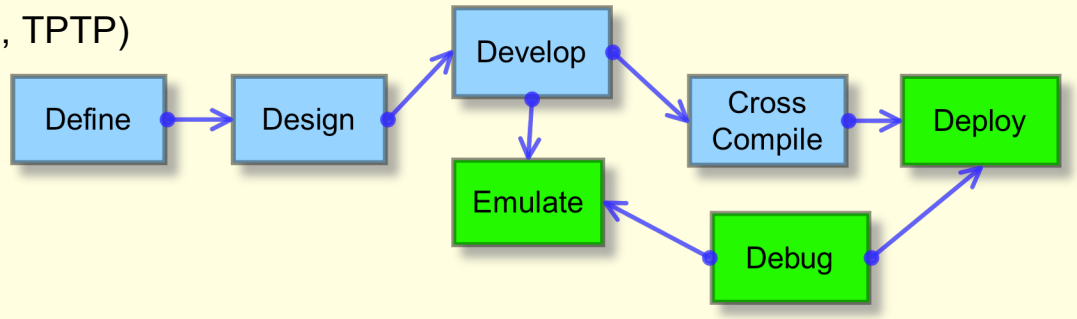
- Large number of target toolchains/libraries to create executables for different devices
- Emulation/Sandboxing of an execution environment to test applications
- Simulation of mobile infrastructure, e.g. SMS backend, OBEX send/receive, mobile network

Project Goals and Scope

- Work with existing Eclipse projects to vertically extend Eclipse into a C/C++ application development environment for the mobile device market.

- Create a Simulation Environment Framework to cover the end-to-end mobile services development
  - Sandboxed Linux Runtime environment
  - Simulated mobile infrastructure, messaging, location based services

- Support the entire software development lifecycle for mobile Linux Applications:
  - **Define** (using/extending EMF)
  - **Design** (using/extending DSDP/NAB, EMF)
  - **Develop** (using/extending CDT)
  - **Deploy** (using/extending DSDP/TM)
  - **Debug** (using/extending DSDP/DD, TPTP)



Multi-target C/C++ Projects

Extensible Framework for multiple (concurrently installed) toolchains and libraries for devices.

Goal is to deploy new toolchains packaged as Eclipse Plug-ins to dynamically enable support for new devices as they become available.

- Multi-target C/C++ projects
  - Management of multiple toolchains and device-specific libraries
  - Does *not* overlap with CDT Managed Make/Standard Make mechanisms
  - May be contributed to CDT in the future

TmL Toolchain framework to register:

- "New Project" Wizard extensions - in case there is unique features/capabilities
- Toolchain and library views
- Sample applications and integrated tutorials for new device features
- Specialized code completion according to available APIs
  - Some APIs may be available in some targets while not available in others
  - New markers to indicate target-specific errors
- Specialized error handling for multi-target projects

Device Emulator Framework

C/C++ Applications developed for mobile devices cannot easily be executed on a developer's PC. Execution on actual device hardware is often hindered by limited availability of hardware, esp. in the early development phase.

A critical piece of the TmL development environment is a Device Emulator.

TmL seeks to define an extensible framework for an emulator. The design goal is to keep the emulator generic enough to accommodate implementations using different technologies like:

- User-mode Linux
- Qemu
- Vmware
- Other proprietary technology

The framework shall provide for ways to:

- Support different launching parameters and configurations
- Infrastructure for Eclipse control components to communicate with emulated devices executing in the emulator context
- Provide toolbar buttons to launch emulators according to different configurations (similar to "Run..." toolbar button on the Eclipse Workbench)
- Provide Properties pages to define emulator configurations and arguments, including arguments to be passed on to the emulated system
- Support launching of and connection to remote emulators

To implement parts of the emulator framework we are looking at the possibility of using existing DSDP/TM features, as well as open source protocols like the "Remote Framebuffer Protocol" (RFB) defined by the VNC team.

Mobile Infrastructure Simulation

Part of the development environment for mobile device applications is the simulations of the interaction with a mobile infrastructure. This infrastructure may include components like:

- Simulated messaging servers
- Deployment servers
- Simulation or capture/replay of location based data (Cell information, GPS, etc.)

TmL aims to create a framework to deploy, register and configure components to simulate devices in this infrastructure. It shall:

- Provide toolbar buttons to start up/shut down simulated services
- Provide a view for visualization of the simulated infrastructure
- Provide a graphic editor (possibly using GEF) to set up simulated infrastructure
- Provide Properties pages to define service properties
- Support remote components of the simulated infrastructure

We will closely cooperate with DSDP/MTJ and the TmL Emulator Framework to ensure maximum code resuse and interoperability with any components developed for the Java MIDLET emulator.

Relationship to other Eclipse projects

The goal for the “Tools for mobile Linux” project is to develop a complete end-to-end development environment for the mobile application development vertical market.

Most existing Eclipse projects have a more general, horizontal focus.

Therefore it is natural that TmL will interact, re-use and extend other existing projects.

Some projects which have been preliminary identified to be used as basis for TmL are:

CDT, DD, TM, TPTP, NAB, GEF, EMF and others as we will define more use-cases.

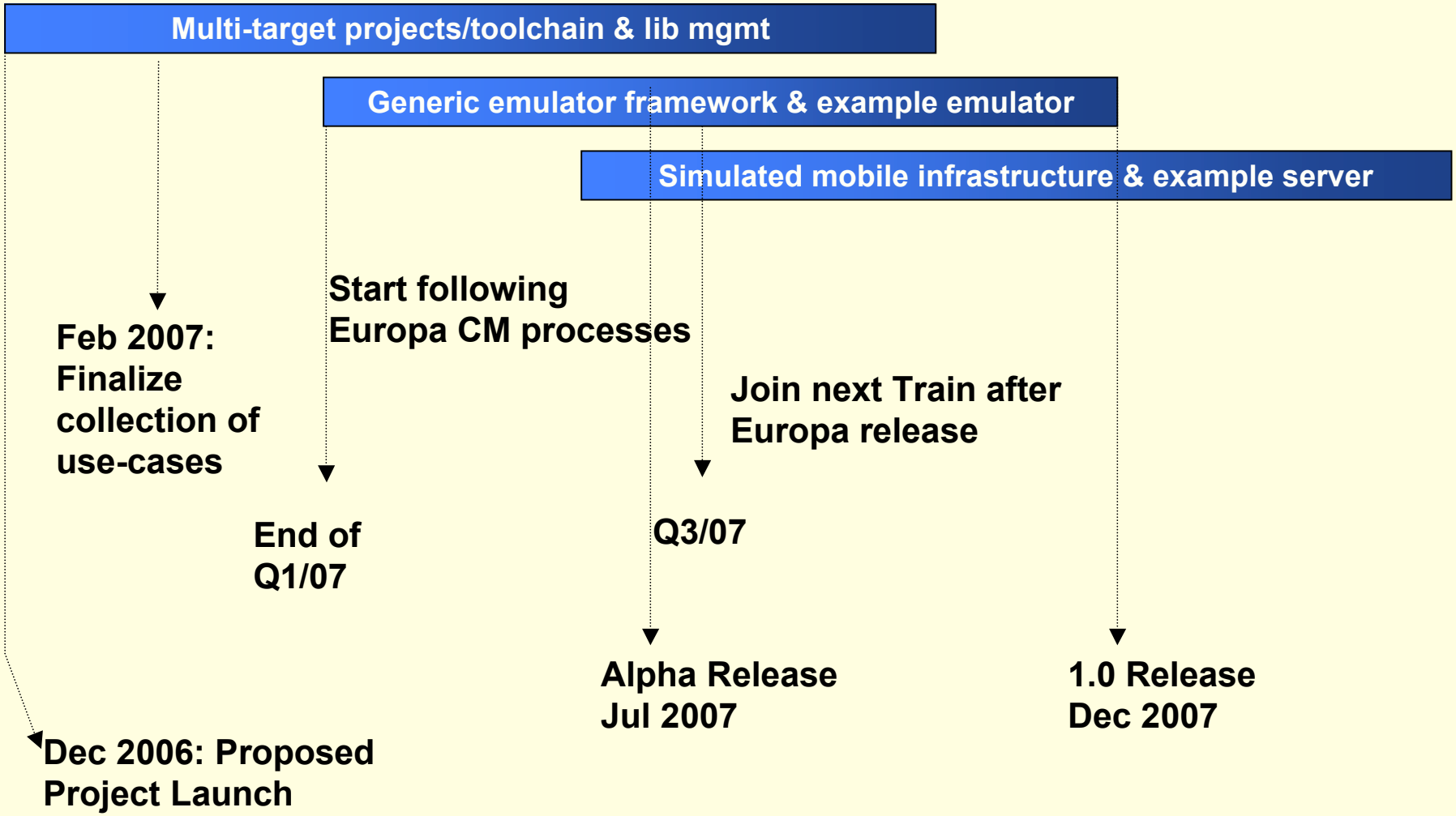
Participation

At this time we have had preliminary talks with several companies and individuals who would be interested in contributing use-cases, time and possibly code.

The interested possible corporate contributors to the project are:

- Motorola
- Windriver Systems
- Symbian
- Nokia
- ARM
- Palm

Initial Roadmap



Current Status

- Collecting Use-cases for TmL IDE
- Planning and architecture definition for Multi-target Project support
- Initial architecture proposal for General Emulator Framework
- Evaluate other projects for suitability to be used as base of TmL Frameworks

Next Steps

- Define requirements for TmL Tools (to be finalized Feb 07)
- Document and contribute source code for Multi target Support to Eclipse CVS
  - Get more community feedback
  - Engage more deeply with CDT project
- Initiate Code contribution for Emulator framework and Linux components (Eclipse and Motorola Legal)