UCL

## The Eclipse Platform

Wolfgang Emmerich
Professor of Distributed Computing
University College London
http://sse.cs.ucl.ac.uk

UCL

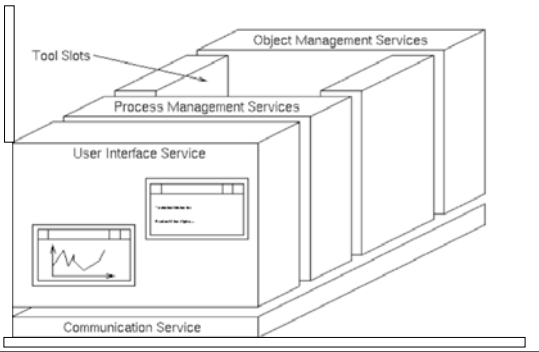
## Learning Objectives

- Many might know Eclipse as a Java interactive development environment
- The aim of this lecture is to review the Eclipse platform on which the Java IDE (and many others) are built with respect to:
  - Its component model
  - The project management support
  - The user interface support
- To give you insights into a very well designed and complex software system



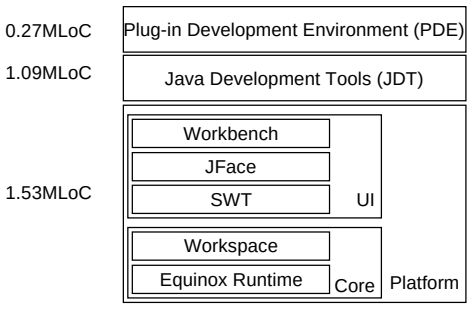
UCL

## Review: The ECMA reference model



UCL

## Overview of the Eclipse Platform




4

UCL

## Bundles / Plug-ins

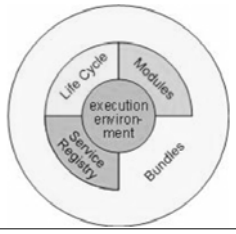
- Eclipse vision is to flexibly customise and evolve the Eclipse platform through plug-ins.
- Plug-ins are structured bundles of code and/or data that contribute functionality to the system. Plug-ins can define extension points, well-defined places where other plug-ins can add functionality.
- Eclipse has moved to adopting OSGi to manage the versions and configurations of plug-ins, control their deployment and avoid interference between them.
- Implemented in Eclipse Platform Equinox Runtime.
- Therefore every Eclipse plug-in is an OSGi bundle.

5

UCL

## Plug-in deployment with OSGi

- Java class loading is too inflexible for realizing a powerful plug-in mechanism. (why?)
- OSGi and its implementation in Equinox overcome this:




6

UCL

## Project Management in Eclipse Workspaces

- Eclipse manages one workspace (at a time)
- Workspace is mapped to a single file system directory
- A workspace contains multiple projects
- Projects are mapped to subdirectories of the workspace directory
- A project contains a number of artifacts (that may be arranged in a hierarchical way themselves)
- Projects are of one or several natures
- A nature controls which tools, perspectives and views are available to artifacts of the project.

7

UCL

## Change management in Eclipse Workspaces

- The same resource may be used by multiple different plug-ins, e.g. Java source is used by program editor, incremental compiler and debugger.
- Workspace provides for plug-ins to track changes to resources.
- E.g. the incremental java compiler can register an interest in a Java file and it will get notified when the file is changed.
- Notification contains a report of differences in the form of a resource delta

8

UCL

## Build management in Eclipse Workspaces

- Platform contains an incremental project builder framework
- Provides build plug-ins with a resource tree delta
- Resource tree delta captures resource differences since last build
- Build plug-in (e.g. a Java compiler) can then incrementally rebuild the affected dependent resources (e.g. Java class files)
- As a result Eclipse projects have an up-to-date executable available almost all the time.

9

UCL

## User Interface Management in Eclipse

- Integrated Software Development Environments are demanding from a user interface point of view
- Different plug-ins should all have same look-and-feel.
- The Eclipse platform provides
  - SWT - Standard Widget Toolkit that provides low-level GUI programming abstractions on top of native operating system GUI libraries.
  - JFace - a GUI programming framework build on SWT with abstractions needed in IDEs
  - Workbench - The UI personality of the Eclipse platform. Provides abstractions for editors, views and perspectives
- Neither Java's AWT nor Swing are used in Eclipse

10

UCL

## Eclipse Standard Widget Toolkit

- In user interface toolkit design: trade-off between portability and speed.
- Extreme ends of the spectrum:
  - Java Swing
  - Windows win32 library
- SWT achieves good compromise
  - Provides standard API for widgets needed in advanced GUIs
  - Has different implementations based on native OS GUIs
  - Provides consistent look-and-feel with all other applications on the OS
  - Insulates remainder of the platform from OS dependencies

11

UCL

## Eclipse JFaces

- Provides:
  - Image and font registries
  - Dialogue framework
  - Preference framework
  - Wizard framework
  - Progress reporting for long running operations
  - Action mechanism
  - Viewer framework
- Implemented solely using SWT
- SWT can still be used directly (e.g. by GEF)

12

UCL

## Eclipse Workbench

- Manages editors, views and perspectives
- Main Eclipse window contains one perspective at a time and users can switch between them.
- Perspective contains a selection of editors and views.
- Workbench handles menu bar to which editors may contribute commands.
- Workbench handles status of each of these (can be maximized, minimized)
- Workbench persists and reloads visualization status.

13

UCL

## Key Points

- Integrated Development Environments are very complex software systems
- Reuse of common platform infrastructure with UI, process support and run-time support
- Eclipse platform arguably the most sophisticated
- Basis of not just the JDE but a large number of commercial development environments available from IBM, Oracle.
- Availability of Platform lowers barrier to entry. Large number of IDE vendors are SMEs.

14

UCL

## References


- E. Clayberg and D. Rubel. Eclipse: Building Commercial-Quality Plug-ins. Pearson. 2006
- http://www.eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.pdf
- http://www.osgi.org/osgi_technology


15