



Agent and Object Technology Lab
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Parma



Eclipse Platform 3.3 (EUROPA)

Alessandro Negri

negri@ce.unipr.it

<http://www.ce.unipr.it/people/negri/>

- ◆ “Eclipse, an extensible development platform and application frameworks for building software”



- ◆ Versione attuale: Eclipse 3.3.1 (Sett. 2007)
- ◆ Scaricabile dal sito ufficiale: www.eclipse.org

- ◆ Eclipse è stato creato dai gruppi OTI e IBM già responsabili dei prodotti:
 - IBM VisualAge/Smalltalk (Smalltalk IDE)
 - IBM VisualAge/Java (Java IDE)
 - IBM VisualAge/Micro Edition (Java IDE)
- ◆ Inizialmente sviluppato da un gruppo di 40 programmatori impegnati a tempo pieno
- ◆ Donato da IBM alla comunità open-source
 - Versione community di WebSphere® Studio Application Developer (WSAD)
- ◆ Diviso in 3 progetti principali
 - Piattaforma (Core/Runtime)
 - JDT (Java Development Tools)
 - PDE (Plug-in Development Environment)

- ♦ Eclipse Public License - v1.0
- ♦ La licenza EPL è stata approvata dall'Open Source Initiative (OSI)



“The Open Source Initiative (OSI) is a non-profit corporation formed to educate about and advocate for the benefits of open source and to build bridges among different constituencies in the open-source community.”

<http://www.opensource.org/licenses/eclipse-1.0.php>

- ♦ Permette agli sviluppatori di modificare il codice (open source) ed anche di riutilizzarlo all'interno di prodotti commerciali (free redistribution)

“in many ways Eclipse is the Emacs for the 21st century“

- ♦ Fornire una piattaforma aperta per lo sviluppo di applicativi che:
 - sia utilizzabile sul più alto numero di sistemi operativi
 - sia utilizzabile con un'interfaccia grafica o senza (GUI o non-GUI)
- ♦ Indipendente dal linguaggio di programmazione
 - Gestisca in modo completo vari tipi di linguaggi: HTML, Java, C/C++, XML, JSP, EJB ...
- ♦ Facilitare l'integrazione di tool specifici
 - Possibilità di inserire nuovi tool al prodotto di base per venire incontro alle esigenze del singolo sviluppatore
- ♦ Ma che cosa è la piattaforma Eclipse?
 - ...

- ◆ Universalmente riconosciuta come “**La**” piattaforma per lo sviluppo in linguaggio Java
- ◆ Con tutti gli strumenti necessari...
 - Language aware editor, viste, ...
 - Supporto al refactoring
 - Unit testing e debugging integrato
 - Compilazione incrementale ed esecuzione
 - Supporto allo sviluppo in team
 - Supporto a database esterni, CVS
 - ...

- ♦ IDE = Integrated Development Environment ovvero un ambiente integrato di sviluppo
- ♦ Eclipse + JDT = Java IDE
 - Framework per Java
 - Language aware editor
 - Compilazione incrementale
 - Debugging integrato
- ♦ Eclipse + CDT = C/C++ IDE
 - First class framework for C/C++
 - Language aware editor
 - Refactoring, ricerca
- ♦ Eclipse + PHP = PHP IDE
- ♦ Eclipse + JDT + CDT + PHP = Java, C/C++, PHP IDE
- ♦ ...

- ◆ Estensibilità attraverso l'implementazione OSGi
 - I plug-in rendono Eclipse esattamente quello che ci serve che sia
- ◆ Il focus è sullo sviluppo di una piattaforma universale di framework e tool
- ◆ I tool estendono la piattaforma Eclipse utilizzando plug-in
 - Business Intelligence and Reporting Tools (BIRT)
 - Eclipse Communications Framework (ECF)
 - Web Tools Project (WTP)
 - Eclipse Modelling Framework (EMF)
 - Graphical Editing Framework (GEF)
 - Test and Performance Tooling Project (TPTP)

- ◆ Rimuovi gli elementi dell'IDE, il supporto al linguaggio Java, il supporto allo sviluppo in team ... e rimani con un semplice e generico application framework
 - Supporto multi-piattaforma
 - Linux, Windows, Mac OSX, UNIX...
 - Insieme di widget per la grafica
 - Integrazione nativa con il sistema operativo (drag and drop, integrazione OLE/XPCOM)
- ◆ Una piattaforma per creare “rich clients” (RCP)

- ◆ Dieci (10) progetti di alto livello
 - The Eclipse Project
 - Tools
 - Web Tools Platform
 - Test & Performance Tools Platform
 - Business Intelligence and Reporting Tools
 - Eclipse Modeling Project
 - Data Tools Platform
 - Device Software Development Platform
 - SOA Tools Platform
 - Technology (Incubators)
- ◆ Decine di sottoprogetti

- ◆ Tutti i progetti di Eclipse sono disponibili gratuitamente per il download
- ◆ Centinaia di contribution
- ◆ Centinaia di plug-in realizzati da aziende, organizzazioni e singoli individui
- ◆ Comunità di blogger
 - <http://www.planeteclipse.org>
- ◆ Numerosi portali dedicati
 - EclipseZone (<http://www.eclipsezone.com>)
 - Eclipse Plug-in Central (<http://www.eclipseplugincentral.com>)
 - Altri (<http://www.eclipse.org/community>)

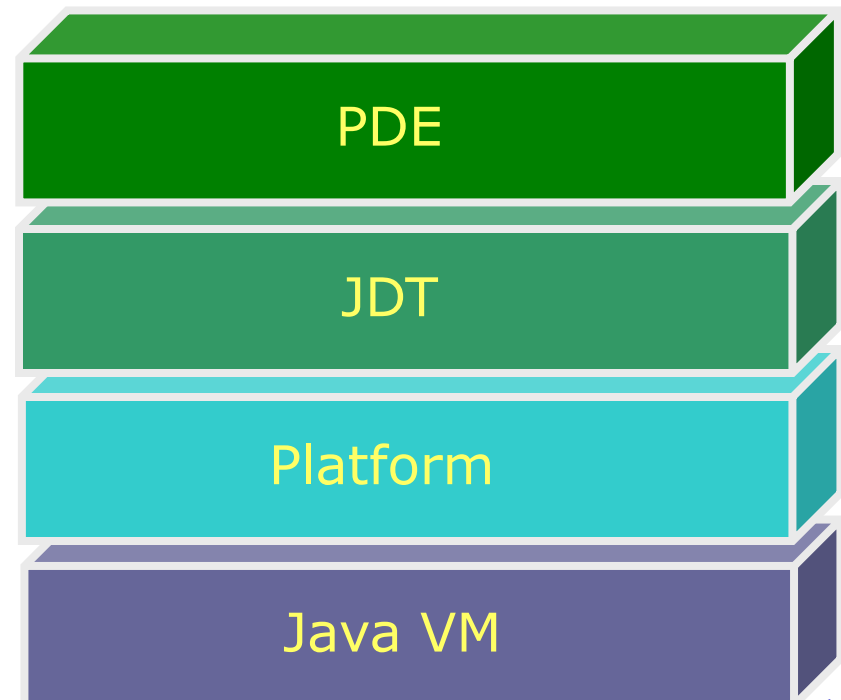
- ♦ Eclipse è una piattaforma universale per integrare tool di sviluppo
- ♦ Ha un'architettura aperta, estendibile, basata su plug-in

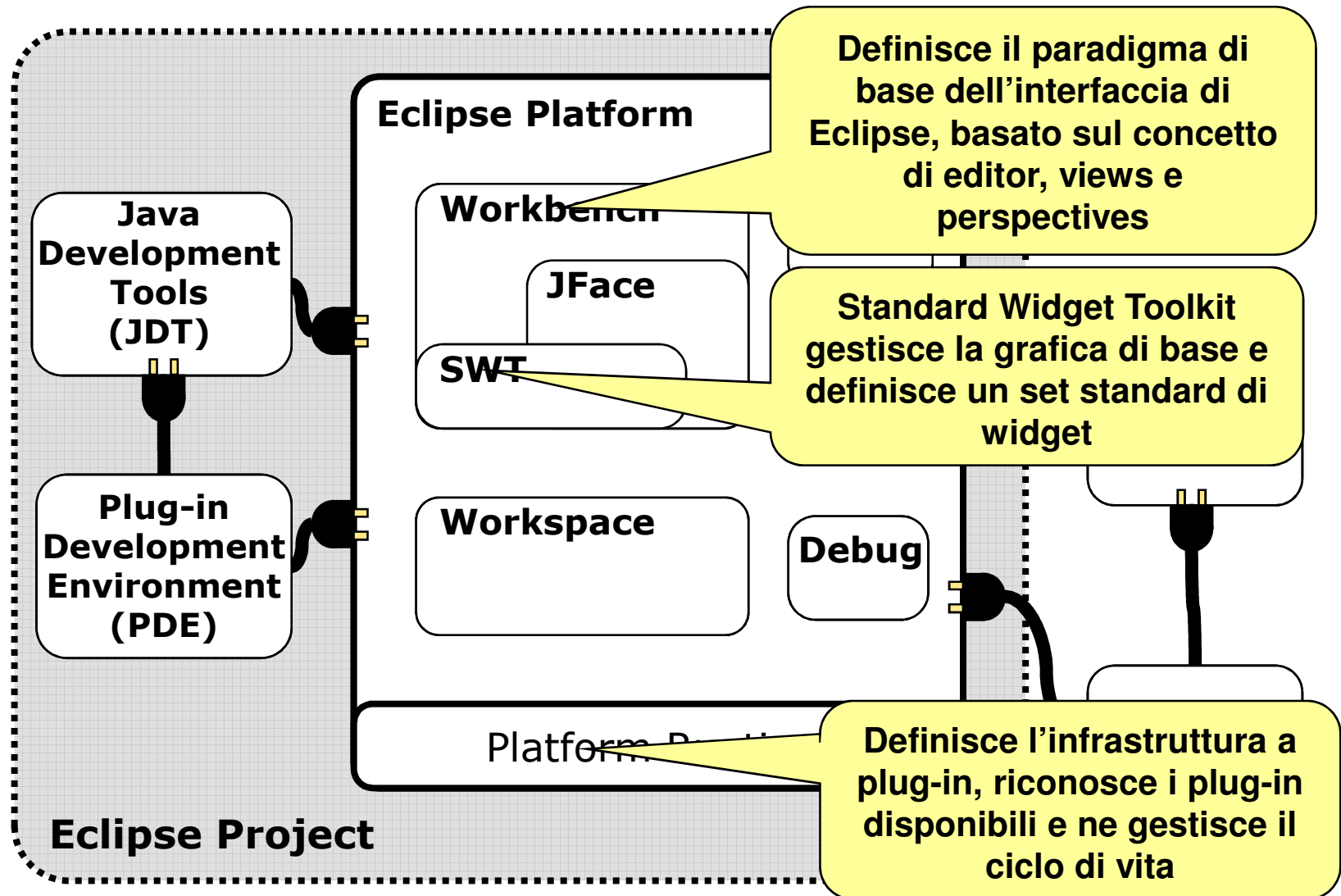
Plug-in Development
Environment

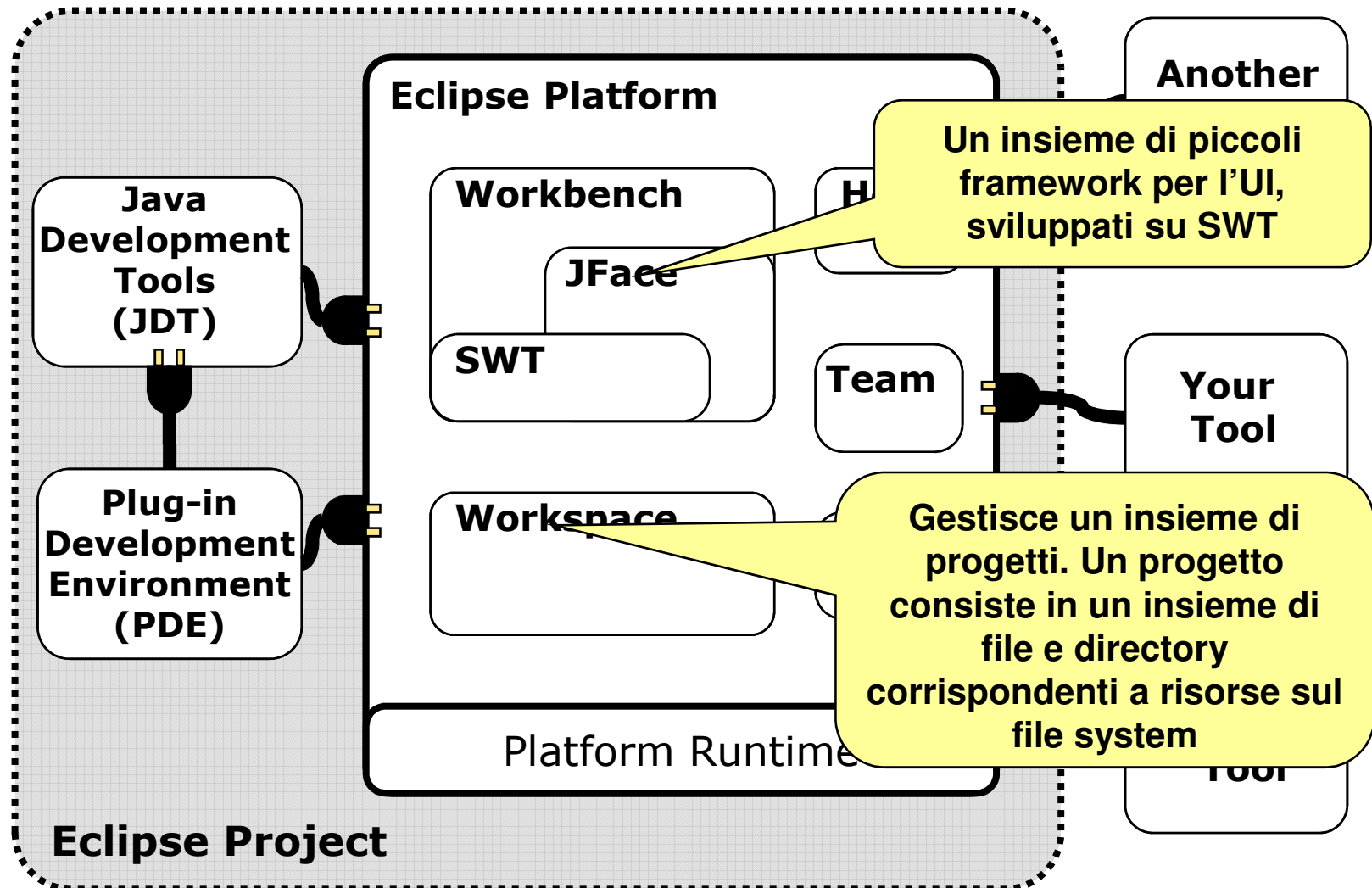
Java Development
Tools

Eclipse Platform

Standard Java2
Virtual Machine







- ◆ Eclipse Platform Runtime è un micro-kernel
 - Tutte le funzionalità sono implementate in plug-in
- ◆ Eclipse Platform Runtime gestisce l'inizializzazione e la creazione dell'ambiente di lavoro
 - Ricerca tutti i plug-in installati sul disco
 - Unisce tutte le Extension con i rispettivi Extension Point
 - Costruisce un registro globale dei plug-in
 - Crea una copia in cache del registro per l'esecuzione successiva

- ◆ Un Extension Point è un punto di aggancio per i contributi dei plug-in
 - “Qualcuno” (un altro plug-in) lo ha dichiarato
 - org.eclipse.ui
 - E' identificato univocamente nella piattaforma (Id)
 - org.eclipse.ui.ActionSets
 - Ha uno schema che regola le informazioni dichiarative che i contributor devono fornire
 - Può contenere un set di Action, ecc...
 - Definisce un'interfaccia che la classe del contributor deve implementare
- ◆ Una Extension è la dichiarazione dalla Contribution
- ◆ Chi ha dichiarato l'Extension Point in genere interroga il registry per sapere chi offre delle Extension
 - Es. il workbench verifica chi fornisce una Extension per gli ActionSets per costruire l'interfaccia utente dell'IDE

- ◆ Contribution Rule: “Everything is a contribution”
 - L'intera piattaforma Eclipse (escluso il Runtime Kernel) non ha nessuna funzionalità built-in, tutto è basato sulla Contribution
 - Eclipse + JDT ≈ 60 plugins
 - IBM Websphere IDE ≈ 600 plugins
 - Il Runtime Kernel è in grado di gestire migliaia di plugin
- ◆ Lazy Loading Rule: ” Contributions are only loaded when they are needed”
 - “Declaration vs Implementation”
 - La piattaforma mantiene le informazioni di tutti i plugin disponibili
 - Manifest file → plugin.xml
 - L'implementazione viene caricata alla prima richiesta di accesso
 - PLUGIN_XXX.jar

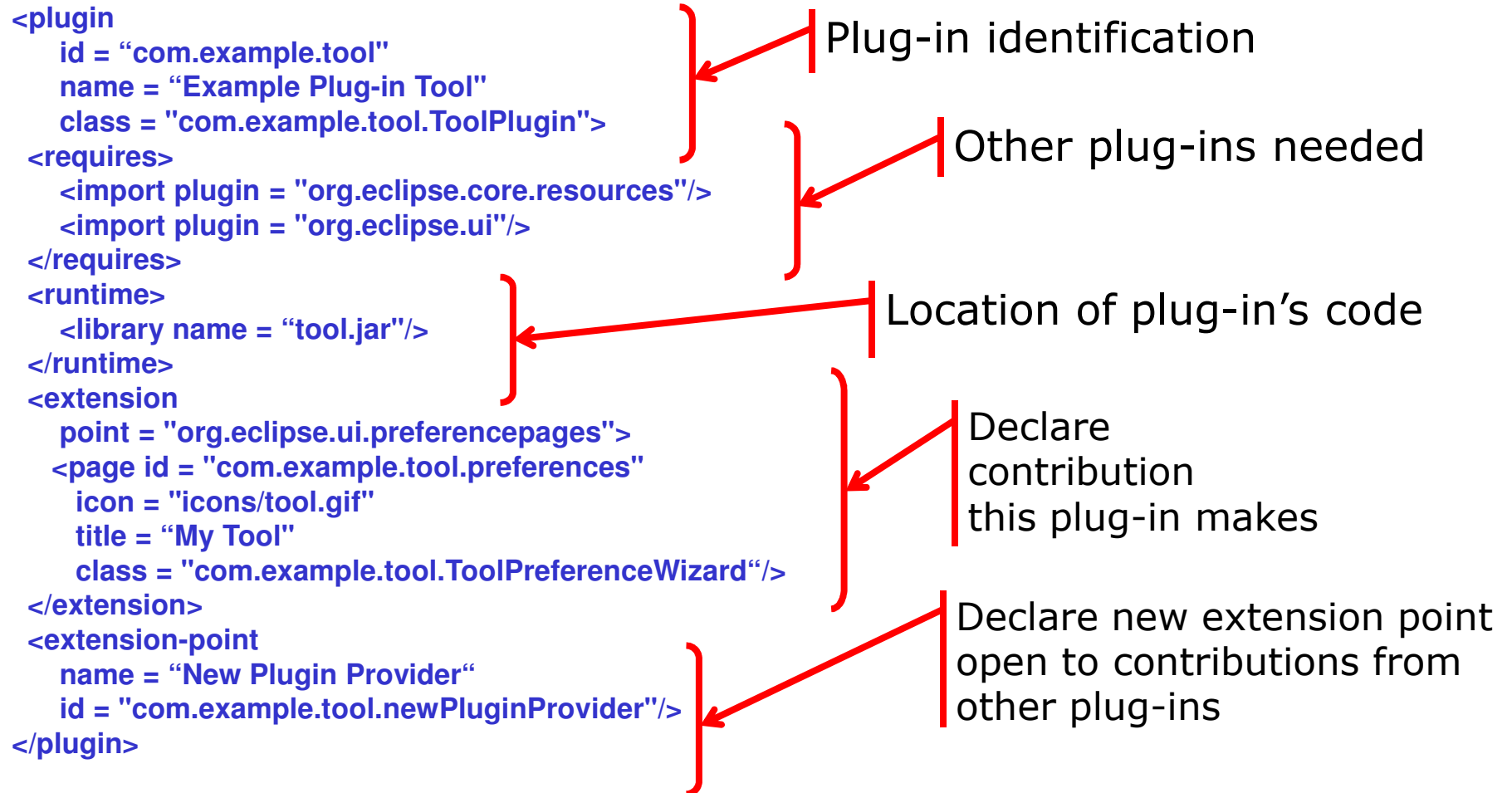
- ♦ Plug-in: la più piccola unità funzionale in Eclipse
 - “a piece of behavior that is outside the run-time kernel”
(Contributing to Eclipse: Principles, Patterns and Plug-Ins by E.Gamma,K.Beck)
 - E' un componente strutturato che descrive se stesso al sistema utilizzando un manifest file (plugin.xml)
 - Può racchiudere numerose funzionalità (es. HTML editor) oppure una semplice funzione (es. l'azione per creare un file zip)
- ♦ Un plugin si concretizza fisicamente in una directory con all'interno:
 - **plugin.xml**: il manifest, una descrizione della Contribution
 - **resources**: risorse utili al plugin (es. icone) - OPZIONALE
 - **Java code**: strutturato in file .jar - OPZIONALE

- ◆ Ha un identificatore univoco – plugin ID
- ◆ Porta una Contribution alla piattaforma
 - Descritta tramite il manifest
 - Implementata (quando serve) attraverso le librerie del plug-in
 - Può estendere un aspetto specifico della piattaforma (**Extensions**), per esempio:
 - Introdurre un nuovo tipo di risorsa o estendere un tipo di risorsa già presente
 - Può contribuire con nuovi punti di estensione (**Extension Points**), per esempio:
 - Validatori di documenti XML
- ◆ Può dipendere da altri plug-in
 - Dipendenza esplicita nella descrizione e gestita in automatico dalla piattaforma

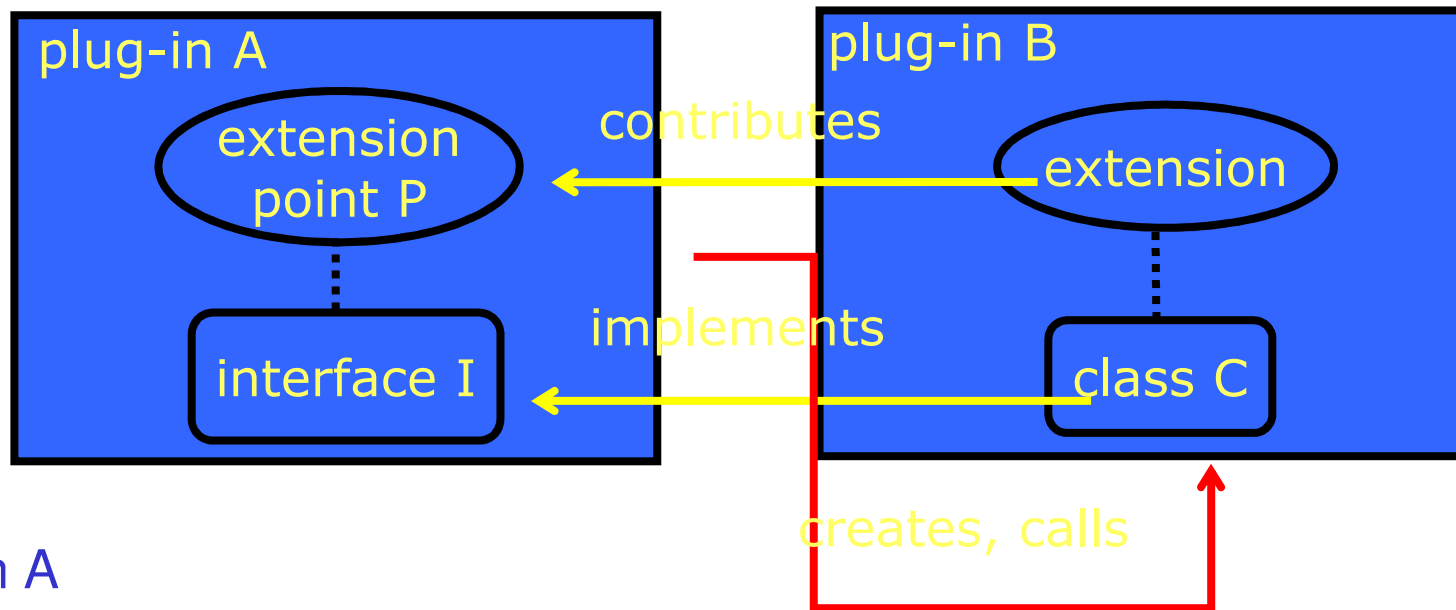
- ◆ Ogni plug-in:
 - Contribuisce ad 1 o più Extension Point
 - Dichiarare 1 o più Extension Point (OPZIONALE)
 - Dipende da un set di altri plug-in (OPZIONALE)
 - Contiene librerie di codice Java o altri file (OPZIONALE)
 - Risiede in una sottodirectory a lui dedicata
- ◆ Plug-in Manifest
 - Dichiarare tutte le “Contributions”
 - Implementa interfacce o fornisce API
 - plugin.xml: descrive le proprietà del plug-in

- ◆ File XML, che permette di dichiarare
 - L'esistenza del plug-in
 - Dati identificativi e descrittivi del plug-in
 - Le librerie che fanno parte del plug-in
 - Jar generati dal codice sorgente
 - Altre librerie incluse nel plug-in
 - Le dipendenze del plug-in
 - Altri plug-in i cui servizi sono necessari in fase di runtime
 - La disponibilità dei plug-in dichiarati viene verificata a runtime
 - I Jar degli altri plug-in vengono inclusi nel classpath del plug-in a runtime
 - I servizi offerti dal plug-in

plugin.xml



Architettura del Plug-in (2)



Plug-in A

- Dichiarare un Extension Point P
- Dichiarare un'interfaccia I per il punto P

◆ Plug-in B

- Implementare l'interfaccia I con la sua classe C
 - Inserisce la classe C come Contribution all'Extension Point P
- ◆ Il Plug-in A istanzia la classe C e richiama i metodi dell'interfaccia I

- ◆ Singleton che rappresenta a runtime il plug-in
- ◆ Estende la classe
 - `org.eclipse.core.runtime.Plugin`
- ◆ Fornisce metodi di hook per la gestione del ciclo di vita del plug-in
 - Start
 - Stop
- ◆ Accentra in genere servizi comuni a tutti gli elementi del plug-in
 - Resource bundle

- ◆ Ogni plug-in ha il proprio Java class loader
 - Si delega l'esecuzione al plug-in stesso
 - Si restringe la visibilità alle API esportate
- ◆ Le contribution sono processate senza l'attivazione dei plug-in
 - Esempio: eventuali menù sono costruiti dal manifest relativo alle varie contribution
- ◆ I plug-ins sono attivati solo al momento in cui sono necessari
 - Esempio: il plug-in è attivato solamente quando l'utente seleziona una voce corrispondente in un menù
 - Soluzione scalabile se si hanno molti plug-in installati
 - Riduce i tempi di startup

- ◆ I Plug-in Fragments memorizzano alcuni files del plug-in
 - Installabili separatamente
- ◆ Ogni fragment ha una propria sottodirectory
 - Manifest separato
- ◆ Logical plug-in = Base plug-in + fragment
- ◆ I Plug-in Fragments sono utilizzati per:
 - Sviluppare caratterizzazioni specifiche per un sistema operativo
 - Gestire la localizzazione (traduzioni)

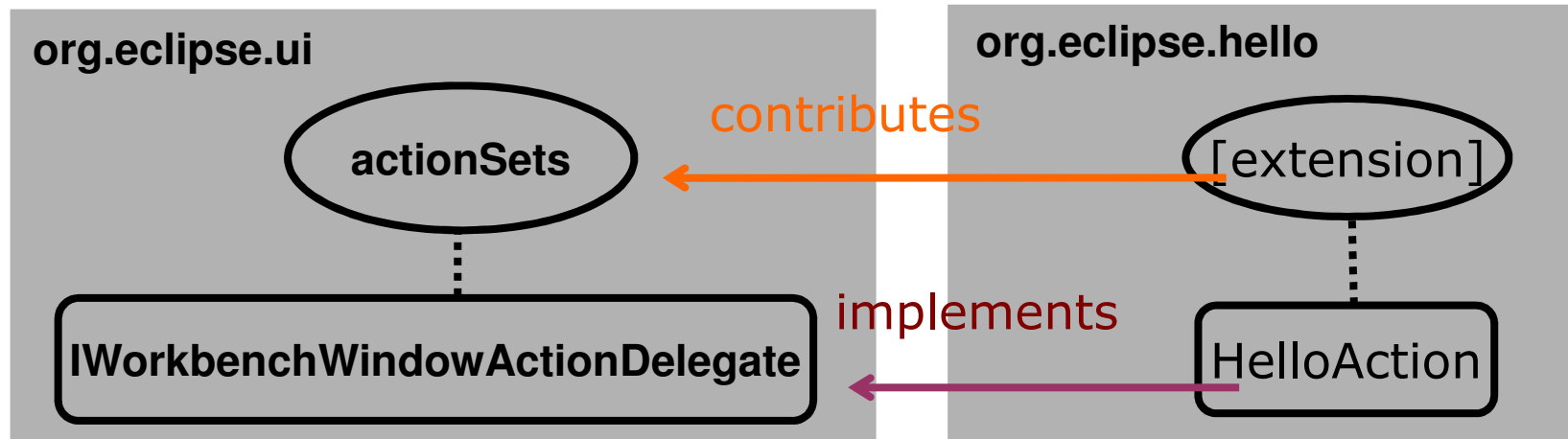
- ◆ Identificare gli extension point
- ◆ Comprendere ed analizzare le specifiche di ogni extension point
- ◆ Definire il plug-in manifest (plugin.xml)
- ◆ Scrivere la class che implementa l'interfaccia definita dall'extension point
- ◆ Installare il plug-in nella rispettiva cartella `plugins`

- ◆ Creare un pulsante “Hello” nella toolbar di Eclipse
- ◆ Aggiungere una action al bottone in modo che possa essere aperta una finestra con la scritta “Hello World”

```
<plugin
  id="org.eclipse.hello"
  name="Hello World"
  version="1.0.0">
  ...
</plugin>
```

- ◆ Problema:
 - come integrare il nostro plug-in all'interno della piattaforma?
- ◆ Soluzione:
 - Eclipse fornisce il plug-in `org.eclipse.ui`
 - Non ha niente a che fare con la User Interface
 - `org.eclipse.ui` ha un extension point `org.eclipse.ui.actionSets`

- ◆ Utilizzato per aggiungere contribution alla piattaforma, in particolare:
 - Nuove voci ai menù
 - Nuovi pulsanti nella toolbar
- ◆ I pulsanti nelle toolbar sono raggruppati in Action Sets
- ◆ Contribuire con un pulsante
 - Definire un action set
 - Definire un'azione (che verrà invocata alla pressione del pulsante)
 - Definire la posizione in cui appare il pulsante (toolbar path)
 - Dichiarare l'implementazione dell'azione (classe Java)
 - Dichiarare l'aspetto del pulsante (icona)



- ♦ **org.eclipse.ui**
 - Ha un extension point **actionSets**
 - Dichiarare l'interfaccia **IWorkbenchWindowActionDelegate**
- ♦ **org.eclipse.hello**
 - Implementa l'interfaccia **IWorkbenchWindowActionDelegate** con la classe **HelloAction**
 - Contribuisce all'extension point **actionSets** con la classe **HelloAction**

```
public class HelloAction implements
    IWorkbenchWindowActionDelegate {

    public void dispose() {...}

    public void init(IWorkbenchWindow window) { ... }

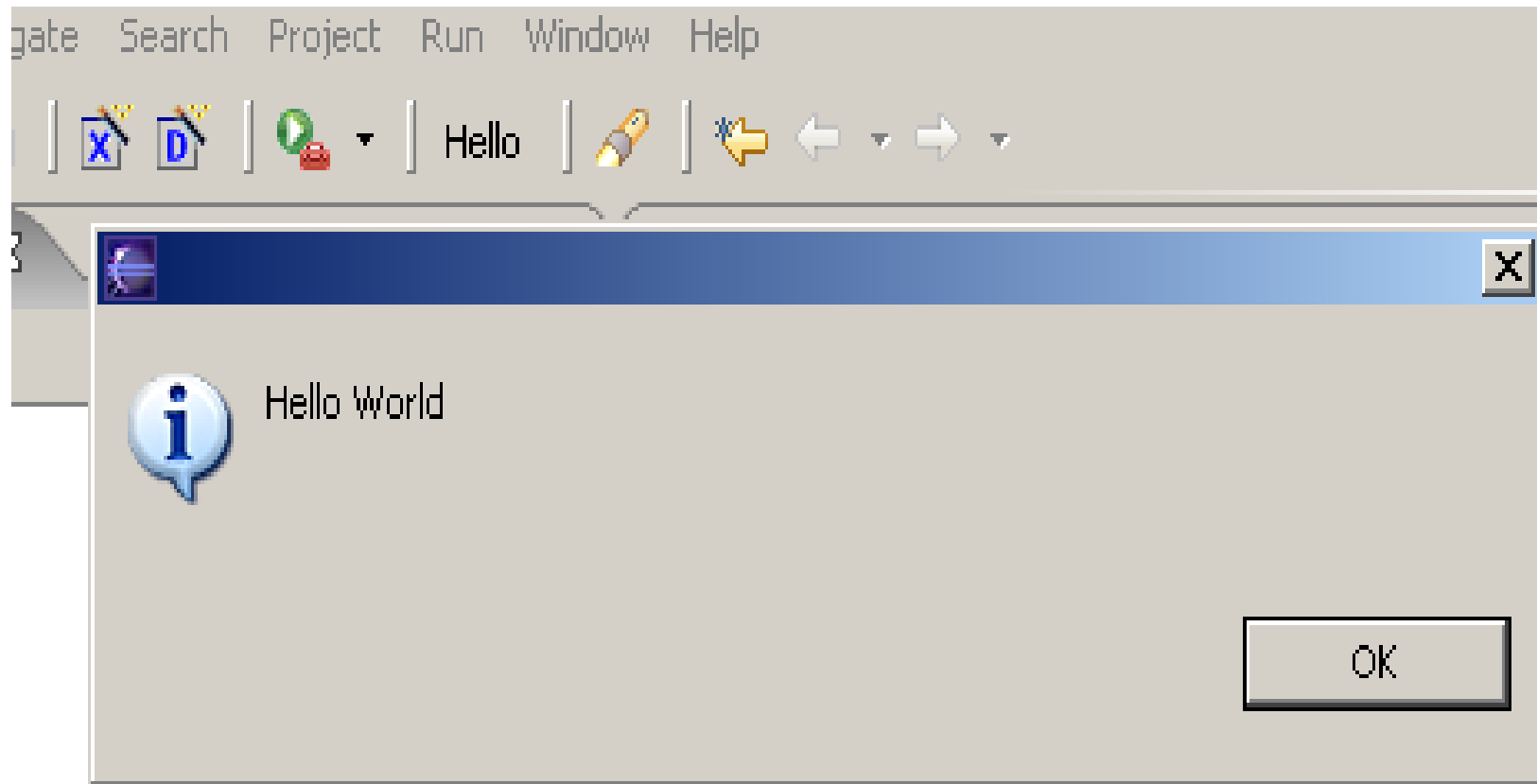
    public void run(IAction action) {
        MessageDialog.openInformation(
            null, null, "Hello World");
    }

    public void selectionChanged(IAction action, ISelection
    selection) {...}
}
```


- ◆ Lista di tutti i plug-in che sono necessari per la compilazione e l'esecuzione del nostro plug-in

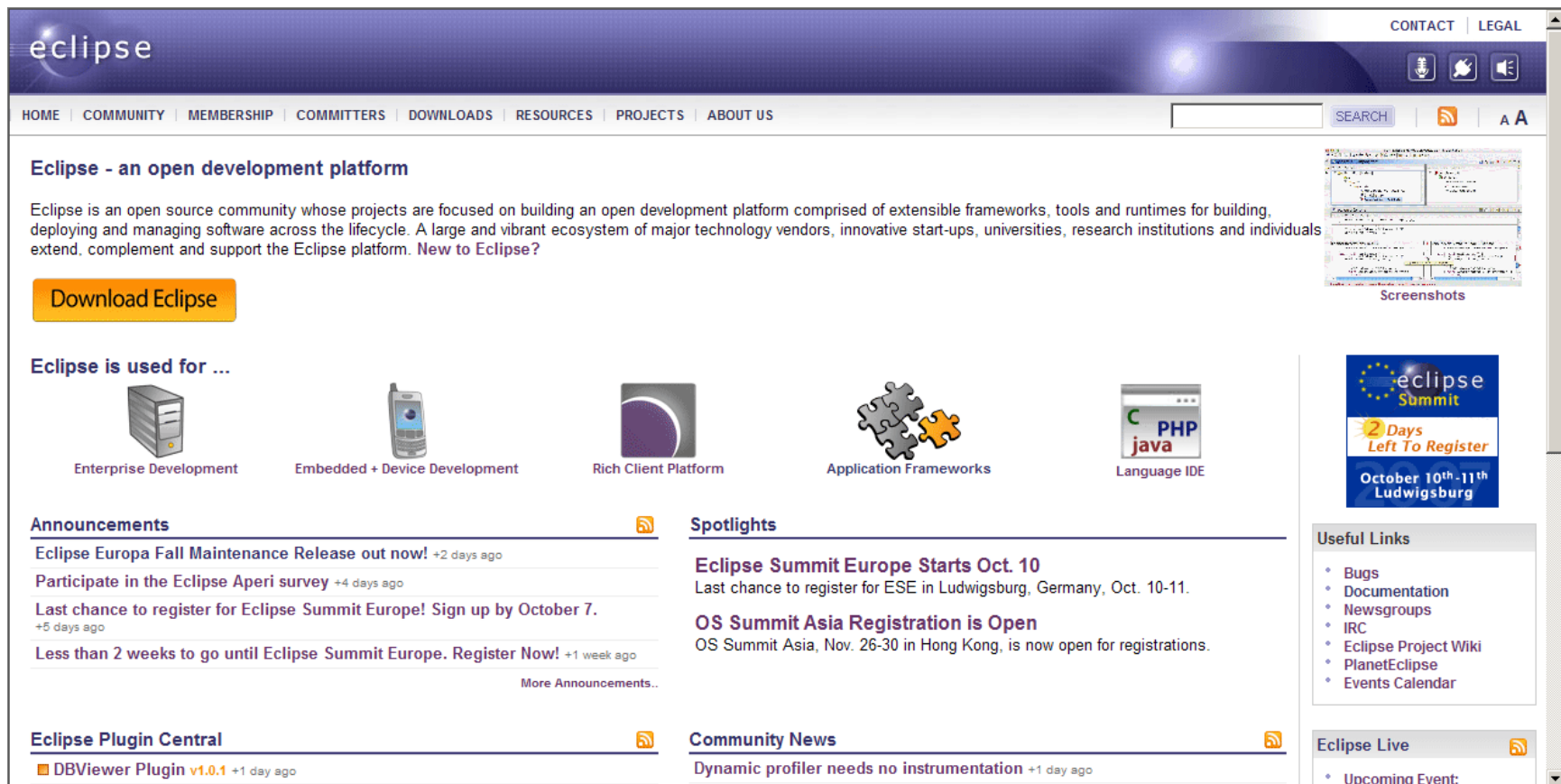
```
<plugin>
  ...
  <requires>
    <import plugin="org.eclipse.ui"/>
  </requires>
  ...
</plugin>
```

```
<plugin>
  ...
  <extension point="org.eclipse.ui.actionSets">
    <actionSet
      id="org.eclipse.hello.actionSet"
      label="Hello Set">
      <action
        id="org.eclipse.hello.HelloAction"
        label="Hello"
        toolbarPath="helloGroup"
        class="org.eclipse.hello.HelloAction">
      </action>
    </actionSet>
  </extension>
  ...
</plugin>
```



- ◆ Un plug-in si presenta come una cartella strutturata, presente nella cartella “plugins” di Eclipse
- ◆ Un plug-in viene rilasciato nella maggior parte dei casi come uno zip
- ◆ Applicazioni medio-grandi sono composte da più plug-in (da qualche unità a qualche centinaio)
 - Feature: un insieme coerente di plugin
 - Con una descrizione
 - Con un meccanismo automatico di installazione
 - Con una licenza

<http://www.eclipse.org>



The screenshot shows the Eclipse website homepage. At the top is a dark blue header with the Eclipse logo and navigation links: HOME, COMMUNITY, MEMBERSHIP, COMMITTERS, DOWNLOADS, RESOURCES, PROJECTS, ABOUT US. There is also a search bar and social media icons. Below the header, the main content area features a large section titled "Eclipse - an open development platform" with a description of the open source community and a prominent orange "Download Eclipse" button. To the right of this section are "Screenshots" of the IDE. Below the main section is a row titled "Eclipse is used for ..." with five icons representing different development areas: Enterprise Development (server rack), Embedded + Device Development (mobile phone), Rich Client Platform (Eclipse logo), Application Frameworks (puzzle pieces), and Language IDE (PHP and Java logos). Further down, there are sections for "Announcements" (listing recent releases and surveys), "Spotlights" (highlighting Eclipse Summit Europe and OS Summit Asia), "Eclipse Plugin Central" (showing the DBViewer Plugin), and "Community News" (mentioning a dynamic profiler). On the right side, there is a "Useful Links" section with links to bugs, documentation, newsgroups, IRC, Eclipse Project Wiki, PlanetEclipse, and Events Calendar. At the bottom right, there is an "Eclipse Live" section with an "Upcoming Event" link. A banner for the "eclipse Summit" is also visible, stating "2 Days Left To Register" for October 10th-11th in Ludwigsburg.



Eclipse IDE for Java Developers (78MB)

Versione specifica per lo sviluppo di applicazioni Java

- **Java editing with validation**
- **Compilazione incrementale**
- **Cross-referencing**
- **Code assist**
- **XML Editor**
- **Mylyn**





Eclipse IDE for Java EE Developers (125MB)


Versione che contiene tutti gli strumenti necessari per creare applicazioni Java Enterprise Edition (Java EE).

- **Java editing**
- **Compilazione incrementale**
- **Supporto a Java EE 5**
- **Editor grafico per HTML/JSP/JSF**
- **Tool per la gestione di database**
- **Supporto integrato per i più conosciuti application servers**





Eclipse IDE for C/C++ Developers (63MB)

Versione studiata appositamente  i sviluppatori in C/C++

- Editor con syntax highlighting ed autocompletamento
- Compilatore e Debugger
- Funzionalità di ricerca
- Generatore di makefile





Eclipse for RCP/Plug-in Developers (153MB)

Versione specifica per lo sviluppo di applicazioni basate sulla piattaforma Eclipse

- Plug-in specifici per lo sviluppo
- Eclipse Rich Client Platform (RCP)
- Java editing con compilazione incrementale
- Plug-in Development Environment (PDE)
- XML Editor
- Mylyn
- Codice sorgente completo della piattaforma










Eclipse Classic (140MB)

Versione tradizionale di Eclipse che contiene tutto quello che è necessario per lo sviluppo di applicazioni

- Eclipse IDE
- Eclipse Rich Client Platform (RCP)
- Plug-in Development Environment (PDE)
- Codice sorgente completo della piattaforma



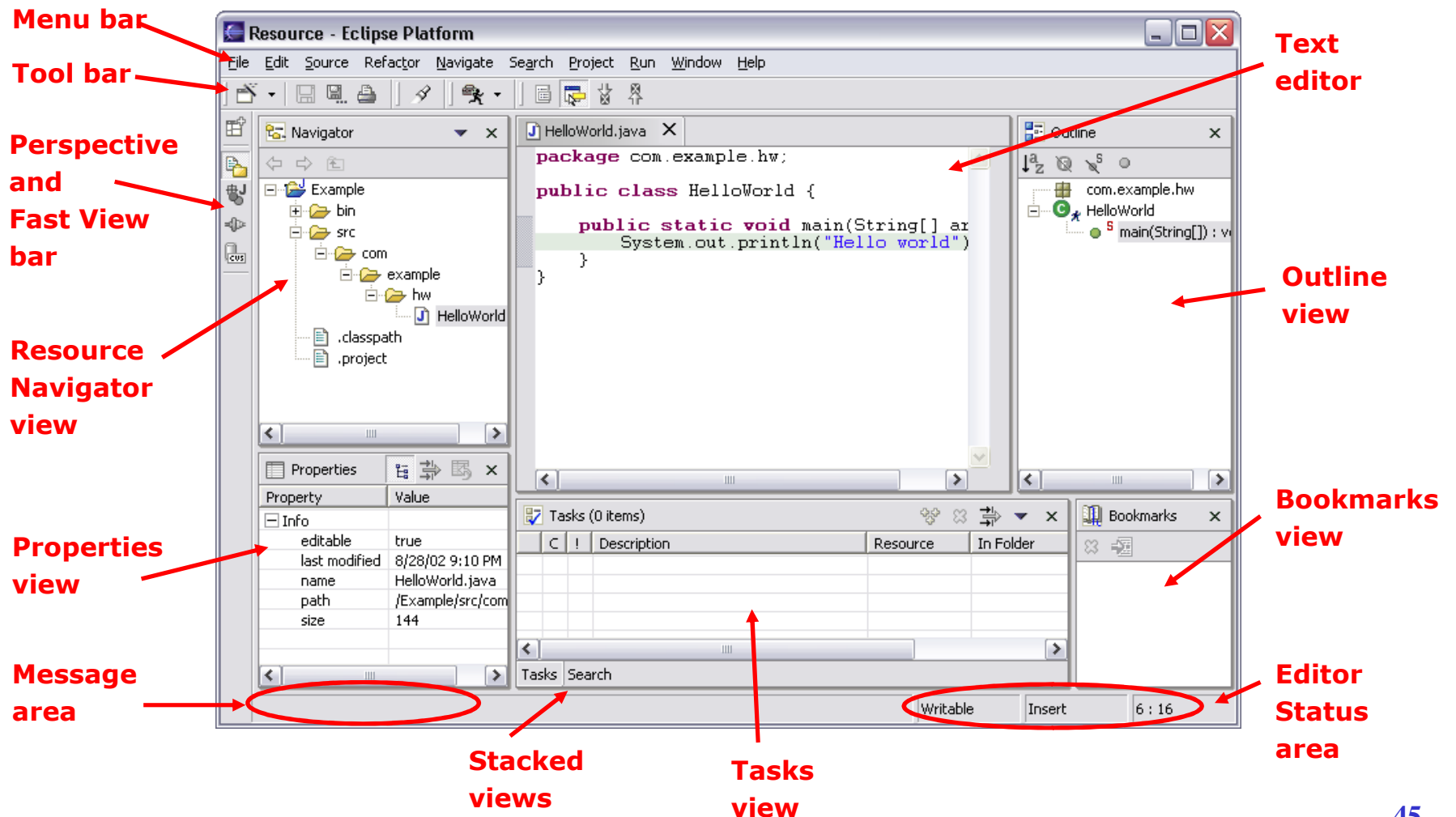
	 Java	 JEE	 C/C++	 RCP/Plugin	 Classic
RCP/Platform	✓	✓	✓	✓ ✓	✓ ✓
CVS	✓	✓	✓	✓	✓ ✓
EMF	✓	✓		✓	
GEF	✓	✓		✓	
JDT	✓	✓		✓	✓ ✓
Mylyn	✓	✓		✓	
WST	✓	✓		✓	
PDE		✓		✓ ✓	✓ ✓
Datatools		✓			
JST		✓			
CDT			✓		

Legend:

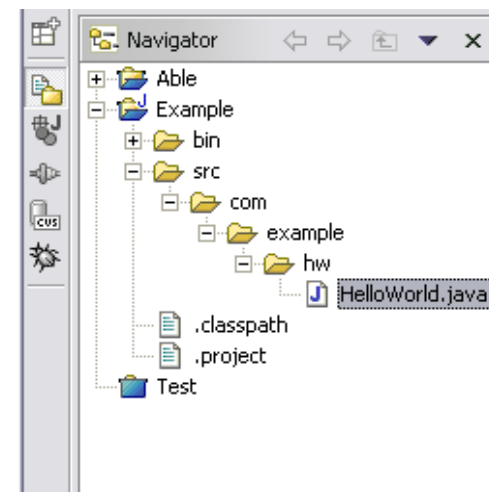
- ✓ ✓ Included (with Source)
- ✓ Included
- ✓ Partially Included

<http://www.eclipse.org/epp/content.php>

- ♦ Il Workbench rappresenta l'interfaccia utente della piattaforma Eclipse
- ♦ Il Workbench usa JFace e vi aggiunge alcune funzionalità più complesse
- ♦ Componenti fondamentali
 - Editors
 - Views
 - Perspectives



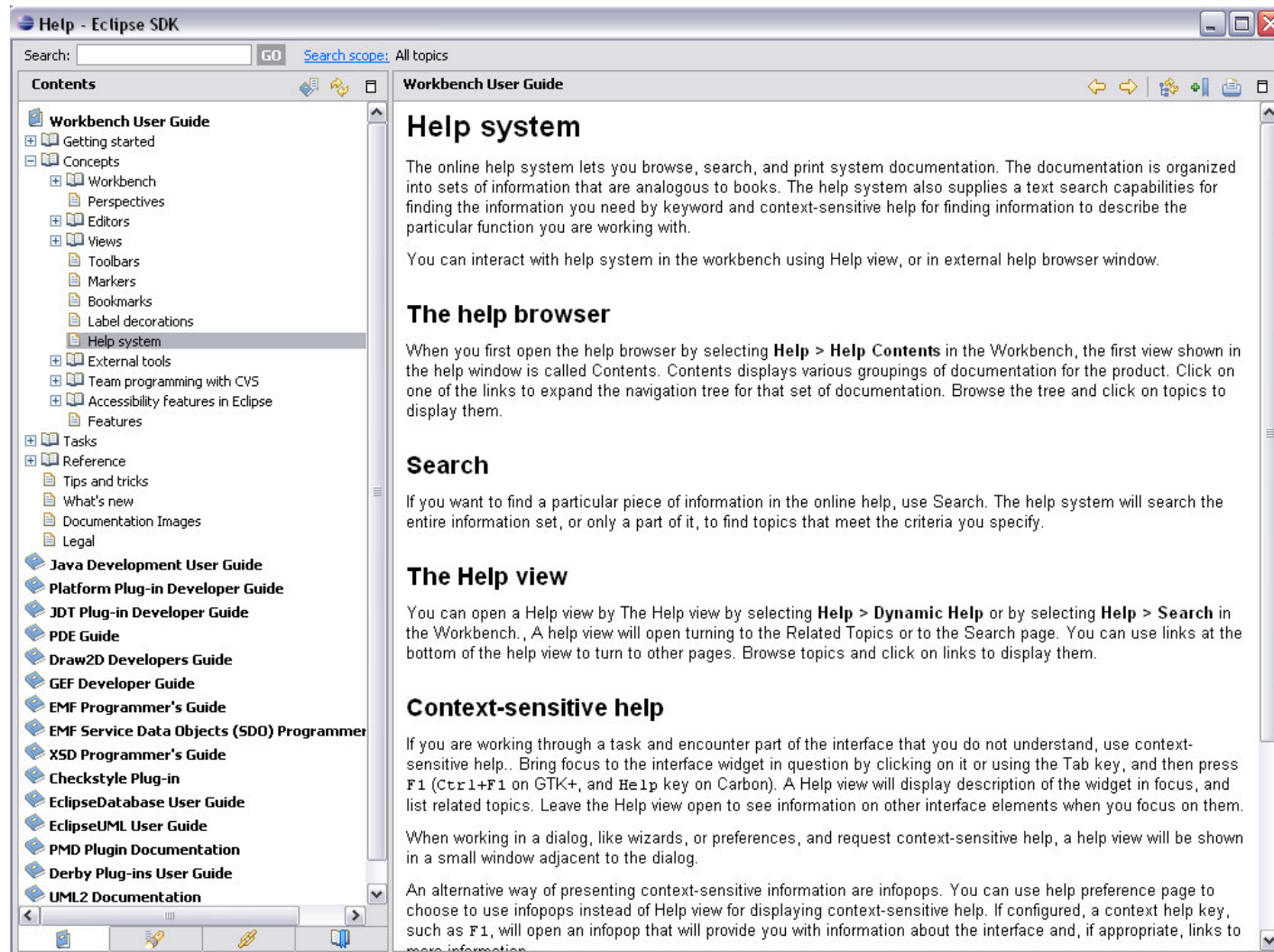
- ◆ **Risorse:** permette di accedere a files, directory o progetti
- ◆ Il Workspace è in grado di gestire 1 o più progetti
- ◆ I progetti corrispondono a directory nel file system
- ◆ Vista Navigator
 - Albero di file e directory
- ◆ Vari tool permettono di leggere, creare, modificare, cancellare risorse nel workspace



- ◆ Compaiono nell'area centrale del workbench
- ◆ Aggiungono azioni ai menu del workbench e alle toolbar
 - Open, edit, save, close lifecycle
- ◆ Il workbench fornisce
 - Extension Point per creare nuovi tipi di editors
 - Esempio: JDT crea un editor per gestire i file Java
 - API per editor di testo estendibili ed un framework

- ◆ Forniscono informazioni su di un oggetto
- ◆ Completano le funzionalità degli editors
 - Esempio: l'Outline View riassume il contenuto dell'editor
- ◆ Completano altre views
 - Esempio: la "Properties View" caratterizza una selezione
- ◆ Il workbench fornisce:
 - Extension Point per definire nuovi tipi di views
- ◆ La piattaforma Eclipse include un certo numero di views standard
 - Esempio: Resource Navigator, Outline, Properties, Tasks, Bookmarks, Search, ...

- ◆ Sono disposizioni di views ed editors
- ◆ Diverse perspectives possono essere utilizzate per gestire diverse operazioni
- ◆ Il workbench fornisce:
 - Extension Point per definire nuove perspectives
- ◆ La piattaforma Eclipse include un certo numero di perspectives standard
 - Esempio: Resource, Debug, ...



- ♦ Sito Ufficiale: **<http://www.eclipse.org/resources/>**
 - Articoli, Webinars, podcasts, libri, presentazioni, esempi di codice, dimostrazioni video, corsi
- ♦ Plug-in: **<http://www.eclipseplugincentral.org>**
- ♦ Tutorial for Beginners:
<http://eclipsutorial.sourceforge.net>
 - Video tutorial
- ♦ Eclipse Live: **<http://live.eclipse.org/>**
 - Webinars, podcast, articoli, presentazioni
- ♦ Eclipse Planet: **<http://www.planeteclipse.org/planet/>**
 - Feed Rss, articoli, blog
- ♦ Eclipse Zone: **<http://www.eclipsezone.com/>**



Agent and Object Technology Lab
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Parma



Eclipse Platform 3.3 (EUROPA)

Alessandro Negri

negri@ce.unipr.it

<http://www.ce.unipr.it/people/negri/>