

Platforma Eclipse: Pluginy, RCP oraz GEF

<http://www.apohllo.pl/texts/eclipse.pdf>

Aleksander Pohl

Akademia Górniczo-Hutnicza, Kraków

26. października 2007

Platforma Eclipse

- ▶ www.eclipse.org
- ▶ jedno z najbardziej popularnych narzędzi do tworzenia oprogramowania (Java, C++, php, Ruby, etc.)
- ▶ bibliotek widgetów SWT – dobra integracja z systemem operacyjnym
- ▶ podprojekty: modelowanie, testowanie, tworzenie aplikacji webowych, etc.
- ▶ www.eclipseplugincentral.com
- ▶ pluginy (wtyczki) pisane w Javie.
- ▶ aktualnie: 961 pluginów

Rich Client Platform

- ▶ wiki.eclipse.org/index.php/Rich_Client_Platform
- ▶ minimalny zestaw pluginów, pozwalający na uruchomienie platformy
- ▶ tworzenie własnych aplikacji opartych na bibliotece SWT i systemie komponentowym Eclipse
- ▶ możliwość wykorzystania innych pluginów (np. Download Manager)
- ▶ dostosowanie elementów graficznych w sposób charakterystyczny dla danego produktu (brak odniesień do Eclipse)
- ▶ eksportowanie projektu jako niezależnego produktu
- ▶ przykład: Aptana, Azureus, RadRails, ...

Graphical Editor Framework

- ▶ www.eclipse.org/gef/
- ▶ szybkie tworzenie graficznego edytora na podstawie istniejącego modelu aplikacji
- ▶ wzorzec Model-View-Controller
- ▶ składa się z 2 pluginów: `org.eclipse.draw2d` oraz `org.eclipse.gef`
- ▶ separacja modułu graficznego od modułu edytującego

Pluginy – wstęp

- ▶ plugin = wtyczka = komponent
- ▶ Equinox – podsystem odpowiedzialny za znajdowanie oraz ładowanie/wyładowywanie pluginów
- ▶ każdy plugin posiada plik `plugin.xml`, w którym opisane są jego własności, jako komponentu
- ▶ pluginy wykrywane są automatycznie przy starcie systemu (wystarczy skopiować plik do katalogu `plugins`)
- ▶ współpraca pomiędzy pluginami odbywa się poprzez mechanizm *rozszerzeń*

Pliki plugin.xml i MANIFEST.MF

- ▶ Identyfikacja wtyczki: nazwa, numer wersji, producent, etc.
- ▶ Zależności: pluginy wymagane do uruchomienia danego komponentu (np. `org.eclipse.gef` oraz wymagane biblioteki zewnętrzne (np. `log4j.jar`)
- ▶ Aktywacja – plik wywoływany przy ładowaniu/wyładowaniu wtyczki
- ▶ definicje *rozszerzeń* (extensions) i *punktów rozszerzeń* (extension points)

Elementy GUI

- ▶ Widoki (views) – elementy GUI, które mogą być niezależnie otwierane zamykane
`org.eclipse.ui.views`
- ▶ Perspektywy (perspectives) – zestawy widoków z określonymi zależnościami przestrzennymi
`org.eclipse.ui.perspectives`
- ▶ Edytory (editors) – główny element GUI, pozwalający na edycję danych. Zazwyczaj aktywny jest tylko jeden edytor, pozwalający na edycję jednego zasobu
`org.eclipse.ui.editors`

Elementy GUI - cd.

- ▶ Akcje (actions) – elementy GUI, które pozwalają użytkownikowi na interakcję z systemem. Mogą być wywoływane z poziomu menu głównego, menu kontekstowego, paska akcji oraz z pomocą skrótów klawiszowych `org.eclipse.ui.actionSets`, `org.eclipse.ui.viewActions`, `org.eclipse.ui.popupMenus`
- ▶ Menu (menus) – główne oraz kontekstowe. Pozwalają na grupowanie akcji.

Draw2d

- ▶ nadbudowany nad SWT
- ▶ odpowiedzialny za tworzenie i zarządzanie elementami graficznymi
- ▶ podstawowe abstrakcje: figury i połączenie
- ▶ figury tworzą hierarchię całość-część
- ▶ elementy dodatkowe: dekoracje, etykiety
- ▶ *layout manager*: odpowiedzialny za rozmieszczenie figur względem siebie
- ▶ *routing manager*: odpowiedzialny za rysowanie połączeń

Plugin GEF – wprowadzenie

- ▶ Draw2d należy do warstwy widoku, natomiast plugin GEF należy do warstwy kontrolera
- ▶ Plugin GEF pośredniczy w interakcji pomiędzy użytkownikiem a elementami modelu i widoku
- ▶ Pozwala na edycję graficznych elementów: dodawanie, usuwanie, przesuwanie, łączenie, etc.
- ▶ Podstawowa abstrakcja – `EditPart`

Elementy GEF

- ▶ `EditPart` – pośredniczą w wymianie informacji pomiędzy widokiem a modelem, np. zmiana el. modelu powoduje zmianę koloru figury; przesunięcie jednej figury na inną powoduje w modelu dołączenie jednego elementu jako potomka innego
- ▶ `EditPolicies` – określają możliwe do wykonania transformacje, przyjmują żądania, generują komendy
- ▶ `Requests` (żądania) – generowane są w wyniku działań użytkownika (np. przesunięcie elementu graficznego). Należą do warstwy widoku.
- ▶ `Commands` (komendy) – zawierają informacje o tym w jaki sposób powinien być zmodyfikowany element modelu w wyniku określonego żądania. Należą do warstwy modelu.

Elementy GEF – cd.

- ▶ Edytor zawiera paletę na której znajdują się narzędzia (tools) oraz szablony obiektów
- ▶ `Tools` (narzędzia) – pozwalają na edytowanie elementów graficznych, np. zaznaczanie, łączenie, przesuwanie, etc.
- ▶ szablony obiektów służą do tworzenia nowych obiektów. Zazwyczaj posiadają nazwę i ikonę pozwalające na szybką identyfikację tworzonego obiektu.

Zasoby

- ▶ system pomocy Eclipse
- ▶ www.eclipse.org
- ▶ www.eclipseplugincentral.com
- ▶ www.eclipse.org/gef/
- ▶ wiki.eclipse.org/index.php/Rich_Client_Platform
- ▶ symon.iisg.agh.edu.pl → Developer Manual → Implementacja → GUI
- ▶ www.apohllo.pl/dydaktyka/eclipse-rcp