






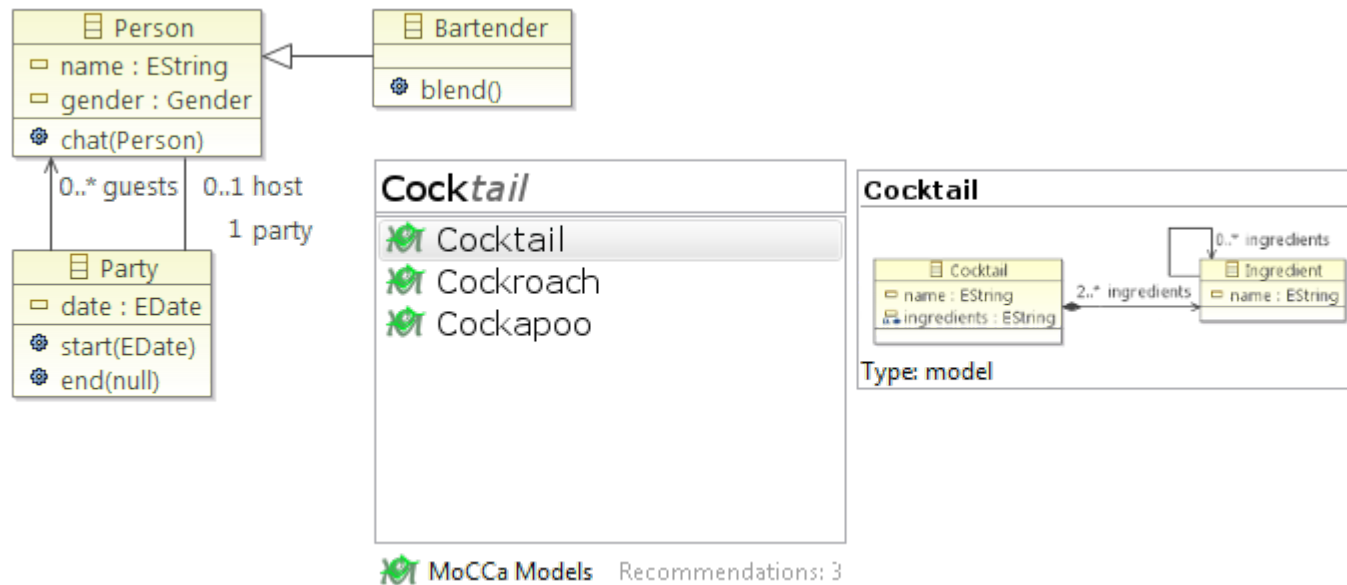
Model Recommenders for Command-Enabled Editors

Andrej Dyck, Andreas Ganser, and Horst Lichter

-  If You Take One Thing
-  Setting the Scene
-  One Vision ...
-  The Details
-  Status and Directions

If You Take One Thing ...

Model recommenders are fancy ;-)



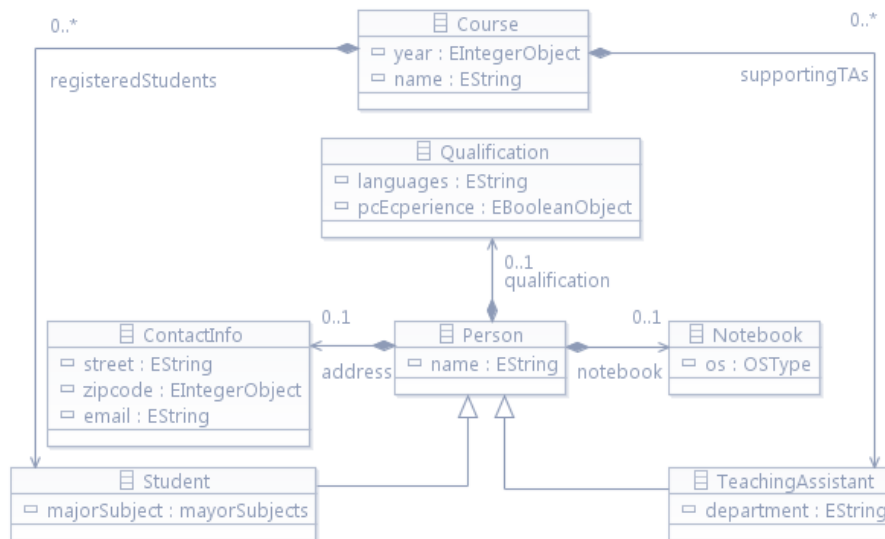
Why should I care ...?

Setting the Scene

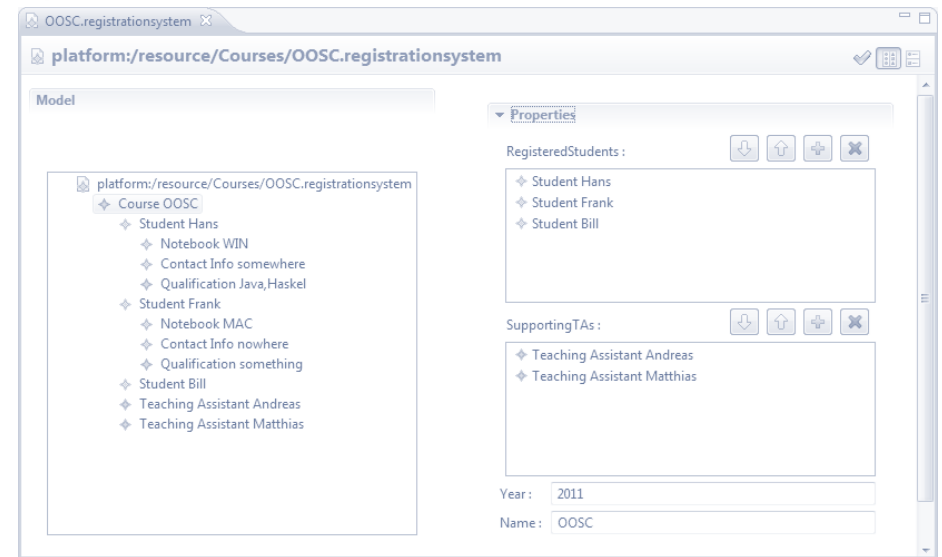
A lecture registration system (EMF/EEF)



Domain Model



Generated Editor (EEF)



How could we aid this ...?

One Vision ...

From content assist to recommender systems

Content Assist

Recommendation

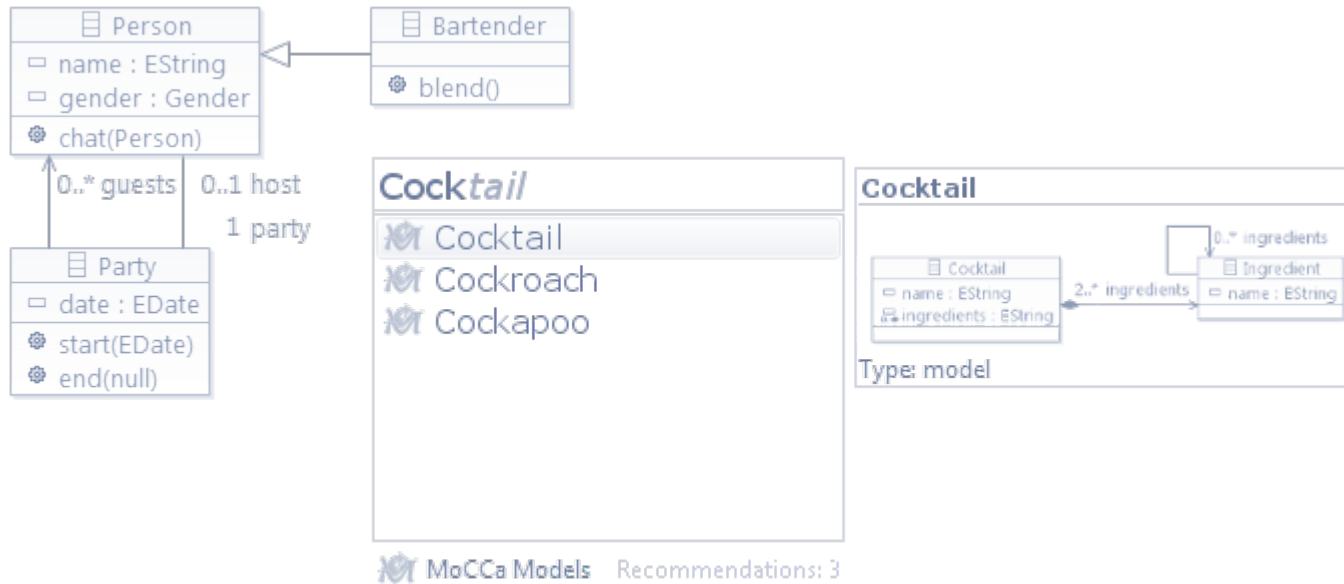
```

public static void main(String[] args) {
    public Text swtText = new Text(container, SWT.NONE);
    String helloAustria = "Hallo Austria";
    String helloEngland = "Hallo England";
    String helloGermany = "Hallo Germany";
    String helloGreece = "Hallo Greece";
    String helloSwitzerland = "Hallo Switzerland";
    System.out.println(helloAustria);
}
    
```

The screenshot shows a list of method suggestions for the `Text` widget, including:

- `setLayoutData(Object layoutData): void - Control - 75 %`
- `setText(String string): void - Text - 44 %`
- `addModifyListener(ModifyListener listener): void - Text - 34 %`
- `addListener(int eventType, Listener listener): void - Widget - 8 %`
- `getText(): String - Text - 7 %`
- `setEnabled(boolean enabled): void - Control - 7 %`
- `setFont(Font font): void - Text - 7 %`
- `handle: long - Control`
- `addControlListener(ControlListener listener): void - Control`
- `addDisposeListener(DisposeListener listener): void - Widget`
- `addDragDetectListener(DragDetectListener listener): void - Control`
- `hasEntry(K key, V value): Matcher<Map<K,V>> - org.hamcrest.Matcher`
- `hasEntry(Matcher<K> keyMatcher, Matcher<V> valueMatcher): Mat`
- `hasItem(Matcher<? extends T> elementMatcher): Matcher<Iterable<`
- `hasItem(T element): Matcher<Iterable<T>> - org.hamcrest.Matcher`

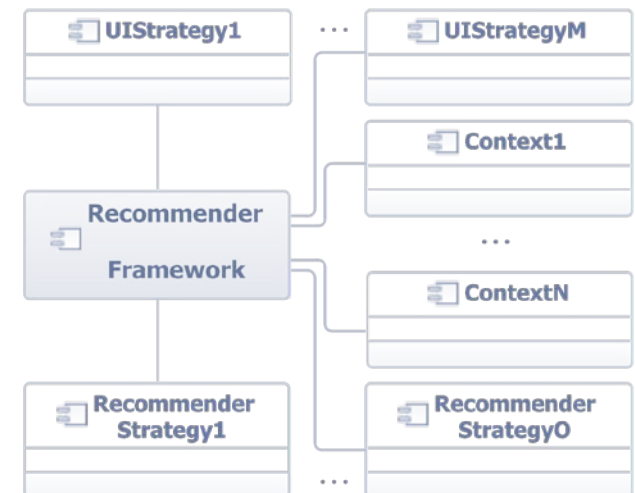
A large blue question mark icon is overlaid on the bottom left of the IDE screenshot, with the text "Isn't it a long way?" written below it.



Producing Recommendations

- a) Read query
- b) Analyze context
- c) Query / evaluate data sources
- d) Rank query results
- e) Produce recommendations

- Focus: Framework



How about an example ...?

An example

WordWeb Online

Dictionary and Thesaurus:

Noun: student *st(y)oo-d(u)nt*

1. A learner who is enrolled in an educational institution
- [pupil](#), [educatee](#)
2. A learned person (especially in the humanities); someone who by long study has gained mastery in one or more disciplines
- [scholar](#), [scholarly person](#), [bookman](#)

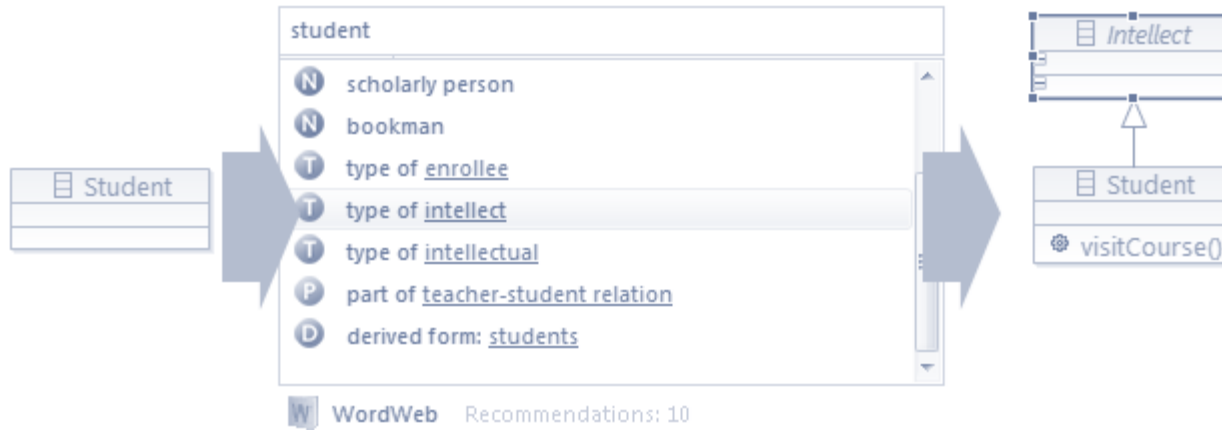
Derived forms: [students](#)

Type of: [enrollee](#), [intellect](#), [intellectual](#)

Part of: [teacher-student relation](#)

Encyclopedia: [Student](#)

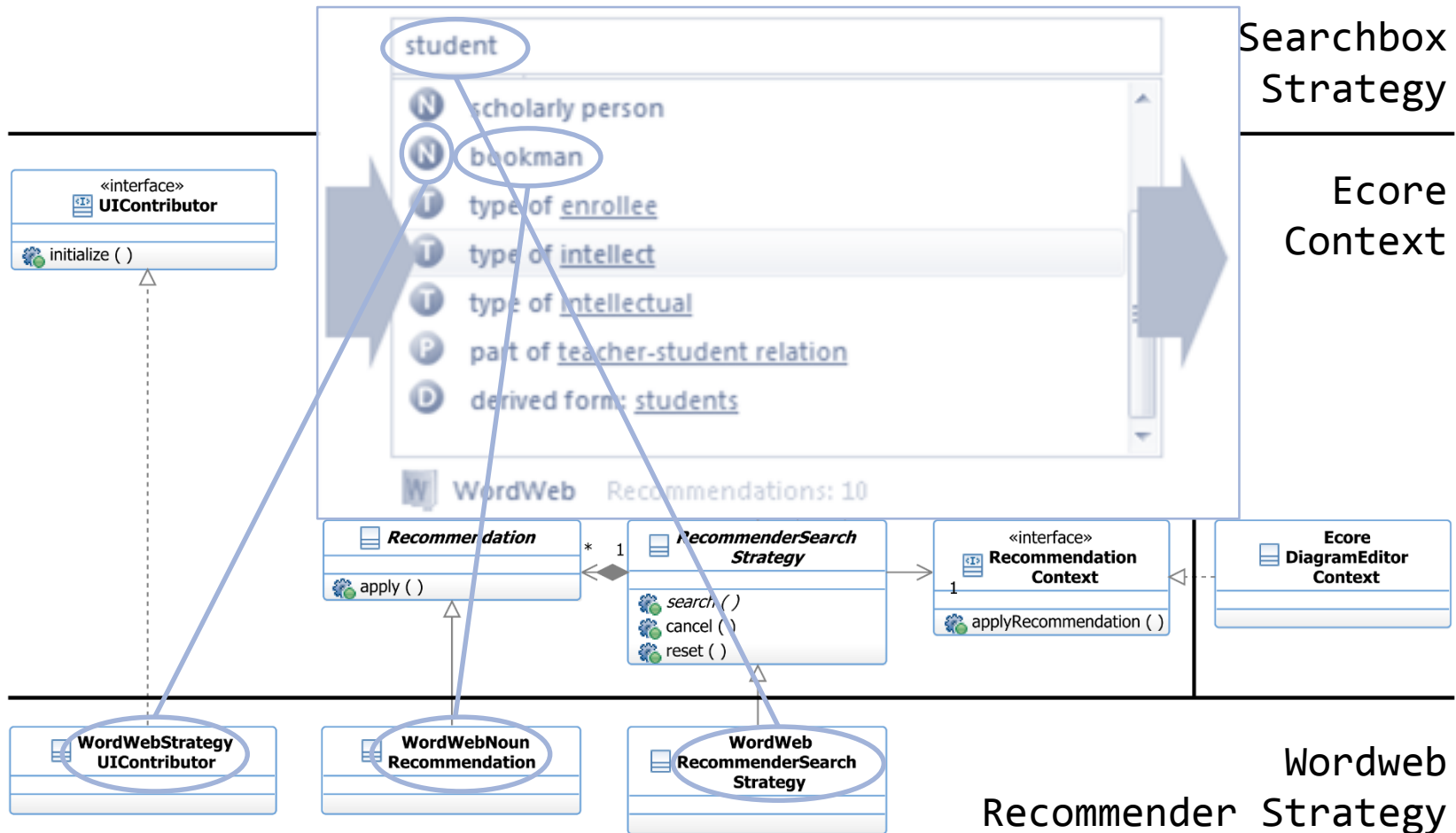
Copyright © WordWeb Software



Is it always that simple ...?

Recommender Framework

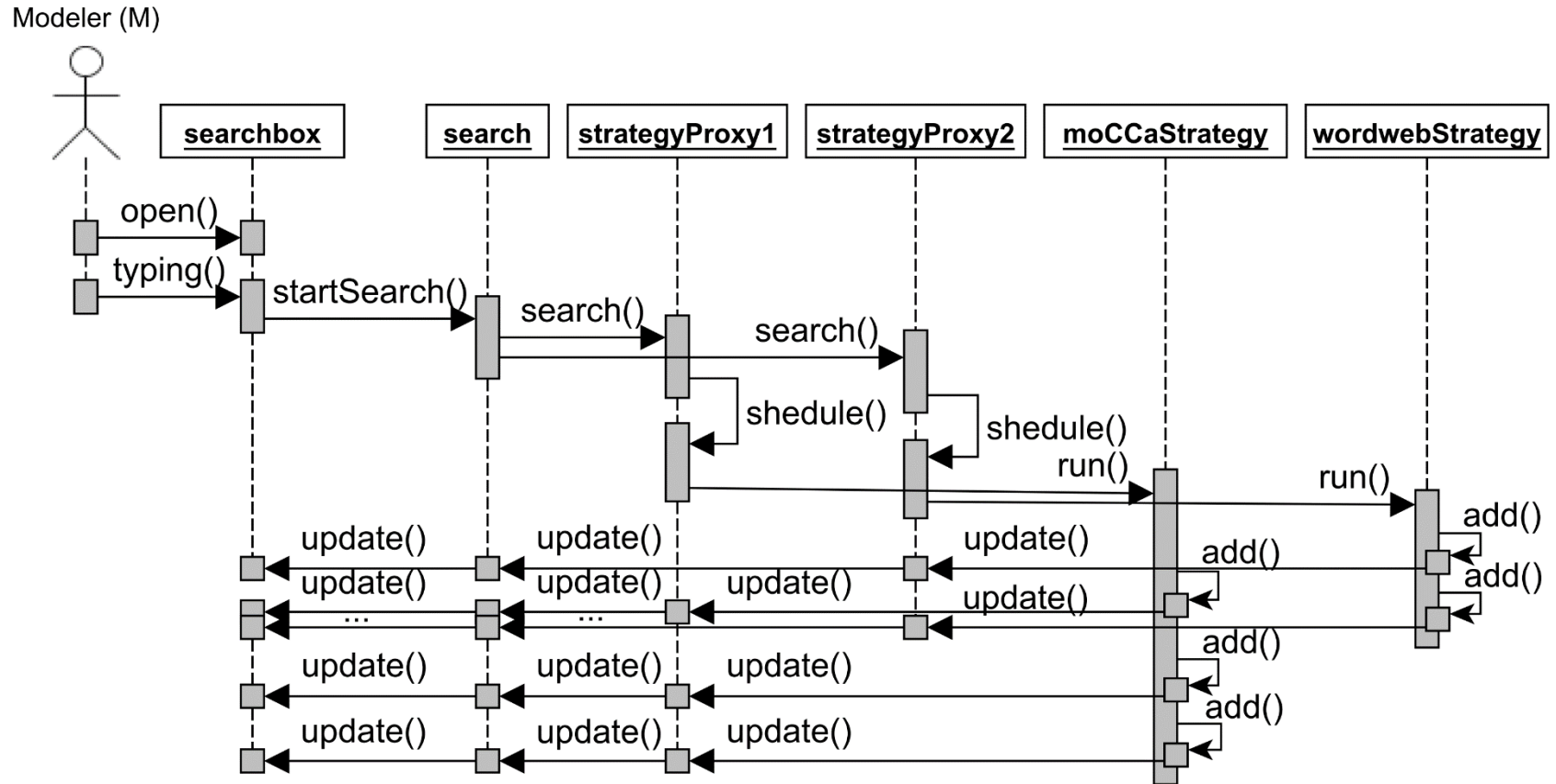
Some more “static” details, if you like ...



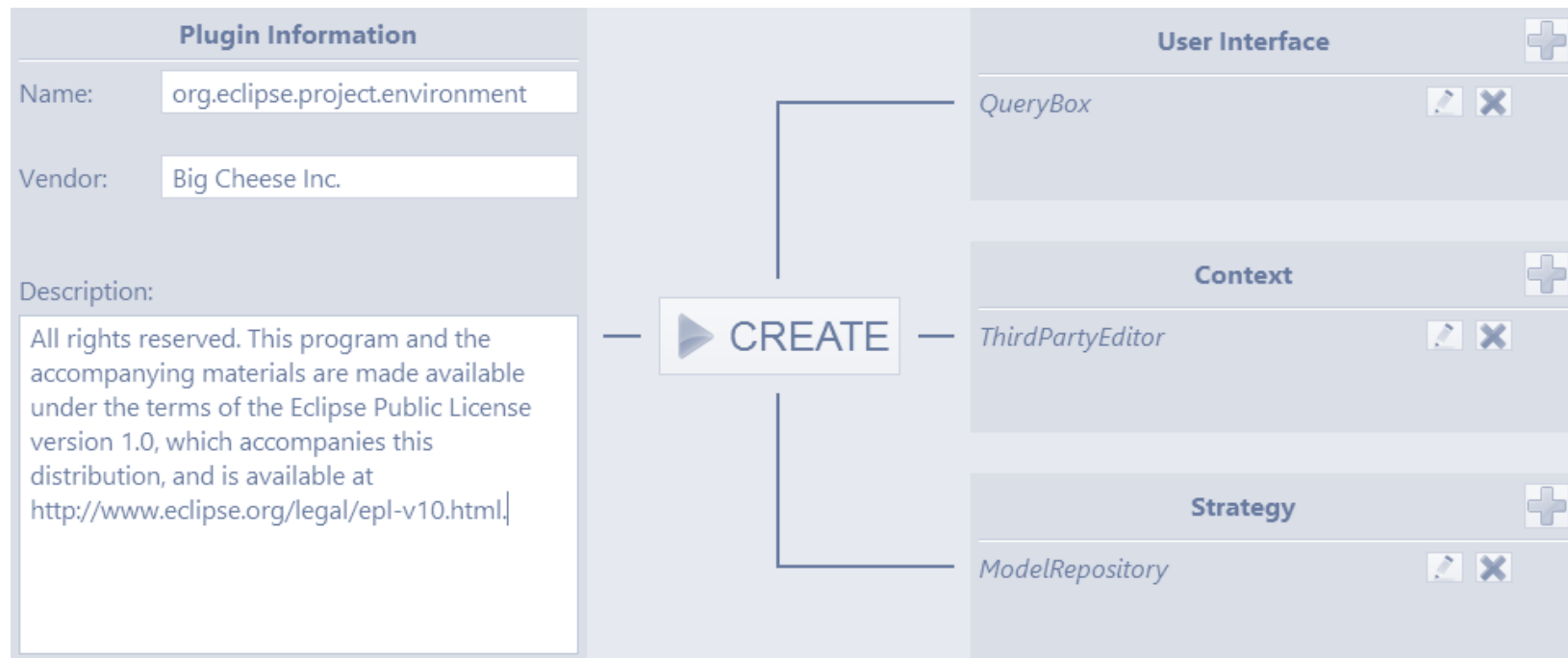
How do the objects flow ...?

Recommender Framework

Some more “dynamic” details, if you like ...



A dashboard to ease the pain ...



The dashboard is divided into several sections:

- Plugin Information:**
 - Name:
 - Vendor:
 - Description:
- Central Action:** A large button labeled "CREATE" with a play icon, connected to the other sections by lines.
- User Interface:** A panel containing a *QueryBox* component with edit and delete icons.
- Context:** A panel containing a *ThirdPartyEditor* component with edit and delete icons.
- Strategy:** A panel containing a *ModelRepository* component with edit and delete icons.

... a Software Prototype

The screenshot displays the MoCCA Evolution software interface. The main window shows a UML class diagram for a "Lecture Registration System". The diagram includes classes such as "Lecture", "Lecturer", "ResearchAssistant", "Student", "Natural Person", "ContactInfo", and "Capability". Relationships are shown with multiplicity and role names like "teacher", "teachingAssistants", and "participants".

On the right side, there is a "MoCCA Stage Monitor" panel. It features a circular diagram with three stages: "vague", "productive", and "elaborating", connected by arrows in a clockwise cycle. Below this diagram are sections for "Model Defects", "Model Smells", "Concerns", "Issues", and "Simple Reviews", each with a corresponding icon.

At the bottom, there is a "MoCCA Graph UI Search" panel. It includes a search input field, a "Type" dropdown menu with options like "infix", "whole words", "Lucene", and "First Exact Then Infix Search", and a search button.

Is that all ...?

Some References



What else is going on ...?

The HERMES Project

HERMES

HARVEST



EVOLVE



REUSE



MODELS EASILY AND SEAMLESSLY

References

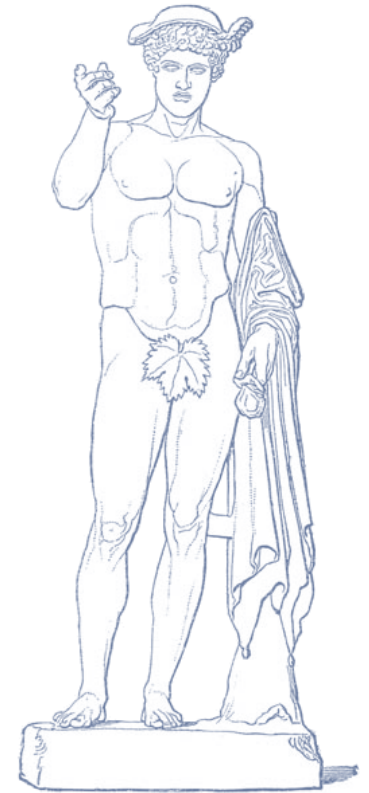
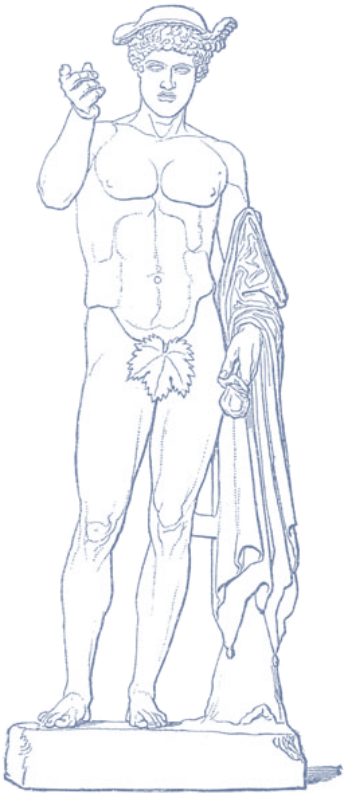
A. Ganser, H. Lichter, Engineering Model Recommender Foundations – From Class Completion to Model Recommendations, (Modelsward 2013, Spain)

A. Ganser, T. N. Viet, H. Lichter, Multi Back-Ends for a Model Library Abstraction Layer, (ICCSA 2013, Vietnam)

A. Dyck, A. Ganser, H. Lichter, Model Recommenders for Command-Enabled Editors, (MoDELS MDEBE 2013, US)

and more to come on

Model Recommender UI Survey,
Framework Internals, Contexts / Scanners



Thanks for your attention

... any questions?

A central collage of software technologies and frameworks. At the top left is the Java logo with a cartoon man holding a coffee cup. To its right is the Graphiti logo. Further right are LOGBack (The Generic, Reliable Fast & Flexible Logging Framework), git (with a red paw print), and trac (Integrated SCM & Project Management). Below these are sselab and Mylyn. In the middle left is HIBERNATE, Blueprints, Gremlin (with $G = (V, E)$), and Rexster. In the middle right is emf (ECLIPSE MODELING FRAMEWORK) and ZEST (SWT/Draw2D ZUI Library). At the bottom left is Neo4j (the graph database), Elasticsearch, and Apache Solr. At the bottom right are Xpand, Xtend, xtext, and Guice. A large purple sphere is in the center. A cartoon man in a suit is in the bottom left corner.