






Modeling Support

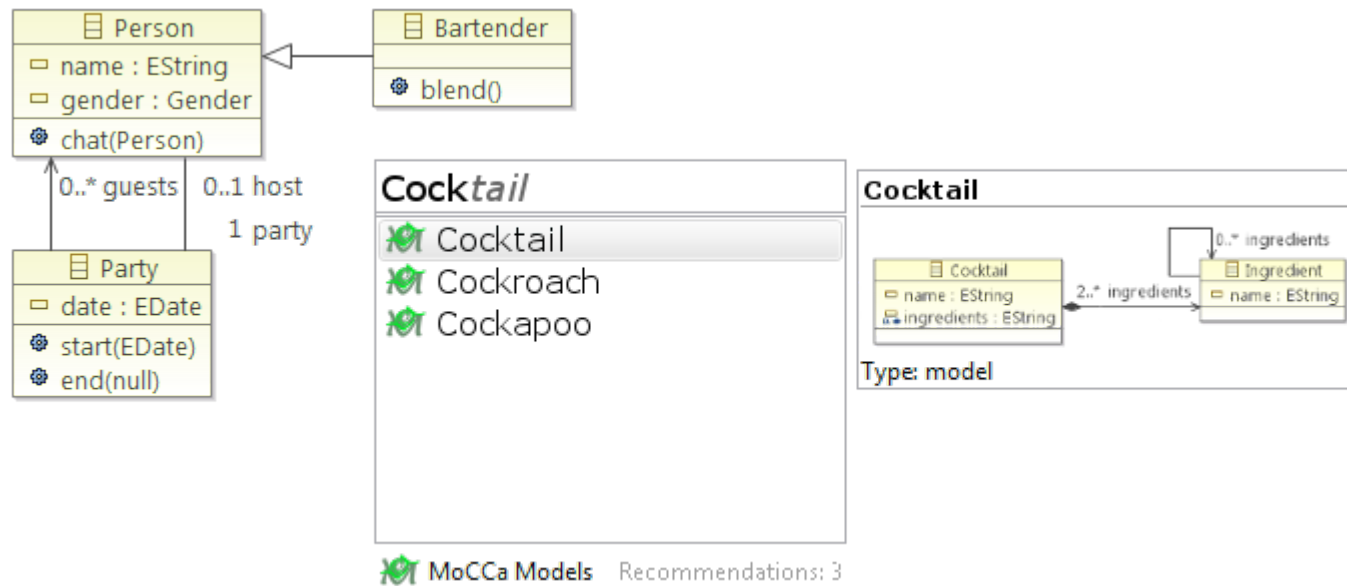
Recommenders Conquering Modeling

Andreas Ganser

-  If You Take One Thing
-  Setting the Scene
-  One Vision ...
-  Some Background
-  Status and Directions

If You Take One Thing ...

Model Recommender are Fancy ;-)



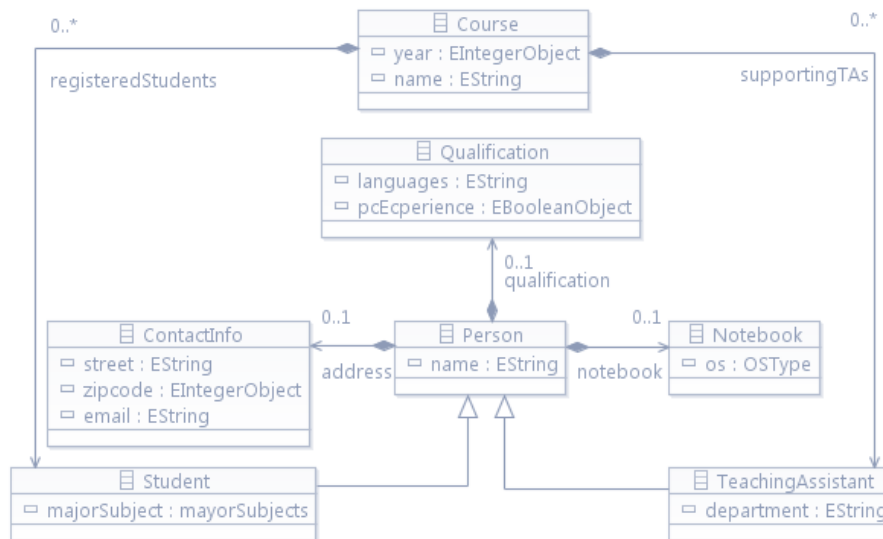
Why Should I Care ...?

Setting the Scene

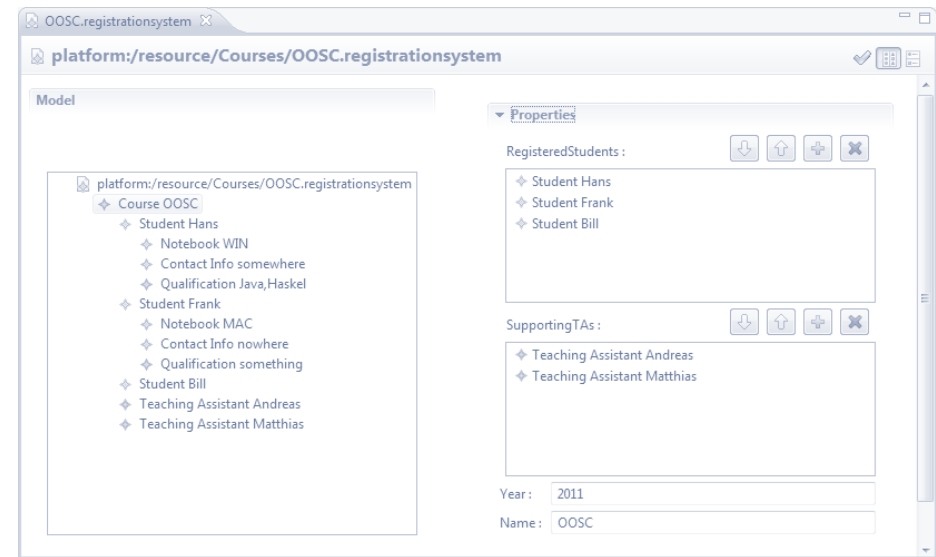
A Lecture Registration System (EMF/EEF)



Domain Model

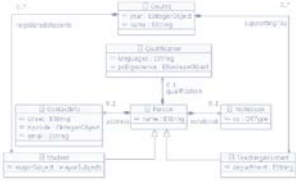


Generated Editor (EEF)



How Did This Work ...?
Which Were the Hard Tasks ...?

Domain Modeling and Programming



Domain Modeling

- Discussing Domain
- Understanding the Domain
 - Iterating Over Concepts
- Experimenting with Concepts
 - Generating Prototypes
- Always From Scratch???

Programming

- Which tasks are at hand?
 - Implementing Final Features
 - Integrating
 - Building
 - Testing
 - ...
- ... Not Covered Here



Why Not Reuse Models? How To Reuse Models?

From Content Assist to Recommender Systems

Content Assist

Recommendation

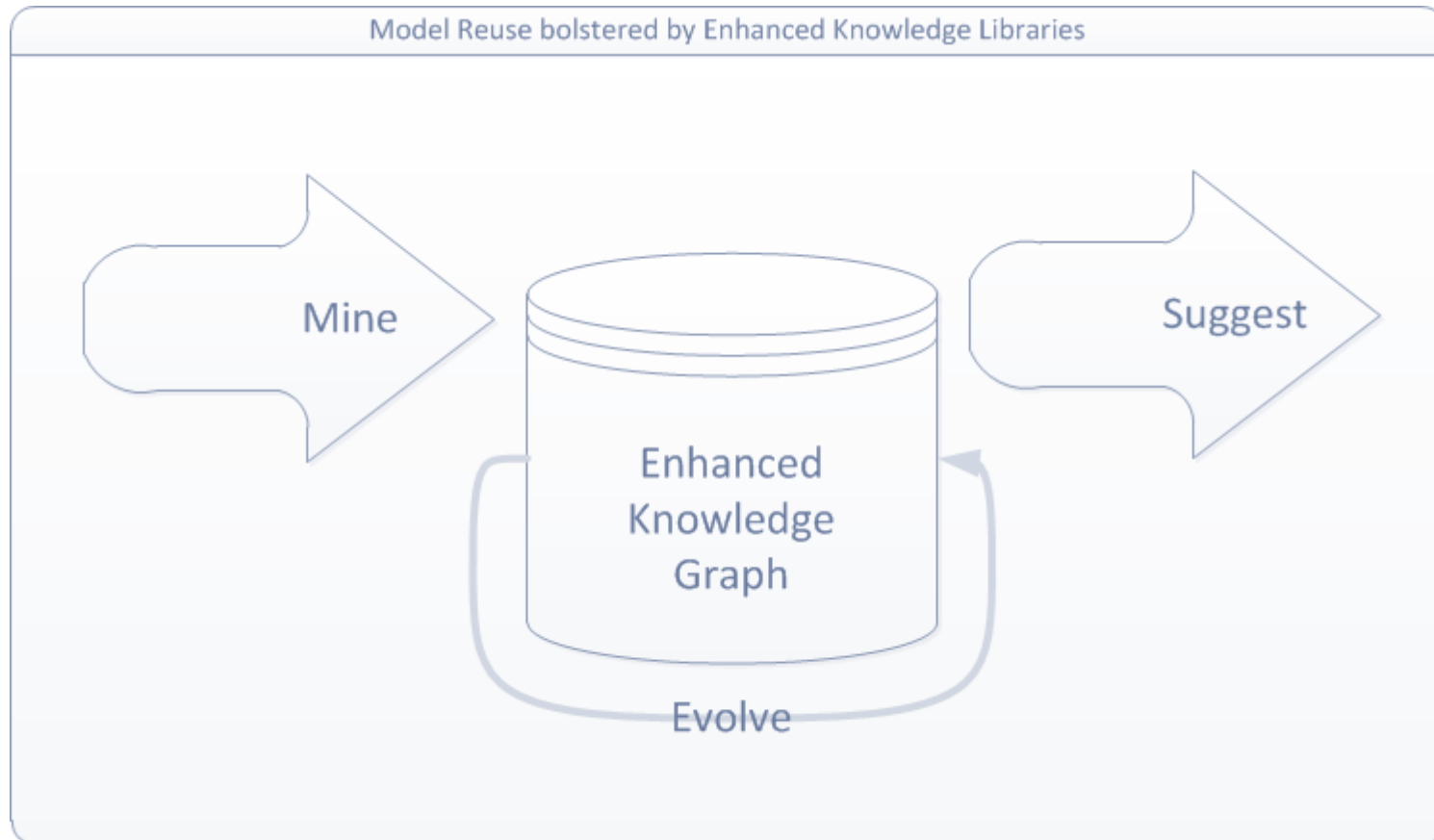
```

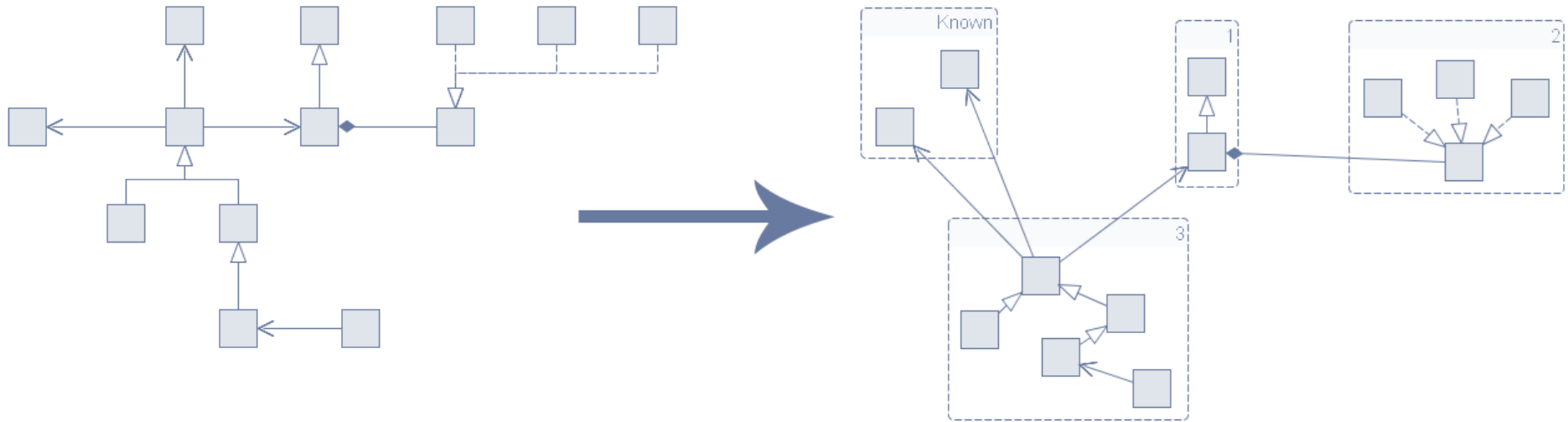
public static void main(String[] args) {
    public Text swtText = new Text(container, SWT.NONE);
    String helloAustria = "Halo Austria";
    String helloEngland = "Halo England";
    String helloGermany = "Halo Germany";
    String helloGreece = "Halo Greece";
    String helloSwitzerland = "Halo Switzerland";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

At the bottom of the image, a large blue question mark is overlaid on a silhouette of a person, with the text: **Isn't It a Long Way?**

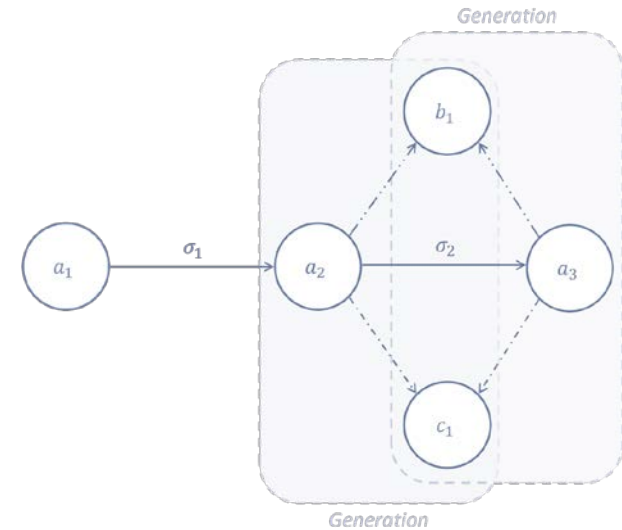
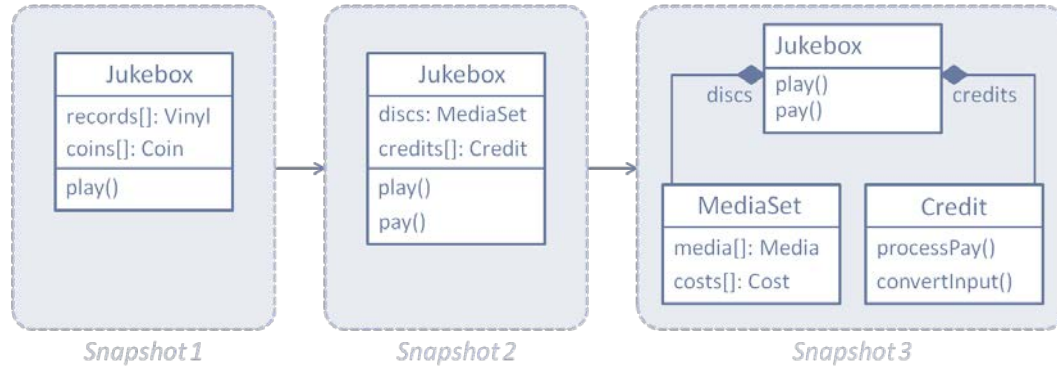




Steps of Mining Model Data

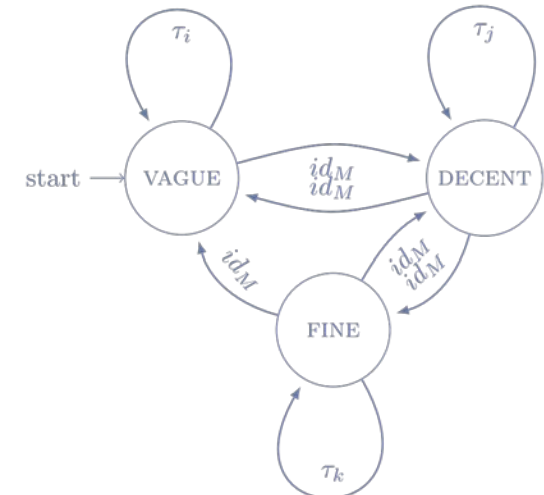
- Find Cliques
- Mark Known Parts
- Extract, Slice, and Split
- Store, Save, and Interlink

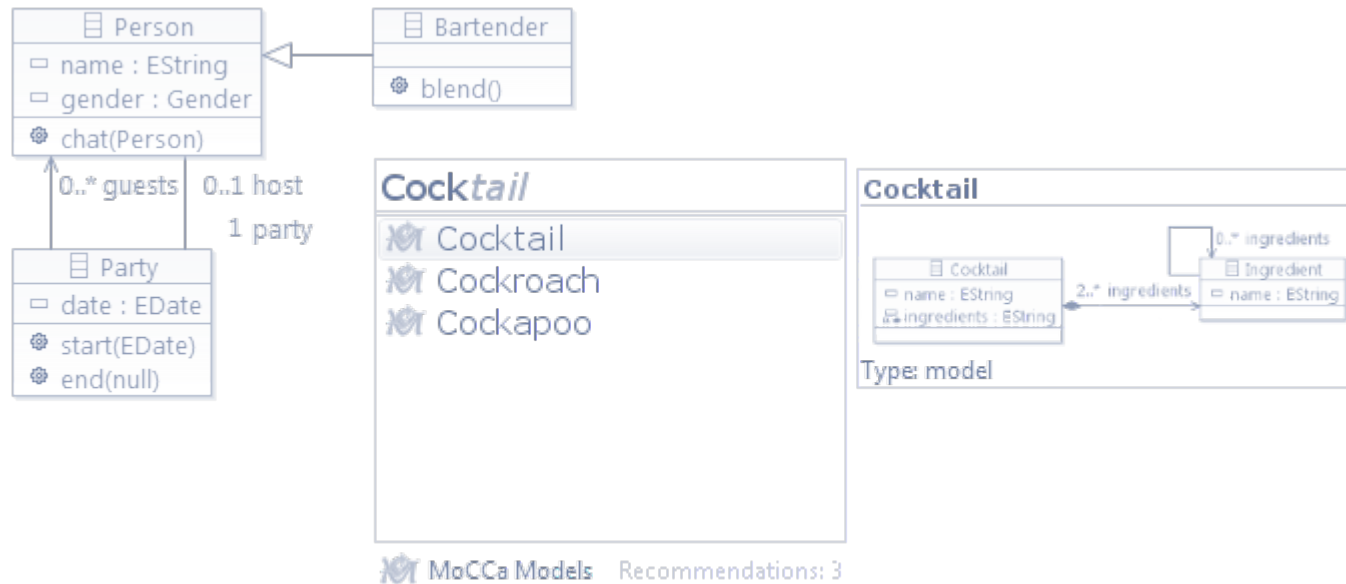
- Focus: Framework



Stages of Evolving Models

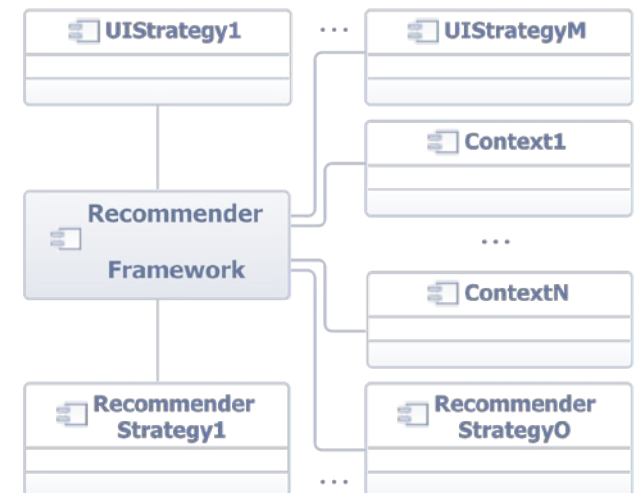
- Put Under Monitoring
- Review and Set Quality Gates
- Resolve Issues and Enhance
- Focus: Workflow





Producing Recommendations

- Take Query
- Analyze Context
- Evaluate Data Sources
- Rank Query Results
- Produce Recommendations
- Focus: Framework



Recommender Add-Ons

Support for Producing Recommendations

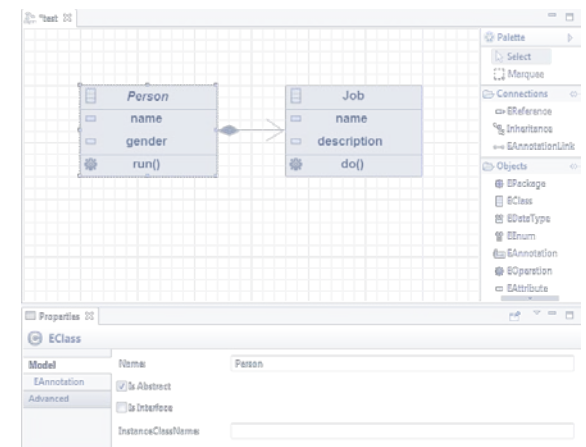
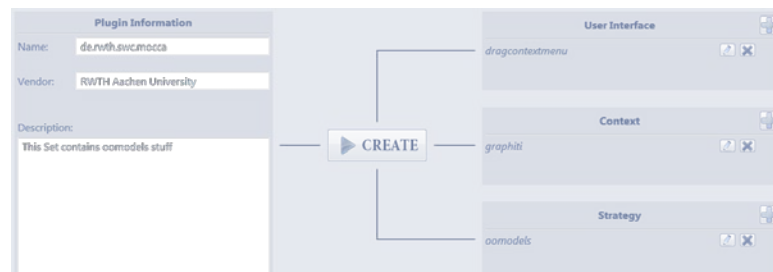


Data Support

- Graph / Relational Database
 - Self-Made and Mining Enhanced
- WordWeb
 - Dictionary Style and Simple Structure
- ReMoDD
 - Complete and Discussed Elements
- oomodels
 - Wiki-Style Documented Models

Tool Support

- Cockpit
 - See! How Easily To Create / Control
- Simulation
 - Check! If Your Strategy Works Right
- Editor
 - Experience! How Everything Integrates



Recommender Add-Ons

Support for Producing Recommendations



Conceptual Support

- Types of Model Recommenders
 - Content-Based
 - Collaborative
 - Hybrid
- Candidate Generation
 - To Match Categories & Queries
- Scoring Candidates
 - To Order Adhering Constraints
- Filtering Candidates
 - To fit (e.g.) Editing Position

Algorithmic Support

- Mining Model Data
 - Marking Known Parts
 - Graph and Data Mining
- Assessing Models (CD!)
 - Model Metrics
- Producing Recommendations
 - Evaluating Similarity Vector
 - Leverage Enhanced K-Grpah

... 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, there are sections for "Model Defects", "Model Smells", "Concerns", "Issues", and "Simple Reviews", each with a corresponding icon. An information icon at the bottom of the panel states: "To get more information on model evolution select a model in the MoCCA database."

At the bottom of the interface, there is a "MoCCA Graph UI Search" section with a search input field and radio buttons for search types: "infix" (selected), "whole words", "Lucene", and "First Exact Then Infix Search".







The Final Steps

- Glue It Together (Coding / Concept)
 - 2nd Quarter
- Evaluate (?)
 - 3rd Quarter
- Write, And Write, And Write ...
 - 2nd & 3rd & 4th Quarter

Publications

- Model Recommenders: A Research Infrastructure (Framework, Cockpit, and Simulation)
- Distributed Model Indexing supporting Model Recommenders
- Quality Staged Model Evolution for Model Recommenders
- Mining Recommendable/Recommenders-Appropriate Models for Libraries
- Model Recommender Building Blocks and Foundations
- Design and Evaluation of a Scalable Preliminary CS Block Course for Programming Novices at University

I had great support, but I'm on my own now ...



Thanks for your attention

... any questions?