



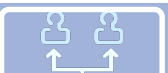


# Engineering Model Recommender Foundations

From Class Completion to Model Recommendations

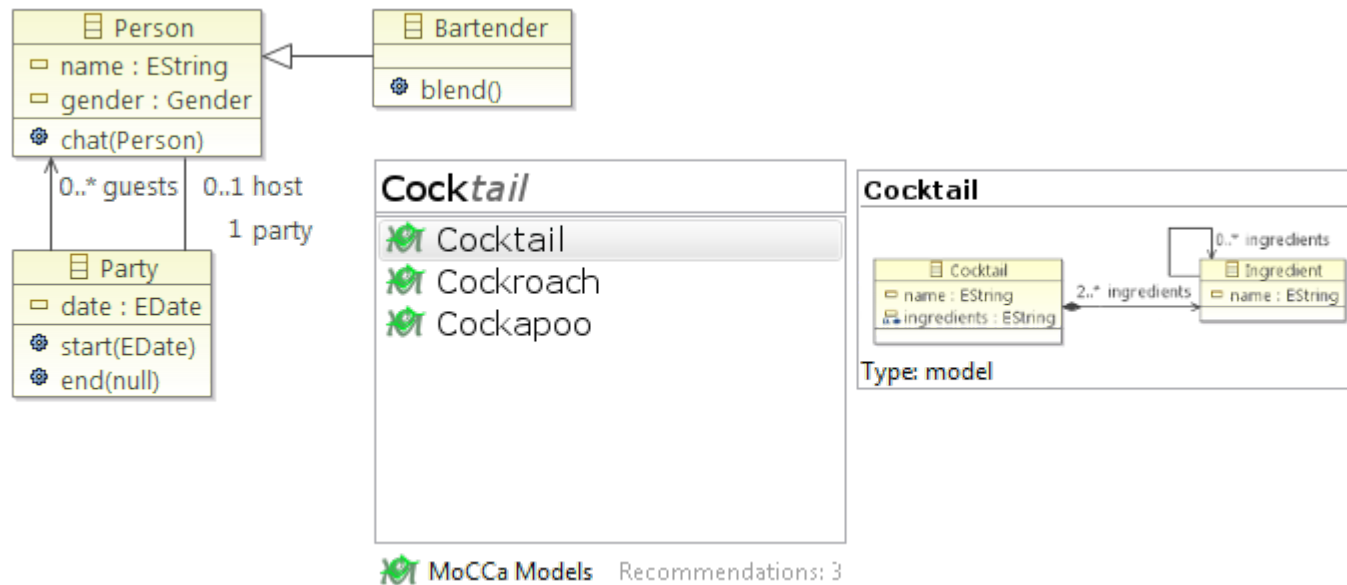
Andreas Ganser and Horst Lichter

RWTH Aachen University, Software Construction

-  If You Take One Thing
-  Setting the Scene
-  One Vision
-  Knowledge Libraries
-  Past, Present, Future

# If You Take One Thing ...

Model Recommender are fancy ;-)



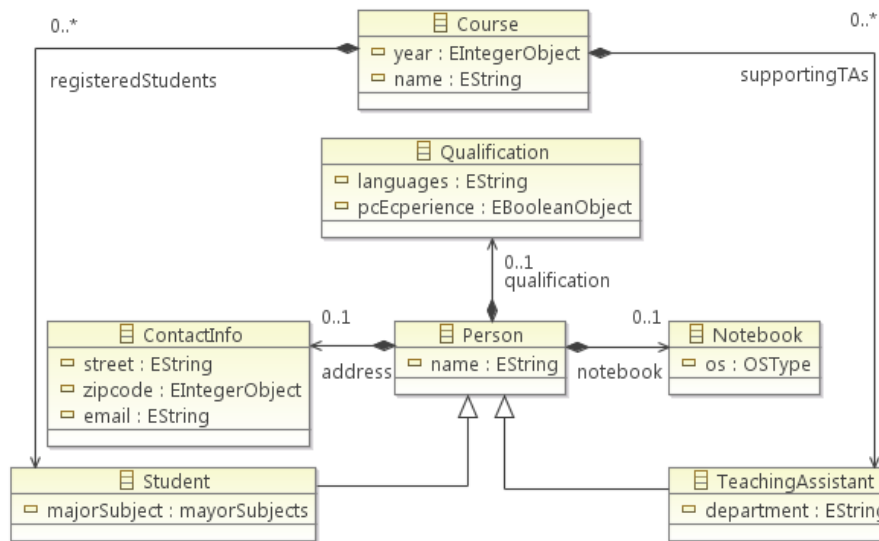
Why should I care ...?

# Example

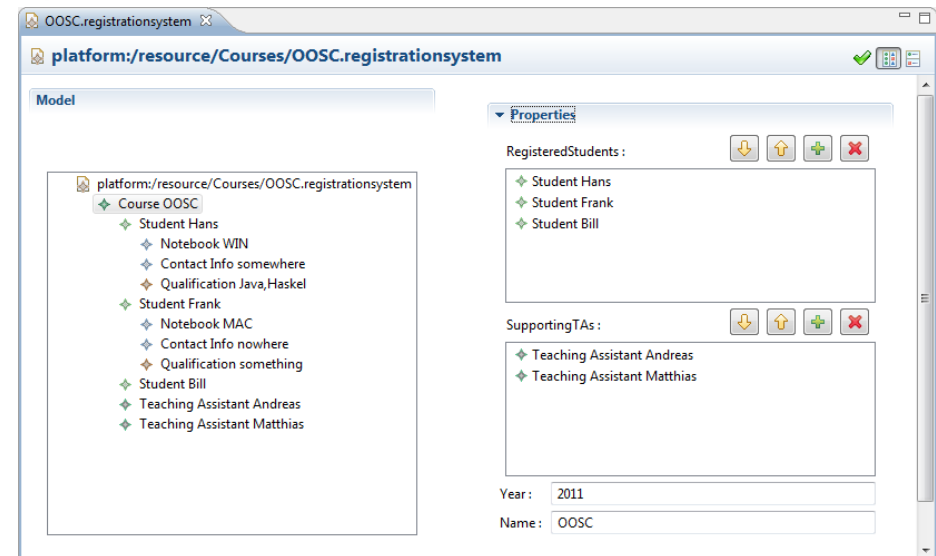
A Lecture Registration System (EMF/EEF)



## Domain Model

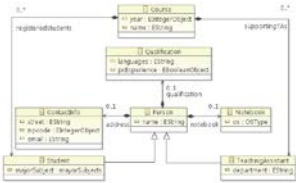


## Generated Editor (EEF)



How did this work ...?  
Which are 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
  - Testing
  - ...
- ... not Covered here



Why not reuse models? How to reuse models?

# Setting The Scene

From Content Assist to Recommender Systems

## Content Assist

## Recommendation

```

public static void main(String[] args) {
    public Text widget(String text, Text text, Container 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);
}
    
```

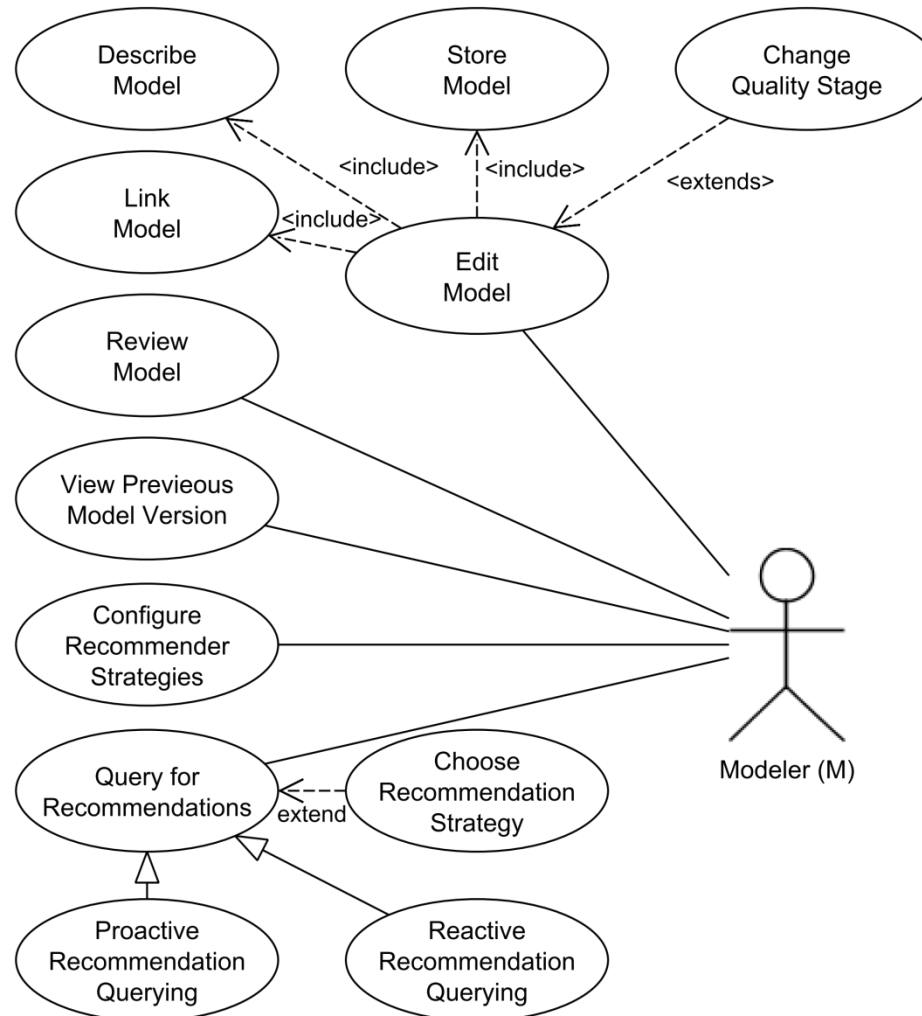
The screenshot shows an IDE with a code editor on the left and a list of method suggestions on the right. The code editor shows a Java class with a `main` method and a `Text` widget. The list of suggestions includes methods like `setLayoutData`, `setText`, `addListener`, `setEnabled`, `setFont`, `handle`, `addControlListener`, `addDisposeListener`, `addDragDetectListener`, `hasEntry`, `hasItem`, and `hasItem`.

... and how could that look like?

Content Assist, Completion, and Recommendations for Models

What are the actual use cases?

# The Use Cases

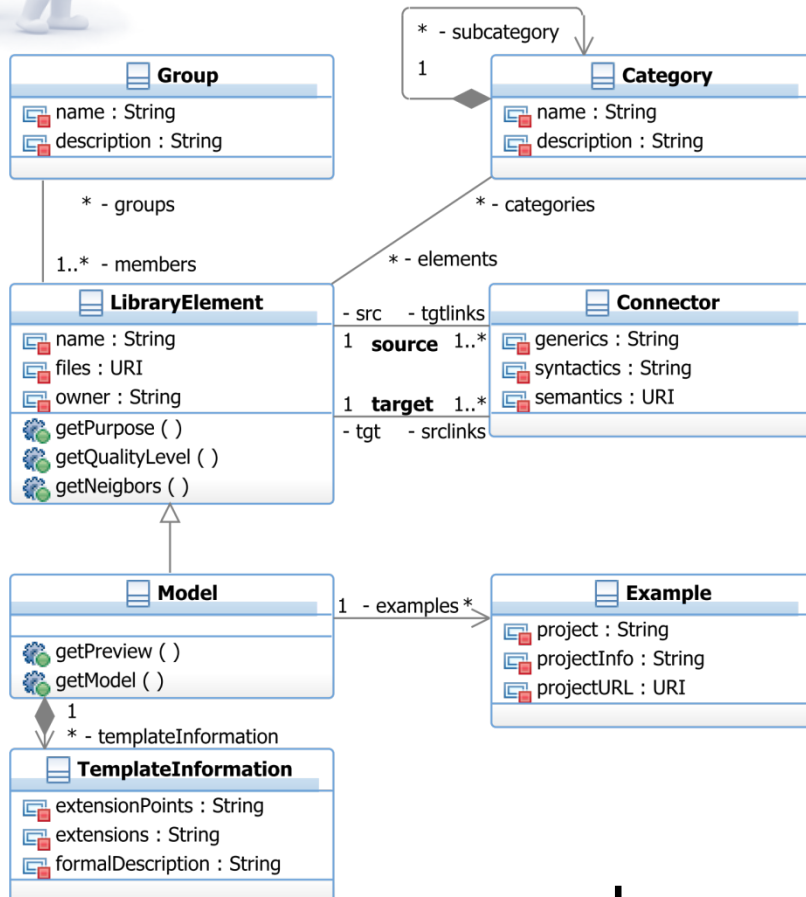


What do we need in terms of foundations?



What is the Data Structure for Suiting Recommenders best?

## Data Model



## Three Levels of Connections

- Generic
  - Description, no details
- Semantic
  - Design rationales, domain information
- Syntactic
  - Model information (UML relationships)

... so, how about recommenders?



# Model Recommendations

Multi and Hybrid-Recommender Systems

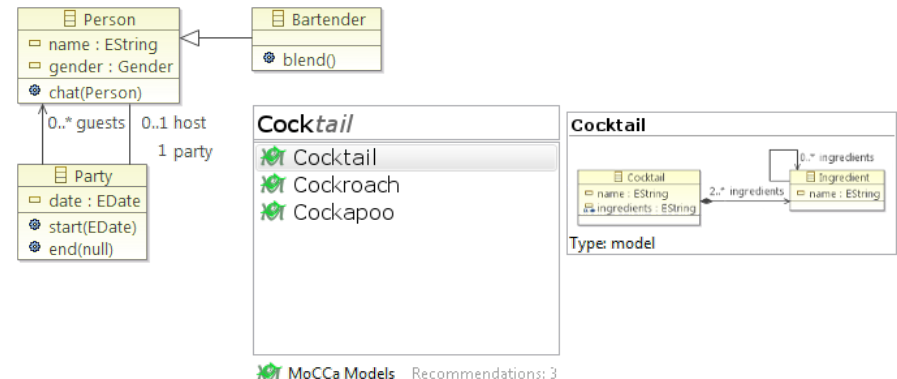


## Multi

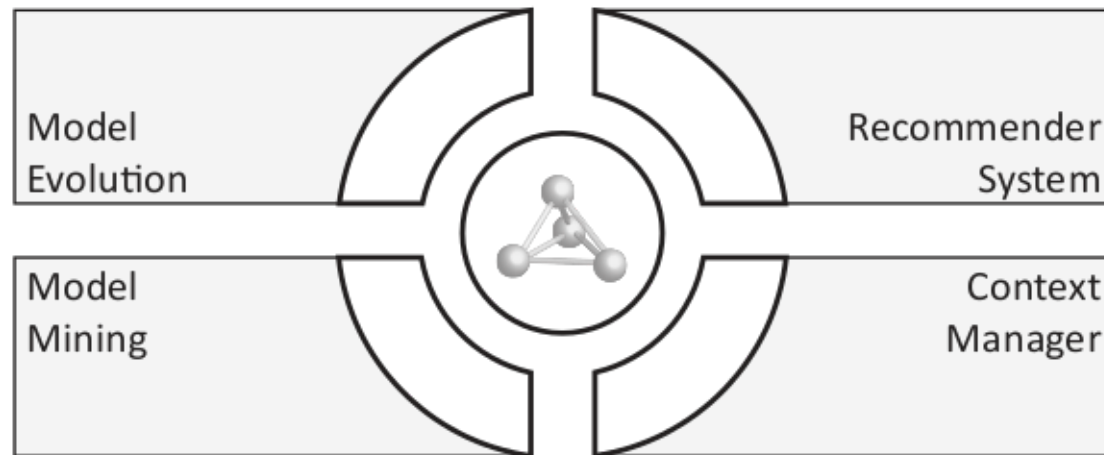
- Which are the data sources?
  - Requirements Specifications
  - Model Libraries
  - Web Services
- How to compose strategies?
  - Glossary + Synonyms
  - Glossary + Model Libraries
  - Glossary + Web Services
- Which misc. data to recommend?
  - Design Patterns
  - Domain Patterns

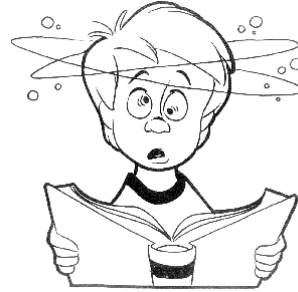
## Hybrid

- Content-Based Recommendations
  - Characteristics of items (cross-content)
- Collaborative Recommendations
  - Profiles disregarding item properties
- Chain Recommendations
  - Enhanced knowledge graph



Chain ... what!?!?





**Thanks for your attention**

... any questions?