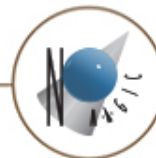


# Data Modeling for Business Analysts

Event Sponsors:



Media Sponsors:



# About Presenter



## Dr. Darius Šilingas

- ✓ *Head of Solutions Department @ No Magic Europe*
- ✓ *Product Manager, Cameo Business Modeler*
- ✓ *Expert in information system and business modeling, lead ~200 training/consulting sessions in 20 countries*
- ✓ *Chair of an annual conference Business Process Management in Practice in Lithuania*
- ✓ *Head of BPM studies at ISM Executive School*



*“... in many cases the BA becomes the DBA but I think that is mainly because the DB becomes the skeleton that drives the entire system. In addition to learning Relational Databases I would also suggest learning Modelling tools which will give you a picture of what the relationships are and how they all fit together. There are some great modelling tools out there ... ”*

Alan Radau

# Agenda

Data Modeling Concepts

Data Modeling Techniques

Using Data Models

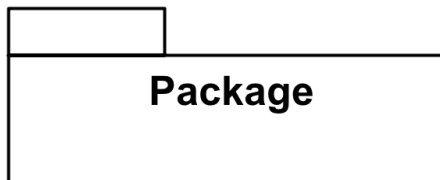
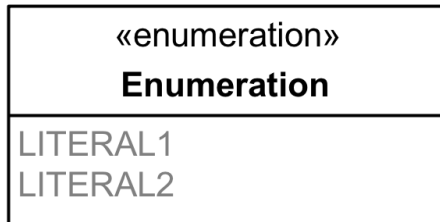
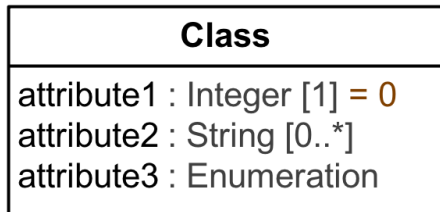
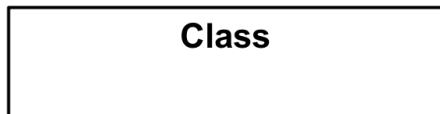
# UML and Data Modeling



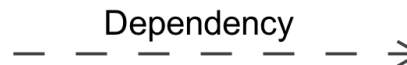
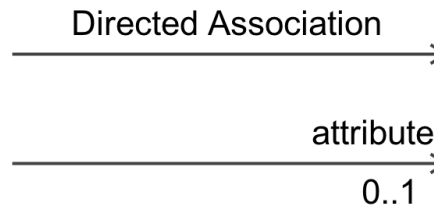
- ✓ UML is *de facto* standard for software modeling
- ✓ UML is complex: 248 metaclasses, 15 diagram types, ...
- ✓ 80% of problems can be solved with 20% of UML
- ✓ Data modeling capabilities in UML are very mature and have been widely used in practice

# UML Concepts for Data Modeling

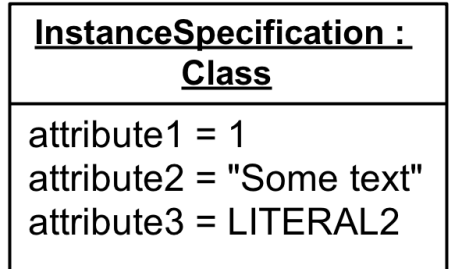
## Data Structures



## Data Relations



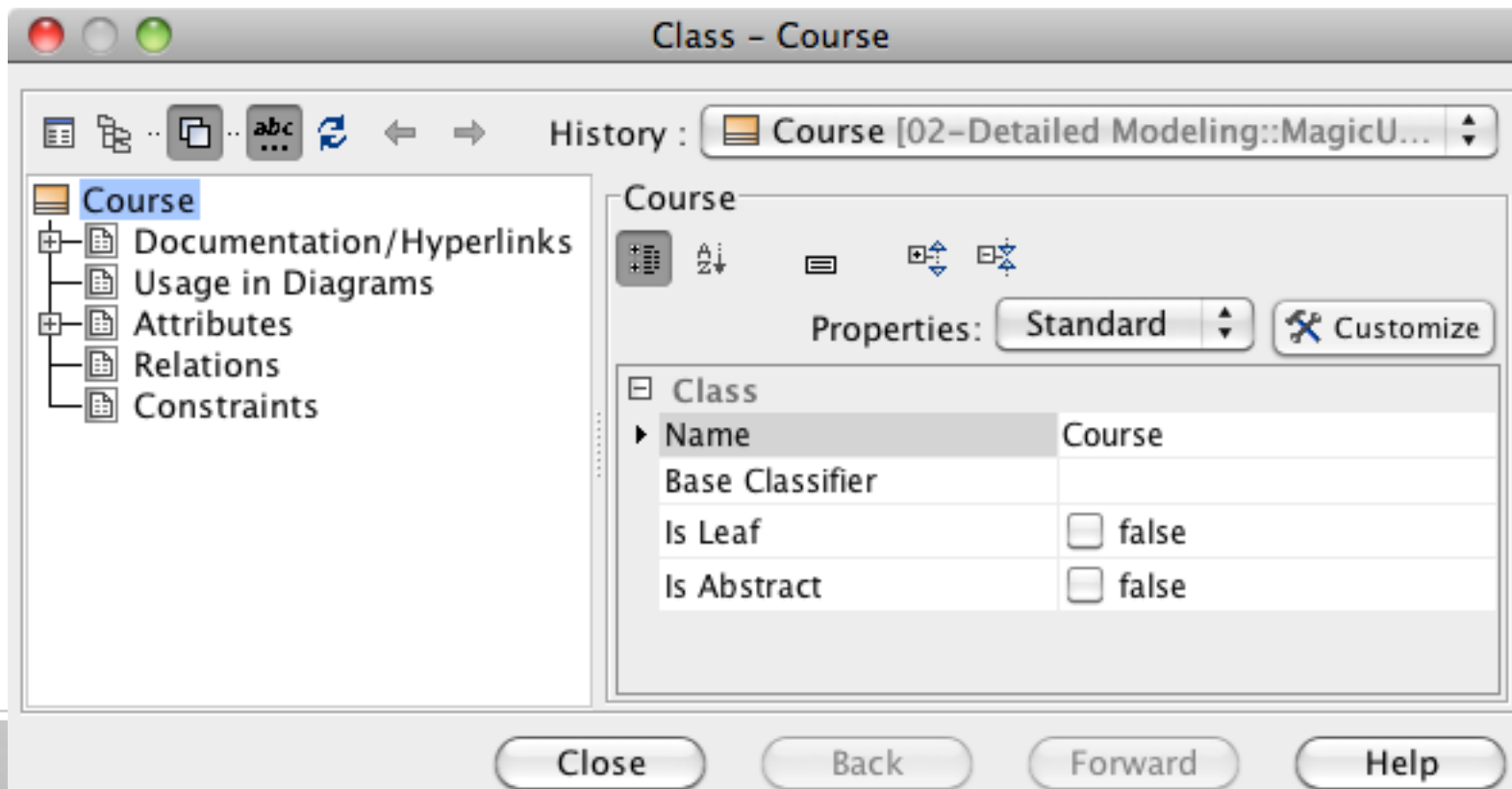
## Data Samples



# Class

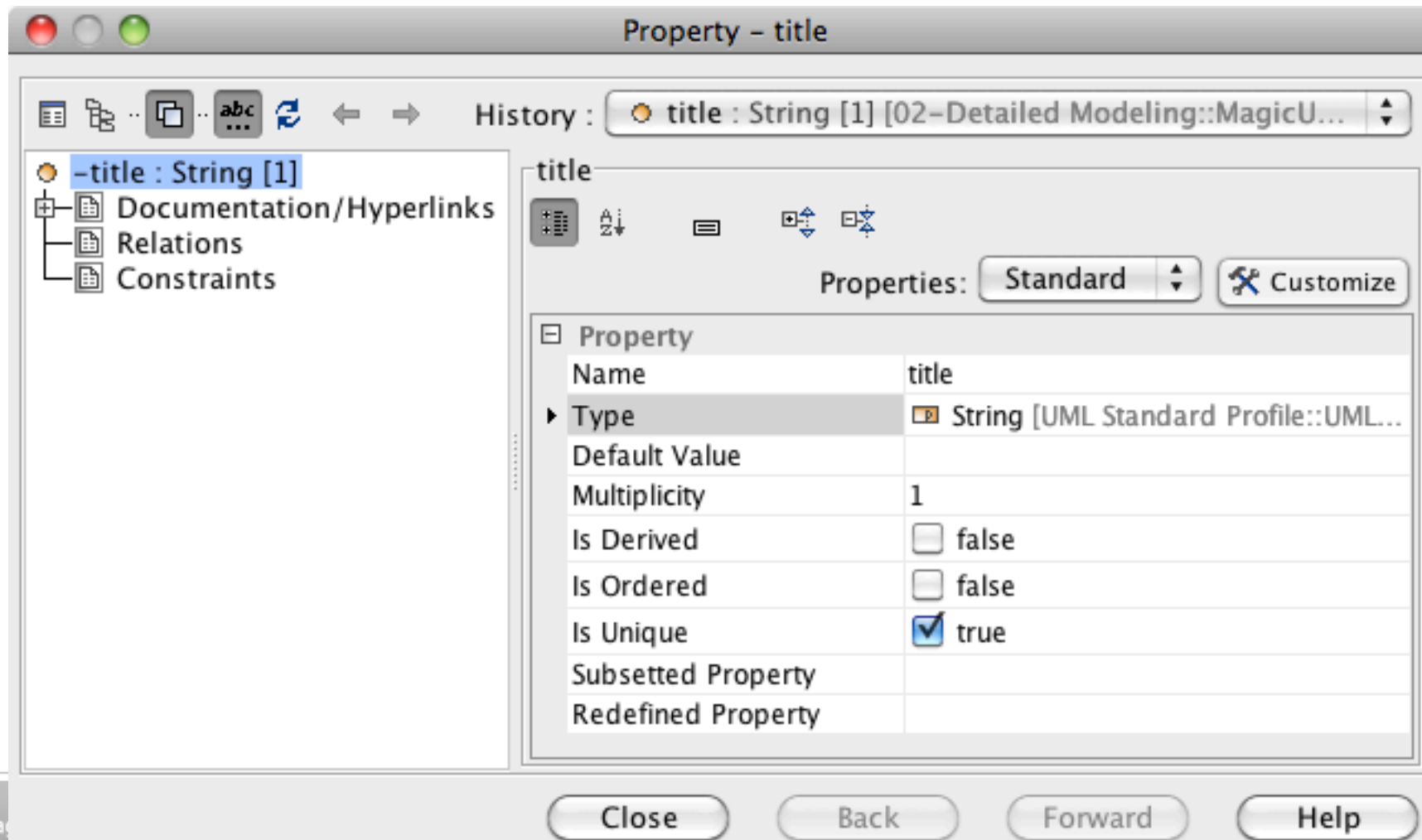
- **Class** provides categorization of objects according to their features\*.

\* *In data modeling, we consider only properties - structural features.*



# Property

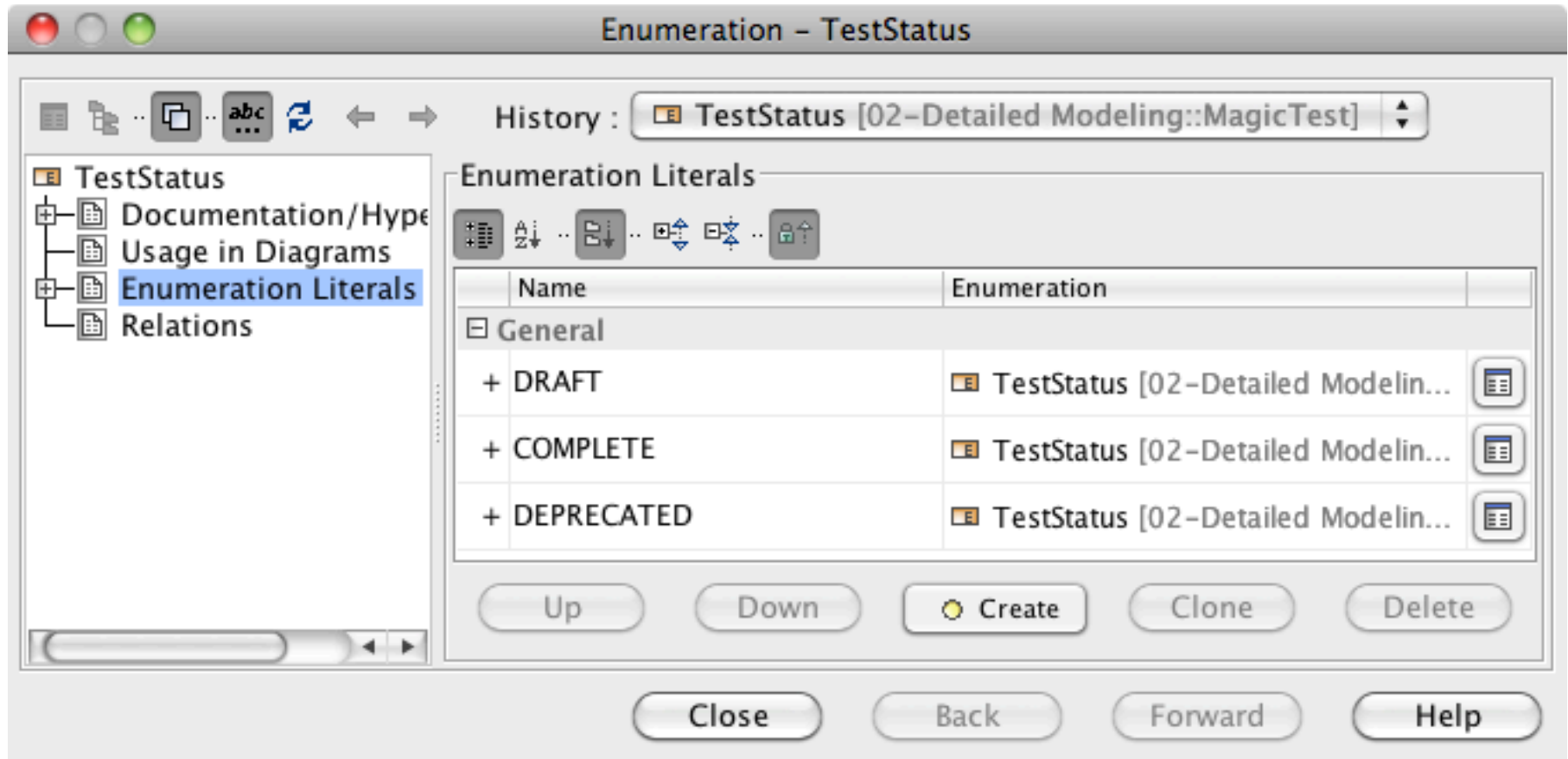
- **Property** is a structural feature that may represent an attribute of a **Class** or and end of **Association**.





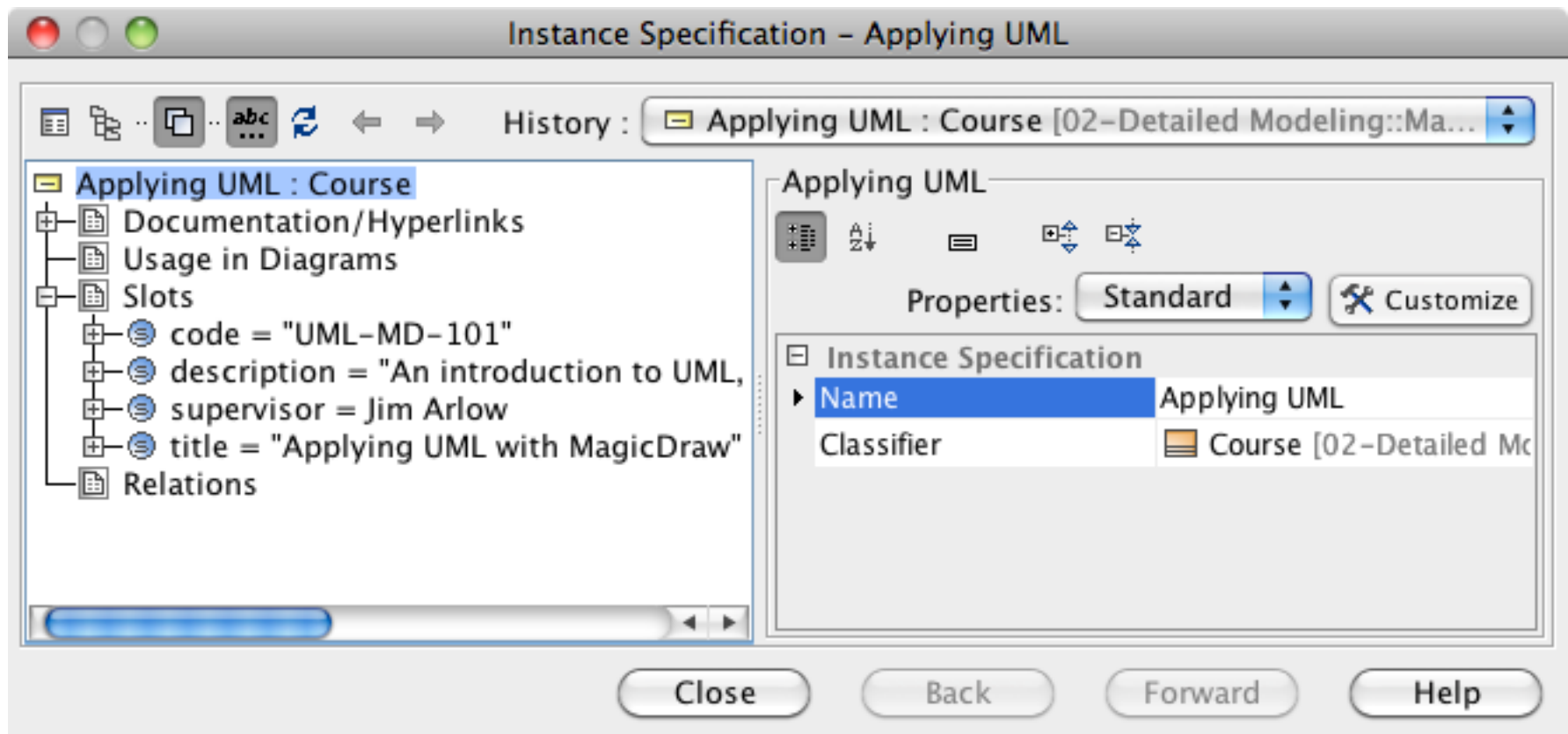
# Enumeration

- **Enumeration** is a data type whose potential values are enumerated as literals.



# Instance Specification

- **Instance Specification** represents the existing object in a modeled system as a snapshot in time with slots representing concrete values for object's attributes.



# Association

- **Association** specifies a semantic relationship that can occur between **two** typed instances.

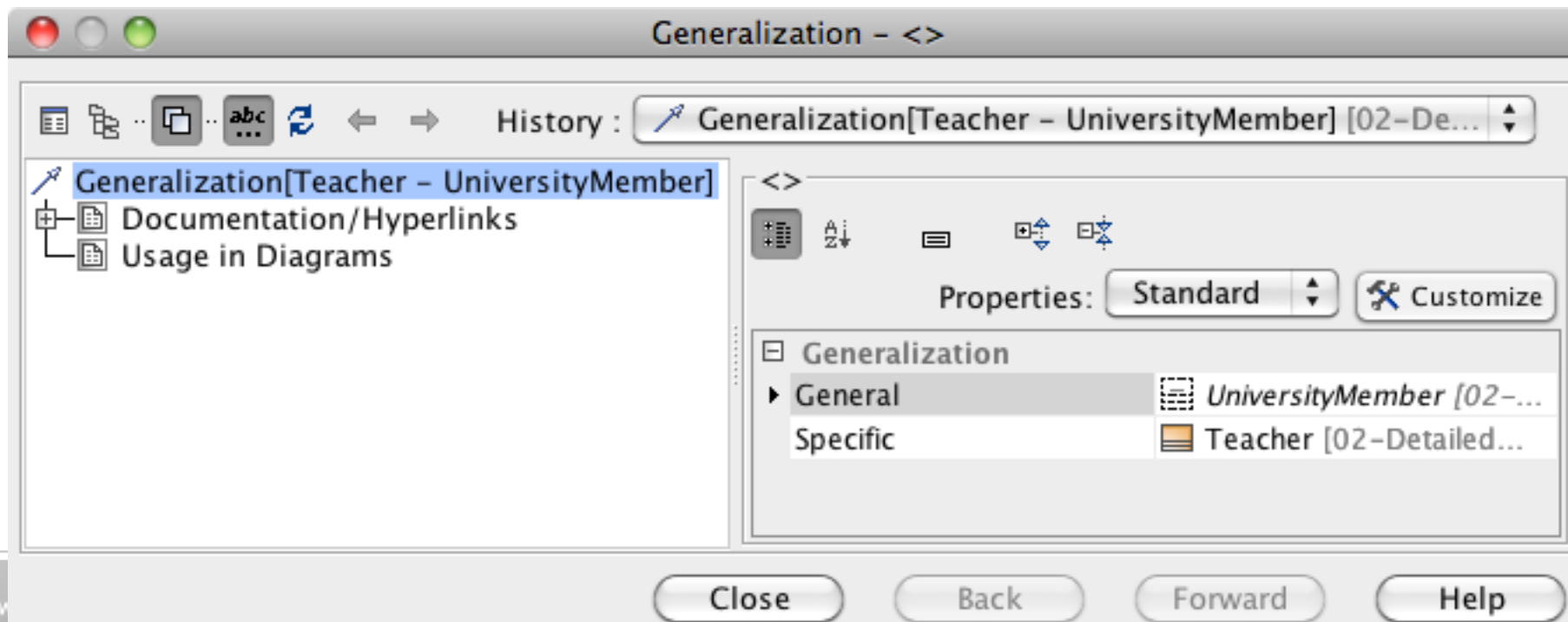
The screenshot shows the 'Association' editor window in MagicDraw. The title bar reads 'Association - <>'. The left sidebar contains a tree view with the following items: 'Association[Course - supervisor:Teacher]', 'Documentation/Hyperlinks', 'Usage in Diagrams', 'Roles', 'Conveyed Information', and 'Item Flow'. The main editor area is divided into two panes. The top pane shows the 'Properties' tab with a dropdown set to 'Standard' and a 'Customize' button. The bottom pane displays the 'Association' properties table.

Association	
Name	
Role of Course	
Name	
Navigable	<input type="checkbox"/> false
Owned By	Association[Course - supervisor:Teacher]...
Multiplicity	0..*
Type	Course [02-Detailed Modeling::MagicUni...
Role of Teacher (supervisor)	
Name	supervisor
Navigable	<input checked="" type="checkbox"/> true
Owned By	Course [02-Detailed Modeling::MagicUni...
Multiplicity	1
Type	Teacher [02-Detailed Modeling::MagicUn...

At the bottom of the editor, there is a search bar with the placeholder text 'Type here to filter properties'. Below the editor, there are four buttons: 'Close', 'Back', 'Forward', and 'Help'.

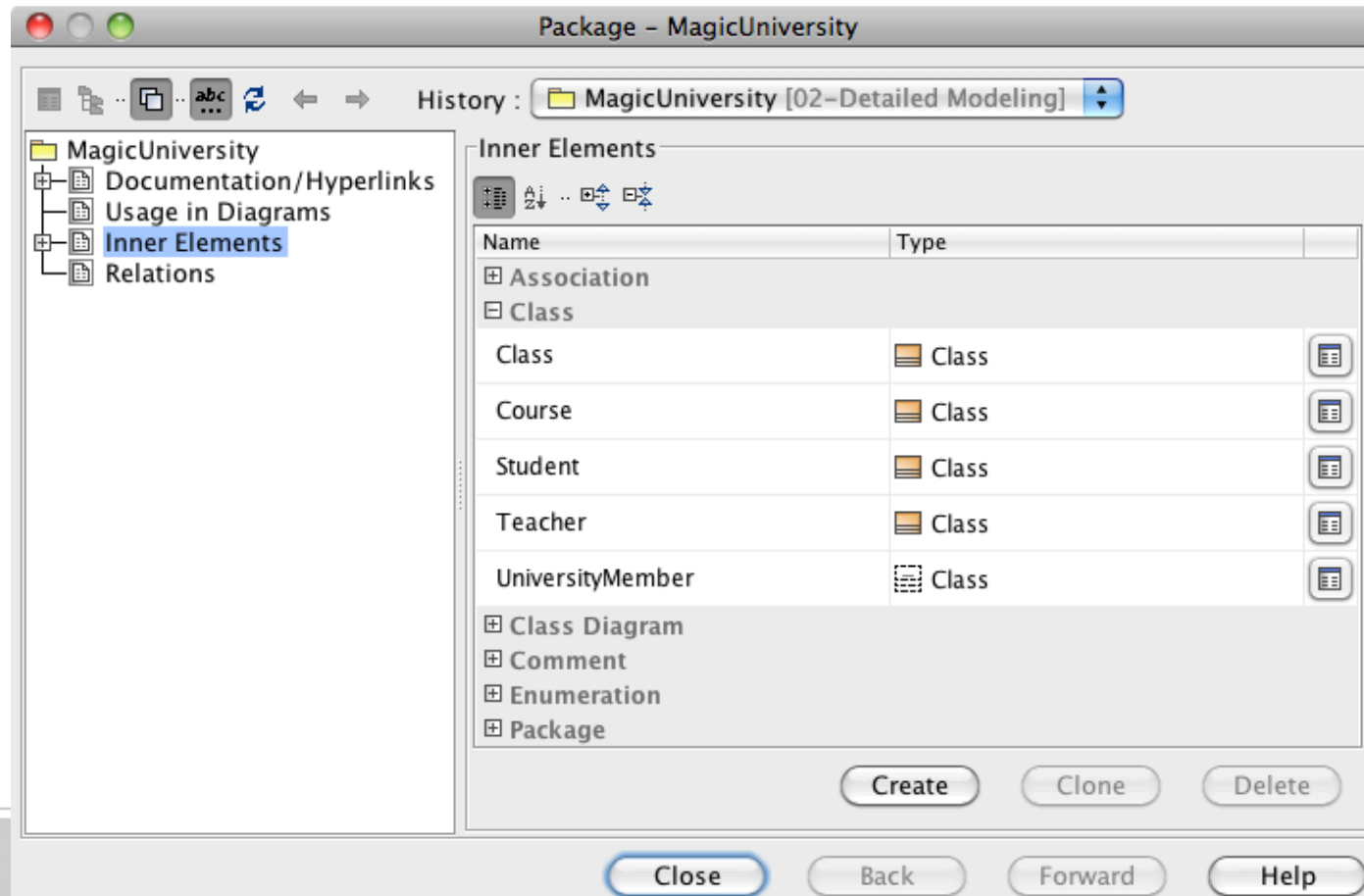
# Generalization

- **Generalization** define generalization/specialization relationships between classes.
- *When a class is generalized, certain members of its generalizations are inherited. An instance of a class is also an (indirect) instance of each of its generalizations. Any constraints applying to instances of the generalizations also apply to instances of the class.*



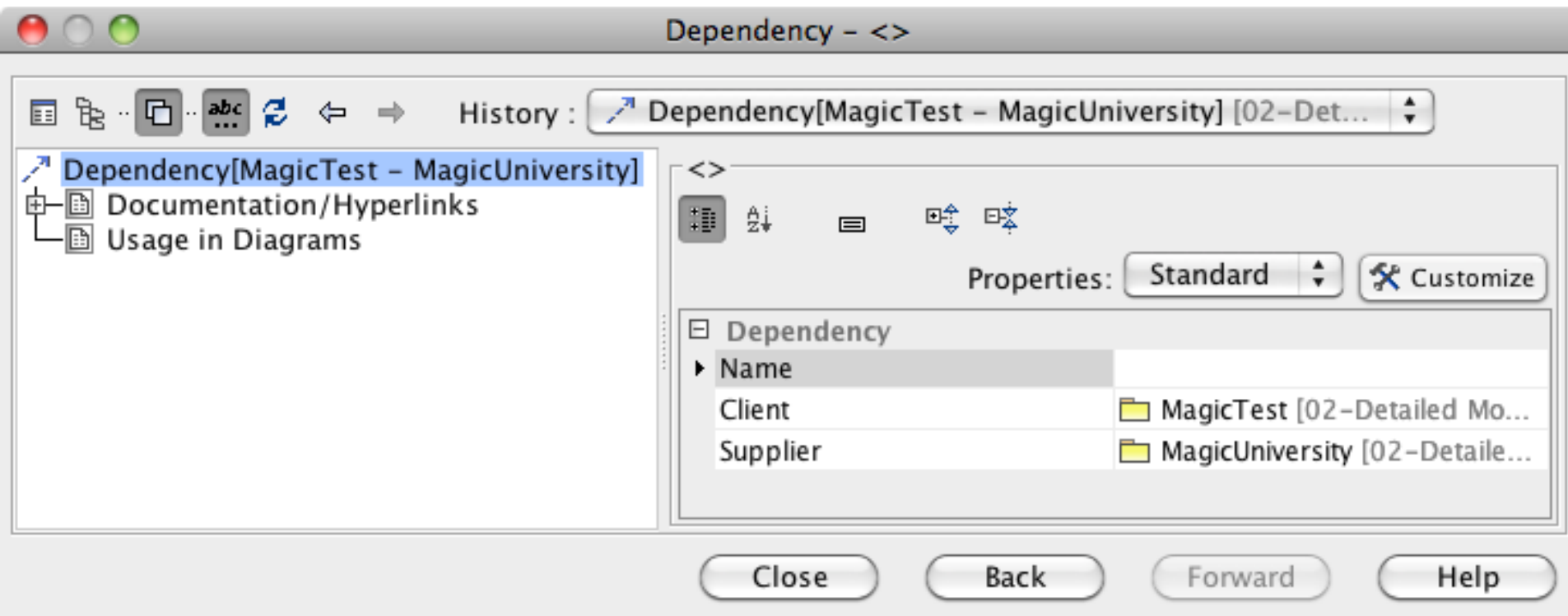
# Package

- **Package** owns other model elements and provides a namespace for them.
- *Packages serve as a means for structuring large models, similar to folders in a file system.*



# Dependency

- **Dependency** represents a supplier/client relationship between model elements where the modification of a supplier may impact the client model elements.



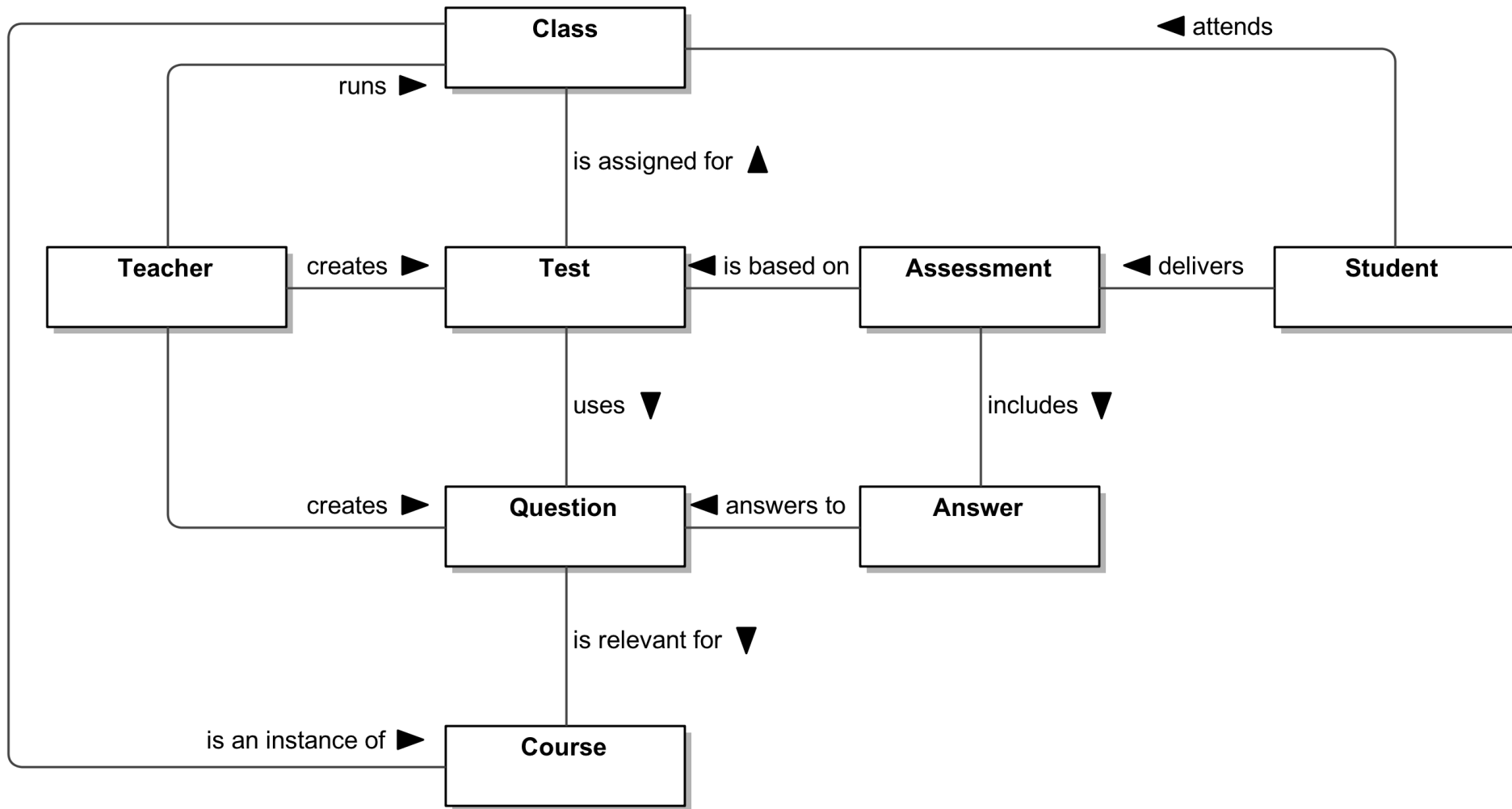
# Agenda

Data Modeling Concepts

**Data Modeling Techniques**

Using Data Models

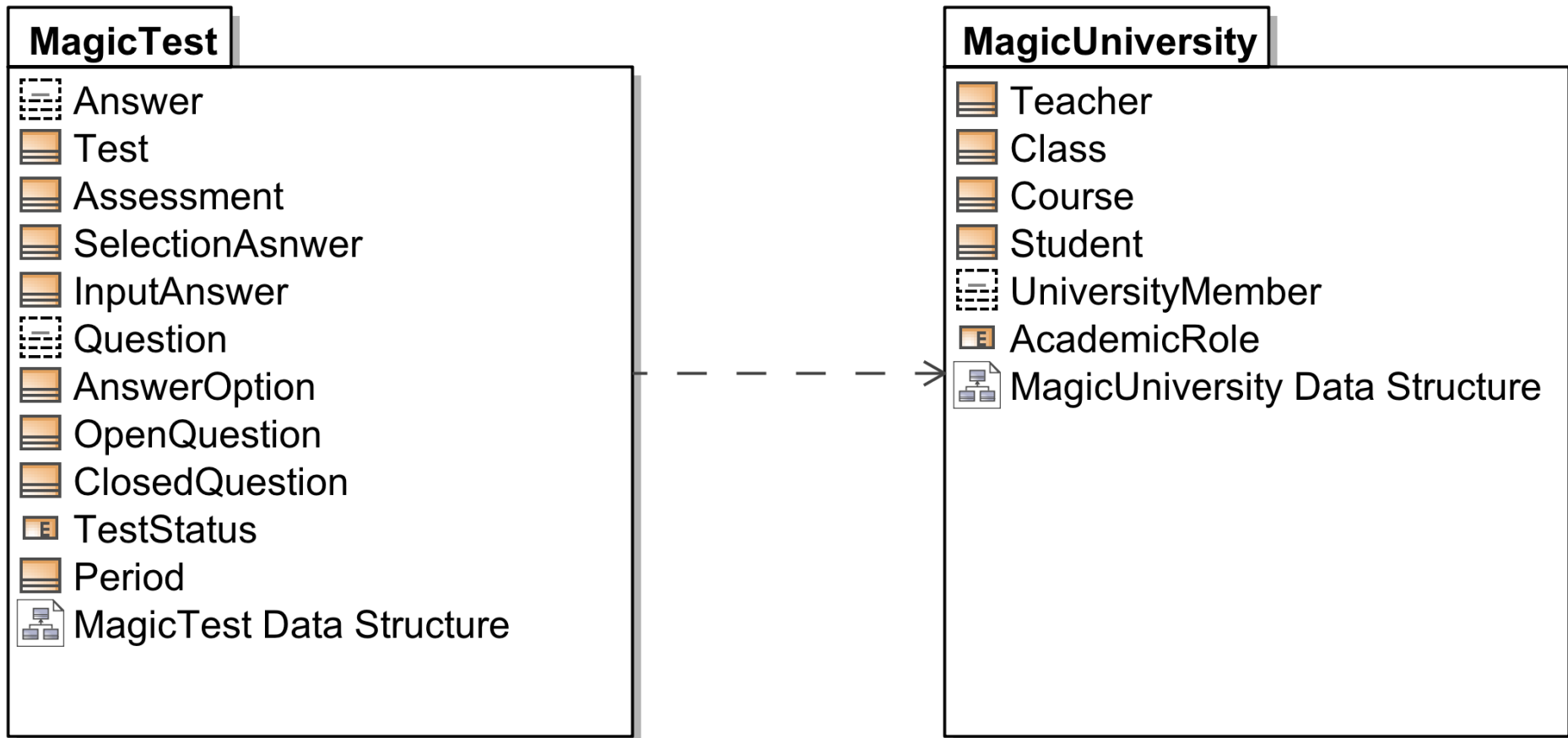
# Conceptual Data Modeling



✓ Conceptual modeling focuses on defining terms and their relations, not so much on precise data properties



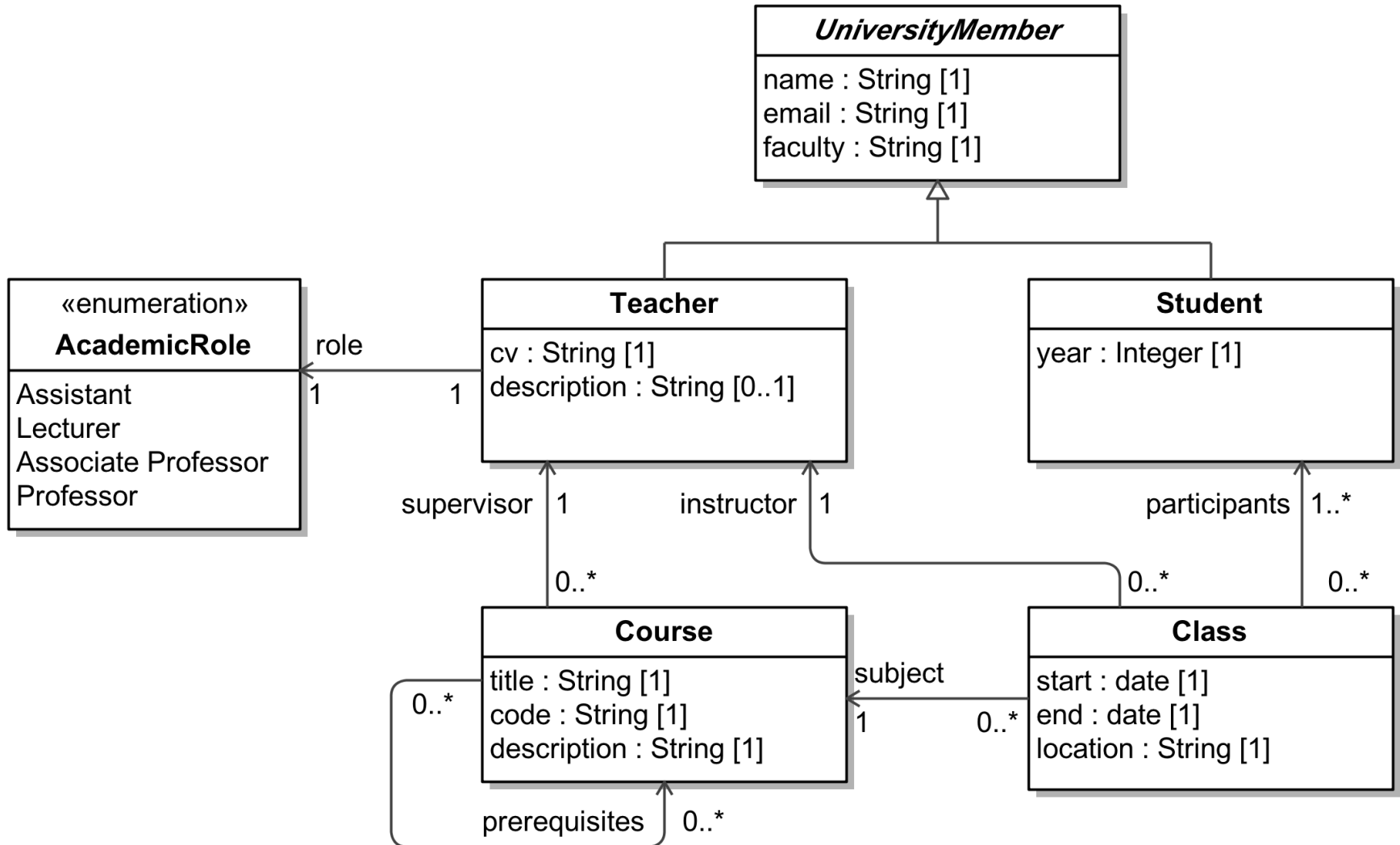
# Detailed Data Modeling: Overview (1)



✓ This small set of UML constructs is sufficient to model data structures completely and precisely

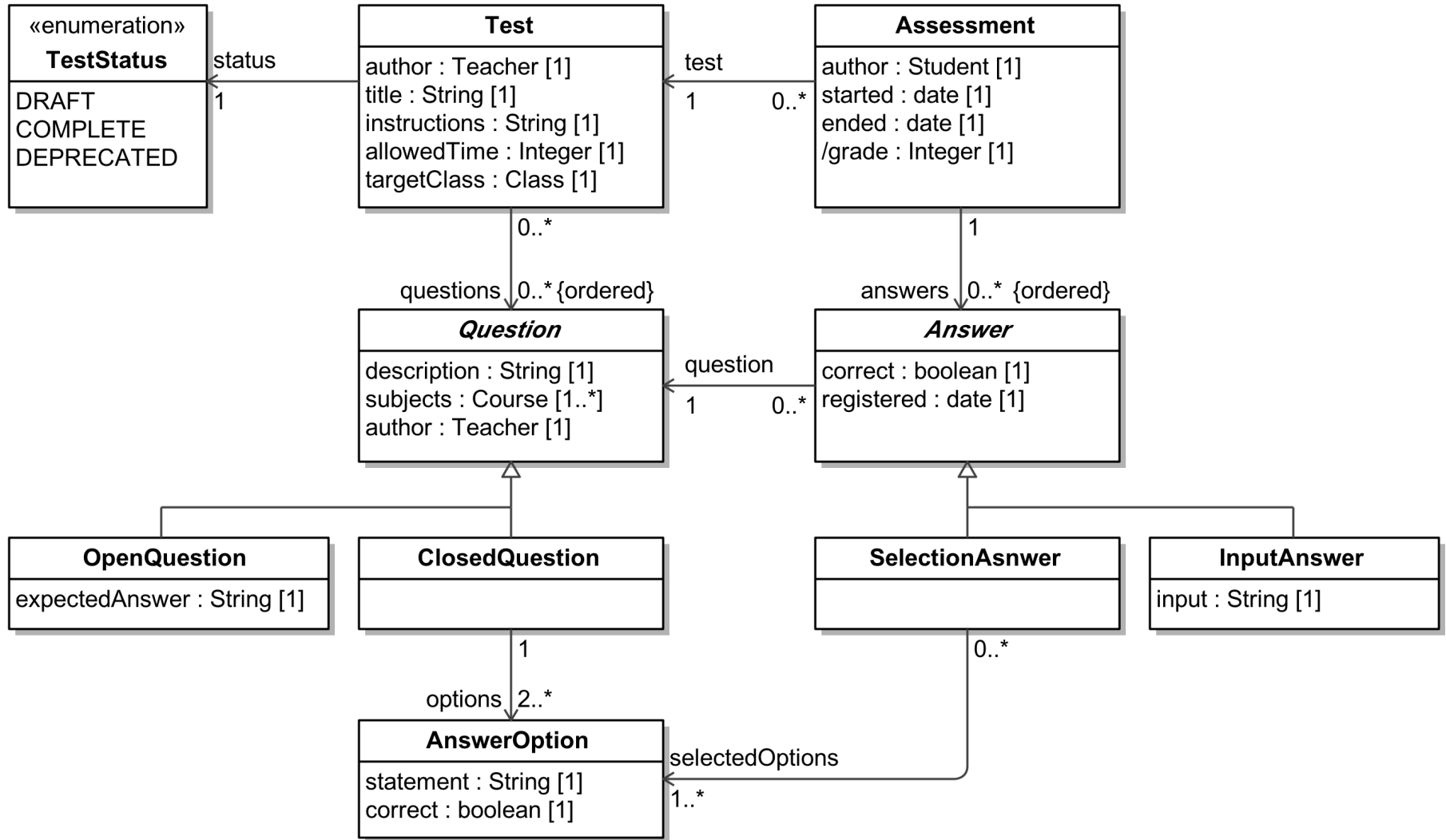
# Detailed Data Modeling: MagicUniversity (2)

package MagicUniversity [  MagicUniversity Data Structure ]

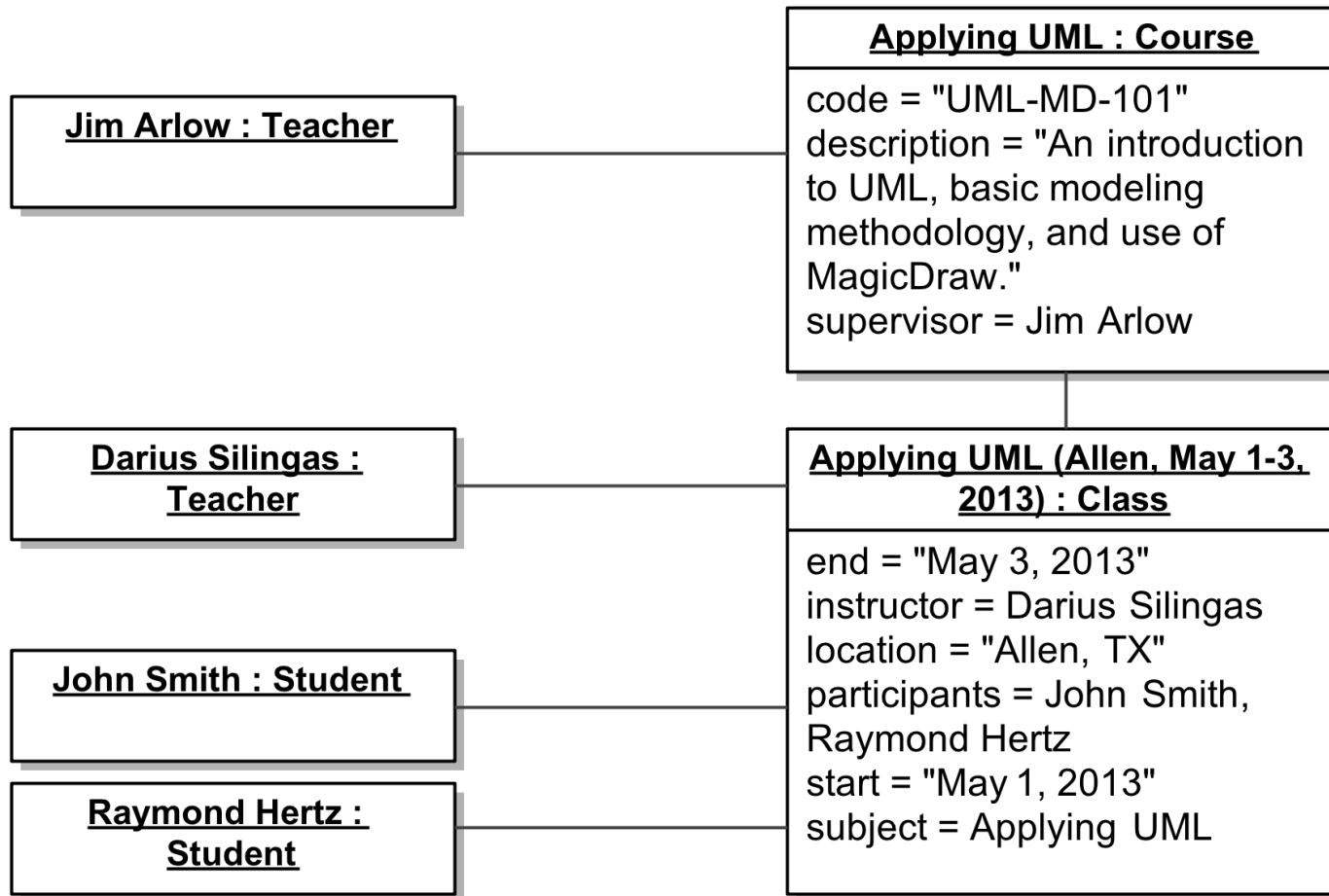


# Detailed Data Modeling: MagicTest (3)

package MagicTest [  MagicTest Data Structure ]

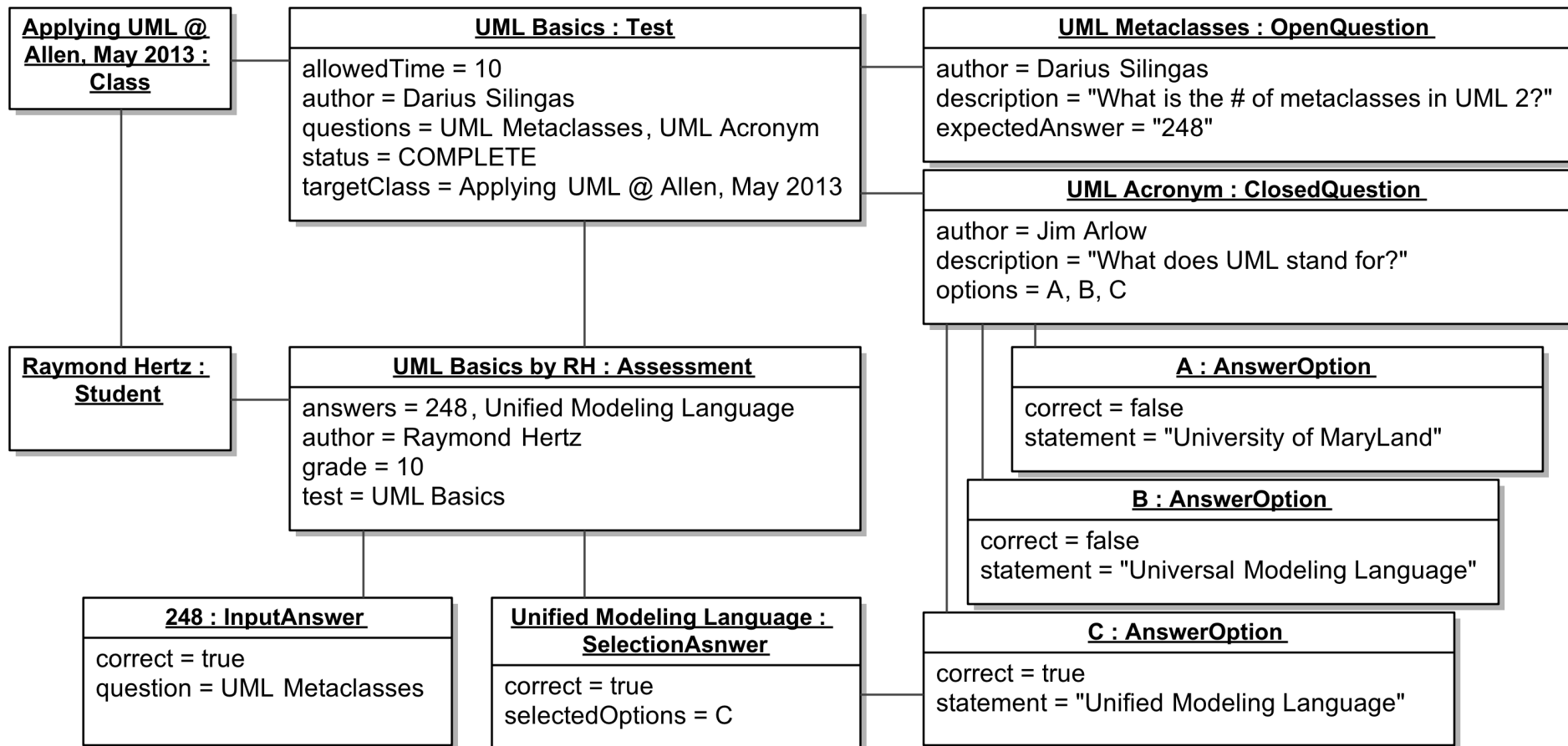


# Testing Data Models with Data Examples (1)



✓ Modeling representative data examples validates data structure design and prepares initial data set up for testing

# Testing Data Models with Data Examples (2)



# Data Model Validation: Correctness and Completeness

Data modeling is an error-prone process

A data model can be either

**incorrect** (*it breaks some modeling rules*) or

**incomplete** (*it lacks some required information*)

Rules defined in UML specification are automated in MagicDraw

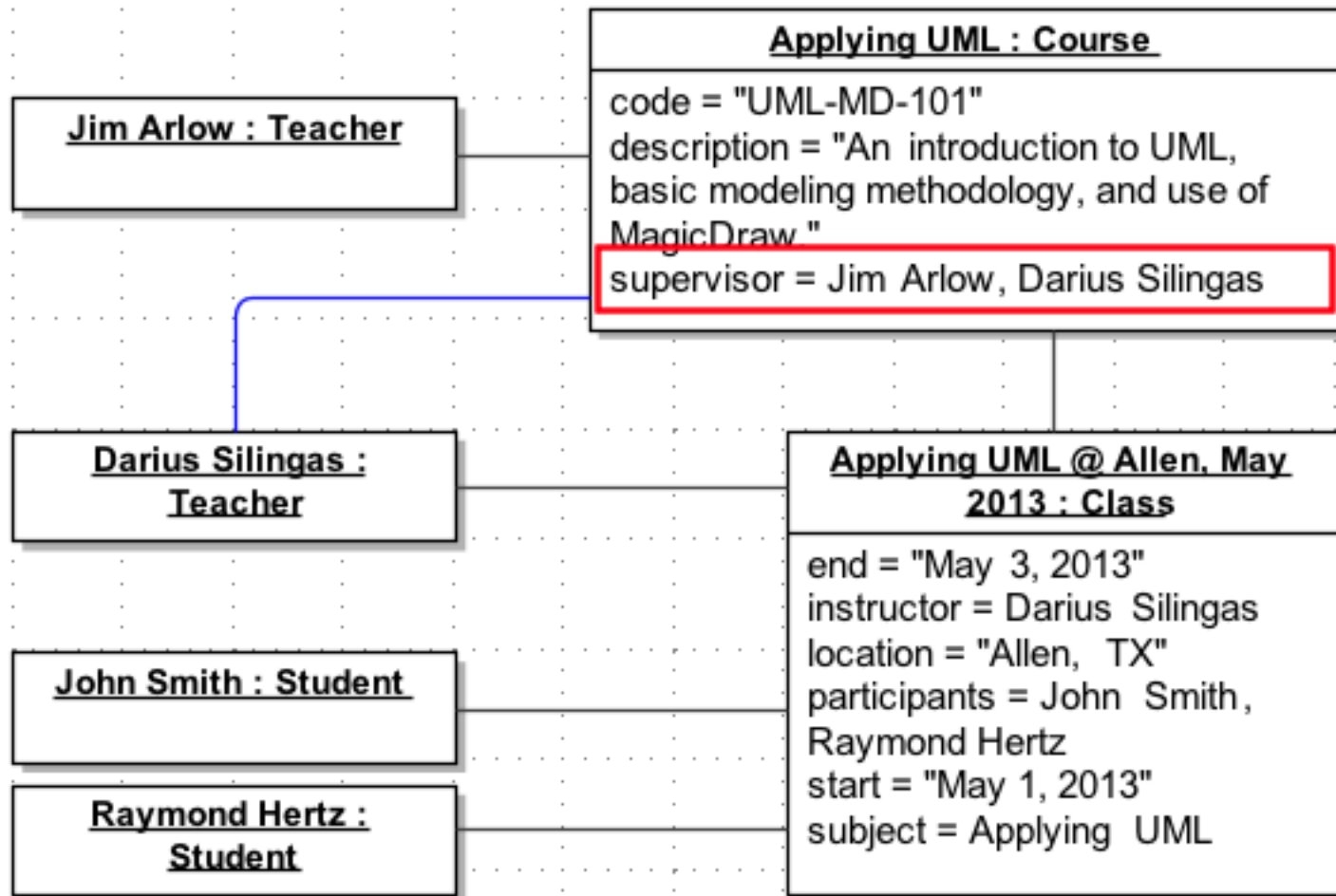
However, a specific modeling method typically implies additional rules

- Restriction to single generalization for classes
- Compulsory role names on navigable association ends
- Each class must be documented with owned comment

MagicDraw enables validating model based on custom validation rules

- Validation rules can be specified on OCL 2.0 or Java

# Validating Data Structure with Data Samples



- ✓ MagicDraw automatically detects model inconsistency
- ✓ Design choice: correct data structure or data example?

# Custom Validation of Model Completeness

The screenshot shows the Conceptual ER Analysis tool interface. The main workspace displays a UML model with the following elements:

- Entities:** Reader, Request, Penalty, and Category.
- Associations:**
  - Reader (1) to Penalty (0..\*) labeled "assigned for overdue".
  - Reader (1) to Request (0..\*) labeled "makes".
  - Request (0..\*) to Category (1..\*) labeled "ask".
  - Category (0..\*) to Category (0..\*) labeled "associated to".
  - Category (0..1) to Category (0..\*) labeled "has parent".

A **Message** dialog box is open, displaying the error: "Conceptual association should be named".

A **Validation** dialog box is also open, showing the following settings:

- Location: Conceptual ER Validation Suite
- Validation Selection: [Dropdown]
- Severity: >=debug
- Buttons: Validation Options, Cancel, Help

The **Validation** dialog box also shows a list of validation rules:

Rule Name	Expression	Severity	Message
NonnameAssociation	name.size() > 0	error	Conceptual association should be named

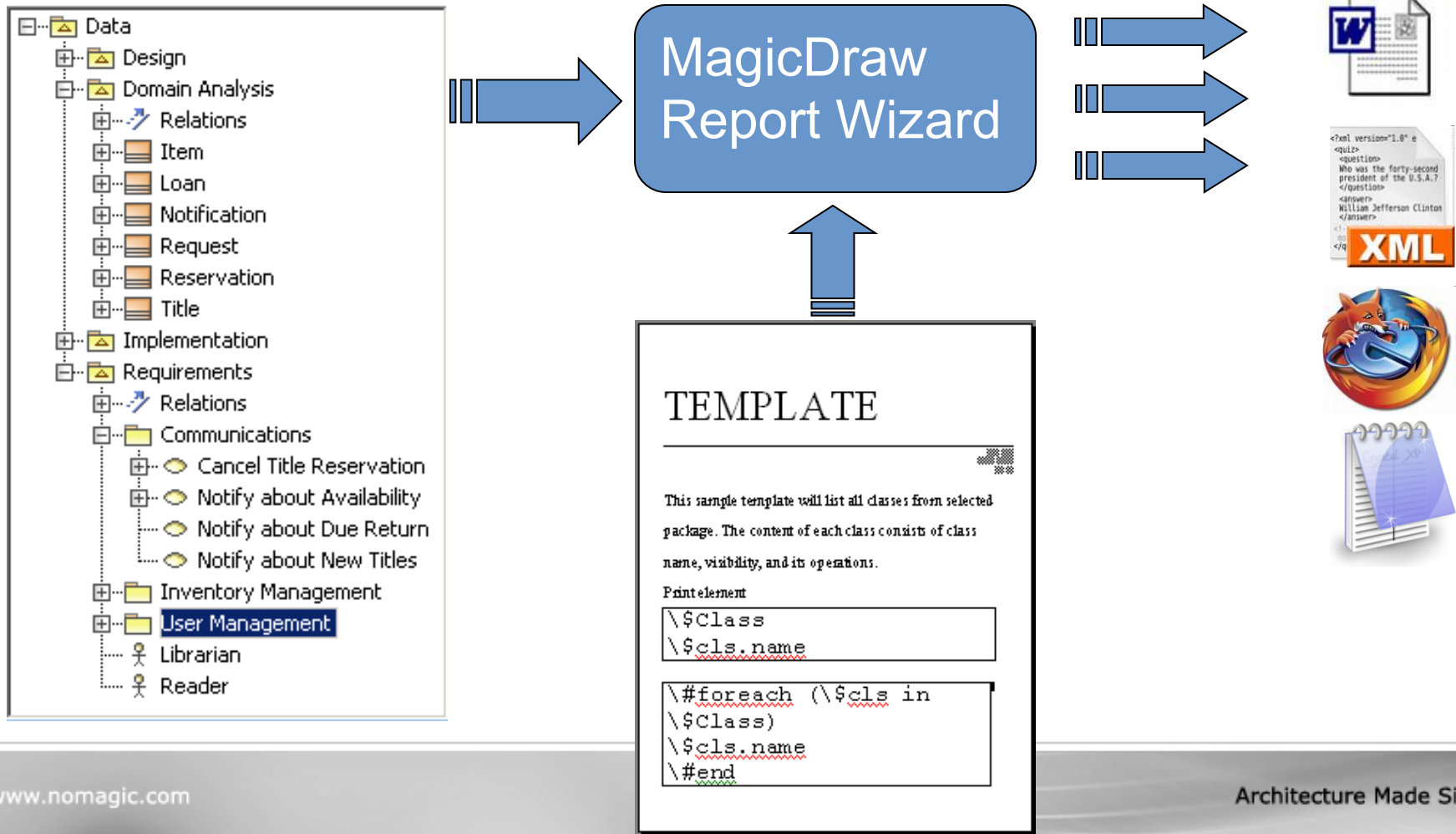
The **Validation** dialog box also shows a list of validation rules:

Rule Name	Expression	Severity	Message
NonnameAssociation	name.size() > 0	error	Conceptual association should be named



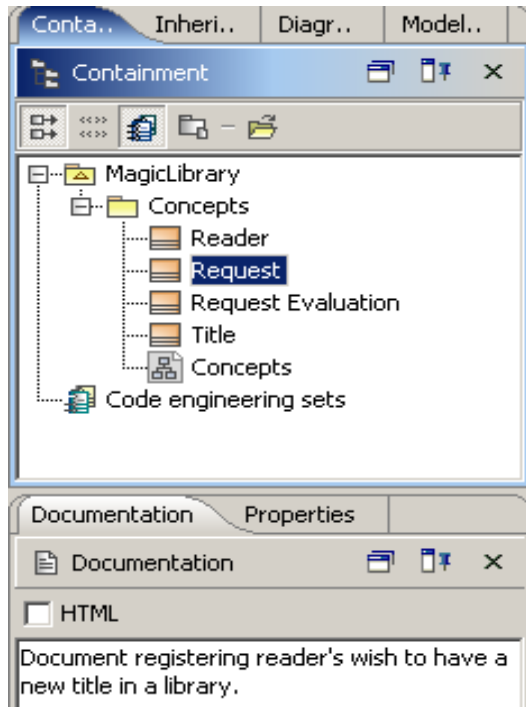
# Transforming Data Model to Documents (1)

Using MagicDraw Report Wizard, you can transform your model into HTML, XML, Microsoft Word, and other formats based on customizable report templates



# Transforming Data Model to Documents (2)

- It is recommended to publish model contents on the web accessible without MagicDraw on a regular basis



Concept	Description
<pre>#forrow (\$class in \$sorter.sort(\$Class, "name")) \$report.getIconFor(\$class) <b>\$class.name</b></pre>	<pre>\$report.getComment(\$class) #endrow</pre>

Concept	Description
<b>Reader</b>	<i>Information about library customer.</i>
<b>Request</b>	<i>Document registering reader's wish to have a new title in a library.</i>
<b>Request Evaluation</b>	<i>Librarian's decision whether to approve or deny reader's request.</i>
<b>Title</b>	<i>Information about a book, journal or another kind of library inventory item. Library may contain multiple copies of the same title.</i>

- You can start with standard Web Publisher 2.0 template, but a custom report template typically gives the best output

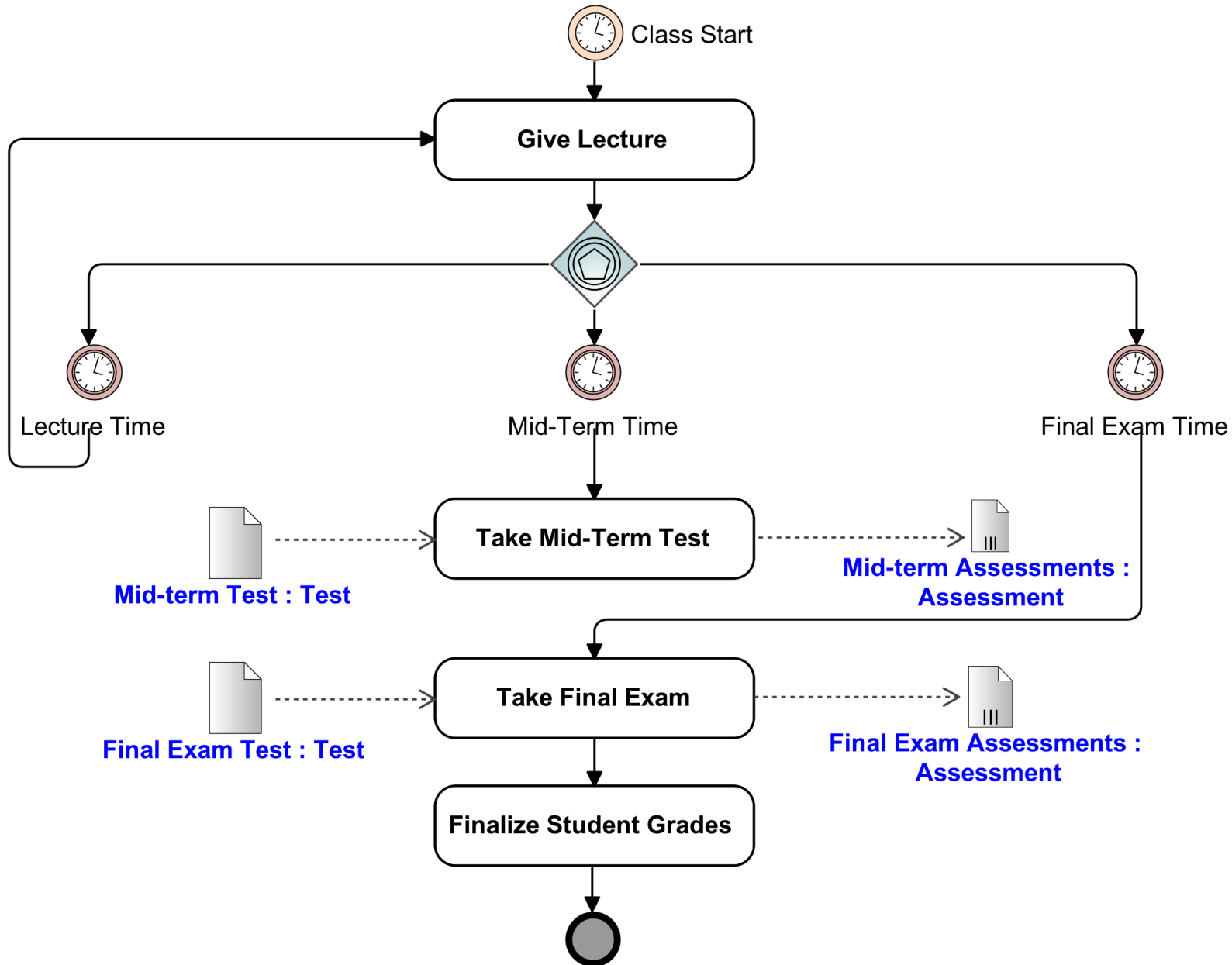
# Agenda

Data Modeling Concepts


Data Modeling Techniques

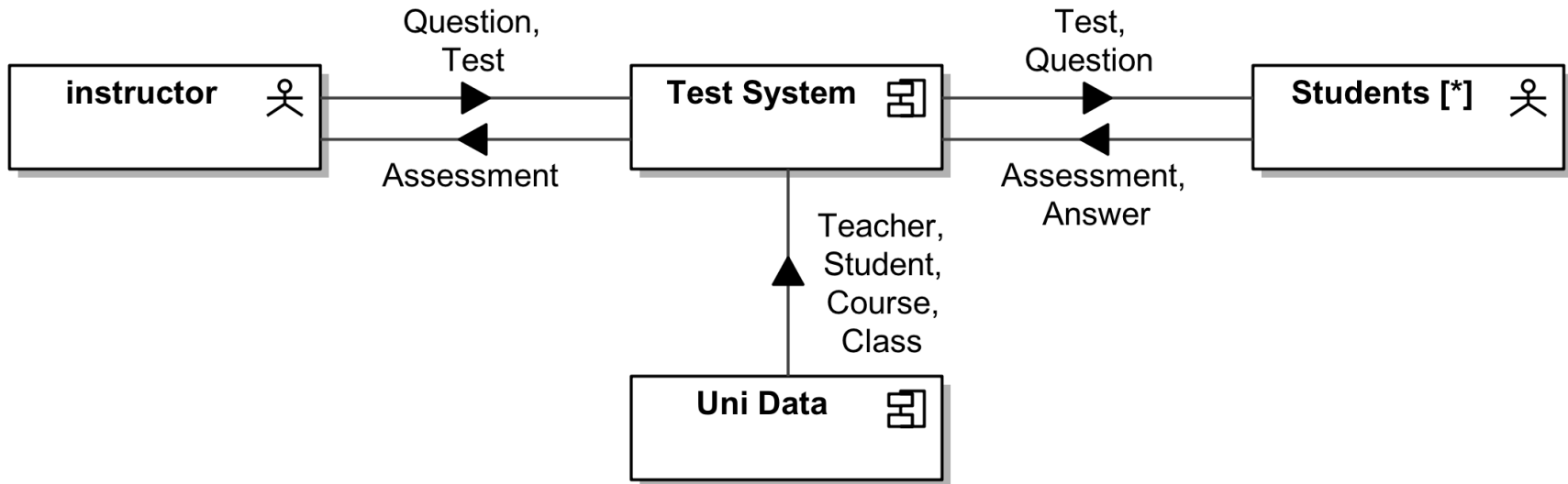
**Using Data Models**

# Using Data Objects in Business Processes



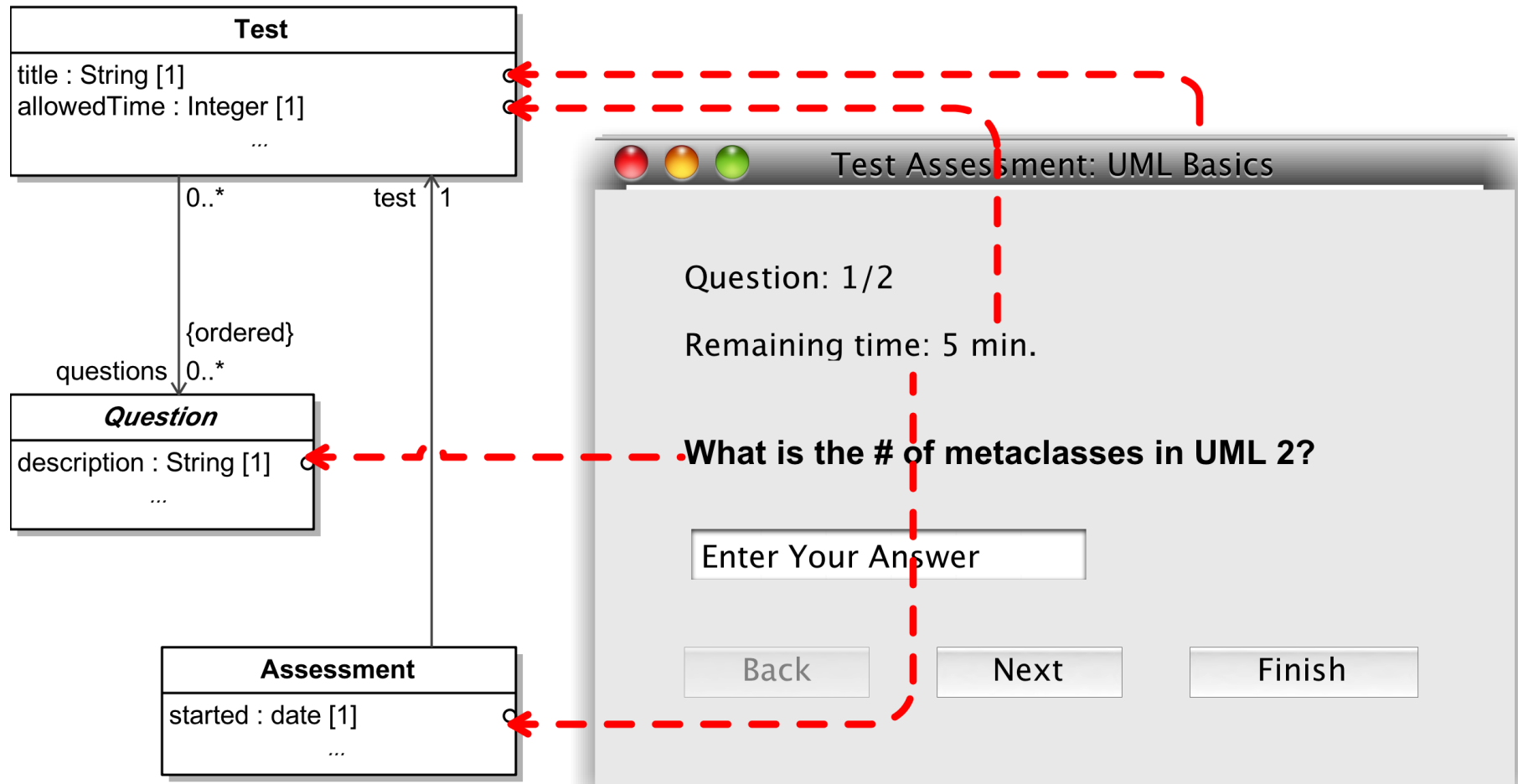
# Modeling System Context with Information Flows

**package** Class Context [  Assessing Students Knowledge in Class ]



✓ System context diagram is often included in project vision in order to understand solution's environment

# Designing GUI Based on Data Models



✓ System analysts often do data modeling via GUI design. They should rather trace GUI elements to data design!

# Tracing Use of Data Structures

	Remaining time: 5 min.	Enter Your Answer	Question: 1/2	What is the # of meta...
[-] [Folder Icon] MagicTest	2		1	1
[-] [Folder Icon] Test	1		1	
-allowedTime : Integer [1]	✓			
-instructions : String [1]				
-questions : Question [0..*]			✓	
-title : String [1]				
[+] [Folder Icon] Assessment	1			
[+] [Folder Icon] Question				1

✓ MagicDraw provides data modelers with an easy way to visualize and edit traces between elements in matrix form

# Agenda

Data Modeling Concepts

Data Modeling Techniques

Using Data Models



# Wrap up

- ✓ UML has mature capabilities for modeling data structures
- ✓ A small subset of UML is enough for data modeling
- ✓ Data models should contain both data structure definitions and representative examples
- ✓ Modeling tool should enable data model analysis, including model validation for correctness and completeness, and transformation to documents and code
- ✓ There are many ways to utilize data model elements in modeling various aspects of system architecture



**Questions?**  
**Thank you!**