Data Modeling for Business Analysts
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
✓ 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**

- **Class**
-  `attribute1` : Integer [1] = 0
-  `attribute2` : String [0..*]
-  `attribute3` : Enumeration

- «enumeration»
-  **Enumeration**

-  **Package**

**Data Relations**

- **Association**
-  **Directed Association**
-  **Generalization**
-  **Dependency**

**Data Samples**

- **InstanceSpecification : Class**
  - `attribute1` = 1
  - `attribute2` = "Some text"
  - `attribute3` = LITERAL2

- **Link**
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 an end of **Association**.
• **Enumeration** is a data type whose potential values are enumerated as literals.
• **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** specifies a semantic relationship that can occur between two typed instances.
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.
Conceptual modeling focuses on defining terms and their relations, not so much on precise data properties.
This small set of UML constructs is sufficient to model data structures completely and precisely.
Modeling representative data examples validates data structure design and prepares initial data set up for testing.
Testing Data Models with Data Examples (2)

Applying UML @ Allen, May 2013: Class

Raymond Hertz: Student

UML Basics: Test
- allowedTime = 10
- author = Darius Silingas
- questions = UML Metaclasses, UML Acronym
- status = COMPLETE
- targetClass = Applying UML @ Allen, May 2013

UML Basics by RH: Assessment
- answers = 248, Unified Modeling Language
- author = Raymond Hertz
- grade = 10
- test = UML Basics

248: InputAnswer
- correct = true
- question = UML Metaclasses

Unified Modeling Language: SelectionAnswer
- correct = true
- selectedOptions = C

UML Metaclases: OpenQuestion
- author = Darius Silingas
- description = "What is the # of metaclasses in UML 2?"
- expectedAnswer = "248"

UML Acronym: ClosedQuestion
- author = Jim Arlow
- description = "What does UML stand for?"
- options = A, B, C

A: AnswerOption
- correct = false
- statement = "University of Maryland"

B: AnswerOption
- correct = false
- statement = "Universal Modeling Language"

C: AnswerOption
- correct = true
- statement = "Unified Modeling Language"
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
It is recommended to publish model contents on the web accessible without MagicDraw on a regular basis.

You can start with standard Web Publisher 2.0 template, but a custom report template typically gives the best output.
Using Data Objects in Business Processes

1. Class Start → Give Lecture
2. Lecture Time
3. Mid-Term Time → Take Mid-Term Test
   - Mid-term Test: Test
   - Mid-term Assessments: Assessment
4. Take Final Exam
   - Final Exam Test: Test
   - Final Exam Assessments: Assessment
5. Finalize Student Grades
System context diagram is often included in project vision in order to understand solution’s environment
System analysts often do data modeling via GUI design. They should rather trace GUI elements to data design!
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!