

INFS4206/7206 Advanced Topics in Database

2006 Semester 2 Part 1 Metamodeling

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

1

Welcome to Metamodeling: Contacts

- ❖ Course Co-ordination
 - Dr. Bob Colomb
 - Office - 78-628
 - Phone 3365 1190
 - E-mail colomb@itee.uq.edu.au
 - www.itee.uq.edu.au/~colomb

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

2

Teaching & Learning: Lectures

- ❖ When
 - Time: 10:00am-12:00pm
 - Day: Thursday
- ❖ Where
 - St Lucia, 78-622

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

3

Teaching & Learning: Consultation

- ❖ Bob Colomb
 - GP South 628
 - by appointment: colomb@itee.uq.edu.au
- ❖ Prepare your questions prior to consultation

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

4

Introduction: Teaching & Learning Resources

- ❖ Web-site
 - www.itee.uq.edu.au/~infs4206/
 - Contains
 - all course details
 - links/references to further reading
- ❖ Text Book
 - Notes and readings available from bookstore
POD service or on the web site

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

5

Teaching & Learning: Assessment

- ❖ Assessment by assignments and examination
 - Assignment due week 7, Thursday 7 September, 25%
 - at lecture
 - Assignment in groups of up to 3
 - Final examination 25%

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

6

Outline of Course

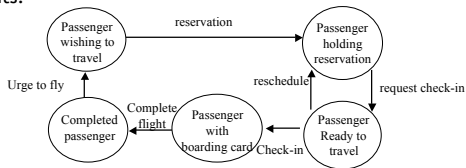
5	Metamodeling
6	MOF Metamodels
7	Model-Driven Architecture
8	Further Topics

Metamodeling

- ❖ A system design is a complex object. Information systems are about keeping track of complex structures. This is what metamodeling is for
- ❖ We will start with an example, developing along the way concepts of **metamodel** and **repository**

Example of State Transition Model

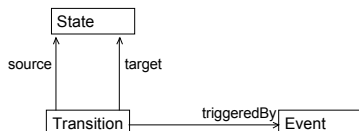
Passengers buy tickets at the time they obtain reservations. At check-in time they obtain boarding cards if there are still seats available. Due to over booking of flights they may be rescheduled on later flights.



Components of Simple STM

- ❖ **State** - Five states in the example
- ❖ **Transition** - Link from source state to target state
 - Six transitions in the example
- ❖ **Event** - Associated with transition
 - One for each transition
- ❖ Real STM is considerably more complicated

UML Model of STM



Repository Schema for STM

- ❖ **State**
 - Name : string
- ❖ **Transition**
 - Source : State
 - Target : State
 - TriggeredBy : Event
- ❖ **Event**
 - Name : string

Population of Repository for Model Instance

❖ State	❖ Event
❖ Name	❖ Name
❖ WishTravel	❖ reservation
❖ Completed	❖ reschedule
❖ HoldRes	❖ reqCheckIn
❖ ReadyTravel	❖ checkIn
❖ WBoardCard	❖ complete
	❖ urgeFly

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

13

Population continued

❖ Transition		
❖ Source	Target	triggeredBy
❖ WishTravel	HoldRes	reservation
❖ HoldRes	ReadyTravel	reqCheckIn
❖ ReadyTravel	HoldRes	reschedule
❖ ReadyTravel	WBoardCard	checkIn
❖ WBoardCard	Completed	complete
❖ Completed	WishTravel	urgeFly

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

14

Query on Repository

- ❖ Sink state has no transition out
- ❖ CREATE VIEW Sink (StateName) AS
- ❖ SELECT State.name FROM State, Transition
- ❖ WHERE
 NOT EXISTS SELECT Transition.Source
 FROM Transition
 Where Transition.Source = State.Name

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

15

SQL-92 not sufficient

- ❖ Can tell if a model is entirely cyclic, since it has neither source nor sink
- ❖ SELECT 'Cyclic' FROM State WHERE
 - NOT EXISTS SELECT * FROM Source AND
 - NOT EXISTS SELECT * FROM Sink
- ❖ But to find a state participating in a cycle or a cycle involving only some states need a recursive query. Not possible in SQL-92, but can in SQL:1999.
- ❖ Model repositories need richer query languages
- ❖ Smaller populations, but more complex

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

16

Edit Model in Repository

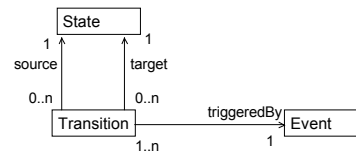
- ❖ Want to use repository to create and edit model instance
- ❖ But want model to be syntactically correct
- ❖ Use integrity constraints to enforce syntax rules
- ❖ Example: A transition in a STM must link two states with an event.
- ❖ Does it make sense to have an event without a transition?
- ❖ A state without a transition?

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

17

Multiplicities Useful



- But also need constraint queries
- No isolated states
 - One source, one sink, no sources nor sinks

INFS4206/INFS206 part 2 week 1, 19 July 2006

Bob Colomb

18

Metamodel defines rendering conventions

- ❖ State : Name within circle
- ❖ Transition : arrow between states
 - Source : no arrowhead
 - Target : arrowhead
 - Event : name associated with arrow
- ❖ Event : not independently rendered
- ❖ There are graph layout algorithms to produce a readable rendering

Summary so far

- ❖ **Metamodel** is conceptual model for syntax of a modeling system
- ❖ Metamodel specifies **schema** for repository
- ❖ Repository stores **model instances**
- ❖ **Constraints** expressed as queries on repository
- ❖ Repository supports model creation, editing, **rendering**, browsing etc.

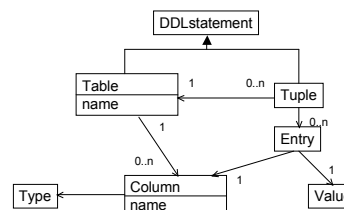
There are many metamodeling systems

❖ Backus-Naur Form metamodel of SQL DDL

- <DDLstatement> ::= (<createTable><insertInto>)*
- <createTable> ::= 'CREATE TABLE' <TName> <ColumnDecs>*
- <TName> ::= string
- <ColumnDecs> ::= null | <ColumnDec><ColumnDecs>
- <ColumnDec> ::= <Aname><Type>
- <insertInto> ::= 'INSERT INTO' <Tname><Values>
- <Values> ::= 'VALUES (' <valueList> ')'
- <valueList> ::= null | <value> <valueList>

A system can have several metamodels

❖ UML Metamodel for SQL Create Table and Insert



Different metamodeling systems allow different things to be said

Natural language as a metamodel

- ❖ Natural language is the most powerful and flexible metamodeling system
- ❖ But more easily misunderstood
- ❖ Hard to get precise
- ❖ So harder to automate
- ❖ Metamodeling needs to support interoperating automated tools
- ❖ Works against natural language, but NL commonly used
- ❖ EG Topic Maps (TMDM) uses UML as illustrations, but
- ❖ *These diagrams are purely informative, and in cases of discrepancy between the diagrams and normative prose, the prose is definitive.*

Repository for SQL DDL

- ❖ Columns
- ❖ TableName ColumnName Type
- ❖ Entry
- ❖ TupleID TableName ColumnName Value

- ❖ In a database manager, the table specifications are managed exactly this way. This is the data dictionary in the system catalog.
- ❖ But entry tables are managed separately, with schemas

❖ TableName

• TupleID ColumnName1 ColumnName2 ...

- ❖ ID1 value11 value12 ...
- ❖ ID2 value21 value22 ...

But the repository can store instances

❖ First few tuples of STM Transition Table

❖ Entry

❖ TupleID	TableName	ColumnName	Value
❖ 001	Transition	Source	WishTravel
❖ 001	Transition	Target	HoldRes
❖ 001	Transition	Event	reservation
❖ 002	Transition	Source	HoldRes
❖ 002	Transition	Target	ReadyTravel
❖ 002	Transition	Event	reqCheckIn
❖ 003	Transition	Source	ReadyTravel
❖ 003	Transition	Target	HoldRes
❖ 003	Transition	Event	reschedule

Classes model vs Instance model

❖ The system being modeled in a metamodeling structure typically has a structure represented as a collection of classes which can be populated by a collection of instances.

❖ So part of the structural features model is considered a **classes model** and part an **instances model**.

❖ The repository will use a generic representation of the classes model and often a model-specific representation of the instances model.

A metamodeling system can be modeled

❖ BNF in BNF

❖ <model> ::= <statement>*

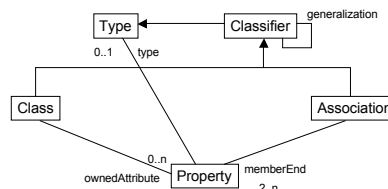
❖ <statement> ::= <nonTerminal> '::<>' (<terminal> | <nonTerminal>)*

❖ <terminal> ::= string

❖ <nonTerminal> ::= '<' string '>'

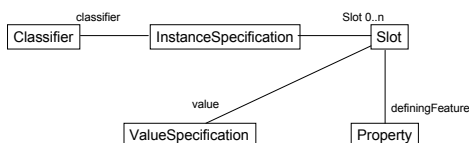
A metamodeling system can be modeled

UML in UML



Note classes model, but no instances model shown
This is called the **Meta Object Facility** or **MOF** - Next week

MOF Instances Model



Summary: Key Terms



- ❖ **Metamodel** is conceptual model for syntax of a modeling system. Specifies **schema** for **repository**
- ❖ **Repository** stores **model instances**. **Constraints** expressed as queries on repository. Repository supports model creation, editing, **rendering**, browsing etc.
- ❖ Metamodel usually has **classes model** and **instances model**

Resources

❖ Readings

- Intro to UML
- UML Basics : The Class Diagram
- UML without pictures

❖ Resources

- Topic Map Data Model
- Topic Maps Handbook
- OWL-S specification