Conceptual modelling using UML class-diagram

UML class-diagram – selected parts

Learning Goals

After this lecture you will be able to:

- Understand and use the syntax of UML class diagrams
- Use UML class diagram to model user domains
Synopsis

From reality to system
Conceptual modeling
UML class diagram and modeling patterns

Architecture of an Information system

Presentation  (Graphical) user interfaces
Applikationlogic  Java servlets executed on a server
Data  Data from a Database Management System
**Presentation**

How does (end-) users retrieve information from the information system? Graphical user interfaces to various applications provide information about data stored in the system (usually in the database management system).

**Query-languages**

- How does application program retrieve data from the database management system?
- Through a query language.

```
SELECT Name 
FROM PRODUKT 
WHERE Type = "Chair"
```
Relational databases

CUSTOMER

<table>
<thead>
<tr>
<th>Name</th>
<th>Gatuadress</th>
<th>Postnummer</th>
<th>Postadress</th>
<th>Telefon</th>
<th>Kontokort</th>
<th>Giltighetstid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pia Eriksson</td>
<td>Storg. 12</td>
<td>171 55</td>
<td>Torget</td>
<td>88 77 86</td>
<td>155 666 777</td>
<td>11/01</td>
</tr>
<tr>
<td>Ulf Eriksson</td>
<td>Storg. 12</td>
<td>171 55</td>
<td>Torget</td>
<td>88 77 86</td>
<td>234 555 678</td>
<td>11/00</td>
</tr>
<tr>
<td>Cla Hansson</td>
<td>Lillg. 2</td>
<td>218 43</td>
<td>Landet</td>
<td>55 44 11</td>
<td>666 222 111</td>
<td>09/01</td>
</tr>
<tr>
<td>Bo Eriksson</td>
<td>Storg. 12</td>
<td>171 55</td>
<td>Torget</td>
<td>88 77 86</td>
<td>432 854 221</td>
<td>11/01</td>
</tr>
<tr>
<td>Bo Johnsson</td>
<td>Erleg. 45</td>
<td>171 55</td>
<td>Torget</td>
<td>98 77 21</td>
<td>123 444 321</td>
<td>11/01</td>
</tr>
<tr>
<td>My Hansson</td>
<td>Lillg. 2</td>
<td>218 43</td>
<td>Landet</td>
<td>55 44 11</td>
<td>666 222 111</td>
<td>09/01</td>
</tr>
</tbody>
</table>

Building the “right” system

Universe of Discourse

Conceptual Model (CM)

Use of CM

Data-base

Education, Reverse Engineering
Conceptual modelling

A conceptual model depicts information – entities and relationships between entities – in a way that is independent of how information and data is represented and stored. How the model is to be implemented shall not effect the way we model it!

Models – how accurate are they?

Model: A structure that depicts some (chosen!) aspects of reality.

An example of a model – a MAP!

Models simplifies
Models distort- Greenland too big, Africa too small?
Models focus – topographical maps, political maps, …
Models and modelling languages

Models: Structure and behaviour

Structural models/structur-diagrams
- Specifies statical aspects of the system, i.e. statical relationships between terms in the system

Behavioural models/behaviour-diagrams
- Specifies dynamic (behavioural) aspects of the system, i.e. specifies the manipulation/change/construction/deletion of the statical relations and in which order this happens
Graphical modelling languages


Some graphical modelling languages are more expressive than others. One reason for this is that they contain more modelling elements (symbols). They can as a result represent more concepts, i.e. more aspects of reality (or the system). A disadvantage is that a model written in a “rich” modelling language with great expressiveness and many symbols is harder for a user to understand and validate.

UML classdiagram for the first time (or revisited)
Class diagram – the center of UML

Class diagrams constitute a majority of UML diagrams – sometimes almost used as a synonym of UML.

Class diagrams describe the **classes** of a domain or a system, and **statical associations** between the classes.

Class diagrams also show the **attributes** of the classes (and **operations**).

Note här, in this context the following words may be used as synonyms:
- type and **class**,
- relation and **association**,
- structural and **static**

Class – variants in notation

This middle notation (somewhat simplified) is usually used in systems analysis.
### Classdiagram and object-diagram

#### Classdiagram

**Student**
- personnr
- namn
- bostadsadress
- epostadress

#### Object-diagram

**nilsHall:Student**
- personnr = "850302-XXXX"
- namn = "Nils Erik Hall"
- bostadsadress = "Rågstigen 3"
- epostadress = "hal@dsv.su.se"

**annaSvan:Student**
- personnr = "770102-XXXX"
- namn = "Anna Cecilia Svan"
- bostadsadress = "Ekvägen 10"
- epostadress = "sva@dsv.su.se"

Object-diagram

- Often called instance-diagram.
- Extended notation for instance names.
- Attributes are assigned values.
- All associations from class diagram are present in object diagrams.
- Can be seen as an observation of one or several objects at a given point in time. Attribute values may change during the life cycle of an object.

### Attributes and the roles of an association have multiplicites (aka cardinality constraints)

<table>
<thead>
<tr>
<th>Student</th>
<th>1..1</th>
<th>0..*</th>
<th>Registration</th>
<th>0..*</th>
<th>1..1</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>personnr</td>
<td>1..1</td>
<td>0..*</td>
<td>registreringsID</td>
<td>1..1</td>
<td>0..1</td>
<td>kursID</td>
</tr>
<tr>
<td>namn</td>
<td>1..1</td>
<td>0..*</td>
<td>datum</td>
<td>1..1</td>
<td>0..1</td>
<td>kursnamn</td>
</tr>
</tbody>
</table>

The **multiplicity** for an attribute or a role of an association indicate how many objects the role of the association refer to or how many values the attribute may take on.

Multiplicity for the roles of associations are placed close to the associated classes.

For multiplicities:
- the smallest (minimum) number is given first
- the larger (maximum) number last
- Two points (full stop symbol) separate the minimum and maximum multiplicity
## Multiplicity – usual combinations

<table>
<thead>
<tr>
<th>Student</th>
<th>Registration</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>personnr 1..1</td>
<td>registrationID 1..1</td>
<td>courseID 1..1</td>
</tr>
<tr>
<td>name 1..1</td>
<td>date 1..1</td>
<td>coursename 0..1</td>
</tr>
<tr>
<td>email 1..*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1..1 At least 1 and at most 1 ("one and only one"). Example: A student have exactly one personal number, i.e. one value on the attribute personal number.

1..* At least 1 and an no upper limit ("one and possibly many more")

0..* 0 is valid and no upper limit ("not necessarily any and and possibly many")

0..1 0 is valid and at most 1 (not necessarily any and at most one)

Example: A course may not have a coursename but if it has there is at most one name of the course.

---

## Multiplicites for ATTRIBUTES in UML – what identifies a class?

### Student

<table>
<thead>
<tr>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>personnr 1..1</td>
</tr>
<tr>
<td>name 1..1</td>
</tr>
<tr>
<td>email 1..*</td>
</tr>
</tbody>
</table>

Example: A student have exactly one personal number, i.e. one value on the attribute personal number.

### Student

<table>
<thead>
<tr>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>personnr 1..1</td>
</tr>
<tr>
<td>name 1..1</td>
</tr>
<tr>
<td>email 1..*</td>
</tr>
</tbody>
</table>

Example: A student have exactly one personal number, i.e. one value on the attribute personal number.
**Multiplicity – rules in an information system vs ‘common sense’ - exercise**

<table>
<thead>
<tr>
<th>CAR ownedBy PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regno: Sträng 1..1</td>
</tr>
<tr>
<td>UNIQUE</td>
</tr>
<tr>
<td>Model Sträng 1..1</td>
</tr>
<tr>
<td>UNIQUE</td>
</tr>
</tbody>
</table>

---

**So how do we start modelling...?**

- In a real situation users are interviewed, manuals are read, legacy systems analysed, etc…
- Sometimes we only have access to textual descriptions of the system or domain to be modelled
- How to address such descriptions? Analyse role of verbs, nouns, adjectives? Modelling patterns, data-abstractions, rule of thumbs, etc....
What is important, what is the focus, ...?

A company wishes to sell its products, various types of furniture, through a web-shop. The customers shall be able to order the furniture via the web-shop. The products shall then be delivered to the customers. To give the customers a good view of the furniture it is important to display information about the size, type, weight and type of material the products are made of; a photo of the various types of furniture is also very important. Price is of course also an important factor. On top of the price of an individual piece of furniture there is also the price for the delivery of the purchased articles – this price is based on the weight of the furniture. The customer may have the products delivered to her home or to an address of her choice. The customer may choose a delivery date – it is however possible that the real date of delivery may deviate from the one the customer asked for. In order to let the customer know when a delivery has been started, information about departure times for various delivered orders must be present on the web-site. The company strives to deliver all products that are part of one order to one customer in one batch but sometimes this is not possible. The customer pay via credit-cards and information about payment must be present on the web-site.

Same word- differenct concepts(homonyms)

In what ways are the word "book" used in the sentences below?

- Jules Verne wrote many books, one of which has the title 'Captain Grant's Children'.
- The library in Vällingby has many books.
- If 'Captain Grant's Children' is printed in A4-format the number of pages in a printed book will be 100.
- The book-shop of Vällingby sells many books, the top selling book was 'Captain Grant's Children' with 2000 sold books.
Same concept – different words (synonyms)

- Product - Article
- Buy – Purchase
- Big – Large
- Fire – Let go?
Modelling patterns - Analysis patterns - Data abstractions

Hierarchical structures are present in reality. As such we need to capture these structure when we create a model of the same reality.

Däggdjur

Djur

Fåglar

Gräsätare

Rovdjur

Pingvin

Hovdjur

Tax

Gnagare

Gnu

Kanin

Inheritance.

A class-schema with so called 'isa'-relationships:

DJUR

DÄGGDJUR

Gräsätare

ROVDJUR

Gnu

FÅGLAR

HARE

Rovdjur

PINGVIN

HOVDJUR

Gnagare

TAX

isa

isa

isa

isa

isa

isa

isa

isa
Inheritance cont.

An inheritance hierarchy consists of sub-classes and super-classes. If the sub-classes span the entire super-class we say that the sub-classes are exhaustive wrt to the super-class. If one and the same instance of a sub-class may not belong to more than one sub-class we say that the sub-classes are mutually exclusive.

MAN and WOMAN are mutually exclusive and exhaustive with respect to PERSON.

DOG and CAT are mutually exclusive but not exhaustive with respect to ANIMAL.

An attribute that has a minimum value = 1 is called a TOTAL attribute. An attribute with a minimum value = 0 is called a PARTIAL attribute.

Inheritance

What would the model look like if we did not use isa-relationships?
Inheritance

Inheritance excersise

- AN INFORMATION SYSTEM IS TO BE DEVELOPED, the domain is animals in zoological gardens. Help the systems analyst to construct a conceptual schema in the form of an UML class-schema that can represent the following types of facts (it is important that all attributes are total, i.e have a minimum value = 1):
  - Shere-Khan is a tiger weighing 250 kg and with a tail length of 1.4 meter.
  - Tweety the penguin has a wing-span of one meter and weighs 10 kg
  - Clyde is a 1.5 ton heavy elephant.
  - Elephants have tail lengths and so has tigers but not penguins. Penguins have wing-span.
  - Elsa the elephant has a tail length of 4 dm.
  - Clyde has a tail length of 6 dm.
  - Tweety's brother Bick weighs 12 kg.
  - Simba, sister of Shere-Khan, weighs 200 kg and has a tail length of 1.5 meter.
  - All animals live in a zoo.
  - ‘London Zoo’ is the name of a zoo.
  - ‘Kolmården’ is the name of another zoo.
  - For tigers it is important to keep information about colour, this is not important for elephants and penguins (all penguins are considered to have the same colour pattern, as is elephants).
Reification

Association 'cures' is M:M. If we want to address information about association 'cures' we have to reify the association, i.e. turn it into an entity. It we do not need to represent additional information about association 'cures' there is, however, no need to reify it!

Reification exercise:

Extend the conceptual schema so that it can represent that a person became a member in a certain club on a certain date!
Summery: modelling mechanisms

- Classification - from individuals in a domain to classes, including attributes and associations.
- Generalisation – inheritance
- Reification
- More? Yes!