



# Object Oriented Analysis and Design

## Chapter 02: Object Modeling using UML

Prof. Hichem Debbi

`hichem.debbi@gmail.com`

October 27, 2024

# The need for expressive notation

## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

In many domains, such as electronics and architecture, there exist unique notations for representing designs. In software industry, the need for standard notation has emerged as well.



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)



In many domains, such as electronics and architecture, there exist unique notations for representing designs. In software industry, the need for standard notation has emerged as well.

- A standard notation makes it possible for an analyst or developer to describe a scenario or formulate an architecture and then unambiguously communicate those decisions to others
- A good notation sets it free to concentrate on more advanced problems
- A good expression makes the task of verifying correctness of our understanding much easier.

# Modeling Methods before 1994 (OO Technology without UML)

## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- Rumbaugh 1991(Object-Modeling Technique, OMT):  
It was based on three main types of models: object, dynamic and functional. Many modeling elements were later incorporated into UML.
- Jacobson 1992 (Object-Oriented Software Engineering, OOSE):  
It is basically considered as a methodology that leads later to the creation of Rational Unified Process(RUP).
- Booch 1992(Booch method) :  
Their models serve as basics for many UML diagrams such as: Object diagram, class diagram, state transition diagram...



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- 1995: Booch, Rumbaugh, and Jacobson joined forces at Rational Software Corporation
- 1996: A visual modeling language was submitted to Object Management Group (OMG) as a first version of UML
- 1997 UML has been adopted as a standard by OMG (a consortium that creates and maintains standards for the computer industry)



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system.



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system.

UML serves as a helpful tool during the software development process despite the methodology in use.



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## UML Diagrams

- UML notation is given in the form of diagrams
- It is impossible to capture all the subtle details of a complex software system in just one large diagram
- The act of drawing a diagram does not constitute analysis or design. A diagram simply captures a statement of a system's behavior (for analysis), or the vision and details of an architecture (for design)



# UML Diagrams

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

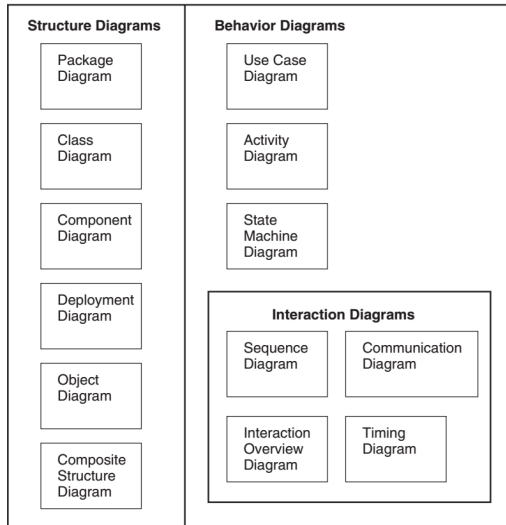


Figure: The Diagrams of the UML

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



| Analysis diagrams    | Design diagrams     |
|----------------------|---------------------|
| Use case diagrams    | Class diagrams      |
| Activity diagrams    | Object diagrams     |
| Class diagrams       | Component diagrams  |
| Interaction diagrams | Deployment diagrams |

Table: Analysis/Design UML diagrams

# Class diagram

In order to determine all the entities forming the systems, their roles and responsibilities as well as the way they collaborate we need a strong visualization diagram that captures all object-oriented notions, which is the class diagram. A class diagram is used to show the existence of classes and their relationships in the logical view of a system.

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



In order to determine all the entities forming the systems, their roles and responsibilities as well as the way they collaborate we need a strong visualization diagram that captures all object-oriented notions, which is the class diagram. A class diagram is used to show the existence of classes and their relationships in the logical view of a system.

## Definition

A class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

During analysis, we use class diagrams to indicate the common roles and responsibilities of the entities that provide the system's behavior. During design, we use class diagrams to capture the structure of the classes that form the system's architecture.

# The Class Notation

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- **Name:** A name of the class must be unique. Begins with capital letter with no spaces
- **Attributes:** are variables that their types must be identified
- **Methods :** the operations that the class could execute

| ClassName                                                                                       |
|-------------------------------------------------------------------------------------------------|
| +Attribute1: Type01<br>-Attribute2: Type02<br>#Attribute3: Type03                               |
| +Operation1(): Return Type 01<br>-Operation2(): Return Type 02<br>#Operation3(): Return Type 03 |

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- **public (+)** : Visible to any element that can see the class
- **Protected (#)** : Visible to other elements within the class and to subclasses
- **Private (-)** : Visible to other elements within the class
- **Internal (Package) (~)** : Visible to elements within the same module (Package)



# Class- Levels of Details

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

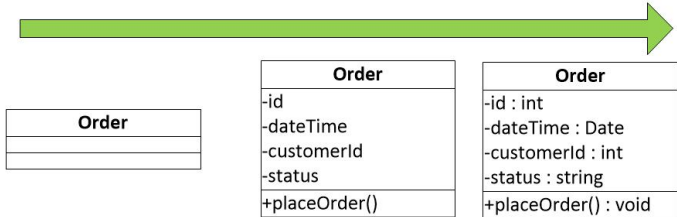
## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Different Levels of Details



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



- **Generalization** : Is an association that indicates that the subclass inherits the structure and behavior of its superclass
- **Association** : A binary association (with two ends) is normally represented as a line
- **Aggregation** : Is an association that indicates if a class is a collection or container of other classes
- **Composition** : When attempting to represent real-world whole-part relationships
- **Dependency** : When one class depends on another class but the contrary is not essentially true.





## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- **1:** Exactly one
- **\***: Unlimited number (zero or more)
- **0..\***: Zero or more
- **1..\***: One or more
- **0..1**: Zero or one
- **1..23**: Specified range



# Class Relationships

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

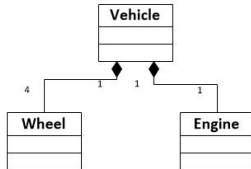
Object Constraint Language (OCL)



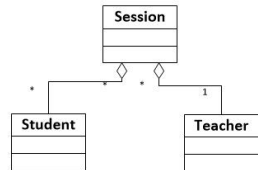
## Generalisation



## Composition



## Aggregation



# Association, Aggregation and Composition



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

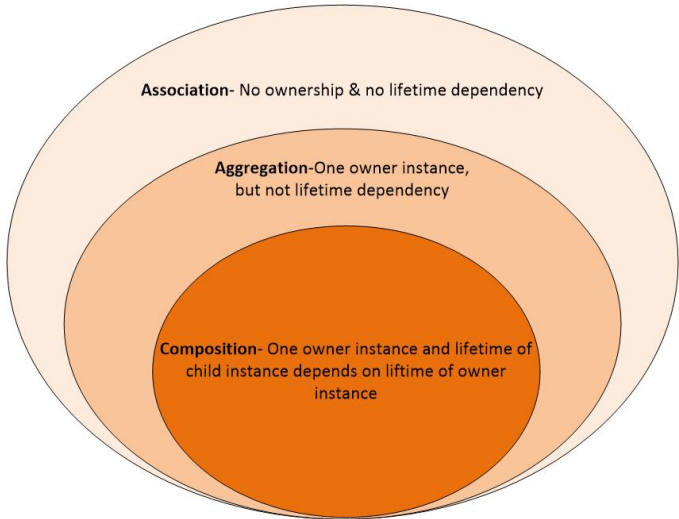
Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



# Association

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



```
class Student
{
    public string Name;
    private List<Teacher> _lstTeachers;

    public void SetTeachers(List<Teacher> teachers )
    {
        _lstTeachers = teachers;
    }
}

class Teacher
{
    public string Name;
    List<Student> _lstStudents = new List<Student>();
}

class Program
{
    static void Main(string[] args)
    {
        Student student = new Student();
        student.Name = "Student";
        List<Teacher> _lstTeachers = new List<Teacher>();
        Teacher teacher1= new Teacher()
        {
            Name = "teacher01"
        };
        Teacher teacher2 = new Teacher()
        {
            Name = "teacher02"
        };
        _lstTeachers.Add(teacher1);
        _lstTeachers.Add(teacher2);
        student.SetTeachers(_lstTeachers);
    }
}
```

# Aggregation

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

```
public class Department
{
    public string Name;
    public List<Teacher> LstTeachers = new List<Teacher>();
}
public class Teacher
{
    public string Name;
}
class Program
{
    static void Main(string[] args)
    {
        Department department = new Department();
        department.Name = "Informatique";
        List<Teacher> lstTeachers = new List<Teacher>();
        Teacher teacher1 = new Teacher()
        {
            Name = "teacher01"
        };
        Teacher teacher2 = new Teacher()
        {
            Name = "teacher02"
        };
        //Informatique department is the owner of teacher1
        //teacher2, but thier lifetime is independent of department instacne
        department.LstTeachers.Add(teacher1);
        department.LstTeachers.Add(teacher2);
    }
}
```

# Composition

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

```
public class University
{
    public string Name;
    private List<Department> _lstDepartments = new List<Department>();
    public List<Department> LstDepartments
    {
        get { return _lstDepartments; }
        set { _lstDepartments = value; }
    }
    public University()
    {
        Department department01 = new Department();
        Department department02 = new Department();
        _lstDepartments.Add(department01);
        _lstDepartments.Add(department02);
        // University here is the owner of dep1 and dep2,
        // their lifetime is dependent on university lifetime
    }
}

public class Department
{
    public string Name;
}

class Program
{
    static void Main(string[] args)
    {
        University university01 = new University { Name = "M'sila" };
    }
}
```



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

```
public class Customer
{
    public int CustomerId { get; set; }
    public String Name { get; set; }
}

public class Order
{
    public int OrderId { get; set; }
    public int CustomerId { get; set; }
    public DateTime OrderDate { get; set; }

    public Order(Customer customer)
    {
        this.CustomerId = customer.CustomerId;
    }
}
```

# Abstract Class & Interface

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

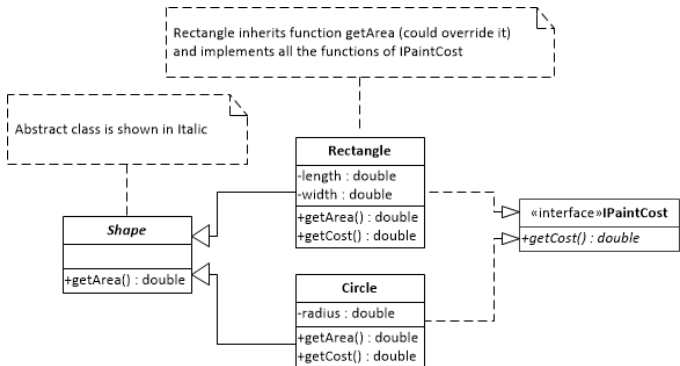
## Specification

Formal and Informal

Specification

Object Constraint

Language (OCL)





# Class Diagram - example

## Introduction

## UML Diagrams

### Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

### Stereotypes

### Tagged Values

### Constraints

## Meta-models and UML Profiles

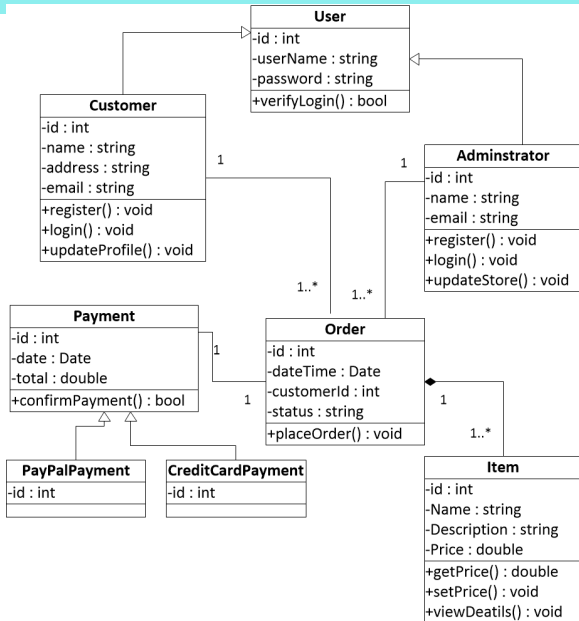
### Metamodels

### UML Profiles

## Specification

### Formal and Informal Specification

### Object Constraint Language (OCL)



The UML consists of four common mechanisms that apply throughout the language that help to meet our needs in object modeling.

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

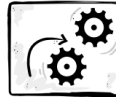
Metamodels

UML Profiles

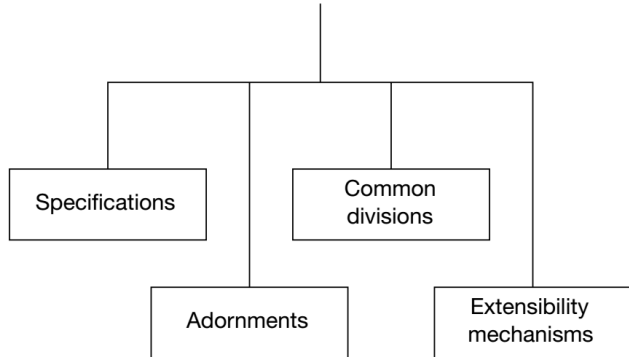
## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

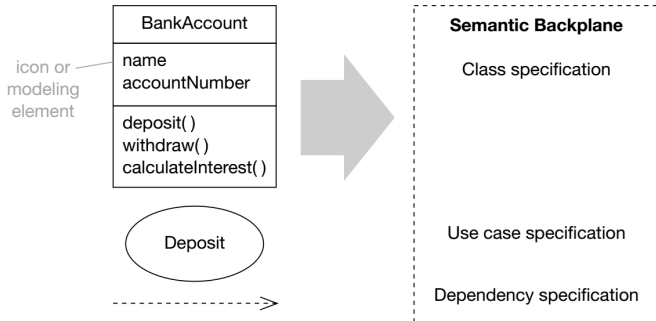


## Common mechanisms



# Specification

- UML consists of two dimensions , graphical (boxes, arrows, ....) and textual dimension.
- Specification represents the textual dimension that helps us to describe the semantics of the modeling element.
- Meaningful model should consist of meaningful specification.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- UML diagrams consist of basic notations elements and such adornments added to them.
- Adornments are textual/graphical items added to the basic elements to give much detail



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



- UML diagrams consist of basic notations elements and such adornments added to them.
- Adornments are textual/graphical items added to the basic elements to give much detail

Adornments highlighting important details



# Common divisions

- UML consists of two common divisions: classifier and instance, and interface and implementation
- Classifier and instance distinguish between abstraction and manifestation (e.g. class/object)
- Interface introduces a contract, and an implementation implements the realization of this contract

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



# Common divisions

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

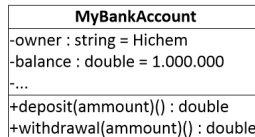
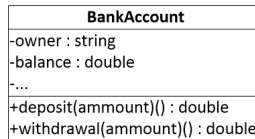
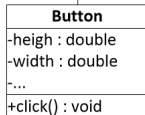
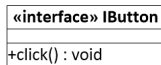
## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



- UML consists of two common divisions: classifier and instance, and interface and implementation
- Classifier and instance distinguish between abstraction and manifestation (e.g. class/object)
- Interface introduces a contract, and an implementation implements the realization of this contract







## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- **Problem:** With UML it's not simply not possible to satisfy everyone's needs about modeling



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- **Problem:** With UML it's not simply not possible to satisfy everyone's needs about modeling
- **Solution:** Extensibility mechanisms are invented to extend UML notation through adding new building blocks, creating new properties and specifying new semantics in order to make UML suitable for any application domain



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- **Stereotypes:** A stereotype allows us to define a new UML modeling element based on an existing one



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- **Stereotypes:** A stereotype allows us to define a new UML modeling element based on an existing one
- **Tagged values:** Provide a way of extending an element's specification by allowing us to add new, ad hoc information to it.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- **Stereotypes:** A stereotype allows us to define a new UML modeling element based on an existing one
- **Tagged values:** Provide a way of extending an element's specification by allowing us to add new, ad hoc information to it.
- **Constraints:** extend the semantics of an element by allowing us to add new rules





## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Definition

Stereotypes allow designers to extend the vocabulary of UML in order to create new model elements, derived from existing ones, but that have specific properties that are suitable for a particular problem domain or otherwise specialized usage.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Definition

Stereotypes allow designers to extend the vocabulary of UML in order to create new model elements, derived from existing ones, but that have specific properties that are suitable for a particular problem domain or otherwise specialized usage.

## Example

When modeling a network you might need to have symbols for representing routers and hubs. By using stereotyped nodes you can make these things appear as primitive building blocks

# Stereotypes

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

### Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

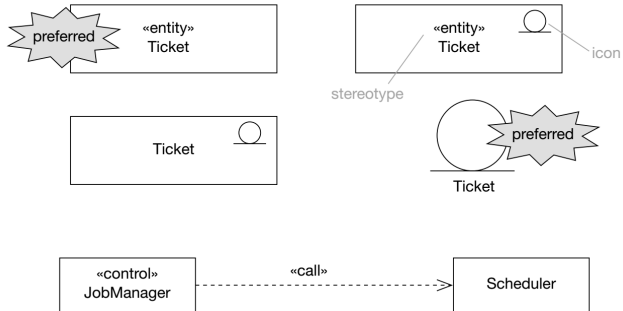
## Specification

Formal and Informal

Specification

Object Constraint

Language (OCL)





## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

### Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

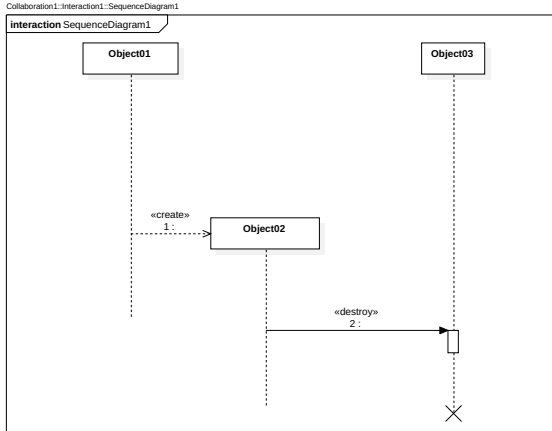
Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



# Stereotypes

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

### Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

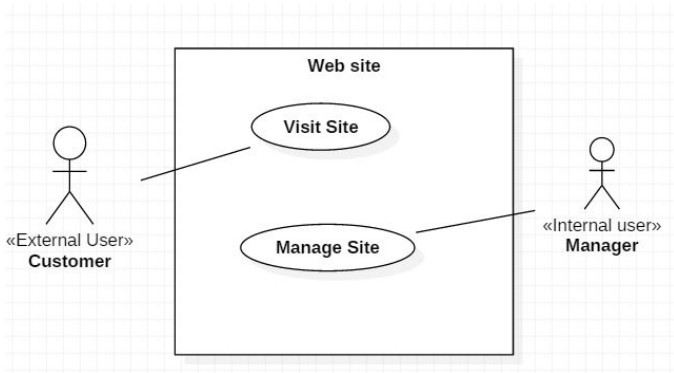
## Specification

Formal and Informal

Specification

Object Constraint

Language (OCL)



# Stereotypes classes

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

### Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

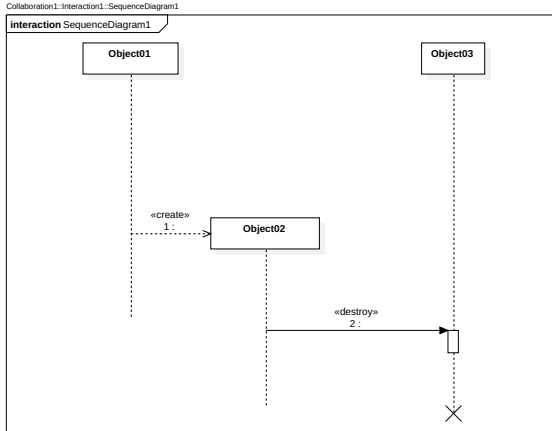
Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)





## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Definition

UML allows you to add your own properties to modeling elements by using tagged values. So simply, a tagged value is just a keyword which can have a value attached. They usually indicate properties of new modeling elements defined by a stereotype.

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Definition

UML allows you to add your own properties to modeling elements by using tagged values. So simply, a tagged value is just a keyword which can have a value attached. They usually indicate properties of new modeling elements defined by a stereotype.

## Example

The syntax for tagged values is shown as: `tag1 = value1, tag2 = value2, . . . , tagN = valueN`

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- Can be defined for existing UML elements
- Can be defined for individual as well as multiple stereotypes
- they are not the same as class attributes
- The tag and the value are usually encoded as strings
- They are applied to the element, not its instance



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

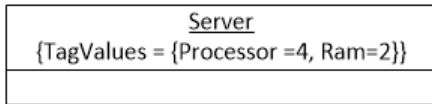
Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Definition

A constraint is simply a text string in braces `{ }` that specifies some condition, restriction or assertion about the modeling element that must be maintained as true. Constraints are usually defined on elements of class diagrams.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Definition

A constraint is simply a text string in braces `{ }` that specifies some condition, restriction or assertion about the modeling element that must be maintained as true. Constraints are usually defined on elements of class diagrams.

## Example

The syntax for constraint is shown as: `Name1 = BooleanExpression1, Name2 = BooleanExpression2`  
Example : `Owner->notEmpty()` and `Balance>=0`

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- The placement of a constraint in a visual diagram depends on the number of diagram elements affected by the constraint



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- The placement of a constraint in a visual diagram depends on the number of diagram elements affected by the constraint
- Preconditions and postconditions are examples of constraints



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- The placement of a constraint in a visual diagram depends on the number of diagram elements affected by the constraint
- Preconditions and postconditions are examples of constraints
- A constraint can be represented in a natural language (textual) or in a more formal language such as a programming language or the UML Object Constraint Language (OCL).



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- The placement of a constraint in a visual diagram depends on the number of diagram elements affected by the constraint
- Preconditions and postconditions are examples of constraints
- A constraint can be represented in a natural language (textual) or in a more formal language such as a programming language or the UML Object Constraint Language (OCL).
- Formal language is desirable to describe constraints



# Constraints

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

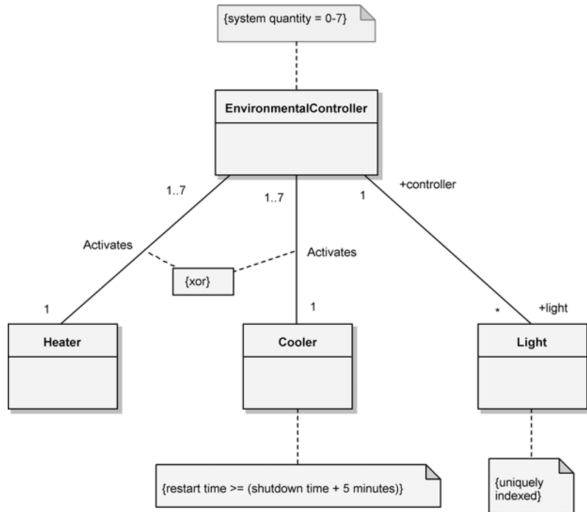
Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

A metamodel is a model of a model, we call the process of generating such metamodels, a metamodeling. While models describe real and observable issues, metamodels describe models.



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)



A metamodel is a model of a model, we call the process of generating such metamodels, a metamodeling. While models describe real and observable issues, metamodels describe models.

- Meta-Object Facility (MOF) is an Object Management Group (OMG) standard that helps to create meta-models



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)



A metamodel is a model of a model, we call the process of generating such metamodels, a metamodeling. While models describe real and observable issues, meatmodels describe models.

- Meta-Object Facility (MOF) is an Object Management Group (OMG) standard that helps tp create meta-models
- UML meatmodels are defined using MOF

## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)



A metamodel is a model of a model, we call the process of generating such metamodels, a metamodeling. While models describe real and observable issues, meatmodels describe models.

- Meta-Object Facility (MOF) is an Object Management Group (OMG) standard that helps tp create meta-models
- UML meatmodels are defined using MOF
- Typical metamodels proposed by OMG are UML, SysML, SPEM or CWM

## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

A metamodel is a model of a model, we call the process of generating such metamodels, a metamodeling. While models describe real and observable issues, metamodels describe models.

- Meta-Object Facility (MOF) is an Object Management Group (OMG) standard that helps to create meta-models
- UML metamodels are defined using MOF
- Typical metamodels proposed by OMG are UML, SysML, SPEM or CWM
- A metamodel describes exactly the semantics of the metamodel elements



# Metamodel example

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

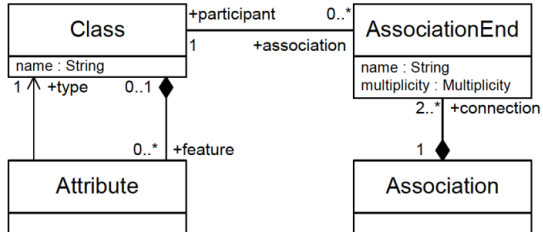
UML Profiles

## Specification

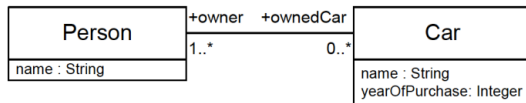
Formal and Informal Specification

Object Constraint Language (OCL)

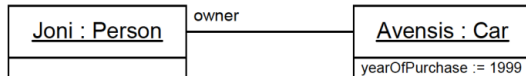
### M2. Metamodel



### M1. Model



### M0. Instance



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Definition

UML profile combines all the previous notions (stereotypes, tagged values and constraints) into one extension collection in order to customize UML models for particular domains (Systems engineering, business process, ...) and platforms (JEE, .Net framework, ....)

## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Definition

UML profile combines all the previous notions (stereotypes, tagged values and constraints) into one extension collection in order to customize UML models for particular domains (Systems engineering, business process, ...) and platforms (JEE, .Net framework, ....)

## Examples

- SysML: (OMG)-standardized profile of Unified Modeling Language which is used for system engineering applications
- SOA, CORBA and EJB architectures
- XSD schemas
- ....



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- UML models did not respond to every detail of all application domains



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- UML models did not respond to every detail of all application domains
- Profiles represent the standard mean to extend UML





## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- UML models did not respond to every detail of all application domains
- Profiles represent the standard mean to extend UML
- A profile defines one or more UML stereotypes.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- UML models did not respond to every detail of all application domains
- Profiles represent the standard mean to extend UML
- A profile defines one or more UML stereotypes.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- UML models did not respond to every detail of all application domains
- Profiles represent the standard mean to extend UML
- A profile defines one or more UML stereotypes.
- Profile uses the same notation as a package



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- UML models did not respond to every detail of all application domains
- Profiles represent the standard mean to extend UML
- A profile defines one or more UML stereotypes.
- Profile uses the same notation as a package
- A profile can define classes, stereotypes, data types, primitive types, enumerations.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

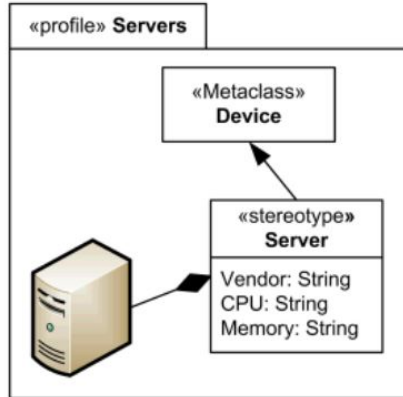


Figure: Simple profile: [uml-diagrams.org](http://uml-diagrams.org)

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

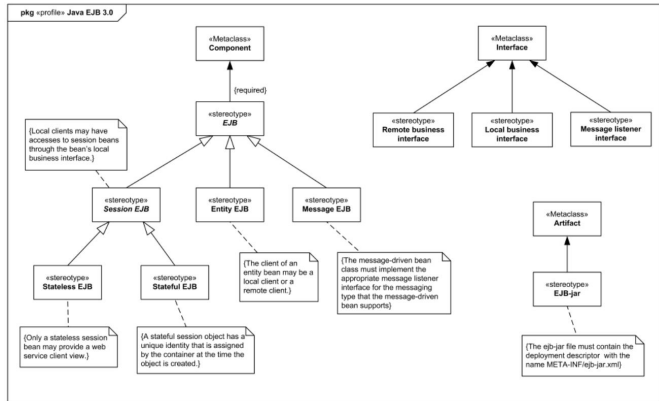


Figure: Java EJB 3.0 Profile : [uml-diagrams.org](http://uml-diagrams.org)

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal

Specification

Object Constraint

Language (OCL)

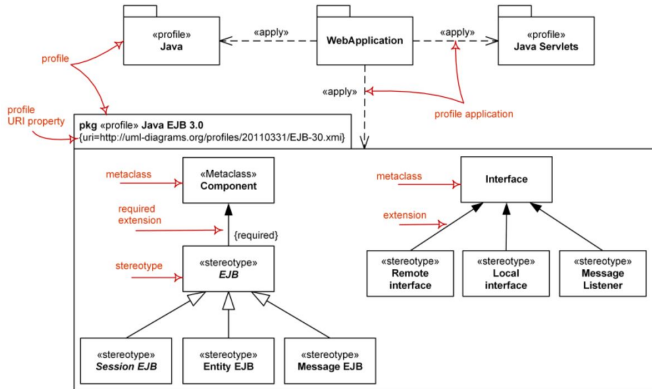


Figure: Using EJB profile: [uml-diagrams.org](http://uml-diagrams.org)

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Definition

A specification is a complete, unambiguous description of the required behavior of a piece of software. The piece of software in question could be an entire system, a subsystem, a layer, a class or a function.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Goals behind using Specification

- Remove the ambiguity remaining after earlier phases
- Improve our understanding of the implementation
- Increase our confidence that the system will work
- Help us debug our software
- Help us test our software
- Provide better documentation



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Some Formal Specification Languages

- Z, B and VDM
- Petri Nets
- Lotos
- PROMELA

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Example-VDM

*squareRoot*( $x : R$ )  $y : R$

**pre**  $x \geq 0$

**post**  $(y^2 = x) \wedge (y \geq 0)$

---

$R$  : Real numbers

**pre** : precondition

**post** : postcondition

$(y^2 = x) \wedge (y \geq 0)$  : logical expression called assertion



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint

Language (OCL)



## Definition

Informal Specification is simply not formal. Informal specification are not mathematical-based but rather are textual.

A good example of informal specification is commenting the methods of a program, in way that they could be understood by different programmers.

## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Definition

Informal Specification is simply not formal. Informal specifications are not mathematical-based but rather are textual.

A good example of informal specification is commenting the methods of a program, in way that they could be understood by different programmers.

## Example

```
/*  
Returns the positive square root of x.  
Preconditions:  $x \geq 0$ .  
*/  
float squareRoot(float x);
```

## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

Object-oriented specification languages such as OCL have been designed specifically for object-oriented software development. they assert objects in terms of their messages and attributes.



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

Object-oriented specification languages such as OCL have been designed specifically for object-oriented software development. they assert objects in terms of their messages and attributes. An object-oriented specification language must allow us to assert the following

- when a message can legally be sent to an object
- the valid parameters for each message
- the effect a message has on the receiving object
- the valid replies from each message
- the conditions that are always met by the object



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- OCL is a declarative language that supplements UML models by providing formal expressions that help better to describe the design of the system. It was developed at IBM, and now is a part of the UML standard.





## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- OCL is a declarative language that supplements UML models by providing formal expressions that help better to describe the design of the system. It was developed at IBM, and now is a part of the UML standard.
- It may be used as well with Meta-Object Facility (MOF) model or meta-model.



## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- OCL is a declarative language that supplements UML models by providing formal expressions that help better to describe the design of the system. It was developed at IBM, and now is a part of the UML standard.
- It may be used as well with Meta-Object Facility (MOF) model or meta-model.
- It represents a powerful mechanism that allow us to describe such constraints that cannot be expressed by diagrammatic notation.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- OCL is a declarative language that supplements UML models by providing formal expressions that help better to describe the design of the system. It was developed at IBM, and now is a part of the UML standard.
- It may be used as well with Meta-Object Facility (MOF) model or meta-model.
- It represents a powerful mechanism that allow us to describe such constraints that cannot be expressed by diagrammatic notation.
- Constraints may be added as notes on the UML model, or in separate file.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- **Invariants:** An invariant is a constraint that must be met by all instances of the class.

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- **Invariants:** An invariant is a constraint that must be met by all instances of the class.
- **Precondition for operations:** Is a condition that must be always verified before executing the operation.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)

- **Invariants:** An invariant is a constraint that must be met by all instances of the class.
- **Precondition for operations:** Is a condition that must be always verified before executing the operation.
- **Postconditions for operations:** Is a condition that must be always verified after executing the operation.  
Precondition and postconditions assert the call of object's methods like contracts, the caller should fulfill precondition, and the object called verify asserts the postcondition.



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



- **Invariants:** An invariant is a constraint that must be met by all instances of the class.
- **Precondition for operations:** Is a condition that must be always verified before executing the operation.
- **Postconditions for operations:** Is a condition that must be always verified after executing the operation.  
Precondition and postconditions assert the call of object's methods like contracts, the caller should fulfill precondition, and the object called verify asserts the postcondition.
- We use context keyword `do` to define which class we are dealing with.
- *self* is used to denote the current context or class.

# A class diagram

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

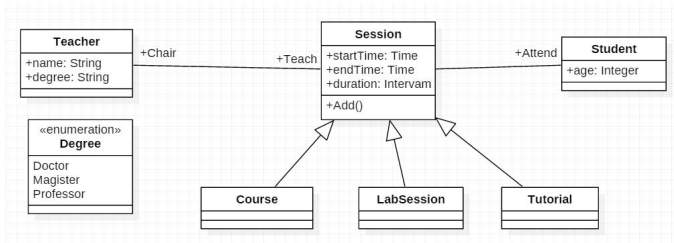
## Specification

Formal and Informal

Specification

Object Constraint

Language (OCL)





## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Example

```
context Session
inv : self.duration=90
```

## Introduction

### UML Diagrams

Class Diagram

### Common Mechanisms

### Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

### Meta-models and UML Profiles

Metamodels

UML Profiles

### Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Example

```
context Session
inv : self.duration=90
```

## Example

```
context Teacher
self.name has type String
```

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



### Example

```
context Session
inv : self.duration=90
```

### Example

```
context Teacher
self.name has type String
```

### Example

```
context Student
inv : self.age>=18
```

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Example

```
context Container::  
  add(o:Object)  
  pre: (o <> null) and not contains(o)  
  post: contains(o) and (size = size@pre + 1)  
context Container::  
  inv: size  $\geq$  0
```

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Session-Student Example

```
context Session::  
add(s: Student)  
pre: (s <> null) and not contains(s)  
post: contains(s) and (size = size@pre + 1)  
context Session::  
inv: size  $\geq$  0  
inv: size  $\leq$  30
```



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Relationships Constraints Example

context Teacher

inv: self.Teach.Course→size <= 2

inv: self.Teach.LabSession→size <= 4

inv: self.Teach.Tutorial→size >= 1

---



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Relationships Constraints Example

context Teacher

inv: self.Teach.Course→size <= 2

inv: self.Teach.LabSession→size <= 4

inv: self.Teach.Tutorial→size >= 1

---

context Course

inv: self.Chair→forAll(t | t.degree <> #magister)

---

## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



## Relationships Constraints Example

context Teacher

inv: self.Teach.Course→size <= 2

inv: self.Teach.LabSession→size <= 4

inv: self.Teach.Tutorial→size >= 1

---

context Course

inv: self.Chair→forAll(t | t.degree <> #magister)

---

context Teacher

inv: Teacher.AllInstances→exists(t | t.degree = #magister)



## Introduction

## UML Diagrams

Class Diagram

## Common Mechanisms

## Extensibility Mechanisms in UML

Stereotypes

Tagged Values

Constraints

## Meta-models and UML Profiles

Metamodels

UML Profiles

## Specification

Formal and Informal Specification

Object Constraint Language (OCL)



- 1 Grady Booch et al, *Object-oriented Analysis and Design with Applications, 3rd edition* , Addison-Wesley Professional, 2007
- 2 Mike O'Docherty, *Object-Oriented Analysis and Design Understanding System Development with UML 2.0* , WILEY, 2005
- 3 Jim Arlow and Ila Neustadt, *UML and the Unified Process Practical Object-Oriented Analysis and Design*, Pearson Education Limited 2002
- 4 Torben Amtoft, Lectures CIS 771: Software Specifications