

Unified Modeling Language (UML) and Its Applications in BIM

Prepared for NIST Project by
Weichao Wang
UNC Charlotte

Learning Objectives

- Understand the reasons that we need a cross-platform, uniform-language representation model for BIM systems
- Learn the basics of UML
- Learn the basics of the UML models for BIM
- Introduction to frequently used UML tools



Outline

- Challenges faced by modern BIM
- What is UML and why we use UML
- Examples of UML models for BIM
- UML Modeling tools



Challenges faced by BIM

- Representation
 - Image based
 - Text based
 - Visualization/VR/AR based
 - User friendly

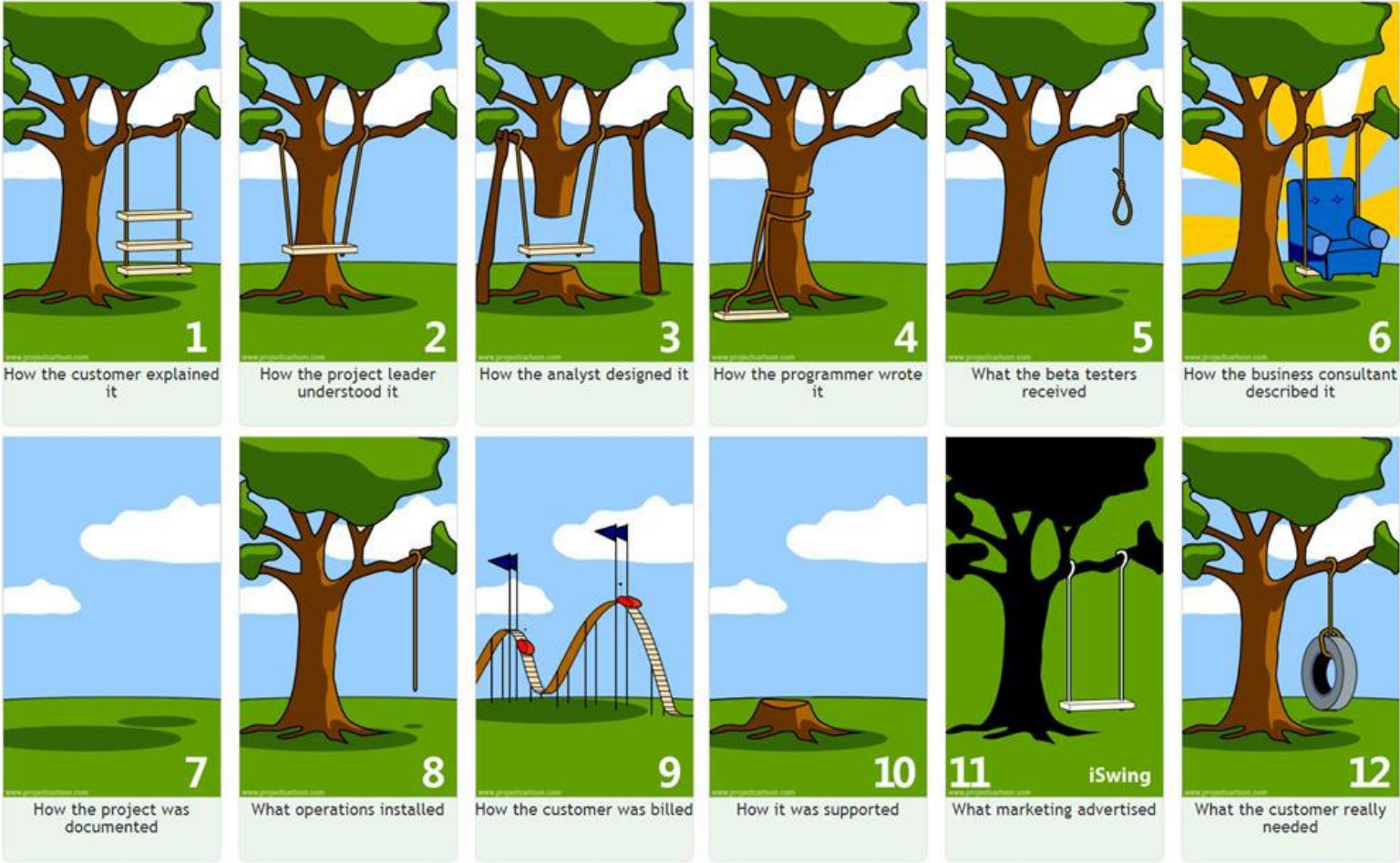
```
<auc:ExteriorFloors>  
  <auc:ExteriorFloor>  
    <auc:ExteriorFloorID IDref="ExteriorFloorSystemType-69909851364720">  
    </auc:ExteriorFloorID>  
  </auc:ExteriorFloor>  
</auc:ExteriorFloors>  
<auc:Foundations>  
  <auc:Foundation>  
    <auc:FoundationID IDref="FoundationSystemType-69909855891160">  
    </auc:FoundationID>  
  </auc:Foundation>  
</auc:Foundations>
```



This figure shows one level of an academic building at UNC Charlotte. Very hard for non-constructional engineering persons to fully understand the building structure.

Challenges faced by BIM

- Communication: text or even imaged based description could be misleading



Challenges faced by BIM

- Sharing/Reuse
 - If a good component has been designed, why re-invent the wheel?
 - Sharing across different software platforms and tools
- Validation
 - How can we verify that all designs are in compliance with the national and state regulations and rules
 - When the designer introduces some changes, we need to verify that:
 - The changes can be done
 - Still in compliance with the regulations



Student Exercise: Need of UML

- Design description and sharing
 - Students form 3-person groups
 - One member will describe a one-story residence house floor plan, while the other two members will draw the floor plan based on the description. The drawings cannot be shown to member 1 during the procedure.
 - Share the drawings with the whole group, discuss how we can reduce misunderstanding during the procedures



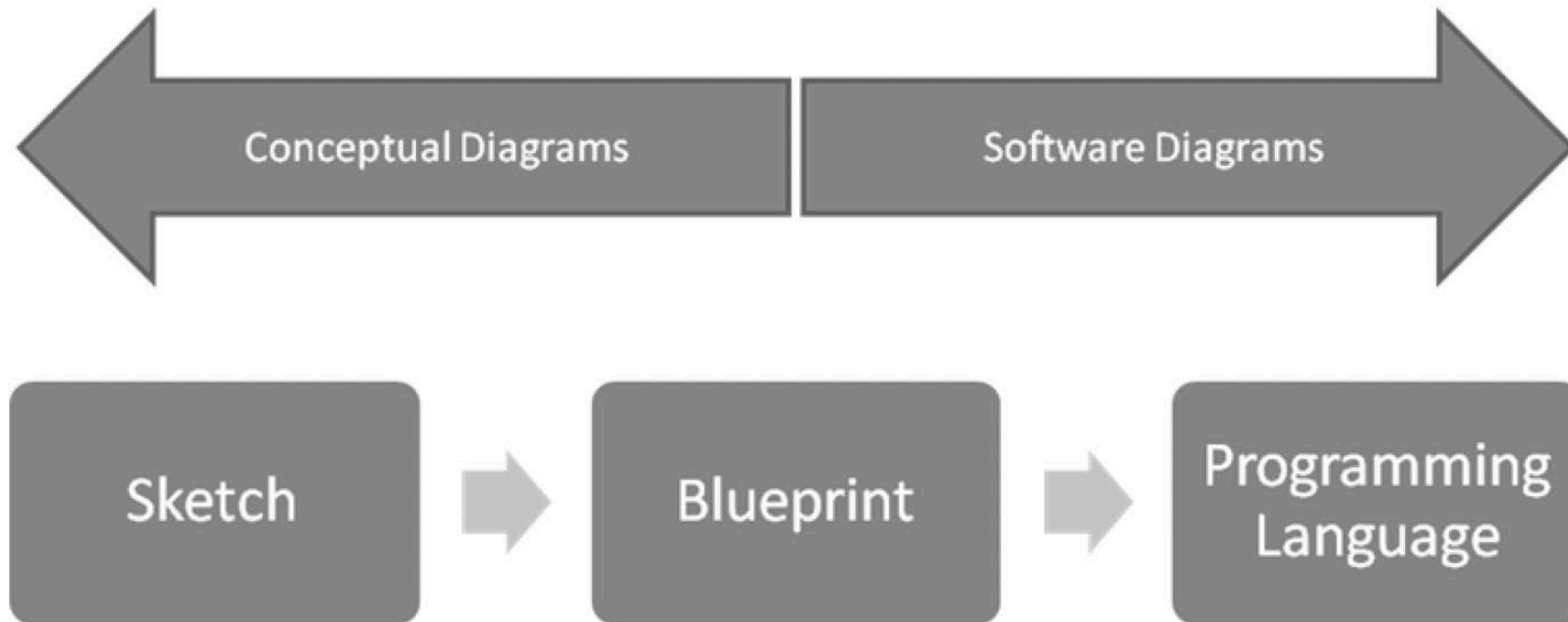
What is UML and Why we use UML?

- UML → “Unified Modeling Language”
- Modeling: Describing a system at a high level of abstraction
- UML is a standardized modeling language that is used for object-oriented analysis
- Unified: UML has become a world standard
 - Object Management Group (OMG): www.omg.org



What is UML and Why we use UML?

- Support multiple types of UML diagrams and these diagrams can be used to accomplish numerous purposes ranging from providing a rough picture of a concept to providing a precise representation of an actual implementation.



What is UML and Why we use UML?

- It is an industry-standard graphical language for specifying, visualizing, constructing, and documenting the artifacts of systems
- The UML uses mostly graphical notations to express the OO analysis and design of projects.
- Simplifies the complex process of design



What is UML and Why we use UML?

- Use graphical notation: more clearly than natural language (imprecise) and code (too detailed).
- Help acquire an overall view of a system.
- UML is *not* dependent on any one language or technology.
- UML moves us from fragmentation to standardization.



How to use UML diagrams to design object-oriented systems?

- Types of UML Diagrams:
 - Use Case Diagram
 - Class Diagram
 - Sequence Diagram
 - Collaboration Diagram
 - State Diagram
- This is only a subset of diagrams ... but are most widely used
- Class diagrams are a type of structure diagrams that are used to show the possible ways that things can go together.



Why do we need Object Oriented Design

- Problems of Procedure based Design
 - Non-standard interfaces and operations
 - No-inheritance
 - Relationship between components
- Objects are used to overcome such challenges



Class diagram

- A class diagram depicts classes and their interrelationships
- Used for describing structure and behavior in the use cases
- Provide a conceptual model of the system in terms of entities and their relationships
- Detailed class diagrams are used for developers

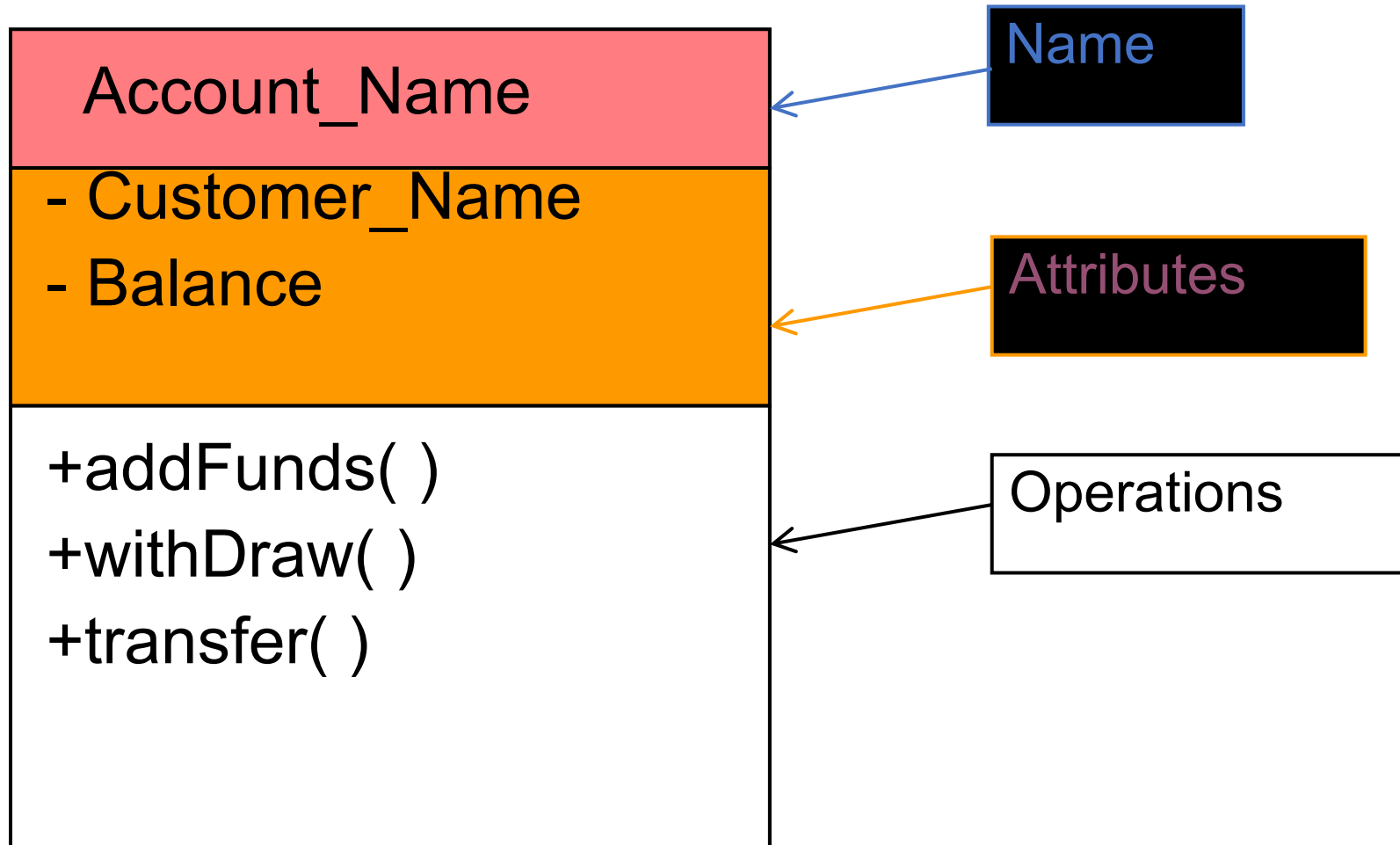


Class diagram

- Each class is represented by a rectangle subdivided into three compartments
 - Name
 - Attributes
 - Operations
- Modifiers are used to indicate visibility of attributes and operations
 - ‘+’ is used to denote *Public* visibility (everyone)
 - ‘#’ is used to denote *Protected* visibility (friends and derived)
 - ‘-’ is used to denote *Private* visibility (no one)
 - “~” means that the scope is package.
- By default, attributes are hidden and operations are visible
 - Avoid manipulation
 - Enable verification



Class diagram

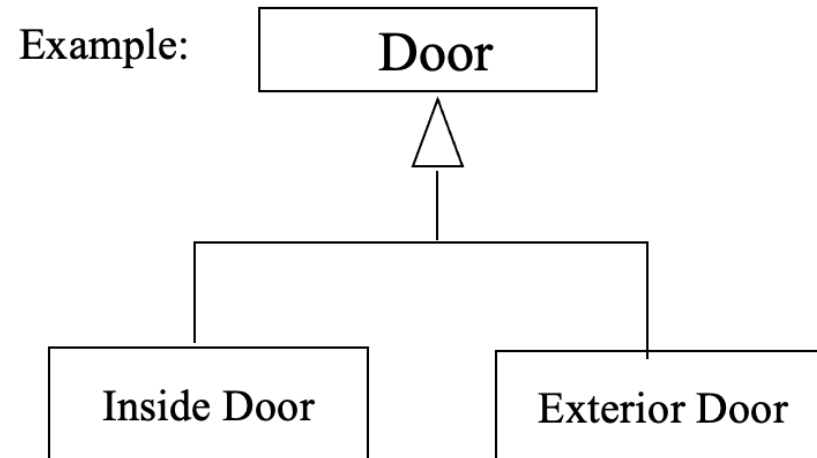
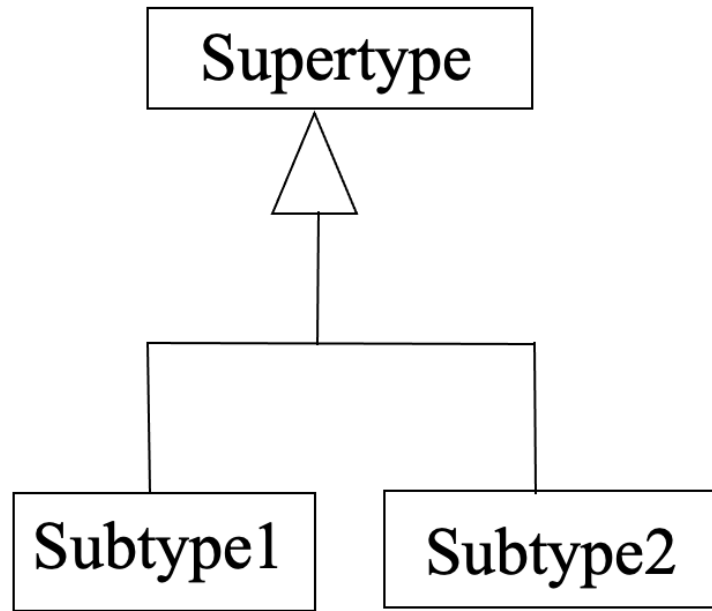


Class diagram

- Example Relationships
 - Generalization (parent-child relationship)
 - Association (Windows installed in a room)
 - Dependency (Cart and products in an e-commerce transaction)
- Associations can be further classified as
 - Aggregation
 - Composition
 - ---
- In some cases, child classes need to further constrain those inherited relationships, which is called “redefines” construct



OO Relationships: Generalization



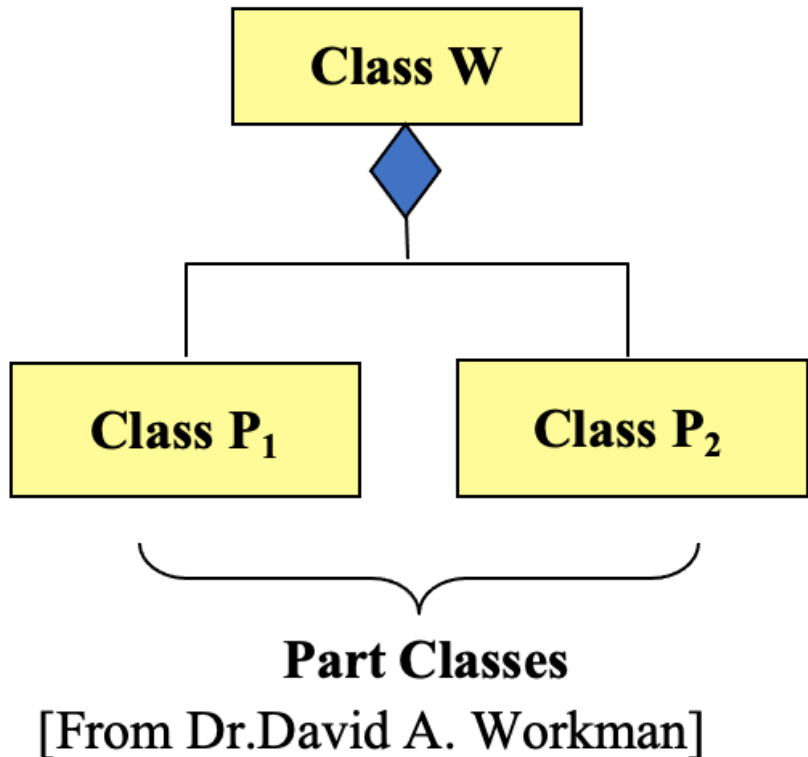
- Inheritance is a required feature of object orientation
- Generalization expresses a parent/child relationship among related classes.
- Used for abstracting details in several layers

OO Relationships: Association

- Represent relationship between instances of classes
 - Windows installed in a room
 - Rooms have windows
 - Rooms have doors
 - Etc.
- Association has two ends
 - Role names
 - Multiplicity (e.g. One room can have multiple windows)
 - Navigability (unidirectional, bidirectional)

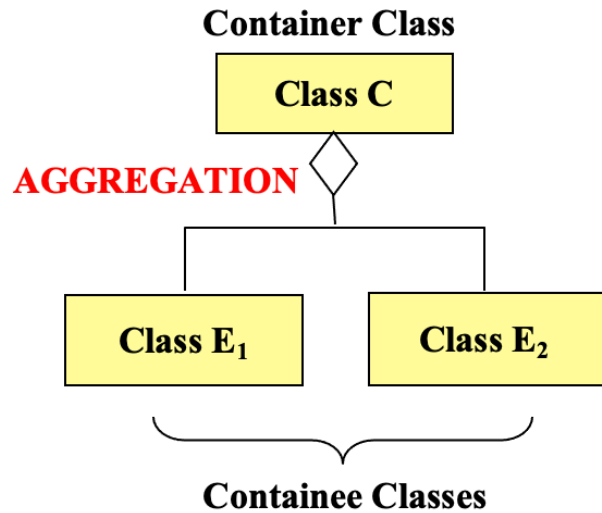


OO Relationships: Composition



- Association
 - Models the part–whole relationship
- Composition
 - Also models the part–whole relationship but, in addition, every part may belong to only one whole, and if the whole is deleted, so are the parts
- Example:
 - A number of different chess boards: Each square belongs to only one board. If a chess board is thrown away, all 64 squares on that board go as well.

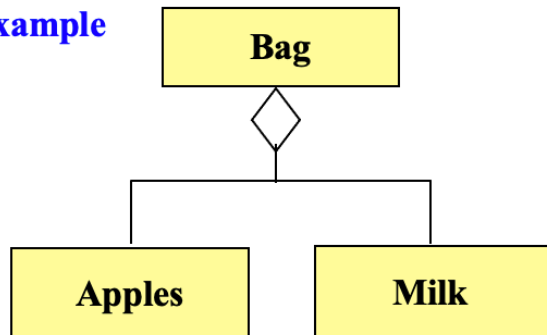
OO Relationships: Aggregation



Aggregation:

- expresses a relationship among instances of related classes. It is a specific kind of Container-Containee relationship.
- express a more informal relationship than composition expresses.
- Aggregation is appropriate when Container and Containees have no special access privileges to each other.

Example



[From Dr. David A. Workman]



Aggregation vs. Composition

- Composition is really a strong form of association
 - components have only one owner
 - components cannot exist independent of their owner
 - components live or die with their owner
 - e.g. Each car has an engine that can not be shared with other cars.
- Aggregations
 - may form "part of" the association, but may not be essential to it. They may also exist independent of the aggregate. e.g. Apples may exist independent of the bag.



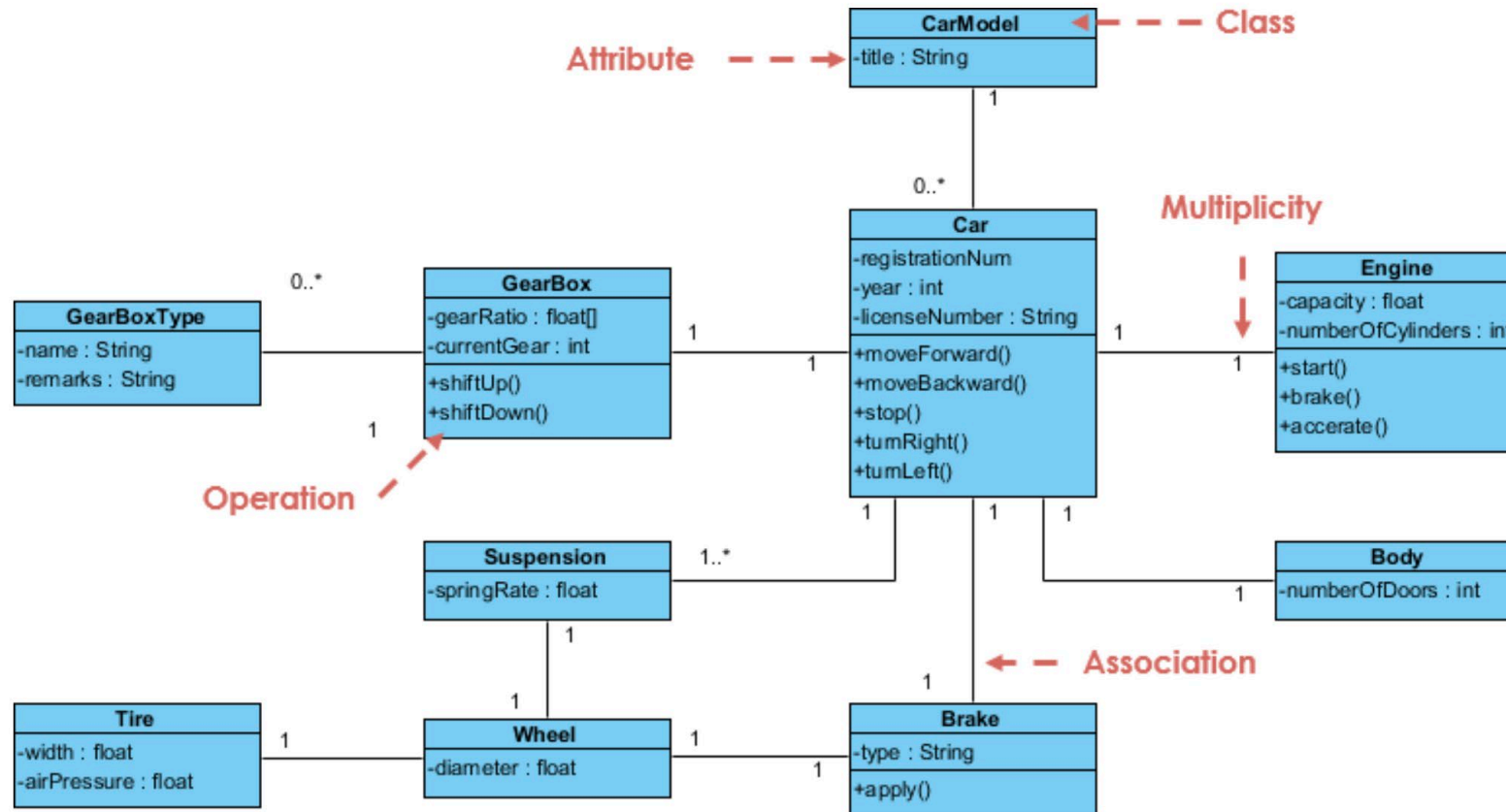
Generalizations, Interfaces, and Packages

- Interfaces are similar to classes except that they do not have any attributes. They only specify a behavior that is to be implemented. Interfaces look like a class symbol with the word “<<interface>>” added.
- Packages are similar to folders in Windows. They allow you to logically group things. In class diagrams, X::Y means that Class Y is contained in package X.



Student Exercise: UML Example

- In the same 3-person group, discuss the following UML model and the relationship between the components.



UML for BIM Systems

- Modeling BIM Object Lifecycle:
 - UML state diagrams can model the lifecycle of BIM objects, showing how they transition through different stages.
 - help facility management teams understand how to interact with building elements during their lifespan
- Integrating BIM with Other Systems:
 - UML can be used to model the integration of BIM with other systems.
 - This integration can streamline workflows, improve efficiency, and enhance decision-making



UML for BIM Systems (continued)

- Modeling Geospatial Information:
 - UML can be used to model geospatial information within BIM, especially when integrating with GIS (Geographic Information Systems).
 - This enables better data sharing and interoperability between BIM and GIS, facilitating tasks like urban planning and infrastructure management
- Data Modeling and Interoperability:
 - UML diagrams can represent the semantic data model for solid building models, which can facilitate data integration between BIM and GIS.
 - This allows for more detailed and accurate representation of building components and their relationships



Example of UML for BIM: Work Orders

Table 1 Example work orders depicting the information in relation to problems and work performed

	Work order 1 (WO1)	Work order 2 (WO2)
Name	Replace a leaky pipe	Repair the supply fan in an AHU
Request Date	05/20/2011	06/01/2011
Finish Date	05/26/2011	06/03/2011
Type	Replace	Repair
Location	Room 105	Mechanical room A11
Component(s)	Pipe, dry wall	Supply fan
Shop/Trade	Plumber, Carpenter	Electrician
Description	Pipe's leaky section was replaced and the wall cut to access the pipe was replaced.	The supply fan in AHU 2 has a belt-relaxation fault and was repaired.
Labor Hour	13:00	6:30
Labor Cost	\$530.18	\$162.50
Material Cost	\$55.00	\$198.99
Total Cost	\$585.18	\$361.49



Example of UML for BIM: Work Orders



Figure 1 UML diagram of the instances of work orders and building elements for illustrating the implemented linkage between work order and BIM



Example 2: Building Condition Assessment

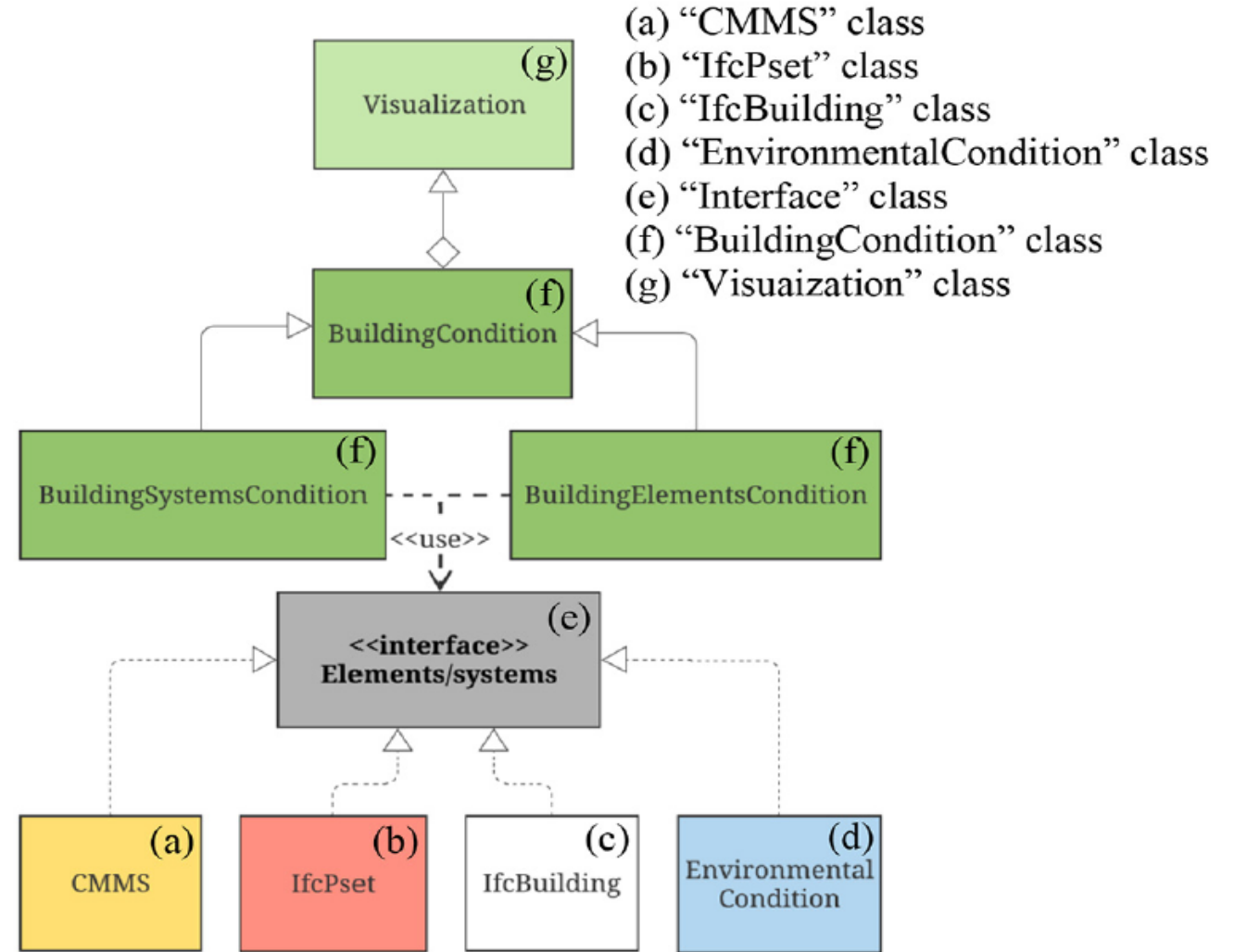


Fig. 2. Conceptual design of the UML diagram for the proposed data model.

Example 2: Building Condition Assessment

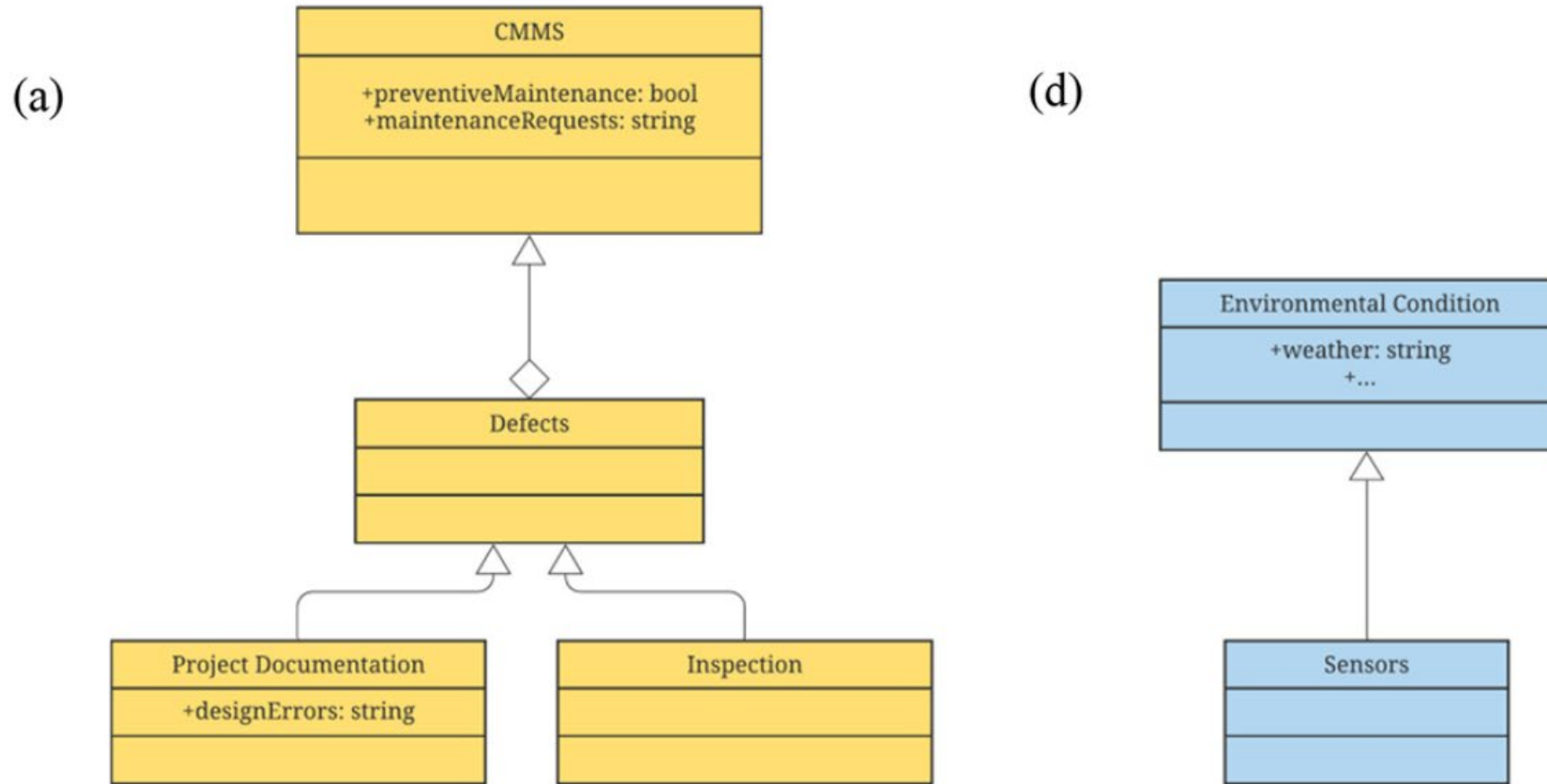


Fig. 3. UML diagram of (a) “CMMS” class and (d) “EnvironmentalCondition” class.

Student Exercise: UML Description of BIM Work Order

- In the same 3-person group, generate a UML model for a BIM work order for “replace a faucet in Room 333 of the Woodward Hall”. Please identify the related BIM packages as well. Please refer to the examples above.



What UML Modeling tools we use today?

- List of UML tools http://en.wikipedia.org/wiki/List_of_UML_tools
- ArgoUML: <https://github.com/argouml-tigris-org/argouml>
- Rational ClearQuest (<https://www.ibm.com/docs/en/rational-clearquest>) by IBM
- Visual Paradigm Software (<https://www.visual-paradigm.com/>)
- Visio by Microsoft (<https://www.microsoft.com/en-us/microsoft-365/visio/flowchart-software>)



Summary

- We learn the challenges that BIM systems face and the reasons that we need a more generic modeling language
- We learn the basics of the UML language
- We show examples of the UML usage in BIM
- We introduce the frequently used modeling tools



Assignment

- Please see the assignment of the UML module



Acknowledgement

- *Object-Oriented and Classical Software Engineering*, Sixth Edition, WCB/McGraw-Hill, 2005 Stephen R. Schach
- UML resource page <http://www.uml.org/>
- Class slides from UCF software engineering courses
- This work was supported by the Standards Coordination Office Curricula Development Cooperative Agreement Program of National Institute of Standards and Technology.

