Copyright Warning

ViSLAB
High-Performance Visualization, Communication, Computing

Object Oriented Design

Week 1-2

OO == Easy Dev.

- When did it start?
  - Two Norwegian (Ole-Johan Dahl and Kristen Nygaard)
  - Simula 67 (1967) : had pretty much all important OO concept.

OO == Soft Dev Techs

- Why would you develop software using Object Oriented approach?

- To make software development process easy.
  - Many people perceive that OO dev is difficult... why?

What are in OO?

- Programming Language (C++, Java, Objective-C)
- Diagram representation (for requirements, design, etc.)
- Reusable software tools/modules (libraries, frameworks, etc.)
- Design Know-how (design pattern)
- Project Management (task analysis, requirement def, etc.)

What are in OO?

- All look different technologies, but
- All for soft development, and
- All for make the soft. dev. easy (agile, safe, accurate, etc.)

OO != magic

- What young expert OO programers would say to make OO sound like “magic”?

- Again, OO is to make your soft dev process easy!

Soft. Dev == Difficult

- Software is a set of instructions for a computer to execute.
- What make soft. dev. difficult?
  - humans' ambiguous tasks/requests
  - inflexible nature of computers

Soft. Dev == Difficult

- humans' ambiguous tasks/requests
- often sequences of tasks are unclear
- human algorithms often include spontaneous ideas
- humans often have a leap in their logic

Soft. Dev == Difficult

- inflexible nature of computers
  - Computers can execute massive amount of instructions very fast.
  - They are capable of everything as long as expressed in the instructions.
  - You even have to specify how/what to output as error messages.

Soft. Dev == Difficult

- When you (a group of programmers) have to write software for a large system...
- might need to write > 10,000, 100,000, or even 1,000,000 loc.
- without a mistake.

OOP to OOT

- Started as OOP, it became OO Technologies over last 30 years.

- Simula67 in 1967
  - Class, Polymorphism, Inheritance
  - First OO Programming Language

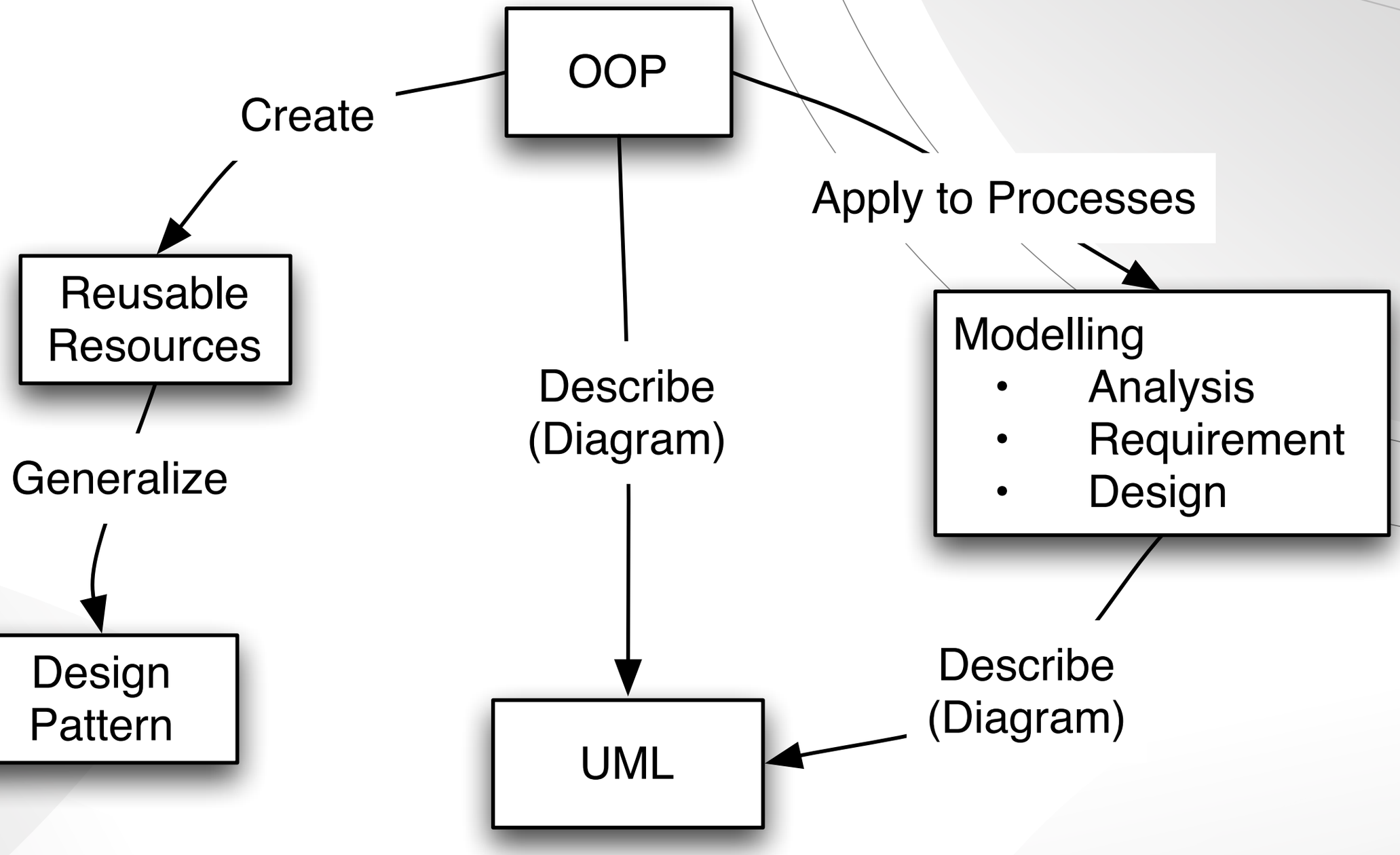
OOP ➔Reusable Resources

- OOP allowed to build large number of "re-usable" software parts.

- Repeated design ideas became "Design Pattern"

RR ➔Modelling

- Developed technologies to describe software structure (created by OOP) using “Diagram”
  - UML
- Applied the concept of OOP to “higher-level dev. schedule management”
  - Modelling of processes (Analysis, Requirement, Design)
- Total Soft. Dev. Process in OO manner.
- Repeated design ideas became “Design Pattern”



Why ppl think OO is difficult...

- So many terms...
  - Class, Object, Superclass, Subclass, Inheritance, Multiple Inheritance, Generalization, Specialization, Interface, Relation, Aggregation, Delegation, Override, Overload, Access Control, Constructor, Destructor, Namespace, Package, Exception, Garbage Collection, Class Libraries, Frameworks, Design Pattern, Use Case Modelling, UML, Refactoring, Agile Process, Extreme Prog., etc.

Why ppl think OO is difficult...

- abuse/misuse of metaphors
  - effective use of metaphor to explain OO is ok...but
- some describes..."You can describe a real-world target(s) in the program using OO"...
- really?

Why ppl think OO is difficult...

- At SIT, lecturers and admin staff members communicate each other to carry out tasks
  - Objects in the computer also send messages to each other to carry out tasks...
- Multiple Inheritance
  - Mammals : Animals, Reptiles: Animals,
    - Platypus : mammals, reptiles
- Message passing
  - Human.Birthday,
    - Tanaka (Human).yourAge() returns his/her age.

Why ppl think OO is difficult...

- EOS :"Everything is Object Syndrome"
  - Many real-world objects can be represented as software objects (classes) but some objects are hard to directly translate into software objects..
  - example?

We will cover...

- OO Programming
- Reusable Resources / Design Patterns
- Modelling (design/UML)