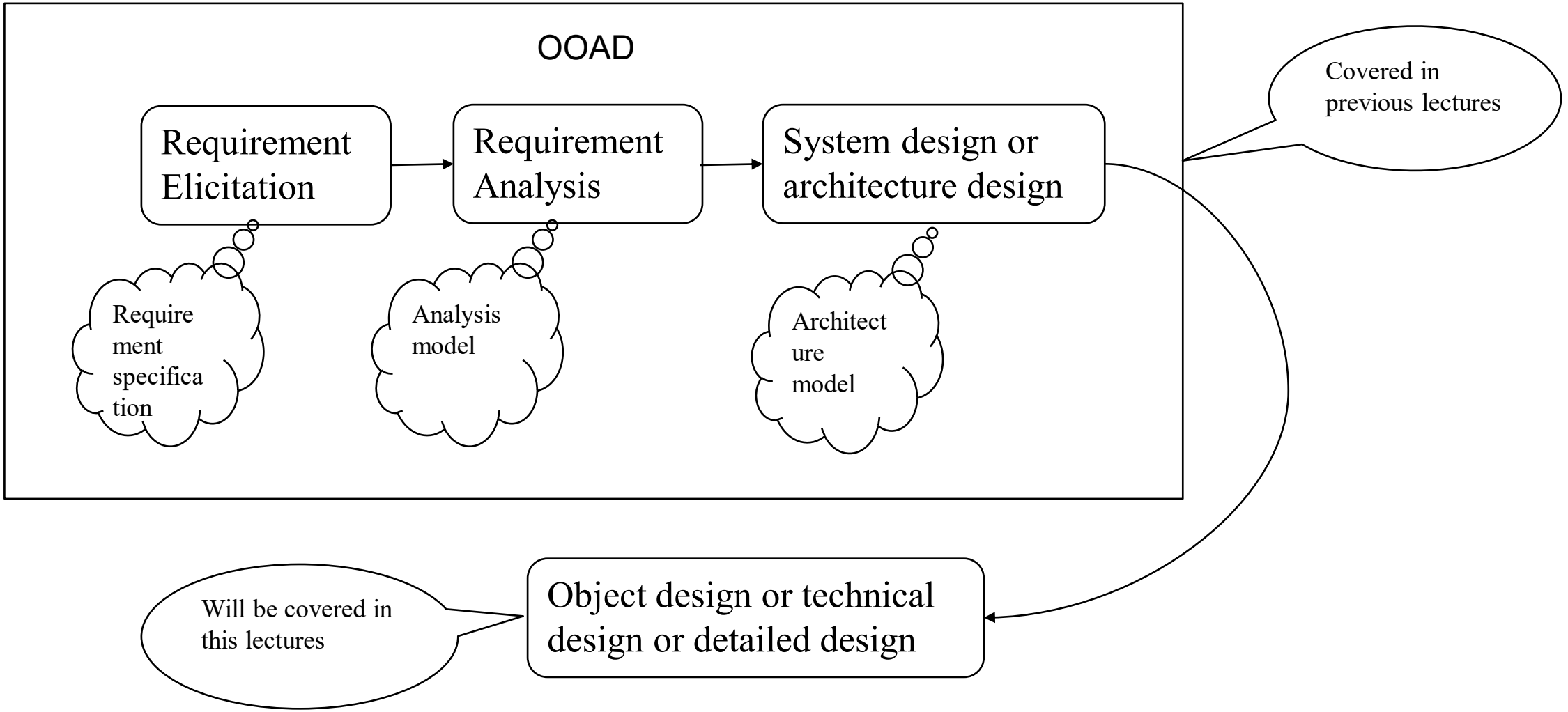
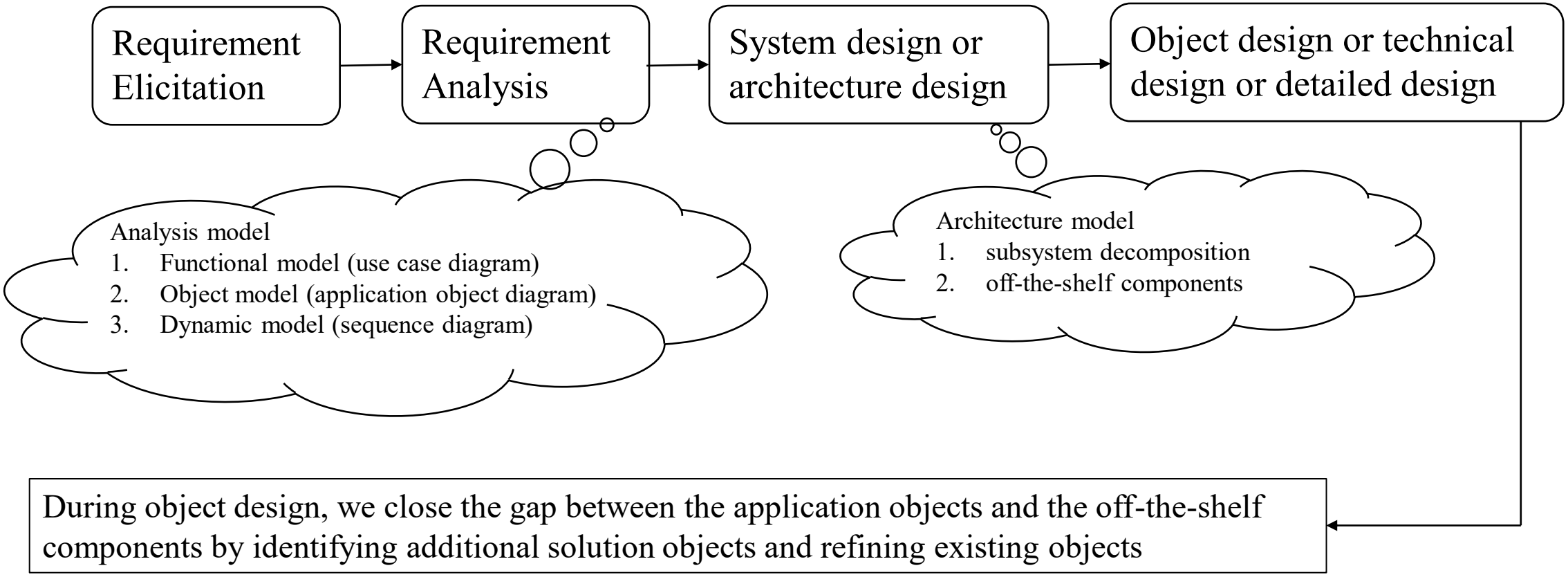
Object-Oriented Analysis and Design using JAVA

B.Tech (CSE/IT) 5th SEM
2021-2022

Lecture-11: Object design

Introduction




Requirement Elicitation
Requirement Analysis
System design or architecture design
Object design or technical design or detailed design
Analysis model
1. Functional model (use case diagram)
2. Object model (application object diagram)
3. Dynamic model (sequence diagram)
Architecture model
1. subsystem decomposition
2. off-the-shelf components
During object design, we close the gap between the application objects and the off-the-shelf components by identifying additional solution objects and refining existing objects

Object design

Object design includes

- ***Reuse*****,** during which we identify off-the-shelf components and design patterns to make use of existing solutions
- ***Service specification***, during which we precisely describe each class interface
- ***Object model restructuring***, during which we transform the object design model to improve its understandability and extensibility
- ***Object model optimization***, during which we transform the object design model to address performance criteria such as response time or memory utilization.

Application Objects and Solution Objects

- **Application objects**, also called "domain objects," represent concepts of the domain that are relevant to the system.
- **Solution objects** represent components that do not have a counterpart in the application domain, such as persistent data stores, user interface objects, or middleware.
- **During analysis**, we identify entity objects and their relationships, attributes, and operations. Most entity objects are application objects that are independent of any specific system.
- **During analysis**, we also identify solution objects that are visible to the user, such as boundary and control objects representing forms and transactions defined by the system.
- **During system design**, we identify more solution objects in terms of software and hardware platforms.
- **During object design**, we refine and detail both application and solution objects and identify additional solution objects needed to bridge the object design gap

Software architect and design roles in industry

Software architect

1. A software architect needs to **interact with clients**, product managers, and developers in order to **envision, model and provide** initial models and designs that can be built. This role also may cover the meeting potential or current customers.

2. A software architect has to constantly **review the code** to ensure the quality of the design by avoiding complexity, **advocating clarity** and to do this with the team. This usually requires hands-on work in terms of developing prototypes, contributing code or **evaluating technologies**.

3. The role of a software architect includes **collaborative working** with a degree of humility and providing **mentoring** as required. Such collaboration also allows the architect to become familiar with the skills and interests in the team and to share their knowledge with the rest of the team. Humility is required to ensure that all the team is listened to, as they may have more specific experience or knowledge for the problem at hand.

4. Taking into account all of the main aspects the software architect role includes, its obvious that this person should have knowledge in programming, management, psychology, communication and even finance. So, what are the main skills and qualities this specialist must have?

Software designer

1. A software designer is responsible for problem-solving and planning for a software solution.

2. After the purpose and specifications of software are determined, software developers will design or employ designers to develop a plan for a solution.

3. It includes low-level component and algorithm implementation issues as well as the architectural view.

4. A software designer is responsible for the documentation of the plan which is usually the product of the design.

5. The software designer works closely with the software developer to ensure that the design meets needs for the solution.

6. It is also important to ensure that the software is able to interact well with other applications that are currently in use.

Competing Qualities and Trade-offs

IEEE Std. 1061	ISO Std. 9126	MITRE Guide to Total Software Quality Control	
Efficiency	Functionality	Efficiency	Integrity
Functionality	Reliability	Reliability	Survivability
Maintainability	Usability	Usability	Correctness
Portability	Efficiency	Maintainability	Verifiability
Reliability	Maintainability	Expandability	Flexibility
Usability	Portability	Interoperability	Portability
		Reusability	

- The main challenge of architecting modern industrial software system is that they need to fulfill quality attributes (QAs) - such as modularity or reusability.
- QAs often are conflicting, which entails difficult trade-offs. As a consequence, even the architecture of closely related software products can differ substantially due to their different priorities on QAs.

Referred sources mentioned in Key References

Key references

Bernd Bruegge & Allen H. Dutoit - Object-Oriented Software Engineering: Using UML, Patterns, and Java

https://syndicode.com/blog/the-role-skills-and-duties-of-a-software-architect/

https://www.coursera.org/lecture/object-oriented-design/1-1-2-software-architect-and-design-roles-in-industry-cIGsa?utm_source=link&utm_medium=page_share&utm_content=vlp&utm_campaign=top_button

https://www.qgcio.qld.gov.au/information-on/workforce-planning/ict-career-streams/software-designer

Thank You