UNIVERSITAS INDONESIA

Veritas, Probitas, Iustitia

FAKULTAS ILMU KOMPUTER

ADDITIONAL- MOVING ON TO DESIGN

ANALISIS DAN PERANCANGAN SISTEM INFORMASI

CSIM603183

Introduction

- Whereas **analysis modeling** concentrated on the **functional requirements** of the evolving system, **design modeling** incorporates the **nonfunctional requirements**.
- That is, design modeling focuses on how the system will operate.
- Design contains steps that guide the project team to constructed system based on analysis model by optimizing the problem-domain information and adding system environment

Steps in Design Modeling

1. Verifying and Validating the Analysis Models
    - Before adding system environment, project team need to **verify and validate** the **analysis model** and ensure that different models (functional, structural, and behavioral) must be consistent.
2. Evolving the Analysis Models into Design Models
    - The analysis model should be reviewed when evolving the analysis model into design model. In addition, **system environment information** must be added to the problem-domain information

Analysis to Design

- Analysis determines the business needs
- Design activities focus on how to build the system
  - Major activity is to evolve the models into a design
  - Goal is to create a blueprint for the design that makes sense to implement
  - Determine how and where data will be stored
  - Determine how the user will interface with the system (user interface, inputs and outputs)
  - Decide on the physical architecture
- Analysis and design phases are highly *interrelated* and may require much "going back and forth"
  - Example: prototyping may uncover additional information

The Design Process

- Verify and validate the analysis models
- Evolve the analysis models into design models
- Create packages and utilize package diagrams
- Decide upon a design strategy

Design Modeling Deliverables

1. The class and method designs
   - illustrated using the class specifications (using CRC cards and class diagrams), contracts (constraints and guarantees), and method specifications.
2. The data management layer
   - addressed by designing the actual database or file structure to be used for object persistence, and a set of classes that will map the class specifications into the object persistence format chosen.

Design Modeling Deliverables

3. The user interface layer design
   - use use scenarios, windows navigation diagrams, real use cases, interface templates, storyboards, windows layout diagrams, and user interface prototypes.
4. The physical architecture layer design
   - is created using deployment diagrams and hardware software specifications.

This collection of deliverables represents the system specification that is handed to the programming team for implementation.

Verifying & Validating the Analysis Models

- Do the analysis models accurately represent the problem domain?
  - Test the fidelity of each model
  - Example: activity diagrams, use-case descriptions and use-case diagrams should all describe the same functional requirements
- Balance the models to ensure consistency between them

Balancing Functional &Structural Models

- A class on a class diagram must be associated with at least one use-case
- An activity in an activity diagram and an event in a use-case description should be related to one or more operations on a class diagram
- An object node on an activity diagram must be associated with an instance or an attribute on a class diagram
- An attribute or an association/aggregation relationship on a class diagram should be related to the subject or object of a use-case

Balancing Functional &Structural Models

- Sequence & communication diagrams must be associated with a use-case
- Actors on sequence & communication diagrams or CRUDE matrices must be associated with actors within a use-case
- Messages on sequence & communication diagrams, transitions on behavioral state machines and entries in a CRUDE matrix must relate to activities on an activity diagram and events in a use-case
- All complex objects in activity diagrams must be represented in a behavioral state machine

Balancing Functional &Structural Models

- Objects in a CRUDE matrix must be associated with classes.
- Behavioral state machine must be associated with objects on a class diagram.
- Objects in sequence and communication diagrams must be associated with objects on a class diagram.
- Messages on sequence and communication diagrams and transitions on behavioral state machines must be associated with operations in a class.
- States in a behavioral state machine must match the different values of an attribute of an object.

Selecting Design Strategies

During design, project team can consider three approaches:

- developing a custom application in-house,
- buying a package and customizing it,
- relying on external vendor to build/support the system

	Use Custom Development When…	Use a Packaged System When…	Use Outsourcing When…
Business Need	The business need is unique.	The business need is common.	The business need is not core to the business.
In-house Experience	In-house functional and technical experience exists.	In-house functional experience exists.	In-house functional or technical experience does not exist.
Project Skills	There is a desire to build in-house skills.	The skills are not strategic.	The decision to outsource is a strategic decision.
Project Management	The project has a highly skilled project manager and a proven methodology.	The project has a project manager who can coordinate the vendor's efforts.	The project has a highly skilled project manager at the level of the organization that matches the scope of the outsourcing deal.
Time frame	The time frame is flexible.	The time frame is short.	The time frame is short or flexible.

Selecting Acquisition Strategies

- The decision must be made regarding the specific type of system needs to be designed.
- It can be help using decision matrix by presenting feasibility information for some candidate solutions.

Selecting Acquisition Strategies

Evaluation Criteria	Relative Importance (Weight)	Alternative 1: Custom Application Using VB.NET	Score (1–5)*	Weighted Score	Alternative 2: Custom Application Using Java	Score (1–5)*	Weighted Score	Alternative 3: Packaged Software Product ABC	Score (1–5)*	Weighted Score
Technical Issues:										
Criterion 1	20		5	100		3	60		3	60
Criterion 2	10		3	30		3	30		5	50
Criterion 3	10		2	20		1	10		3	30
Economic Issues:										
Criterion 4	25	Supporting	3	75	Supporting	3	75	Supporting	5	125
Criterion 5	10	Information	3	30	Information	1	10	Information	5	50
Organizational Issues:										
Criterion 6	10		5	50		5	50		3	30
Criterion 7	10		3	30		3	30		1	10
Criterion 8	5		3	15		1	5		1	5
TOTAL	100			350			270			360

* This denotes how well the alternative meets the criteria. 1 = poor fit; 5 = perfect fit.