IF2261 Software Engineering

Analysis Modeling

Program Studi Teknik Informatika
STEI ITB

Analysis Model

- the first technical representation of a system.
- uses a combination of text and diagrams to represent software requirements (data, function, and behavior) in an understandable way.
- helps make it easier to uncover requirement inconsistencies and omissions.
- two types are commonly used:
  - structured analysis and
  - object-oriented analysis.

** SEPA 6th ed, Roger S. Pressman*

Analysis Model Guidelines

- Analysis products must be highly maintainable, especially the software requirements specification.
- Problems of size must be dealt with using an effective method of partitioning.
- Graphics should be used whenever possible.
- Differentiate between the logical (essential) and physical (implementation) considerations.
- Find something to help with requirements partitioning and document the partitioning before specification.
- Devise a way to track and evaluate user interfaces.
- Devise tools that describe logic and policy better than narrative text.

** SEPA 6th ed, Roger S. Pressman*

Analysis Model Objectives

- Describe what the customer requires.
- Establish a basis for the creation of a software design.
- Devise a set of requirements that can be validated once the software is built.

** SEPA 6th ed, Roger S. Pressman*

Analysis Model Rules of Thumb

- The model should focus on requirements that are visible within the problem or business domain and be written as a relatively high level of abstraction.
- Each element of the analysis model should add to the understanding of the requirements and provide insight into the information domain, function, and behavior of the system.
- Delay consideration of infrastructure and non-functional models until design.
- Minimize coupling throughout the system.
- Be certain the analysis model provides value to all stakeholders.
- Keep the model as simple as possible.

** SEPA 6th ed, Roger S. Pressman*

Structured Analysis Model Elements

- Data dictionary
  - contains the descriptions of all data objects consumed or produced by the software
- Entity relationship diagram (ERD)
  - depicts relationships between data objects
- Data flow diagram (DFD)
  - provides an indication of how data are transformed as they move through the system; also depicts functions that transform the data flow (a function is represented in a DFD using a process specification or PSPEC)
- State diagram (SD)
  - indicates how the system behaves as a consequence of external events, states are used to represent behavior modes. Arcs are labeled with the events triggering the transitions from one state to another (control information is contained in control specification or CSPEC)

** SEPA 6th ed, Roger S. Pressman*

Data Modeling (ERD)

- Elements:
  - Data object - any person, organization, device, or software product that produces or consumes information
  - Attributes - name a data object instance, describe its characteristics, or make reference to another data object
  - Relationships - indicate the manner in which data objects are connected to one another
- Cardinality and Modality (ERD)
  - Cardinality - in data modeling, cardinality specifies how the number of occurrences of one object are related to the number of occurrences of another object (1:1, 1:N, M:N)
  - Modality - zero (0) for an optional object relationship and one (1) for a mandatory relationship

** SEPA 6th ed, Roger S. Pressman*

Creating Entity Relationship Diagrams

- Customer asked to list "things" that application addresses, these things evolve into input objects, output objects, and external entities
- Analyst and customer define connections between the objects
- One or more object-relationship pairs is created for each connection
- The cardinality and modality are determined for an object-relationship pair
- Attributes of each entity are defined
- The entity diagram is reviewed and refined

** SEPA 6th ed, Roger S. Pressman*

Functional Modeling and Information Flow (DFD)

- Shows the relationships of external entities, process or transforms, data items, and data stores
- DFD's cannot show procedural detail (e.g., conditionals or loops) only the flow of data through the software
- Refinement from one DFD level to the next should follow approximately a 1:5 ratio (this ratio will reduce as the refinement proceeds)
- To model real-time systems, structured analysis notation must be available for time continuous data and event processing (e.g., Ward and Mellor or Hately and Pirbhai)

** SEPA 6th ed, Roger S. Pressman*

Creating Data Flow Diagram

- Level 0 data flow diagram should depict the system as a single bubble
- Primary input and output should be carefully noted
- Refinement should begin by consolidating candidate processes, data objects, and data stores to be represented at the next level
- Label all arrows with meaningful names
- Information flow must be maintained from one level to level
- Refine one bubble at a time
- Write a PSPEC (a "mini-spec" written using English or another natural language or a program design language) for each bubble in the final DFD

** SEPA 6th ed, Roger S. Pressman*

Behavioral Modeling (STD)

- State transition diagrams represent the system states and events that trigger state transitions
- STD's indicate actions (e.g., process activation) taken as a consequence of a particular event
- A state is any observable mode of behavior
- Hatley and Pirbhai control flow diagrams (CFD) and UML sequence diagrams can also be used for behavioral modeling

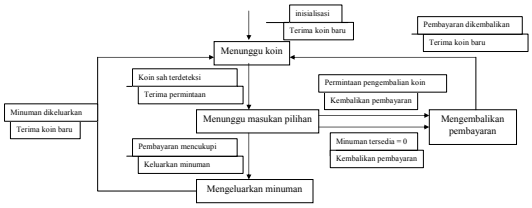
** SEPA 6th ed, Roger S. Pressman*

Behavioral Modeling

- Mendeskripsikan status sistem yang dapat muncul ketika perangkat lunak digunakan
- mendeskripsikan kelakuan sistem
- Tools:
  - State Transition Diagram
  - Control Specification
- Umumnya digunakan pada sistem waktu-nyata

State Transition Diagram

- Contoh STD untuk mesin otomatis penjual minuman
  (tidak ada hubungannya dengan contoh sebelumnya):



Control Specification

- Fungsi C-SPEC sama dengan P-SPEC namun berisi deskripsi dari setiap status yang dapat muncul pada sistem

Creating Control Flow Diagrams

- Begin by stripping all the data flow arrows form the DFD
- Control items (dashed arrows) are added to the diagram
- Add a window to the CSPEC (contains a SD that is a sequential specification of the behavior) for each bubble in the final CFD

** SEPA 6th ed, Roger S. Pressman*