Chapter 7: System Design Methodology

Instructor:
Navin Gautam
Lecturer, Kantipur Engineering College
Email: navingautam84@hotmail.com
Date: September 14, 2010

Introduction

- A system development methodology is a set of activities, methods, best practices, deliverables, and automated tools that stakeholders use to develop and continuously improve information systems and software
- Modeling is an act of drawing one or more graphical representation of the system->aka waterfall model

Model driven development

- Process modeling
- Data modeling
- Object modeling
  - Object models are diagrams that document a system in terms of its objects and their interactions.

Advantages of model driven development

- It maintains the planning overhead because all the phases are planned up front.
- Requirement analysis tends to be more through and better documented.
- Alternative technical solution tends to be more thoroughly analyzed.
- System designs tend to be more sound, stable, adaptable, and flexible because they are model based and more thoroughly analyzed before they are built.
- The approach is effective for systems that are well understood but so complex that they require large teams to compare
- The approach is more suitable when fulfilling user expectations and quality are more important than cost and schedule.

Disadvantages of model driven development

- It is a long process: it is takes time to collect the facts, draw the models, and validate those models
- The models can only be as good as the user's understanding of those requirements.
- Most of the users are interested in the working modules/ software of the project, not on the models and the pictures.
- Inflexible; because user must fully specify the requirements before design; design must fully be documented before construction and so forth.

Model driven development technique

- Structured Analysis and Design
- Information Engineering
- Object-Oriented Analysis and Design

Structured analysis and design

- process-centered technique
- Data Flow Diagram (DFD), used to illustrate business process requirements
- Structured design introduced tool called structure charts, used to illustrate software structure to fulfill business requirement
- Data flow diagrams and structure chart contribute significantly to reducing the communication gap that offers between the non-technical and technical groups

Information Engineering

- A model-driven and data-centered, but process-sensitive technique for planning, analyzing, and designing information systems
- IE models are pictures that illustrate and synchronize the system's data and processes

Object-oriented analysis (OOA)

- A model-driven technique that integrates data and process concerns into constructs called objects
- OOA models are pictures that illustrate the system's objects from various perspectives such as structure and behavior, and interactions of the objects
- Object is the encapsulation of the data (called properties) that describes a discrete person, object, place, event, or thing, with all the processes (called methods) that are allowed to use or update the data and properties
- The only way to access or update the object's data is to use the object's predefined processes

Checklist methodologies (1)

- Drafting checklist
  - use simple language, avoid using jargon;
  - define terms, symbols and acronyms;
  - do not expect the specification to be read by experts; write it so that a layman will understand;
  - use a logical structure;
  - be as concise as possible, but keep the meaning clear;
  - plan and analyze your needs;
  - arrange the components of the requirement into a logical form matching the evaluation model; a good way of doing this is to set out a skeleton structure with the main headings and then add in sub-headings as necessary;
  - do not embed requirements in background information –suppliers may miss them – keep requirements in their own sections;
  - list the most important elements of the requirement first and work through to the least important; and
  - discuss the requirement with colleagues, other users and procurement staff. During this process you may also identify other topics you need to include.

Checklist methodologies (2)

- Types of checklist
  - Specification checklist
  - Technical Review Checklist
  - Specification checklist
    - state the requirement specification completely, clearly, concisely, logically and unambiguously;
    - focus on outputs not how they are to be met;
    - contain enough information for potential suppliers to decide and cost the goods or services they will offer, or in the case of negotiated route arrive at realistic budgetary
    - costs;
    - permit offered goods or services to be evaluated against defined criteria by examination, trial, test or documentation;

Checklist methodologies (3)

- Specification checklist (contd..)
  - state the criteria for acceptance by examination, trial, test or documentation;
  - contain only the essential features or characteristics of the requirement;
  - provide equal opportunity for all potential suppliers to offer a product or service which
  - satisfies the needs of the user and which may incorporate alternative technical solutions;
  - and comply with any legal obligations e.g. under UK law, the EEC Treaty, an EC Directive or the GATT Agreement on Government procurement

Checklist methodologies (4)

- Technical review checklist sample
  - DESIGN REFERENCE
    - What are the applicable Specification/Proposal paragraphs
    - What design data is required?
    - Is all data available?
    - Assumptions made when data is not available
  - DESIGN APPROACH
    - Describe the operational system
    - Describe the simulation approach
      - Simulation requirements
      - Any prior design used?
      - What alternatives were considered?
      - Was a block diagram provided?
      - Were interface requirements discussed and did they cover the scope of the requirement?

Checklist methodologies (5)

- Technical review checklist sample (contd..)
  - DESIGN APPROACH (contd..)
    - Describe the simulation approach (contd..)
      - If required in the spec, were malfunction simulations addressed?
      - Were any deviations to the spec noted or requested?
      - Were risks and risk mitigation addressed?
    - Simulation Design Analysis
      - Were the equations used in the design presented?
      - Was a mathematical analysis of the design presented?
      - Were the computational methods used in the design discussed?

Checklist methodologies (6)

- Technical review checklist sample (contd..)
  - DESIGN APPROACH (contd..)
    - Were the following program areas discussed?
      - Standardization
      - Reliability
      - Maintainability
      - EMI
      - Value Engineering
      - Safety
      - Human Factors
      - COTS, NDI(New Development items)

Checklist methodologies (7)

- Technical review checklist sample (contd..)
  - SOFTWARE PROGRAM REQUIREMENTS
    - Was a simplified flow chart and description provided?
    - Were these methodologies addressed?
      - Function generation
      - Initialization parameters
      - Data table layout
      - Subroutine linkage
      - Runtime versus special requirement
      - total memory versus spec requirement
  - PROBLEM AREAS
    - Risks
    - Recommendations

Process Oriented System Development Methodology

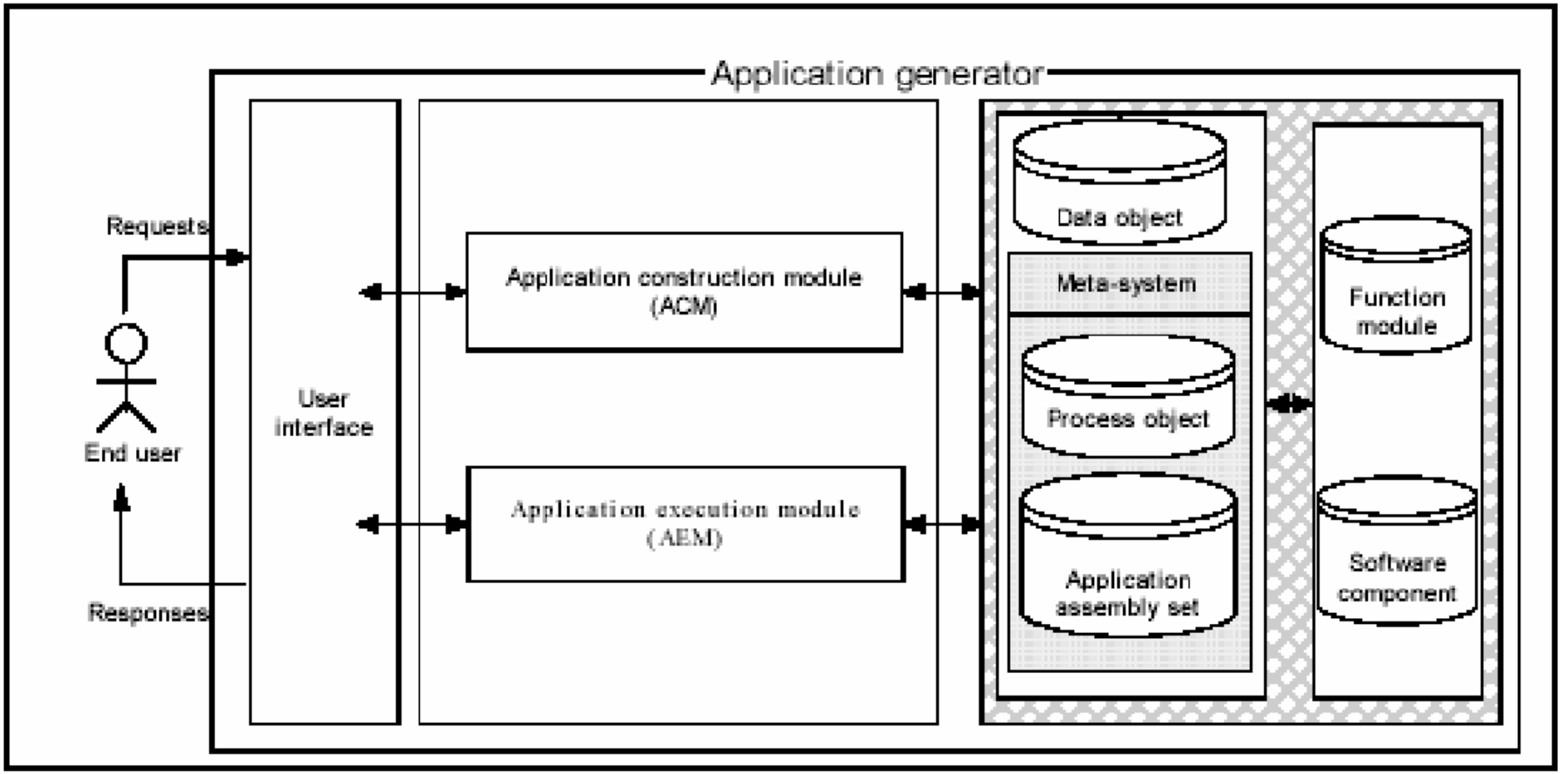
- A traditional approach in which an information system design is based upon what the system is supposed to do; i.e. the focus is on output and processing logic
- The assumption is that we could easily derive all data
- requirements from all known system deliverables
- traditional process oriented approach suffers from the problems like, redundancy, scalability etc

Application Generation

- Application generators have been championed because they can be used to shorten a system's development cycle and therefore lower its development costs.
- Application generators are standardized building blocks that can be used to assemble, rather that develop, information systems
- The methodology of Information System Development using Application Generation consists of three steps:
  - User requirement modeling,
  - Specification construction
  - Application construction
- One of the most visible benefits of the Application Generation approach of System Development is the Software Reusability. Primary concern of reusability
  - Developing software form prefabricated parts
  - Reusing those parts in other applications,
  - Maintaining and customizing those parts to produce new functions and features

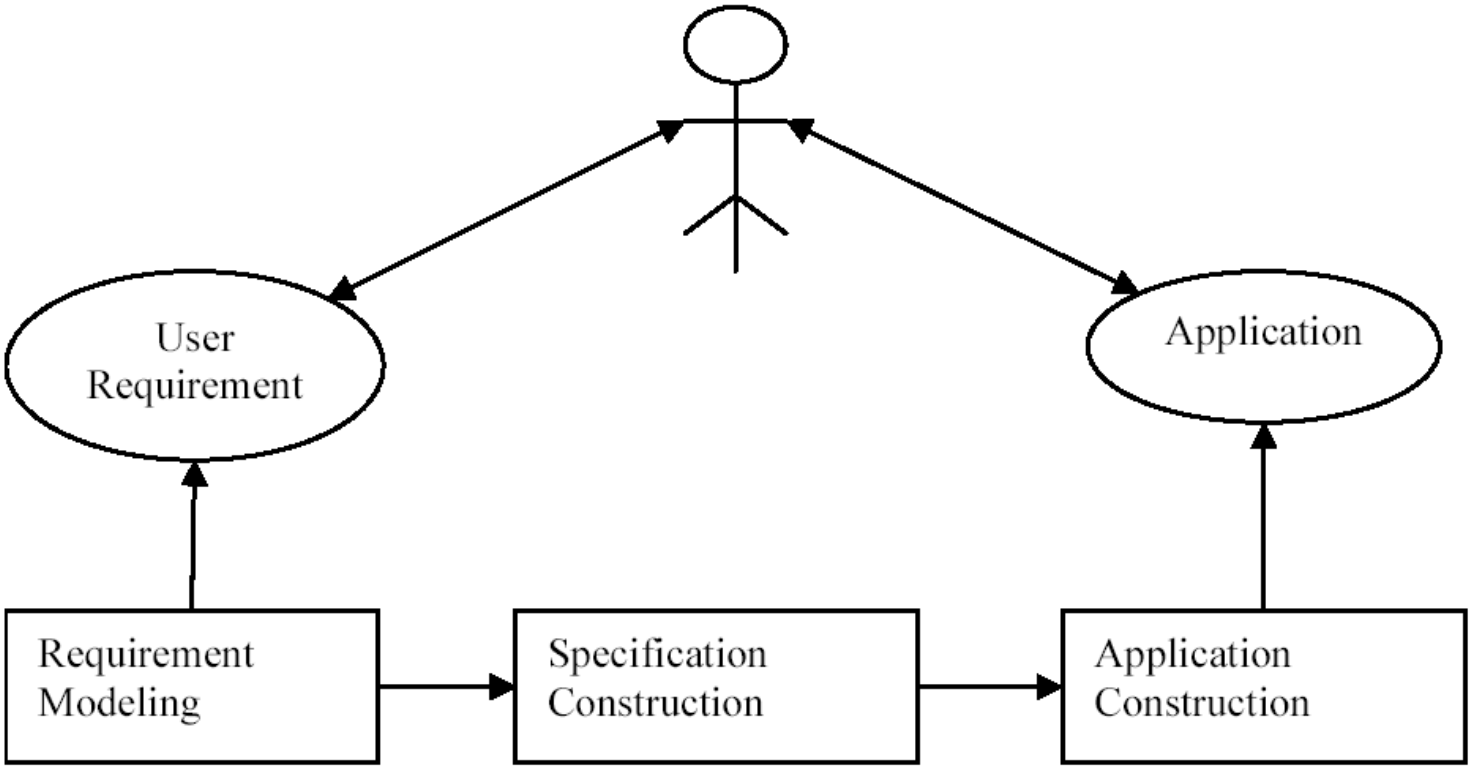
Application Generator Architecture

- The application generator architecture consists of three subsystems: User Interface,



IS Construction Methodology Using Application Generation

- The method includes three major stages: *requirement modeling, specification construction, application construction*



Structured Design (1)

- Structure is a fundamental characteristic of computer software that permits decomposition of a large system into smaller, more manageable units with well-defined relationships to the other units in the system
- Structured design was developed by Constantine as a top-down technique for architectural design of software system.
- The basic approach in structured design is systematic conversion of call and return architecture from data flow diagrams within the required model.
- The aim of structured design is to transform the results of structured analysis (i.e. DFD representation) into a structure chart
- Structure Chart represents the software architecture, various modules making up the system, module dependency (i.e. which module calls which other modules), parameters that are passed among the different modules.

Structured Design (2)

- Steps to build Structure Chart
  - Draw a context DFD to establish initial project scope.
  - Draw a functional decomposition diagram to partition the system into subsystems.
  - Create an event-response or use-case list for the system to define events for which the system must have a response.
  - Draw an event DFD (or event handler) for each event.
  - Merge event DFDs into a system diagram (or, for larger systems, subsystem diagrams).
  - Draw detailed, primitive DFDs for the more complex event handlers.
  - Document data flows and processes in the data dictionary.

Structured Design (3)

- Context diagram (Self)
- Event diagram (Self)
- DFD (Self)