KBS : Knowledge-Based System Design

- Motivation
- Objectives
- Chapter Introduction
  - Review of relevant concepts
  - Overview new topics
  - Terminology
- ES Development Life Cycle
  - Feasibility Study
  - Rapid Prototype
  - Refined System
  - Field Testable
  - Commercial Quality
  - Maintenance and Evolution
- Software Engineering and ES Design
  - Software Development Life Cycle
- Linear Model ES Life Cycle
  - Planning
  - Knowledge Definition
  - Knowledge Design
  - Knowledge Verification
- Important Concepts and Terms
- Chapter Summary

Material

[Awad 1996]
- Chapter 5: Expert System Development Life Cycle
- Chapter 15: Verification and Validation
- Chapter 17: Implementing the Expert System
- Chapter 18: Organizational and Managerial Impact

[Durkin 1994]
- Chapter: 8: Designing Backward-Chaining Rule-Based Systems
- Chapter 10: Designing Forward-Chaining Rule-Based Systems
- Chapter 15: Designing Frame-Based Expert Systems
- Chapter 18: Knowledge Engineering

Material

[Sommerville 2001]
- Chapter 3: Software processes
  - waterfall model
  - evolutionary development
    - spiral model
  - formal methods
  - reuse-based methods
- Chapter 8: Software prototyping
  - rapid prototyping techniques

[Jackson 1999]
- Chapter 14, 15: Constructive Problem Solving
- Chapter 16: Designing for Explanation

Motivation

- reasons to study the concepts and methods in the chapter
  - main advantages
  - potential benefits
- understanding of the concepts and methods
- relationships to other topics in the same or related courses

Objectives

- regurgitate
  - basic facts and concepts
- understand
  - elementary methods
  - more advanced methods
  - scenarios and applications for those methods
  - important characteristics
    - differences between methods, advantages, disadvantages, performance, typical scenarios
- evaluate
  - application of methods to scenarios or tasks
- apply
  - methods to simple problems

ES Development Methods

- commercial quality systems require a systematic development approach
  - ad hoc approaches may be suitable for research prototypes or personal use, but not for widely used or critical systems
- some software engineering methods are suitable for the development of expert systems

Problem Selection

- the development of an expert system should be based on a specific problem to be addressed by the system
- it should be verified that expert systems are the right paradigm to solve that type of problem
  - not all problems are amenable to ES-based solutions
- availability of resources for the development
  - experts/expertise
  - hardware/software
  - users
  - sponsors/funds

Project Management

- activity planning
  - planning, scheduling, chronicling, analysis
- product configuration management
  - product management
  - change management
- resource management
  - need determination
  - acquisition resources
  - assignment of responsibilities
  - identification of critical resources

ES Development Stages

- feasibility study
  - paper-based explanation of the main idea(s)
  - no implementation
- rapid prototype
  - quick and dirty implementation of the main idea(s)
- refined system
  - in-house verification by knowledge engineers, experts
- field test
  - system tested by selected end users
- commercial quality system
  - deployed to a large set of end users
- maintenance and evolution
  - elimination of bugs
  - additional functionalities

Error Sources in ES Development

- knowledge errors
- semantic errors
- syntax errors
- inference engine errors
- inference chain errors
- limits of ignorance errors

Knowledge Errors

- problem: knowledge provided by the expert is incorrect or incomplete
  - reflection of expert's genuine belief
  - omission of important aspects
  - inadequate formulation of the knowledge by the expert
- consequences
  - existing solution not found
  - wrong conclusions
- remedy
  - validation and verification of the knowledge
    - may be expensive

Semantic Errors

- problem: the meaning of knowledge is not properly communicated
  - knowledge engineer encodes rules that do not reflect what the domain expert stated
  - expert misinterprets questions from the knowledge engineer
- consequences
  - incorrect knowledge, inappropriate solutions, solutions not found
- remedy
  - formalized protocol for knowledge elicitation
  - validation of the knowledge base by domain experts

Syntax Errors

- problem: rules or facts do not follow the syntax required by the tool used
  - knowledge engineer is not familiar with the method/tool
  - syntax not clearly specified
- consequences
  - knowledge can't be used
- solutions
  - syntax checking and debugging tools in the ES development environment

Inference Engine Errors

- problem: malfunctions in the inference component of the expert system
  - bugs
  - resource limitations
    - e.g. memory
- consequences
  - system crash
  - incorrect solutions
  - existing solutions not found
- remedy
  - validation and verification of the tools used

Inference Chain Errors

- problem: although each individual inference step may be correct, the overall conclusion is incorrect or inappropriate
  - causes: errors listed above; inappropriate priorities of rules, interactions between rules, uncertainty, non-monotonicity
- consequences
  - inappropriate conclusions
- remedy
  - formal validation and verification
  - use of a different inference method

Limits of Ignorance Errors

- problem: the expert system doesn't know what it doesn't know
  - human experts usually are aware of the limits of their expertise
- consequences
  - inappropriate confidence in conclusions
  - incorrect conclusions
- remedy
  - meta-reasoning methods that explore the limits of the knowledge available to the ES

KBS and Software Engineering

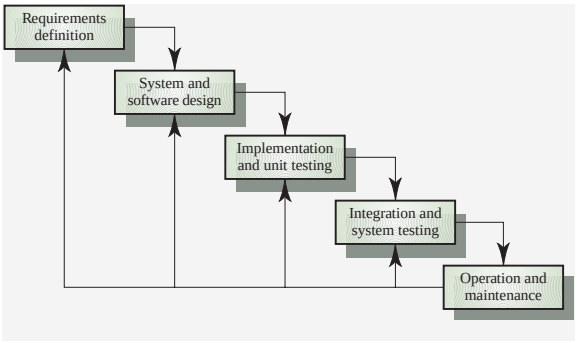
- software process models
  - waterfall
  - spiral
- use of SE models for ES development
- ES development models
  - evolutionary model
  - incremental model
  - spiral model

Generic Software Process Models

- waterfall model
  - separate and distinct phases of specification and development
- evolutionary development
  - specification and development are interleaved
- formal systems development
  - a mathematical system model is formally transformed to an implementation
- reuse-based development
  - the system is assembled from existing components

[Sommerville 2001]

Waterfall Model



[Sommerville 2001]

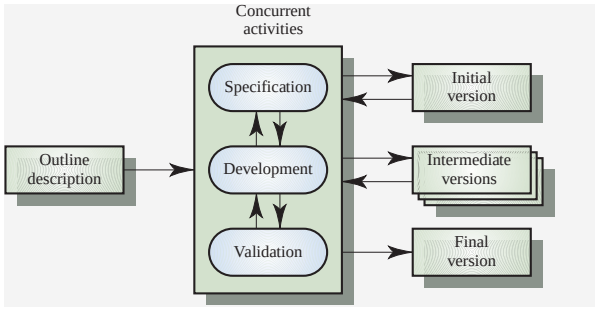
Suitability of Software Models for ES Development

- the following worksheets help with the evaluation of software models for use in the development of expert systems
  - identify the key differences between conventional software development and ES development
    - with respect to a specific model
  - what are the positive and negative aspects of the model for ES development
  - evaluate the above issues, and give the model a score
    - 10 for perfectly suited, 0 for completely unsuitable
  - determine the overall suitability
    - high, medium low
    - explanation

Evolutionary Development

- exploratory development
  - objective is to work with customers and to evolve a final system from an initial outline specification. should start with well-understood requirements
- throw-away prototyping
  - objective is to understand the system requirements. should start with poorly understood requirements

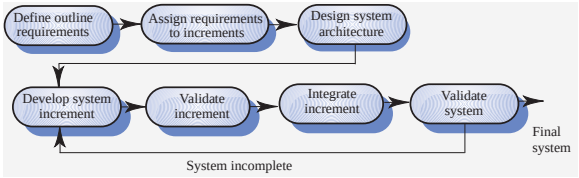
Evolutionary Development



Incremental Development

- development and delivery is broken down into increments
  - each increment delivers part of the required functionality
- user requirements are prioritised
  - the highest priority requirements are included in early increments
- once the development of an increment is started, the requirements are frozen
  - requirements for later increments can continue to evolve

Incremental Development



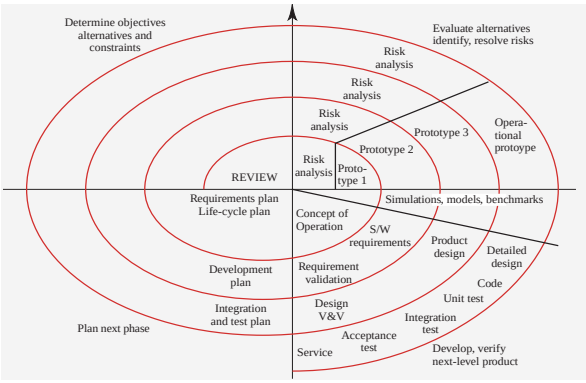
Spiral Development

- process is represented as a spiral rather than as a sequence of activities with backtracking
  - each loop in the spiral represents a phase in the process.
  - no fixed phases such as specification or design
    - loops in the spiral are chosen depending on what is required
  - risks are explicitly assessed and resolved throughout the process
- similar to incremental development

Spiral Model Sectors

- for quadrants in the coordinate system represent specific aspects
  - objective setting
    - specific objectives for the phase are identified
  - risk assessment and reduction
    - risks are assessed and activities put in place to reduce the key risks
  - development and validation
    - a development model for the system is chosen which can be any of the generic models
  - planning
    - the project is reviewed and the next phase of the spiral is planned

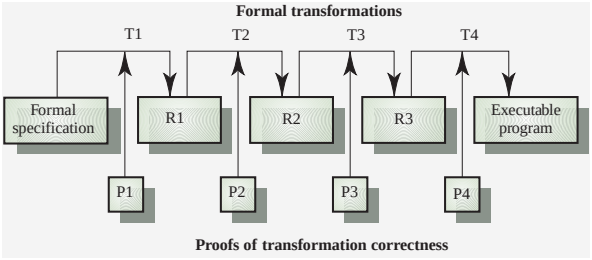
Spiral Model



Formal systems development

- based on the transformation of a mathematical specification through different representations to an executable program
- transformations are ‘correctness-preserving’
  - it is straightforward to show that the program conforms to its specification
- embodied in the ‘cleanroom’ approach to software development

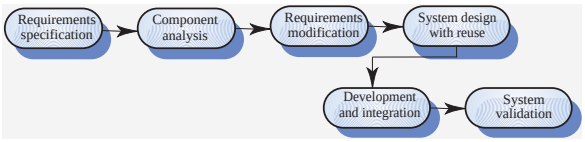
Formal Transformation Model



Reuse-Oriented Development

- based on systematic reuse
  - systems are integrated from existing components or COTS (commercial-off-the-shelf) systems
- process stages
  - component analysis
  - requirements modification
  - system design with reuse
  - development and integration
- this approach is becoming more important but still limited experience with it

Reuse-oriented development

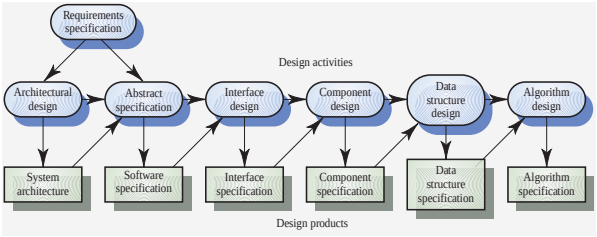


Model Worksheets

Aspect	*Evaluation*	*Score*
key differences		
positive		
negative		

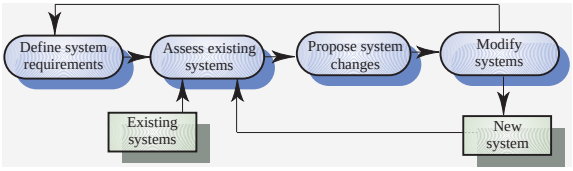
- overall suitability: high medium low
  - explanation

Generic System Design Process



[Sommerville 2001]

System Evolution

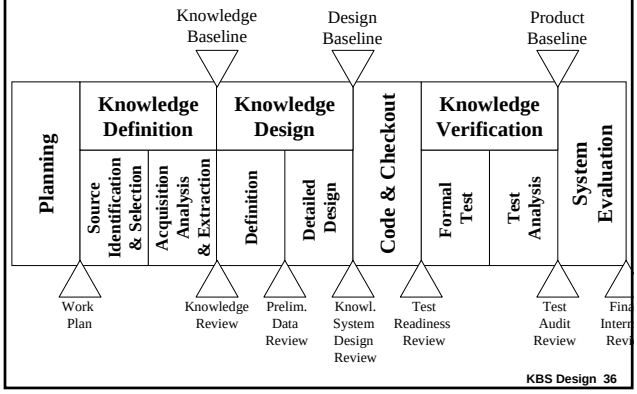


[Sommerville 2001]

Linear Model of ES Development

- the life cycle repeats a sequence of stages
  - variation of the incremental model
  - once iteration of the sequence roughly corresponds to one circuit in the spiral model
- stages
  - planning
  - knowledge definition
  - knowledge design
  - code & checkout
  - knowledge verification
  - system evaluation

Linear Model Diagram



Planning

- feasibility assessment
- resource management
- task phasing
- schedules
- high-level requirements
- preliminary functional layout

Knowledge Definition

- knowledge source identification and selection
  - source identification
  - source importance
  - source availability
  - source selection
- knowledge acquisition, analysis and extraction
  - acquisition strategy
  - knowledge element identification
  - knowledge classification system
  - detailed functional layout
  - preliminary control flow
  - preliminary user's manual
  - requirements specifications
  - knowledge baseline

Knowledge Design

- knowledge definition
  - knowledge representation
  - detailed control structure
  - internal fact structure
  - preliminary user interface
  - initial test plan
- detailed design
  - design structure
  - implementation strategy
  - detailed user interface
  - design specifications and report
  - detailed test plan

Code & Checkout

- coding
- tests
- source listings
- user manuals
- installation and operations guide
- system description document

Knowledge Verification

- formal tests
  - test procedures
  - test reports
- test analysis
  - results evaluation
  - recommendations

System Evaluation

- results evaluation
  - summarized version of the activity from the previous stage
- recommendations
  - as above
- validation
  - system conforms to user requirements and user needs
- interim or final report

Linear Model Exercise

- apply the linear model to your team project
  - map activities, tasks, milestones and deliverables that you have identified to the respective stages in the linear model
  - use the linear model to sketch a rough timeline that involves two iterations
    - first prototype
    - final system
  - estimate the overhead needed for the application of the linear model in our context

Summary Expert System Design

- the design and development of knowledge-based systems uses similar methods and techniques as software engineering
  - some modifications are necessary
  - the linear model of ES development is an adaptation of the incremental SE model
- possible sources of errors are
  - knowledge and limits of knowledge errors
  - syntactical and semantical errors
  - inference engine and inference chain errors

Important Concepts and Terms

- evolutionary development
- expert system (ES)
- expert system shell
- explanation
- feasibility study
- inference
- inference mechanism
- If-Then rules
- incremental development
- knowledge
- knowledge acquisition
- knowledge base
- knowledge-based system
- knowledge definition
- knowledge design
- knowledge representation
- knowledge verification
- limits of ignorance
- linear model ES life cycle
- maintenance
- rapid prototyping
- reasoning
- rule
- semantic error
- software development life cycle
- spiral development
- syntactic error
- waterfall model