Software Engineering

Session 4 – Main Theme
Requirements Model Engineering
Dr. Jean-Claude Franchitti

New York University
Computer Science Department
Courant Institute of Mathematical Sciences
1. Session Overview
2. Requirements Model Engineering
3. Summary and Conclusion
What is the class about?

- Course description and syllabus:
  - [http://www.nyu.edu/classes/jcf/g22.2440-001/](http://www.nyu.edu/classes/jcf/g22.2440-001/)

- Textbooks:
  - *Software Engineering: A Practitioner’s Approach*
    Roger S. Pressman
    McGraw-Hill Higher International
    - Recommended:
      - Code Complete: A Practical Handbook of Software Construction, 2nd Edition
      - The Mythical Man-Month: Essays on Software Engineering, 2nd Edition
Requirements Model Engineering in Brief

- Requirements Engineering Processes
- Tools-Driven Approaches
- Summary and Conclusion
  - Readings
  - Individual Assignment #1 (due)
  - Team Assignment #1 (ongoing)
  - Course Project (ongoing)
Icons / Metaphors

- Information
- Common Realization
- Knowledge/Competency Pattern
- Governance
- Alignment
- Solution Approach
Agenda – Requirements Engineering Processes

2 Requirements Model Engineering

- Requirements Engineering Processes
- Tools-Driven Approaches
The goal of the analysis phase is to truly understand the requirements of the new system and develop a system that addresses them -- or decide a new system isn’t needed.

The line between systems analysis and systems design is very blurry.
Gathering – Information Overview

- Interviews
- Joint Application Design (JAD)
- Questionnaires
- Document Analysis
- Observation
Interviews - Five Basic Steps

- Selecting Interviewees
- Designing Interview Questions
- Preparing for the Interview
- Conducting the Interview
- Post-Interview Follow-up
Selecting Interviewees

- Based on Information Needed
- Often Good to Get Different Perspectives
  - Managers
  - Users
  - Ideally, All Key Stakeholders
## Types of Questions

<table>
<thead>
<tr>
<th>Types of Questions</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Closed-Ended Questions</strong></td>
<td>* How many telephone orders are received per day?</td>
</tr>
<tr>
<td></td>
<td>* How do customers place orders?</td>
</tr>
<tr>
<td></td>
<td>* What additional information would you like the new system to provide?</td>
</tr>
<tr>
<td><strong>Open-Ended Questions</strong></td>
<td>* What do you think about the current system?</td>
</tr>
<tr>
<td></td>
<td>* What are some of the problems you face on a daily basis?</td>
</tr>
<tr>
<td></td>
<td>* How do you decide what types of marketing campaign to run?</td>
</tr>
<tr>
<td><strong>Probing Questions</strong></td>
<td>* Why?</td>
</tr>
<tr>
<td></td>
<td>* Can you give me an example?</td>
</tr>
<tr>
<td></td>
<td>* Can you explain that in a bit more detail?</td>
</tr>
</tbody>
</table>
Designing Interview Questions

- Unstructured interview
  - Broad, Roughly Defined Information
- Structured interview
  - More Specific Information
### Questioning Strategies

<table>
<thead>
<tr>
<th>Level</th>
<th>Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Level</td>
<td>TOP DOWN</td>
</tr>
<tr>
<td>Very General</td>
<td></td>
</tr>
<tr>
<td>Medium-Level</td>
<td></td>
</tr>
<tr>
<td>Moderately</td>
<td>BOTTOM UP</td>
</tr>
<tr>
<td>Specific</td>
<td></td>
</tr>
<tr>
<td>Low-Level</td>
<td></td>
</tr>
<tr>
<td>Very Specific</td>
<td></td>
</tr>
</tbody>
</table>
Interview Preparation Steps

- Prepare General Interview Plan
  - List of Question
  - Anticipated Answers and Follow-Ups
- Confirm Areas of Knowledge
- Set Priorities in Case of Time Shortage
- Prepare the Interviewee
  - Schedule
  - Inform of Reason for Interview
  - Inform of Areas of Discussion
Conducting the Interview

- Appear professional and unbiased
- Record all information
- Check on organizational policy regarding tape recording
- Be sure you understand all issues and terms
- Separate facts from opinions
- Give interviewee time to ask questions
- Be sure to thank the interviewee
- End on time
Conducting the Interview - Practical Tips

- Don’t Worry, Be Happy
- Pay Attention
- Summarize Key Points
- Be Succinct
- Be Honest
- Watch Body Language
Post-Interview Follow-Up

- Prepare Interview Notes
- Prepare Interview Report
- Look for Gaps and New Questions
# INTERVIEW REPORT

Interview notes approved by: ____________

Person interviewed: ______________
Interviewer: ______________
Date: ______________
Primary Purpose:

Summary of Interview:

Open Items:

Detailed Notes:
- Invented by IBM late 1970s
- Structured Meeting of 10-20 users
- ~30 minutes per agenda item
- frequent breaks
Selecting participants
Designing the session
Preparing for the session
Conducting the session
Follow-Up
JAD Key Ideas

- Allows project managers, users, and developers to work together
- May reduce scope creep by 50%
- Avoids requirements being too specific or too vague
Joint Application Design (JAD) Important Roles

- Facilitator
- Scribe
Joint Application Design (JAD) Setting

- U-Shaped seating
- Away from distractions
- Whiteboard/flip chart
- Prototyping tools
- e-JAD
The JAD Session

- Tend to last 5 to 10 days over a three week period
- Prepare questions as with interviews
- Formal agenda and ground rules
- Facilitator activities
  - Keep session on track
  - Help with technical terms and jargon
  - Record group input
  - Help resolve issues
- Post-session follow-up
Managing Problems in JAD Sessions

- Reducing domination
- Encouraging non-contributors
- Side discussions
- Agenda merry-go-round
- Violent agreement
- Unresolved conflict
- True conflict
- Use humour
Structured Meeting
Facilitator and scribe + 10-20 users
Attempts to overcome usual problems with groups
Only one person talks at once
Every opinion is valued
Questionnaire Steps

- Selecting participants
  - Using samples of the population
- Designing the questionnaire
  - Careful question selection
- Administering the questionnaire
  - Working to get good response rate
- Questionnaire follow-up
  - Send results to participants
Begin with non-threatening and interesting questions

Group items into logically coherent sections

Do not put important items at the very end of the questionnaire

Do not crowd a page with too many items

Avoid abbreviations

Avoid biased or suggestive items or terms

Number questions to avoid confusion

Pretest the questionnaire to identify confusing questions

Provide anonymity to respondents
Document Analysis

- Provides clues about existing “as-is” system
- Typical documents
  - Forms
  - Reports
  - Policy manuals
- Look for user additions to forms
- Look for unused form elements
Observation

- Users/managers often don’t remember everything they do
- Checks validity of information gathered other ways
- Behaviours change when people are watched
- Careful not to ignore periodic activities
  - Weekly … Monthly … Annual
Criteria for Selecting the Appropriate Techniques

- Type of information
- Depth of information
- Breadth of information
- Integration of information
- User involvement
- Cost
- Combining techniques
Selecting the Appropriate Techniques

<table>
<thead>
<tr>
<th></th>
<th>Interviews</th>
<th>JAD</th>
<th>Questionnaires</th>
<th>Document Analysis</th>
<th>Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of Information</strong></td>
<td>As-Is</td>
<td>As-Is</td>
<td>As-Is</td>
<td>As-Is</td>
<td>As-Is</td>
</tr>
<tr>
<td></td>
<td>Improve. To-Be</td>
<td>Improve. To-Be</td>
<td>Improve. To-Be</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Depth of Information</strong></td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Breadth of Information</strong></td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Integration of Info.</strong></td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td><strong>User Involvement</strong></td>
<td>Medium</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>Medium</td>
<td>Low-Medium</td>
<td>Low</td>
<td>Low</td>
<td>Low-Medium</td>
</tr>
</tbody>
</table>
Agenda – Requirements Engineering Processes

2 Requirements Model Engineering

- Requirements Engineering Processes
- Tools-Driven Approaches
Sub-Section Objectives

- Describe Requirements Model Engineering Activities for a Selected Tools-Driven Approach
Requirements and Definitions Traceability Graph

- Strategy Definition
- Concept Definition
- Functional Requirements
- Non-Functional Requirements
- Business Use Case Requirements
- Business Model Requirements
  - Location Requirements
  - Organizational Requirements
  - Process Model Requirements
  - Business Rules Requirements
  - Workflow Rules Requirements
Reasoning About Business Entities and Their Dependencies and Goals

Legend:
- : GRL Modeling Constructs

![Diagram]

- **High-Level Business Requirements** (bus. obj. & informal reqs)
- **Business Model Reqs** (org, location, process)
- **Value Flows Between Actors**
- **Goals & Subgoals**
- **Actors**
- **Dependencies**
- **Help Engineer**
- **EAMF Process Patterns**
- **High-Level Tasks**

**Legend:***

1. Modeled by
2. Represented In Terms of
3. Motivated By
4. Have
5. Indicate
6. Have (Based on Goals Achievement)
7. Refined into
Pattern Language Structure for Agent Patterns Selection
(http://www.scs.carleton.ca/~weiss/papers/aois03-revised.pdf)
Requirements Model Engineering Activities

1. Map Business Requirements into Requirements Model
2. Document Traceability between Project Requirements and Req. Model Definitions/Requirements
3. Draw Actor and Dependency Diagram(s)
4. Draw Goal and Task Diagram(s)
5. Document Modeling Constructs
6. Populate Requirements Model Categories
7. Document New Requirements Traceability
8. JUCMNav Diagram
9. EAMF Patterns

Legend:
1. ReqPro Doc(s)
2. ReqPro Req(s)
3. ReqPro Traceab. Info*
4. EA Doc Ref(s)*
5. EA Def(s)
6. EA Req(s)
7. EA Traceab. Info*
8. JUCMNav Diagram
9. EAMF Patterns

Shared File System
SparxSystems EA Repository
EAMF Catalog
ReqPro Reqs Repository

Sparx Systems EA
Req(s) Traceability Info
Def(s) and Req(s) Traceability Info
<definition>all
<requirement>some
<requirement>Business Entity
EAMF Process Patterns
<requirement>Business Entity
EAMF Process Patterns
<requirement>Business Entity
<requirement>Business Use Case
<requirement>Organizational
<requirement>Location
Modeling Constructs To Req, Model Defs Traceability Information
Req(s) Traceability Info

Document New Requirements Traceability
Document Traceability Between Modeling Constructs and Req. Model Definitions

Business Analyst
Business Architect

Map Business Requirements into Requirements Model
Document Traceability between Project Requirements and Req. Model Definitions/Requirements

<analysis input> Project Requirements

ReqPro

Draw Goal and Task Diagram(s)
Use of Sparx Systems EA to Engineer an EAMF Requirements Model

<table>
<thead>
<tr>
<th>Requirements Model Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Bus. Vocabulary Definition</td>
</tr>
<tr>
<td>Strategy Definition</td>
</tr>
<tr>
<td>Concept Definition</td>
</tr>
<tr>
<td>Business Use Case Requirements</td>
</tr>
<tr>
<td>Business Model Requirements</td>
</tr>
<tr>
<td>Business Entity Requirements</td>
</tr>
<tr>
<td>Location Requirements</td>
</tr>
<tr>
<td>Organizational Requirements</td>
</tr>
<tr>
<td>Process Model Requirements</td>
</tr>
<tr>
<td>Business Process Requirements</td>
</tr>
<tr>
<td>Bus/Workflow Rules Reqs</td>
</tr>
<tr>
<td>Requirements Definition</td>
</tr>
<tr>
<td>Functional Requirements</td>
</tr>
<tr>
<td>Non-Functional Requirements</td>
</tr>
</tbody>
</table>

**EAMF Pattern, Product and Enterprise Solutions Catalogs**
- Pattern/Product/Enterprise Solutions Catalog
- Enterprise Glossary
- Enterprise Business Rules
- Enterprise Solution Patterns
- Enterprise Strategies
- Enterprise Projects (e.g., Ent. Worker Services)
- EAMF Catalogs and Enterprise Requirements Model Categories

**EAMF Enterprise Requirements Model**
- Enterprise Projects (EAMF-Compliant)

**Project Requirements Model Categories**
- Analysis input
- Requirements Model Within EA EAMF-Compliant Project Template
- <definition> Project Business Vocabulary
- <definition> Strategy Definition
- <definition> Concept Definition
- <requirement> Business Entity Requirements
- <requirement> Business Use Case Requirements
- <requirement> Business Model Requirements
- <requirement> Requirements Definition
Use of IBM Rational ReqPro for the Requirements Model Engineering Phase

**Enterprise Project Req Model Docs Types are the same as Project Req Model Docs Types**

**Pattern/Product/Enterprise Solutions Catalog**
- Enterprise Glossary
- Enterprise Business Rules
- Enterprise Solution Patterns
- Enterprise Strategies
- Enterprise Projects (e.g., Ent. Worker Services)

**EAMF Catalogs and Enterprise Requirements Model Categories**

**Project Requirements Model Categories**
- Business Use Case Requirements
- Business Entity Requirements
- Business Model Requirements
- Concept Definition
- Functional Requirements
- Non-Functional Requirements
- Process Model Requirements
- Process Requirements
- Location Requirements
- Conceptual Requirements
- Organizational Requirements
- Concept Definition

**RequisitePro Defs & Reqs Types Usage**
- TERM
- SRQ
- BUD
- BUS
- EVE, FEA
- ELT, LNK
- USE
- LOC
- ACT
- BPR
- BUR, WFR
- FNC
- NFE, NFP, NFR
- (PAT, PRD, SOL).CAT
- TERM
- BUR, WFR
- PAT
- SRQ

**RequisitePro Document Types Usage**
- .AGL
- .SRD
- .CDD
- .HUC
- .DUC
- .HBM
- .DBM
- .HRD
- .DRD
- .RMS
- .CAT
- .AGL
- .BDD
- .PAT
- .SRD

**Enterprise Projects**
- Enterprise Projects (e.g., Ent. Worker Services)

**Known Document Type**
- (Known Document Type)
jUCMNav GRL Modeling Constructs
(http://www.scs.carleton.ca/~weiss/papers/MCeTech05.pdf)

(a) GRL Elements

(b) GRL Satisfaction Levels

(c) Link Composition

(d) GRL Links

(e) GRL Contributions Types
jUCMNav GRL Modeling Constructs (continued)

(http://www.jrpit.acs.org.au/jrpit/JRPITVolumes/JRPIT36/JRPIT36.4.259.pdf)

Contribution Links

Correlation Links

Entities

Softgoal
Goal
Task
Resource
Belief
Sample Actor-Dependency Diagram
(Early Requirements Discipline)
Business Architecture Analysis Using EAMF

Requirements Model Engineering

Traceable Artifacts

Software Development Lifecycle Phases

Relationships

Entities

BUC Mdl

Loc. Mdl

Org. Mdl

URN Mdl

Capab. Matrix

Bus. Problem Force Hierarchies

Candidate Bus Pattern Hierarchies

Candidate Bus. Arch. Styles

Candidate Reference Projects

Bus. Arch. Analysis Artifacts

Pattern/Product/Enterprise Solutions Catalog

Enterprise Glossary

Enterprise Business Rules

Enterprise Solution Patterns

Enterprise Strategies

Enterprise Projects (e.g., Ent. Worker Services)

EAMF Catalogs and Enterprise Requirements Model Categories

Project Bus. Vocabulary Definition

Strategy Definition

Concept Definition

Business Objective

Features and Events

Business Entity Requirements

Business Use Case Requirements

Location Requirements

Organizational Requirements

Business Process Requirements

Bus/Workflow Rules Reqs

Functional Requirements

Non-Functional Requirements

Requirements Model Categories

Business Architecture Analysis Using EAMF

Software Development Lifecycle Phases

Traceable Artifacts

Project Bus. Vocabulary Definition

Strategy Definition

Concept Definition

Business Objective

Features and Events

Business Entity Requirements

Business Use Case Requirements

Location Requirements

Organizational Requirements

Business Process Requirements

Bus/Workflow Rules Reqs

Functional Requirements

Non-Functional Requirements

Requirements Model Categories

Pattern/Product/Enterprise Solutions Catalog

Enterprise Glossary

Enterprise Business Rules

Enterprise Solution Patterns

Enterprise Strategies

Enterprise Projects (e.g., Ent. Worker Services)

EAMF Catalogs and Enterprise Requirements Model Categories
UCM Notations Summarized
(http://www.usecasemaps.org/pub/sugarloafplop01.pdf)
<table>
<thead>
<tr>
<th>UCM Notation</th>
<th>Notation Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Point - Path - End Point</td>
<td>Basic path. The basic path is the most basic, complete unit. The path represents scenario flow. Paths connect start points, responsibilities, and end points. A path may have any shape as long as it is continuous (can cross itself). The start points represent preconditions or triggering causes. The end points represent post-conditions or resulting effects.</td>
</tr>
<tr>
<td>Do something</td>
<td>Responsibility point. Represents generic processing (actions, tasks, or functions to be performed). Responsibilities may be bound to a component.</td>
</tr>
<tr>
<td></td>
<td>Direction (optional). In general, the positioning of the start and end points of a path indicate direction. In certain cases, it is useful to show the direction on a complicated map.</td>
</tr>
<tr>
<td>Waiting Place - Continuation Path</td>
<td>Waiting place. Represents a waiting place along a path. Propagation along the path stops at the waiting place until the trigger arrives. Waiting places can be triggered by a trigger path as shown or by the environment.</td>
</tr>
<tr>
<td>Timer - Continuation Path</td>
<td>Timer. A special waiting place that expresses the idea that there is a time limit on waiting. When propagation along the waiting path reaches the timer, the timer is set. Propagation along the continuation path continues if the timer release arrives. Propagation along the timeout path continues if the timeout occurs.</td>
</tr>
<tr>
<td>AND Fork and AND Join</td>
<td>For concurrent paths (two or more).</td>
</tr>
<tr>
<td>OR Fork and OR Join</td>
<td>An OR fork indicates that the path proceeds in only one out of two or more directions. Labels may identify alternative paths or guarding conditions. An OR Join indicates a common causal segment of two or more paths.</td>
</tr>
<tr>
<td>Static stub</td>
<td>Contains only one plug-in (sub UCM), hence enabling hierarchical decomposition of complex maps.</td>
</tr>
<tr>
<td>Dynamic stub</td>
<td>May contain several plug-ins, whose selection can be determined at run-time according to a selection policy (often described with pre-conditions).</td>
</tr>
<tr>
<td>Generic component</td>
<td>Represents an architectural entity.</td>
</tr>
<tr>
<td>Slot</td>
<td>Placeholder for dynamic components as operational units. Dynamic responsibilities can move dynamic components from a path into a slot or out of a slot onto a path.</td>
</tr>
</tbody>
</table>

Create   Delete   Move out   Move into
Inter-Scenario Relationships Design Patterns Used in EAMF BPM Approach


Inter-Scenario Relationships
- Scenario Interactions
  - Mutually Exclusive Scenarios
  - Scenario Composition
  - Scenario Aborting
- Scenario Dependency
- Scenario Clustering
- Concurrent Scenarios
- Waiting Place
- Timed Waiting Place

Scenario Clustering
Concurrent Scenarios

Combining Goal Oriented & Scenario-Based Modeling

(http://www.cs.toronto.edu/km/GRL/from-r2a/fromr2a/straw01.pdf)
Use of Sparx Systems EA to Analyze the Business Architecture

- Project Bus. Vocabulary Definition
- Strategy Definition
- Concept Definition
- Business Use Case Requirements
- Business Model Requirements
- Business Entity Requirements
- Business Process Requirements
- Process Model Requirements
- Business/Workflow Rules Reqs
- Business Objective
- Features and Events
- Requirements Definition
- Location Requirements
- Organizational Requirements
- Functional Requirements
- Non-Functional Requirements
- Requirements Model Categories
- Requirements Model Engineering
- Bus. Arch. Analysis

- Entities
  - BUC Mdl
  - Loc. Mdl
  - Org. Mdl
  - URN Mdl
- Capab. Matrix
- Bus. Problem Force Hierarchies
- Candidate Bus Pattern Hierarchies
- Candidate Bus Arch. Styles
- Candidate Reference Projects
- Bus. Arch. Analysis Artifacts

- Within EAMF Framework
- Within EAMF Enterprise Solutions Catalog

- EAMF Pattern, Product and Enterprise Solutions Catalogs
- EAMF Enterprise Requirements Model
- Enterprise EAMF-Compliant Project

- EAMF Catalogs and Enterprise Requirements Model Categories
- Pattern/Product/Enterprise Solutions Catalog
  - Enterprise Glossary
  - Enterprise Business Rules
  - Enterprise Solution Patterns
  - Enterprise Strategies
  - Enterprise Projects (e.g., Ent. Worker Services)
Use of Sparx Systems EA to Design the Business Architecture

Bus. Arch. Engineering

Analysis

Design

Relationships

Entities

BUC Mdl

Loc. Mdl

Org. Mdl

URN Mdl

Capab. Matrix

Bus. Problem

Force

Hierarchies

Candidate

Bus Pattern

Hierarchies

Candidate

Bus. Arch.

Styles

Candidate

Reference

Projects

Bus. Arch.

Analysis Artifacts

Domain Model

BUC Mdl

Loc. Mdl

Org. Mdl

Process Mdl

Bus. Arch. Model

Bus. Arch. Patterns/Reuse Constraints

Reference Business Arch(s)

Reference EAMF Project(s)

Within EAMF Framework

Within EAMF Enterprise Solutions Catalog

<view> Analysis within <perspective> Business Architecture of EAMF-Compliant Project Template

<view> Design within <perspective> Business Architecture of EAMF-Compliant Project Template
From Requirements Engineering to BA Engineering Using EAMF

Glossary
Stakeholder Requests
Business Objectives
Features and Events
Use Cases
Location
Organization
Process
Workflow Rules
Business or Functional Requirements
Non Functional Requirements
Requirements Types

Requirements Engineering

Project Bus. Vocabulary Definition
Strategy Definition
Concept Definition
Business Objective
Features and Events
Business Entity Requirements

Requirements Model Engineering

Business Use Case Requirements
Location Requirements
Organizational Requirements
Process Model Requirements
Bus/Workflow Rules Reqs
Functional Requirements
Non-Functional Requirements
Requirements Definition

Bus. Arch. Engineering Analysis

Relationships
Entities
BUC Mdl
Loc. Mdl
Org. Mdl
URN Mdl
Capab. Matrix
Bus. Problem Force Hierarchies
Candidate Bus Pattern Hierarchies
Candidate Bus Arch. Styles
Candidate Reference Projects

Bus. Arch. Engineering Design

Domain Model
BUC Mdl
Loc. Mdl
Org. Mdl
Process Mdl
Bus. Arch. Model
Bus. Pattern Hierarchies
Reference Business Arch(s)
Reference EAMF Project(s)
Bus. Arch. Patterns/Reuse Styles
Constraints
Candidate Projects
Bus. Arch. Analysis Artifacts
Bus. Arch. Design Artifacts

Traceable Artifacts

Requirements

Software Development Lifecycle Phases

Proj. Bus. Directives
Functional Requirements
Non-Functional Requirements

Pattern/Product/Enterprise Solution Reqs
Enterprise Business Vocabulary Reqs
Business Rules Reqs Repository
Enterprise Solution Patterns Requirements
Business Strategy and Innovation Reqs
Enterprise Project Requirements
Enterprise Requirements

EAMF Catalogs and Enterprise Requirements Model Categories

Enterprise Glossary
Enterprise Business Rules
Enterprise Solution Patterns
Enterprise Strategies
Enterpris Projects (e.g., Ent. Worker Services)
Actor/Dependency and Goal/Plan Diagram (Late Requirements Discipline)
Composite UCM Root Map for ESLM
Plug-In Map for HandleAllEvents in Root Map
Generic Plug-In Map for HandlePolicyServiceEvents and HandleInvoiceServiceEvents
Generic Plug-In Map for ProcessAllRequests Stub in Root Map
Gather Information Involves
- Interviews
- Joint Application Design (JAD)
- Questionnaires
- Document Analysis
- Observation

Tools-Driven Approach Involves
- Requirements Engineering Tools
- Users and Goals Elicitation Tool
- Use Case Maps
Course Assignments

- Individual Assignments
  - Reports based on case studies / class presentations
- Project-Related Assignments
  - All assignments (other than the individual assessments) will correspond to milestones in the team project.
  - As the course progresses, students will be applying various methodologies to a project of their choice. The project and related software system should relate to a real-world scenario chosen by each team. The project will consist of inter-related deliverables which are due on a (bi-) weekly basis.
  - There will be only one submission per team per deliverable and all teams must demonstrate their projects to the course instructor.
  - A sample project description and additional details will be available under handouts on the course Web site.
Team Project

- Project Logistics
  - Teams will pick their own projects, within certain constraints: for instance, all projects should involve multiple distributed subsystems (e.g., web-based electronic services projects including client, application server, and database tiers). Students will need to come up to speed on whatever programming languages and/or software technologies they choose for their projects - which will not necessarily be covered in class.
  - Students will be required to form themselves into "pairs" of exactly two (2) members each; if there is an odd number of students in the class, then one (1) team of three (3) members will be permitted. There may not be any "pairs" of only one member! The instructor and TA(s) will then assist the pairs in forming "teams", ideally each consisting of two (2) "pairs", possibly three (3) pairs if necessary due to enrollment, but students are encouraged to form their own 2-pair teams in advance. If some students drop the course, any remaining pair or team members may be arbitrarily reassigned to other pairs/teams at the discretion of the instructor (but are strongly encouraged to reform pairs/teams on their own). Students will develop and test their project code together with the other member of their programming pair.
Document Transformation methodology driven approach

- Strategy Alignment Elicitation
  - Equivalent to strategic planning
    - i.e., planning at the level of a project set

- Strategy Alignment Execution
  - Equivalent to project planning + SDLC
    - i.e., planning at the level of individual projects + project implementation

Build a methodology Wiki & partially implement the enablers

Apply transformation methodology approach to a sample problem domain for which a business solution must be found

Final product is a wiki/report that focuses on
  - Methodology / methodology implementation / sample business-driven problem solution
• Document sample problem domain and business-driven problem of interest
  • Problem description
  • High-level specification details
  • High-level implementation details
  • Proposed high-level timeline
Assignments & Readings

- **Readings**
  - Slides and Handouts posted on the course web site
  - Textbook: Part Two-Chapter 5

- **Individual Assignment (due)**
  - See Session 3 Handout: “Assignment #1”

- **Team Project #1 (ongoing)**
  - Team Project proposal (format TBD in class)
  - See Session 2 Handout: “Team Project Specification” (Part 1)

- **Team Exercise #1 (ongoing)**
  - Presentation topic proposal (format TBD in class)

- **Project Frameworks Setup (ongoing)**
  - As per reference provided on the course Web site
Any Questions?
Next Session: Introduction to Software Analysis and Design