




**lero** THE IRISH SOFTWARE  
ENGINEERING RESEARCH CENTRE

## Architecture Knowledge Management: Challenges, Approaches, Tools

M. Ali Babar<sup>1</sup> & Ian Gorton<sup>2</sup>  
<sup>1</sup>Senior Researcher, Lero, Ireland  
<sup>2</sup>Chief Architect, Data Intensive Computing,  
PNNL, USA



**Introductions** THE IRISH SOFTWARE  
ENGINEERING  
RESEARCH CENTRE

- **Tell us**
  - Who you are
  - Educational / Professional Background
  - Research interests
  - Past experiences with architecture knowledge management
  - Expectations of tutorial

Date 5/22/2007 ICSE 2007, Minneapolis, USA 2

## Tutorial Goals



- Comprehend the role and challenges involved in managing software architecture knowledge
- Understand domain modeling techniques for characterizing architecture knowledge
- Gain knowledge of the process for capturing and managing architecture knowledge
- Learn how to utilize architecture knowledge to improve the architecture design and evaluation process
- Appreciate the processes and tools required to build and manage a knowledge repository

## Overview



- Section 1 – Architecture knowledge management
- Section 2 – A Tool and Approaches to Managing architecture knowledge
- Section 3 - Domain modeling for architecture knowledge
- Section 4 - Utilizing architecture knowledge during design and evaluation

## Architecture Knowledge Management



- What is architecture knowledge?
- What is architecture knowledge management?
- Why is architecture knowledge management important?
  - Challenges
  - Strategies
  - Expected benefits
- State-of-the art
  - Tooling support
  - R&D
- Case study description

## Architecture Knowledge and its Management



- Architecture knowledge

Software architecture design knowledge can be characterized by information that supports software architecture processes.
- Architecture knowledge management

Software architecture knowledge management is an approach to improving software architecture process outcomes by introducing practices for identifying and capturing architecture knowledge and expertise, and making it available for reuse across projects.

## Architecture Knowledge Management Issues



- Unavailability of architecture design knowledge
  - System evolution becomes hard
  - Difficult to identify design errors
- Use of COTS without fully understanding the assumptions
- Situation is worst in case of FLOSS components
- Lack of support in terms of practices and tools
  - What types of architecture knowledge are useful?
  - How to store and manage the knowledge?
  - Make knowledge capture cost-effective



Date 5/22/2007

ICSE 2007, Minneapolis, USA

7

## Categories of Factors



- Organization
  - Leadership
  - Processes
  - Platform
  - Structure/Culture
- Human/Behavioral
  - Training
  - Common language
  - Culture
  - Motivation
- Technological
  - KMSS
  - Communication
  - Collaboration
  - Coordination

Date 5/22/2007

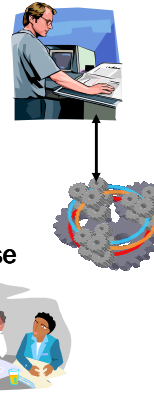
ICSE 2007, Minneapolis, USA

8

## Knowledge Management Strategies



- **Codification**
  - High quality, reliable, and rapid use of knowledge
  - High IT investment
  - Rewards for sharing & reusing codified knowledge
- **Personalization**
  - Connecting people & channelling individual expertise
  - Moderate IT investment
  - Reward for sharing knowledge face to face
- **Hybrid approach**
  - Codify organization-wide knowledge
  - Personalize local & context specific knowledge



Date 5/22/2007

ICSE 2007, Minneapolis, USA

9

## Knowledge Management (KM) Task Model



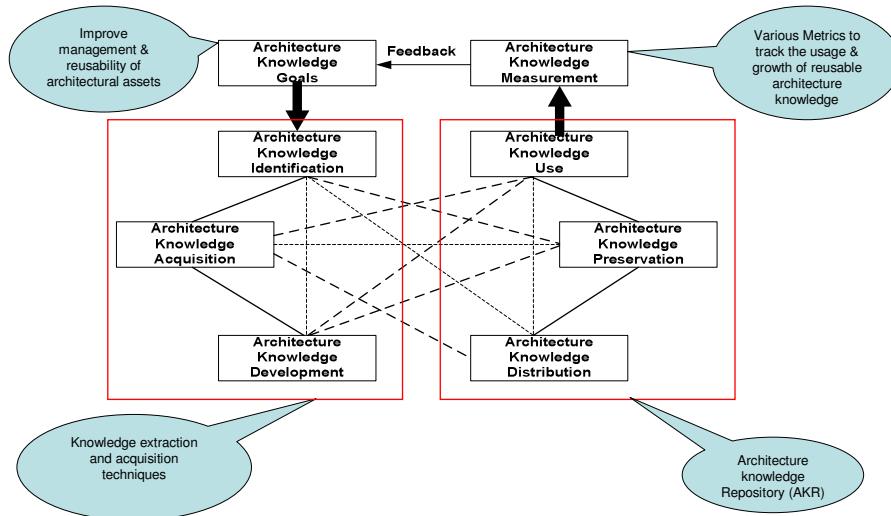
- **KM task model consists of eight tasks**
- **Strategic tasks**
  - Architecture knowledge goals
  - Architecture knowledge measurement
- **Operational tasks**
  - Architecture knowledge identification
  - Architecture knowledge acquisition
  - Architecture knowledge development
  - Architecture knowledge distribution
  - Architecture knowledge use
  - Architecture knowledge preservation

Date 5/22/2007

ICSE 2007, Minneapolis, USA

10

## Supporting Knowledge Management Tasks



Date 5/22/2007

ICSE 2007, Minneapolis, USA

11

## Metrics for Knowledge Management

Some of the measures that can be used for most of the KM initiatives

### Outcome

- Time, money or personal time saved
- Percentage of successful project compared to those before KM
- Time and number of employees trained

### Output

- Usefulness surveys where users evaluate how useful KM initiative have been in helping them accomplish their objective
- Usage anecdotes where users describe how the initiative has contribute to their daily tasks and overall efficiency and effective of organisation

### System

- Dwell time per page or section
- Latency (response time)
- Number of downloads
- Number of site access
- Usability survey
- Frequency of use
- Navigation path analysis
- Number of helpdesk calls
- Number of users
- Frequency of use and/or update
- Percentage of total employees using system

Date 5/22/2007

ICSE 2007, Minneapolis, USA

12

## Benefits of Architecture Knowledge Management



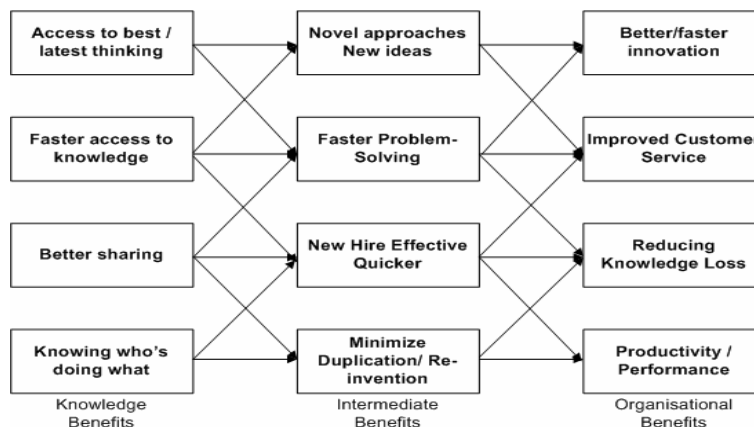
- **Expected benefits:**
  - Capture rationale for architecture decisions
  - Help build architectural capabilities
  - Improve architectural reusability
  - Support better quality decisions
  - Minimize architecture risks
  - Minimize reoccurrence of design mistakes
  - Avoid dependency on key individuals
  - Gain competitive advantage
  - Encourage best architectural practices
  - Improve efficiency of architectural processes
  - Support case-based reasoning

Date 5/22/2007

ICSE 2007, Minneapolis, USA

13

## Knowledge Management Benefits Tree



Source: David Skyrme associates

Date 5/22/2007

ICSE 2007, Minneapolis, USA

14

## A Few Useful Tools



- **Protégé**
  - Creation & manipulation of Ontologies for domain modeling
  - Knowledge acquisition & management
- **Chimaera**
  - Creating & maintaining distributed Ontologies
  - Support for loading, browsing, reorganizing Ontologies
- **Swoop**
  - Hypermedia inspired Ontology editor
- **BRedB**
  - Support for capturing and contextualizing knowledge
  - Knowledge management and reuse

## R&D in Architecture Knowledge



- Grady Booch – Handbook of software architecture
- Philippe Kruchten – Design decision ontology and use case model for tool support
- GRIFFIN Project – Architecture knowledge modeling, representation, and acquisition approaches
- Anton Jansen & Jan Bosch – Describing architecture as design decisions in Archium
- Antony Tang & Jun Han – Architecture rationale and elements linkage (AREL) for capturing and traversing rationale to support design decision reasoning



## A Preview of our R&D



- Repository of architecture knowledge
  - templates for capturing design elements and diagrams
- Tool support for design decision Ontology and Use Cases
- Architecture knowledge model, Use case model, and knowledge acquisition approaches
- Describing architecture as design decisions using templates – different levels of abstractions
- Our approach complements AREL's UML based rationale with templates and search mechanism

Date 5/22/2007

ICSE 2007, Minneapolis, USA

17

## Case Study Description



- Organizational context
- Domain description
- Software architecture evaluation challenges
- Software architecture knowledge management initiative
- Logistics
- Progress so far!

Date 5/22/2007

ICSE 2007, Minneapolis, USA

18

## Summary of Section 1



- What is architecture knowledge?
- What is architecture knowledge management?
- Why is architecture knowledge management important?
  - Expected benefits
  - Challenges
  - Strategies
- State-of-the art
  - Tooling support
  - R&D
- Case study description

## A Tool and Approaches for Managing Knowledge



- An Introduction to BRedB – An Architecture Knowledge Management Infrastructure
  - Architecture
  - Features to support architecture design, documentation, and evaluation activities
- Capturing software architecture knowledge
  - Approaches
  - Advantages and disadvantage
- Using BRedB on the case study

## Tool for Managing Architecture Knowledge

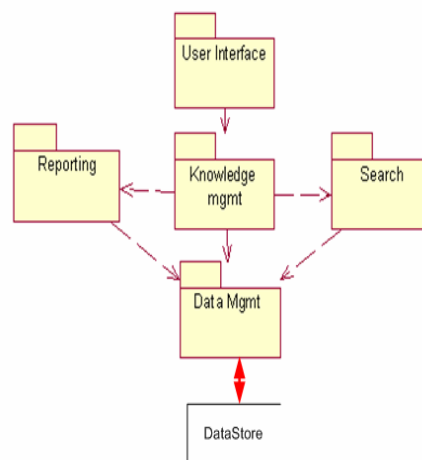


- BRedB: A Process-based Software Architecture Knowledge Management Infrastructure
  - A model of architecture knowledge management
  - Templates for capturing and structuring architecture knowledge
  - Repository of architectural knowledge and experiences
  - Resource for software architects
  - Source of adaptive software architecture processes
  - Support for owning technical and process knowledge

## Component View of PAKME Architecture



- User Interface
- Knowledge Management
- Knowledge Search
- Reporting
- Data Management
- Knowledge Repositories



## GUI for Start Using BRedB



Start Collaborative Tools Contact Management Project Management Content Production Corporate Library Project Collaboration Configuration

► Home | Knowledge-Based | Project-Based | Search

### Project Collaboration

#### Knowledge Based

The knowledge-based involves capturing various experience artefacts that users may *create, modify, delete* and *search*. These artefacts include the following:

- General Scenarios
- Patterns
- Analysis Model
- Architecturally Significant Requirement

#### Project-Based

The project-based involves building database of artefacts for specific projects to support their software architecture evaluation. This can be achieved through creating new artefacts and/or extracting them from the knowledge-based. All newly created artefacts will be added to the knowledge repository. The project-based artefacts include:

- General Scenario
- Concrete Scenario
- Architecturally Significant Requirement
- Quality Factor
- Architecture Decision
- Alternative Decision
- Finding

#### Search

Search provides essential functionalities to allow users to seek for their desired artefacts. The two different types of search are:

- Field-based
- Keyword-based

Date 5/22/2007

ICSE 2007, Minneapolis, USA

23

## Support for Architecture Design



- Support for case-based reasoning by human expert
- Repository of reusable architectural artifacts
- Capture/access rationale for design decisions
- Catalogue of architecture and design patterns/tactics
- Search architectural artifacts and knowledge

Date 5/22/2007

ICSE 2007, Minneapolis, USA

24

Date 5/22/2007

## Searching Design Decision Cases



hipergate :: Attach Design Option - Microsoft Internet Explorer provided by Desktop Services, I. T. Division

**Design Option Search and Listing for Web Tier**

**Search Criteria:**

Keywords to Search for:

Application Type:

Project Domain:

Design Option is: ☐ Used, ☐ Considered

Display Result Sort by:

The following Design Options are current:

Design Option	Description	Project	Number of occurrences used in Architecture Decisions	Show Selected Application Type Domain (only)	Decision	Inspire New	Delete Option
Application Type Domain							
All Design Options (unsorted)							
Unused Design Options (only)							
Percentage Match							

[List all Design Options](#) OR [Add New Design Options](#)

\*Delete function will only remove the Design Option from the considered list, not from knowledge database.

Date 5/22/2007

ICSE 2007, Minneapolis, USA

27

## Using a Design Decision Case



hipergate :: Design Option Listing - Microsoft Internet Explorer

**Design Option Search and Listing for "Database Server"**

**Search Criteria**

Keywords to Search for:  OR  OR

Display Results and Sort by:

The following Design Options are current:

Design Option Name	Description	Project	Number of occurrences used in Architecture Decisions	Show Selected Application Type Domain (only)	Decision	Inspire New	Delete Option
Database Server	Introduce a dedicated server as a database service provider. This reduces the workload on other systems and offers a centralized database. <a href="#">[more...]</a>	1	BCS Project	research	100 %	<a href="#">Modify</a>	<a href="#">Delete</a>
Application Server	Have a dedicated Application Server to provide application service to the clients. Hence reduces the workload to other parts of the system. <a href="#">[more...]</a>	0	None	None	100 %	<a href="#">Modify</a>	<a href="#">Delete</a>
Multiple Server System	Introduce different servers to provide different services for the client. Hence would greatly reduce the workload the current servers. <a href="#">[more...]</a>	0	None	None	100 %	<a href="#">Modify</a>	<a href="#">Delete</a>

Number of Results found from search: 15

[List all Design Options](#)  
[Add New Design Options](#)

Date 5/22/2007

ICSE 2007, Minneapolis, USA

28

## Navigating the Knowledge Base



The screenshot shows the hipergate web application interface. On the left, the 'Advanced Search' page is visible with various filters. A red arrow points from the 'Pattern' tab in the search results to the 'View Pattern' page. The 'View Pattern' page displays details for the 'Business Delegate' pattern, including its description, context, problem, solution, parent, forces, tactics, affected attributes, general scenario, and usage examples.

Date 5/22/2007

ICSE 2007, Minneapolis, USA

29

## Template for Capturing and Representing Patterns



The screenshot shows the 'View Pattern' page for the 'Business Delegate' pattern. The page is structured as follows:

<b>Name</b>	Business Delegate
<b>Type</b>	Design pattern
<b>Description</b>	This pattern reduces coupling between tiers and provides an entry point for accessing the services that are provided by another tier. It may also provide results caching for common requests to improve performance. It typically uses a Service Locator to locate a service.
<b>Context</b>	In a distributed system, clients may be exposed to the complexity of dealing with the distributed components that provide services.
<b>Problem</b>	Presentation-tier components interact directly with business services, which exposes the implementation details of the services to the clients. Such a direct interaction makes the clients vulnerable to any changes in the business services.
<b>Solution</b>	Use Business Delegate to reduce coupling between presentation-tier clients and business services. The Business Delegate hides the underlying implementation details of the business service.
<b>Parent</b>	No Parent Available
<b>Forces</b>	1) <a href="#">Business Service</a>
<b>Tactics</b>	1) <a href="#">Delegate Proxy</a> 2) <a href="#">Delegate Adapter</a>
<b>Affected Attributes</b>	<p><b>Positively</b></p> <p>1) <a href="#">Performance</a></p> <p><b>Negatively</b></p> <p>1) <a href="#">Complexity</a> 2) <a href="#">Introduce new layer</a></p>
<b>General Scenario</b>	1) <a href="#">BD-S6</a> 2) <a href="#">BD-S2</a>
<b>Usage Examples</b>	1) <a href="#">E-Commerce</a>

Date 5/22/2007

ICSE 2007, Minneapolis, USA

30

## Support for Architecture Documentation



- Templates for documenting design decisions
- Store architectural models and documents
- Support for standards such as IEEE 1471-2000
- Represent architectural decisions using views
- Attach process knowledge to architectural artifacts

Date 5/22/2007

ICSE 2007, Minneapolis, USA

31

## Structuring and Representing Architecture Decisions



hipergate :: View Architecture Decision - Business Component Tier - Microsoft Internet Explorer ...

**View Architecture Decision**

<b>Name</b>	Business Component Tier		
<b>Concrete Scenario</b>	<a href="#">Changes to the platform occur</a>		
<b>Quality Factor</b>	<a href="#">Modifiability</a>		
<b>Description</b>	The business components constitute the core business logic for the application. The business components are realized by Enterprise JavaBeans, the software component model supported by J2EE.		
<b>Comment</b>	This frees the application programmer from cluttering the business logic with code to handle system and environmental issues.		
<b>Architecture Description</b>	Functions/processes are divided between clients and server.		
<b>Contractor</b>	<a href="#">NICTA</a>		
<b>Compliant</b>	Complied		
<b>Ranking</b>	5		
<b>Decision Considered</b>	<a href="#">JAVA Enterprise Design</a>		
<b>Architecture Decision</b>	<b>Present Rationale</b>		
	<b>Date</b>	<b>Time</b>	<b>View Rationale</b>
	2006-12-24	13:23:55	<a href="#">[detail...]</a>
<b>Design History</b>	<b>Past Rationales</b>		
	<b>Date</b>	<b>Time</b>	<b>View Rationale</b>

Date 5/22/2007

ICSE 2007, Minneapolis, USA

32



## Modifying an Architecture Decisions

hipergate :: Edit Architecture Decision - Compliance with the High Level Architecture (HLA) - Mi...

**Edit Architecture Decision**

**Name** Compliance with the High Level Architecture

**Concrete Scenario** Reconfigure Wargame 2000

**Quality Factor** Reconfigurability

**Description** The HLA was developed under the leadership of the Defense Modeling and Simulation Office (DMSO) to support reuse and interoperability across the large numbers of different types of simulations developed and maintained by the DoD.

**Comment** HLA compliance in Wargame 2000 is achieved via an HLA gateway to interface with other systems that are HLA compliant.

**Architecture Description** Functions/processes are divided between clients and server.

**Contractor** NICTA

**Compliant** Complied

**Ranking** No ranking entered

**Chosen Design Option** Provide a virtual gaming site

**Select Design Option** Provide a virtual gaming site

**Attach Design Option** [View/Attach More Design Options](#)

**Attached Documents** (Please add documents.) [Add Documents](#)

**Architecture Relationship** (Please add relationships.) [Add Architecture Relationships](#)

**Keyword** (Please add keywords.) [Add Keywords](#)

[Save](#) [Cancel](#)

Date 5/22/2007

ICSE 2007, Minneapolis, USA

33

## Modifying a Pattern

hipergate :: Edit Pattern - Microsoft Internet Explorer

**Edit Pattern**

**Name** Value Object

**Type** Design pattern \* (create Pattern Type if none)

**Description** This pattern provides best techniques and strategies to exchange data across tiers. It attempts to reduce the network overhead by minimizing the number of network calls to get data from the business tier.

**Context** Application clients need to exchange data with components residing on multiple tiers. Usually data is exchanged with enterprise beans.

**Problem** A client object has to make multiple calls to a business object's method to obtain all the attribute values. Calls are usually remote, which causes network overhead and degraded performance.

**Solution** Use a value object to encapsulate the business data, which can be sent to and obtained from enterprise bean in a single call. The value object is constructed, populated, and passed to the clients by the bean.

**Child of**

**Forces** (Please add factors affecting the problem and solution.) [Add Forces](#)

**Available Tactics** (Please add tactics which are used to implement the solution.) [Add Tactics](#)

**Affected ASR** [Add Positively Affected](#) [Add Negatively Affected](#)

**General Scenario** (Please add supported general scenarios.) [Add Supported General Scenario](#)

**Usage Examples** (Please add examples of usage to solve the problem.) [Add Usage Example](#)

**Keywords** (Please add keywords.) [Add Keywords](#)

Date 5/22/2007

ICSE 2007, Minneapolis, USA

34

## Support for Architecture Evaluation



- Repository of general scenario to support QAWs
- Capture rationale and contextual information surrounding design decisions
- Search and view rationale for previous design decisions
- Documenting findings of evaluating design decisions
- Categorize findings in suitable risk themes
- Generate evaluation reports for the management

Date 5/22/2007

ICSE 2007, Minneapolis, USA

35

## General Scenario Captured in BRedB



General Scenario Listing				
<a href="#">New</a> <a href="#">Delete</a> <a href="#">Accept</a> <a href="#">Reject</a> <a href="#">Name</a> <input type="text"/> <a href="#">Search</a> <a href="#">Discard</a> <a href="#">Show</a> 50 results				
Proposed General Scenarios				
<a href="#">Previous</a>				
Name	Description	Source	Date Entered	Logs
SEC-1	QVS shall accept online payments for the services that means the transactions between the QVS and financial institutions must be protected.	User-Defined	Sun 24 Dec 2006 19:47	
SEC-2	QVS provides secured storage to customers' credit details and other information.	User-Defined	Sun 24 Dec 2006 19:47	
SEC-3	QVS shall be able to identify different users and verify their access privileges according to their memberships of different user groups.	User-Defined	Sun 24 Dec 2006 19:49	
SEC-4	QVS shall be able to detect and prevent Denial Of Service (DOS) attacks. The system shall be able to run reliably most of the time.	User-Defined	Sun 24 Dec 2006 19:49	
SEC-5	QVS is an evolving system that shall be easily modifiable to introduce changes in the security policy and other security checks.	User-Defined	Sun 24 Dec 2006 19:50	
Accepted General Scenarios				
<a href="#">Previous</a>				
Name	Description	Source	Date Entered	Logs
Performance	Require bounded response time and/or certain system throughput.	User-Defined	Tue 19 Dec 2006 11:28	
<a href="#">Changing the hardware platform</a>	Change physical location of service with minimal impact on the rest of the system.	Pattern	Tue 19 Dec 2006 10:23	
<a href="#">Changes number of users</a>	Number of users changes while maintaining other qualities such as performance.	Pattern	Tue 19 Dec 2006 10:50	
<a href="#">Change the implementation</a>	Implementation details change without affecting much of the rest of the system.	Pattern	Tue 19 Dec 2006 11:49	
<a href="#">Addition of functionality</a>	Addition of functionality without impacting the rest of the system.	Pattern	Tue 19 Dec 2006 14:49	
<a href="#">Availability</a>	An internal or external component fails and the system is able to recognize the failure and has strategies to compensate for the fault.	Pattern	Tue 19 Dec 2006 12:04	
<a href="#">Not prepared event</a>	An event arrives at the system for which it was not prepared.	User-Defined	Tue 19 Dec 2006 15:20	
Q	Q	User-Defined	Thu 21 Dec 2006 10:12	
Rejected General Scenarios				
<a href="#">Previous</a>				
Name	Description	Source	Date Entered	Logs
BC-S3	Changes in the business services implementation shall not require corresponding changes in their clients residing in other tier.	User-Defined	Fri 26 Jan 2005 11:02	
BC-S1	Presentation tier components shall not be exposed to the implementation details of the business services they use.	Pattern	Fri 26 Jan 2005 11:01	
BC-S6	Different clients, such as devices, web clients, and thick clients need access to business services.	Pattern	Fri 26 Jan 2005 11:16	
BC-S4	Services calls across network or tiers shall be minimized to avoid degraded performance.	Pattern	Fri 26 Jan 2005 11:02	

Date 5/22/2007

ICSE 2007, Minneapolis, USA

36

## Utility Tree of Concrete Scenarios



The screenshot shows two windows from the hipergate application. The left window, titled "hipergate :: Utility Tree for BCS Project - Microsoft Internet Explorer", displays a tree structure under "Performance" > "Response [time < 1 second]". It lists several scenarios, with "Internode message transfer completes < 1 second" highlighted. The right window, titled "hipergate :: View Concrete Scenario - Internal...", shows the details for this selected scenario.

View Concrete Scenario	
Name	Internode message transfer
Description	Internode message transfer completes < 1 second
Quality Factor	Response [time < 1 second]
Complexity Level	High
Importance	Medium

Date 5/22/2007

ICSE 2007, Minneapolis, USA

37

## Viewing Details about a Concrete Scenario



The screenshot shows the "View Concrete Scenario" window in Microsoft Internet Explorer. It displays a table with various attributes for a scenario named "Run simulations with debug enabled".

Name	Run simulations with debug enabled.				
Description	Run simulations with debug enabled.				
Quality Factor	Meet real-time requirements				
Complexity Level	Low(Default)				
Importance	Low(Default)				
Context					
Stimulus					
Response					
Source of Stimulus					
Date Proposed	Tue 19 Dec 2006 16:42				
Status	Proposed				
User	Administrator				
General Scenario					
Analysis Model					
Classification	Unclassified(Default)				
References					
Documents	<table border="1"> <thead> <tr> <th>Name</th> <th>Created By</th> </tr> </thead> <tbody> <tr> <td>AnalyzingEnterpriseJavaBeans.pdf</td> <td>Administrator</td> </tr> </tbody> </table>	Name	Created By	AnalyzingEnterpriseJavaBeans.pdf	Administrator
Name	Created By				
AnalyzingEnterpriseJavaBeans.pdf	Administrator				
Tactics	1) Tag View Management Strategy				
Findings	No Finding Associated				

Date 5/22/2007

ICSE 2007, Minneapolis, USA

38

## Documenting and Viewing Findings

The screenshots show the hipergate application interface for managing findings. The 'New Finding' window includes fields for Concrete Scenario (BCS), Architecture Decision (ABCS), Chosen Design Option (B), Description, Complied (Not Complied), and Ranking (No ranking entered). The 'Edit Finding' window shows the same fields with updated values: Concrete Scenario (BCS), Architecture Decision (ABCS), Design Option Chosen (B), Complied (Not Complied), Ranking (5), and Description (BCS-ABCS-B-F). It also includes a 'Revision Reason' field and buttons for 'Attach Documents' and 'Risks/Non-Risks'. The 'View Finding' window displays a table of findings with columns for Concrete Scenario, Architecture Decision, Chosen Design, Finding Description, Complied, Ranking, and Documents.

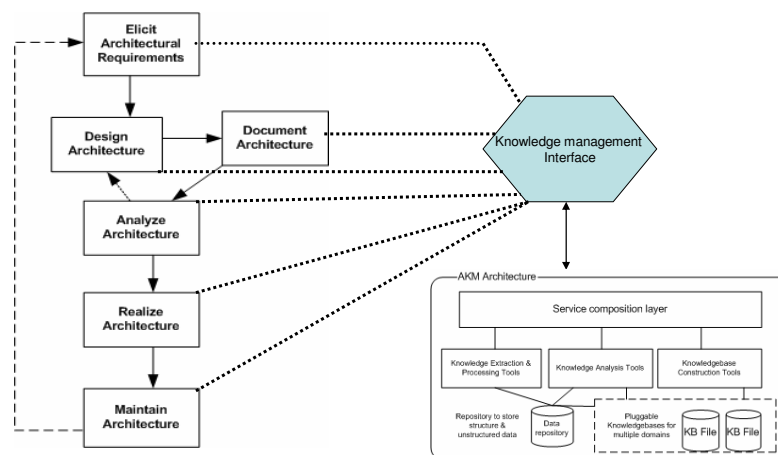
Concrete Scenario	Architecture Decision	Chosen Design	Finding Description	Complied	Ranking	Documents
BCS	ABCS	B	BCS-ABCS-B-F	NOT Complied	5	None Associated

Date 5/22/2007

ICSE 2007, Minneapolis, USA

39

## Support for Architecture-Centric Development

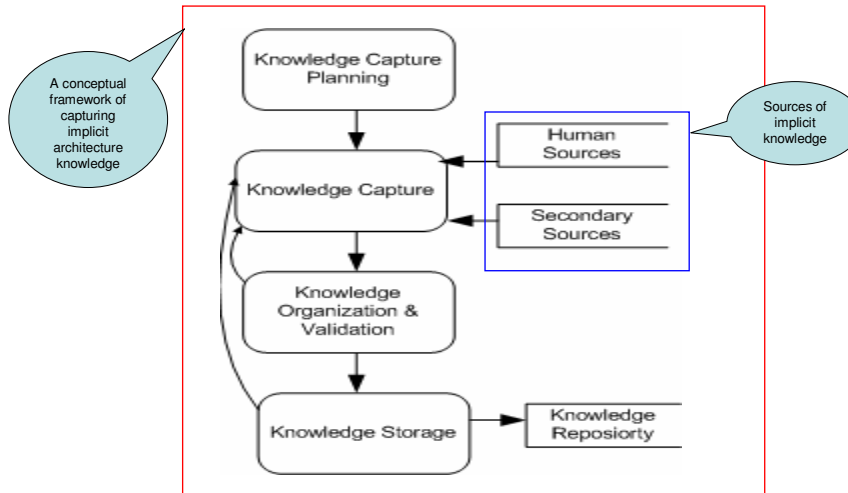


Date 5/22/2007

ICSE 2007, Minneapolis, USA

40

## Capturing Architecture Knowledge



## Source of Architecture Knowledge

- Human sources
  - Stakeholders, BA, architects, developers
- Secondary sources
  - Design patterns, books, case studies

## Knowledge Acquisition from Human Sources



Capturing  
architecture  
knowledge from  
human sources

### Individual Knowledge Acquisition Techniques

Interviewing  
Questionnaire  
Observation  
Protocol analysis  
Repertory grid analysis

### Team Knowledge Acquisition Techniques

Brainstorming  
Architecture reviews  
Focus group interviews  
Delphi technique  
Group repertory grid analysis  
Group support systems

Date 5/22/2007

ICSE 2007, Minneapolis, USA

43

## Individual Knowledge Acquisition Approaches



- Interviewing
- Questionnaires
- Observations
- Prototype analysis
- Repertory grid analysis

Date 5/22/2007

ICSE 2007, Minneapolis, USA

44

## Team Knowledge Acquisition Approaches



- Brainstorming
- Nominal group technique (NGT)
- Focus group interviews
- Delphi technique
- Group repertory grid analysis
- Architecture reviews

Date 5/22/2007

ICSE 2007, Minneapolis, USA

45

## Disadvantages



- Individual Knowledge Acquisition Approaches
  - Time consuming
  - Subjective opinion or interpretation
  - Need considerable domain knowledge
  - May result in fabricated information because of defensive attitude
  - Inconsistent knowledge because of integration issue
- Team Knowledge Acquisition Approaches
  - Expensive activities with multiple logistical issues
  - Require an expert moderator
  - May cause non-participation and conflicts

Date 5/22/2007

ICSE 2007, Minneapolis, USA

46

## A Comparative View of Other Approaches

Method category	Description	Advantages	Disadvantages
Method of familiar tasks	Analysis of the tasks that the expert usually performs.	The expert feels comfortable	Can be fairly time-consuming
Structured and unstructured interviews	The expert is queried with regard to knowledge of facts and procedures.	For a first-and second pass at a data base, it can generate much information.	Typically very time consuming
Limited information tasks	A familiar task is performed, but the expert is not given certain information that is typically available.	Can be tailored to extract information on selected sub-domains of knowledge	Expert feels uncomfortable and is hesitant to make judgements
Constrained processing tasks	A familiar task is performed, but the expert must do so under time or other constraints.	Can be tailored to extract information on selected sub-domains of knowledge, or on the expert's strategies	Expert feels uncomfortable and is hesitant to make judgements.
Method of "tough cases"	Analysis of a familiar task that is conducted for a set of data that presents a "tough case" for the expert.	Can yield information about refined reasoning	Occur unpredictably, the knowledge acquirer may not be present

Source: Experimental Psychology by Robert R. Hoffman, AI Magazine, summer 1987.

Date 5/22/2007

ICSE 2007, Minneapolis, USA

47

## Pattern-Based Architecture Knowledge

- Exploiting relationships among scenarios, quality attributes, and patterns as architecture knowledge
- Patterns widely codified
  - Links among scenarios, quality attributes and patterns
- We have observed that patterns
  - Include implicit scenarios and links between quality attributes and patterns
  - Provide a previously untapped source of reusable architecturally significant information
- Extracting the information
  - Relatively time-consuming and required expertise

Date 5/22/2007

ICSE 2007, Minneapolis, USA

48



## Mining Patterns for Architecture Knowledge



- A mechanism for extracting (*mining*) architecture data from patterns
  - General process model
    - Steps involved in mining patterns with guidelines
  - Templates to record
    - Generic architecture information
    - Project related information relating to concrete scenarios
- Intended to reduce the time and expertise needed to extract architecturally significant information from patterns

Date 5/22/2007

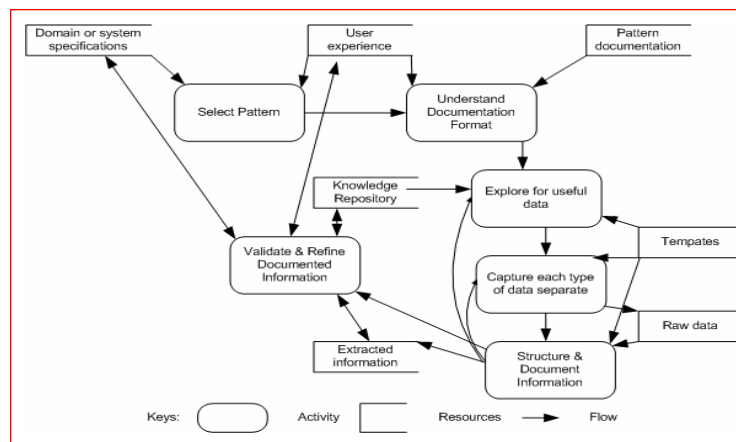
ICSE 2007, Minneapolis, USA

49

## Pattern-Mining Process Model



Process model for mining patterns for architecture knowledge



Date 5/22/2007

ICSE 2007, Minneapolis, USA

50

## Pattern-Mining Process



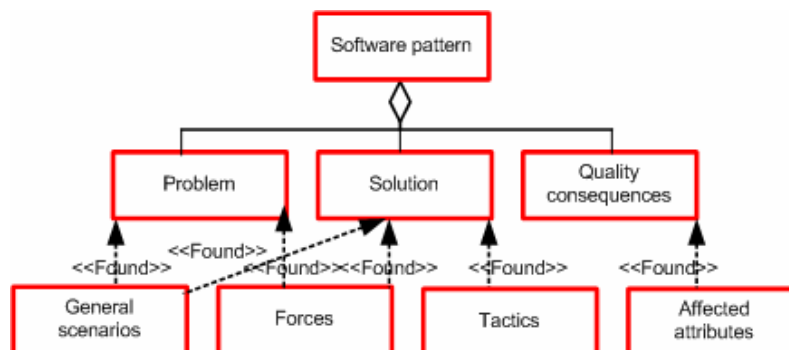
- Select software pattern to be explored for architectural information
- Understand the pattern documentation format
- Identify architectural information described in a pattern's documentation
- Capture each type of information separately
- Organise the extracted information using the provided template
- Validate and refine documented information

Date 5/22/2007

ICSE 2007, Minneapolis, USA

51

## Diagrammatic Guidelines for Mining Pattern



Date 5/22/2007

ICSE 2007, Minneapolis, USA

52

## Documenting the Extracted Information

Pattern Name: <i>Name of the pattern</i>		Pattern Type: <i>Architecture, design, or style</i>	
Description		<i>A brief description of the pattern.</i>	
Context		<i>The situation for which the pattern is recommended.</i>	
Problem		<i>What types of problem the pattern is supposed to address?</i>	
Suggested		<i>What is the solution suggested by the pattern to address the problem?</i>	
Forces		<i>Factors affecting the problem and solution and pattern's justification.</i>	
Tactics		<i>What tactics are used by the pattern to implement the solution?</i>	
Affected Attributes		Positively	Negatively
		<i>Attributes supported</i>	<i>Attributes hindered</i>
General scenarios	S	<i>A textual, system independent specification of a quality attribute.</i>	
	S		
Example		<i>Some known examples of the usage of the pattern to solve the problems.</i>	

Date 5/22/2007

ICSE 2007, Minneapolis, USA

53

## Using the Template

Pattern Name: <i>Business Delegate</i>		Pattern Type: <i>Design pattern</i>	
Brief description		<i>This pattern reduces coupling between tiers by providing an entry point for accessing the services another tier. It also supports results caching to improve performance...</i>	
Context		<i>A client may be exposed to the complexity of dealing with the distributed components...</i>	
Problem description		<i>Presentation-tier components interact directly with business services. Such a direct interaction makes the clients vulnerable to any changes in the business services...</i>	
Suggested solution		<i>Reduce coupling between presentation-tier clients and business services. The Business Delegate hides the underlying implementation details of the business service...</i>	
Forces		<i>Presentation-tier clients require access to business service. It is desirable to minimize coupling to hide implementation details from clients.</i>	
Available tactics		<i>Delegate Proxy and Delegate Adapter</i>	
Affected Attributes		Positively	
		Negatively	
		<i>Reduce coupling, manageability, performance</i>	<i>Introduce new layer, increased complexity</i>
General scenarios	S1	<i>Presentation-tier components shall not be exposed to the implementation details of the business services they use.</i>	
	S2	<i>System shall provide a caching mechanism to improve response to business service request.</i>	
	S3	<i>Services calls across network or tiers shall be minimized to avoid degraded performance.</i>	
Examples		<i>E-commerce portals, online content providers, sports websites.</i>	

Date 5/22/2007

ICSE 2007, Minneapolis, USA

54

## A Template for Documenting Concrete Knowledge



Project Name: <i>Which project needs this scenario?</i>		Date: <i>When was proposed?</i>
Project domain: <i>Domain of the project</i>		Scenarios No: <i>Serial number assigned to the scenario</i>
Business goals	<i>Which business goals does this scenario achieve?</i>	
Stakeholders	<i>Which class of the stakeholders did suggest this scenario?</i>	
Attributes	<i>Which quality attributes are required by this scenario?</i>	
Description	<i>A brief description of the scenario.</i>	
Concrete scenario	Stimulus	<i>A condition that needs to be considered when it arrives at a system.</i>
	Context	<i>A system's condition when a stimulus occurs, e.g. overloaded, running etc.</i>
	Response	<i>A measurable action that needs to be undertaken after the arrival of the stimulus</i>
	Complexity	<i>How complex is this scenario to realize? (Effect on macro or micro architecture)</i>
	Priority	<i>How important is this scenario?</i>
Pattern/Style	<i>Name of the architectural pattern or style that can support this scenario.</i>	
Design tactics	<i>What are the design tactics used by the pattern/style to support the scenarios?</i>	
Design rational	<i>What are reasons for using the patterns/tactics? How does it provide the desired quality attributes?</i>	

Date 5/22/2007

ICSE 2007, Minneapolis, USA

55

## Documenting Concrete Knowledge Example



Project Name: <i>Qualification Verification System</i>		Date: <i>12/06/2005</i>
Project domain: <i>E-Commerce application</i>		Scenarios No: <i>Serial number assigned to the scenario</i>
Business goals	<i>Customer satisfaction and process efficiency.</i>	
Stakeholders	<i>Business Manager, System sponsors, and End User.</i>	
Attributes	<i>Improved performance</i>	
Description	<i>The response to a business service request shall be improved to avoid users' frustration and system shall be able to handle up to 1000 users concurrently without any delay in the response time.</i>	
Concrete scenario	Stimulus	<i>A user request needs to be processed.</i>
	Context	<i>There are 1000 users, who may request for a service simultaneously.</i>
	Response	<i>The system shall be able to respond to a request within 2 seconds.</i>
	Complexity	<i>Medium</i>
	Priority	<i>High</i>
Pattern/Style	<i>Business Delete</i>	
Design tactics	<i>Delegate proxy and Caching</i>	
Design rational	<i>This pattern exposes an interface to the business service API by using proxy function to pass the client methods to the session bean. It can cache any necessary data and references to the session bean's home or remote objects to improve performance by reducing the number of lookups.</i>	

Date 5/22/2007

ICSE 2007, Minneapolis, USA

56

## Value of Mining Pattern



- During architecture design
  - Helps identify patterns used in a software architecture and understand the reasons for using those patterns.
  - Helps identify appropriate patterns for system design.
- During architecture analysis
  - Provide confidence in an architecture's capability of supporting certain scenarios
  - Help develop concrete scenarios based on general scenarios extracted from the patterns in the architecture

## Empirical Evidence



- Research Program
  - Observational study and controlled experiments
- Pattern-Mining Framework
  - Framework is effective and it takes 30-45 minutes to mine one pattern
- Understanding and designing architecture
  - ASIP compared with standard pattern documentation is more helpful in understanding and designing architectures
- Architecture evaluation
  - Improves scenario gathering activity by helping stakeholders to develop better quality scenarios

## Trial of the Technology



- Tailoring BRedB for DSTO
  - Define domain-specific evaluation criteria
  - Capture knowledge underpinning COTS' architectures
  - Support standards compliance analysis
  - Document rationale for COTS acquisition decisions
  - Rank and compare architectural solutions
  - Track architecture decision makers and evaluators
  - Generate reports for management decision making

## Summary of Section 2



- An Introduction to BRedB – An Architecture Knowledge Management Infrastructure
  - Architecture
  - Features to support architecture design, documentation, and evaluation activities
- Approaches to capturing architectural knowledge
- Using BRedB on the case study

## Domain Modeling for Architecture Knowledge



- Knowledge modeling
- Knowledge modeling process
- Knowledge modeling techniques
- Architecture knowledge constructs & relationships
- Models characterizing architecture knowledge
- Some example queries for case study

Date 5/22/2007

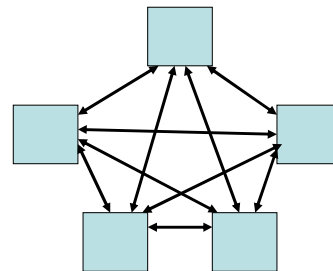
ICSE 2007, Minneapolis, USA

61

## Knowledge Modeling



- Why do we need models?
- Importance of knowledge modeling
- Role of models in knowledge acquisition and validation
- Modeling in software engineering and knowledge engineering: similarities and differences
- How to model knowledge?



Date 5/22/2007

ICSE 2007, Minneapolis, USA

62

## A Generic Process of Modeling Knowledge



- Domain analysis for identifying constructs and their attributes
- Identify relationships among different constructs
- Determine properties of relationships
- Apply some formal modeling approach, e.g. Entity-Relation models
- Refine the conceptual model using the principles of a formal modeling approach
  - Avoid fully normalizing the model
- Assessment – many ways such as expert opinion or theoretical

## Knowledge Modeling Techniques



- CommonKADS
- Protégé
- UML & OCL
- Multi-perspective modeling



## Comparison of Knowledge Modeling Techniques



Technique Features	CommonKADS	Protégé 2000	Multi-Perspective	UML
K.E. methodology	X			
O-O approach	X	X	X	X
Platform independent	X	X	X	X
Hybrid approach	X		X	X
Editor tool		X		
Standard modeling language				X
Documentation	X	X	X	X
Evolving		X	X	X
Domain	Medical, Business, & Legal	Medical, Business, & Legal	Medical, Business, & Legal	Medical, Business, & Legal
Other features (OKBC, RDF, Semantic Web)		X		

Source: M.S. Abdullah et al. 3<sup>rd</sup> European Conf. on KM, 2002

Date 5/22/2007

ICSE 2007, Minneapolis, USA

65

## Exercise – Domain modelling



- Form a group of 2-3 and identify constructs, their properties, and relationships that characterise architecture knowledge in your domain/industry.

Date 5/22/2007

ICSE 2007, Minneapolis, USA

66

## Architectural Knowledge – Some Constructs



- Architecture decision
- Design option
- Design rationale
- Pattern – (Architectural, design, platform)
- Tactic
- Architectural significant requirement (ASRs)
- Scenarios
  - Abstract
  - General

Date 5/22/2007

ICSE 2007, Minneapolis, USA

67

## Some of the Sources for Identifying Constructs



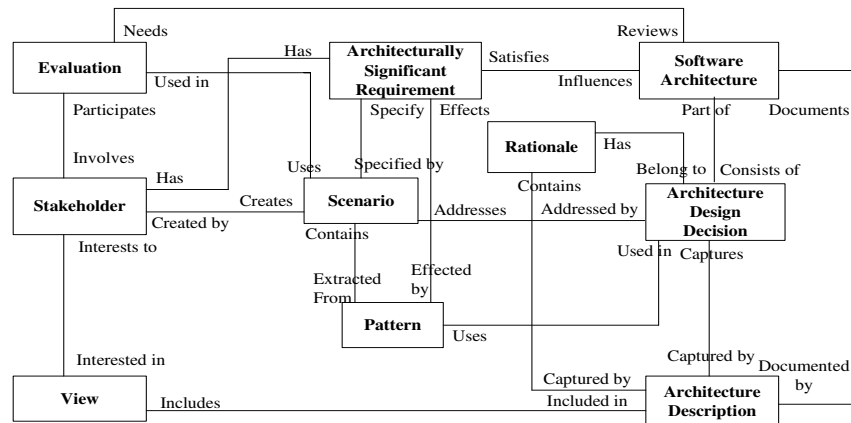
- SARA report
- IEEE 1471 standards
- Views and Beyond
- Books on software architectures
- Literature on software architecture

Date 5/22/2007

ICSE 2007, Minneapolis, USA

68

## A Schema for Architecture Knowledge

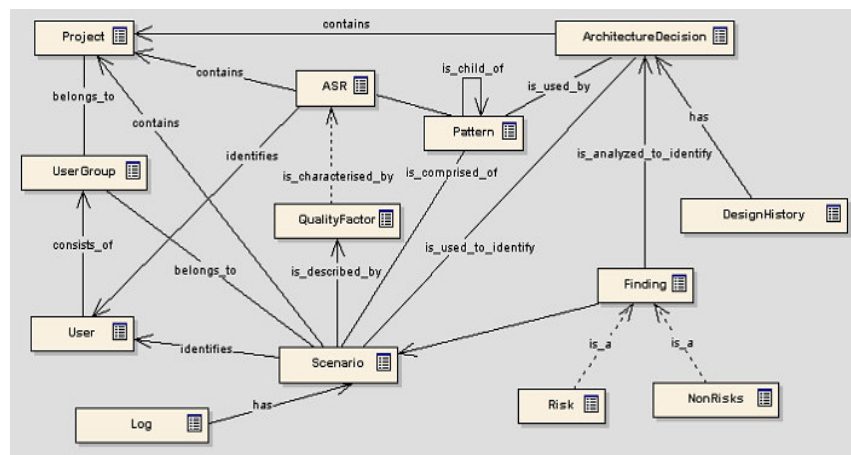


Date 5/22/2007

ICSE 2007, Minneapolis, USA

69

## A partial model of Architecture Knowledge

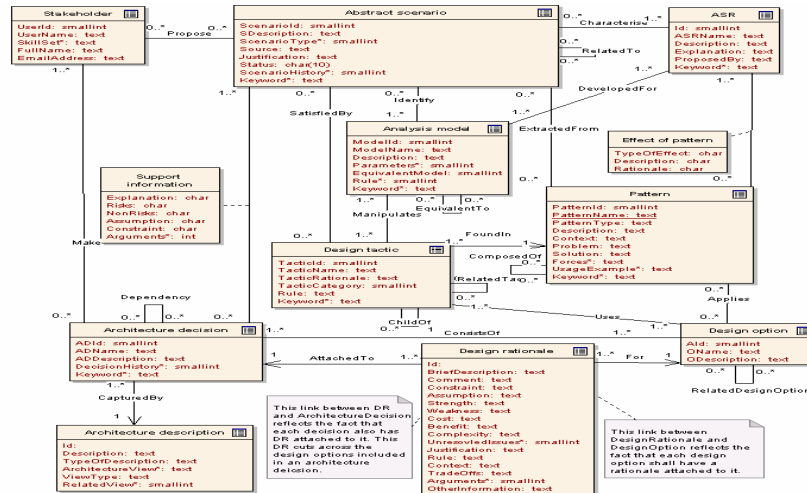


Date 5/22/2007

ICSE 2007, Minneapolis, USA

70

## Characterising Architecture Knowledge



Date 5/22/2007

ICSE 2007, Minneapolis, USA

71

## A Few Heuristics for Modeling Knowledge



- identify most-frequent, synonyms or related terms
- Identify relationships between terms and models
- Identify basic constraints
  - Pre-requisite constraint
  - Temporal constraint
  - Mutually inclusive constraint
  - Mutually exclusive constraint
- Identify domain constraints
- Identify domain dependencies

Date 5/22/2007

ICSE 2007, Minneapolis, USA

72

## Summary of Section 3



- Knowledge modeling
- Knowledge modeling techniques
- Architecture knowledge modeling process
- Architecture knowledge constructs & relationships
- Models characterizing architecture knowledge
- Some example queries for case study

## Using knowledge during design & evaluation



- Industrial case study – Avionics Architectures
  - Problem domain description
  - Current evaluation methods
- A framework for avionic architecture evaluation using knowledge management
  - Quality attributes
  - Evaluation framework
  - Risk management
- Demonstration: Using BRedB for avionics architecture evaluation

## Airborne Mission Systems (AMS) Branch



- Research, analytical studies and experimental work in Avionics and AMS.
- Development of innovative methods and tools to aid the understanding of advanced avionics technologies
- Provide short-term advice to Australian Defence Force (ADF) project teams and develop long-term strategies to support the ADF on mission systems acquisitions.

Date 5/22/2007

ICSE 2007, Minneapolis, USA

75

## Airborne Mission Systems (AMS) Branch



- Responsible for performing Technical Risk Assessments (TRA) of mission systems
  - critical for the ADF acquisition process
- AMS is required to understand and organize large amounts of architecture design knowledge
- “Software intensive projects are historically considered the most risk prone in the Defence domain”
- Building capabilities in systematically evaluating architectures and maintaining architecture knowledge

Date 5/22/2007

ICSE 2007, Minneapolis, USA

76

## ADF Acquisitions and Software related Issues



- Navy's \$100m chopper can't fly in bad light
  - \$1.1b were spent before finding that Seasprite had major software problems
- Billions wasted on 'Collin Class Submarines'
  - Software and architectural issues made them 'Duds'
- Airborne Early Warning and Control project has 4 million lines of code
- Software literacy is vital but quite low
- Poor record keeping makes it impossible to trace reasons for bumbles in acquisition projects

Date 5/22/2007

ICSE 2007, Minneapolis, USA

77

## Trial's Background



- Collaborative R&D project between NICTA and DSTO has been undertaken during 2006
- Exploit NICTA architecture evaluation methods and tools for improving AMS capabilities
- Codifying architecture evaluation process and design knowledge and rationale is a vital goal of this project
- 8 members team (4 on the each side)

Date 5/22/2007

ICSE 2007, Minneapolis, USA

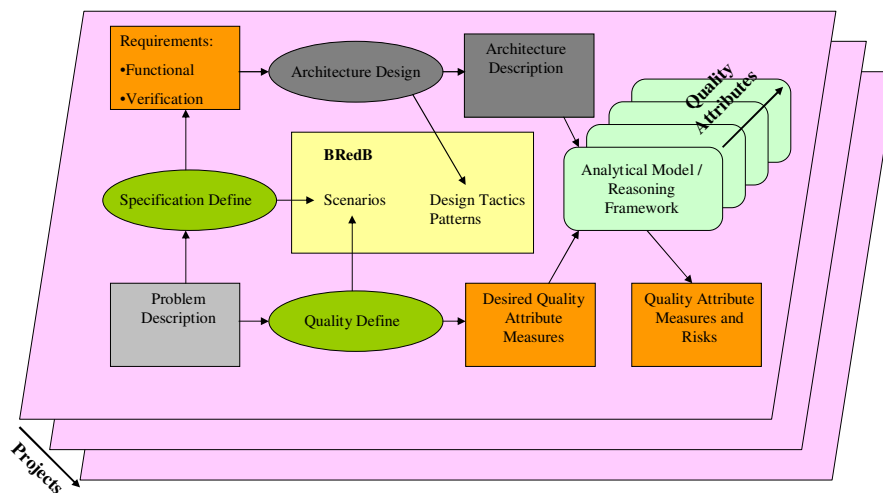
78

## Trial's Objectives



- Support a systematic and repeatable evaluation process
- BRedB is expected to help AMS to achieve several goals:
  - Capture rationale for architecture decisions
  - Help build architectural capabilities
  - Improve architectural reusability
  - Provide an audit trail for TRA findings
  - Reduce demands on subject matter experts
  - Encourage best architectural practices
  - Improve efficiency of architectural processes
  - Accelerate the training process of new employees

## BRedB Supported Architecture Evaluation Process





- AMS doesn't design - it evaluates architectures proposed by contractors in response to a Request for Proposal (RFP)
- AMS only needs BredB features that support architecture evaluation activities
- Gathering customization requirements by analysing activities/tasks of the current process

- High priority requirements implemented:
  - Classification of project data according to the Defence classification scheme.
  - Mechanism for recording compliance of architectural decisions with respect to requirements.
  - Store and evaluate several contractors' proposals for the same set of scenarios within one project.
  - Different levels of access to project data based on the Defence security scheme.
  - Ability to import/export data from the tool based on the classification scheme.
  - Integration with requirements management and architecture modelling tools.

## AMS Case Study



- Assessing customized BRedB by supporting architecture evaluation process of AMS
- Evaluating the proposed architecture for an airborne mission system
- Goal was to investigate BRedB's role in the architecture evaluation process and how it could help capture and manage architecture knowledge and rationale

Date 5/22/2007

ICSE 2007, Minneapolis, USA

83

## Case Study – Use of the Knowledge Base



- BRedB was populated with AMS domain knowledge, starting with a preliminary domain specific quality model.
- Quality model was based on ISO 9126 and AMS domain knowledge elicited from domain experts
- Quality model is used to assess the potential risks of architectural designs
- Quality model consists of six quality attributes:
  - Performance
  - Reliability
  - Usability
  - Maintainability
  - Portability
  - Functionality

Date 5/22/2007

ICSE 2007, Minneapolis, USA

84

## Quality Framework - Performance



Latency	What are average/worst case latencies for behavior such as: Radar to display, Algorithm used to calculate tracks/bearings, and Video frame updates.
Accuracy	Accuracy relates to timestamping of events. Check the architecture for: place of time reference, multiple clocks are maintained/synchronized, clock signal distribution through aircraft?
Capacity	Look for the following data related to capacity: Number of tracks a system can support/process and database size/storage, and processor capacity.
Throughput	Check if the proposed processor and bus architecture can support the likely throughput in terms of events per second (both average and worst case)
Resource usage	Look to assess several things in terms of resource usage: CPU (peak CPU load, spare capacity), Power consumption, Cooling, Memory, Disk (usage for each disk), and I-O rates.
scalability	Various questions to be asked such as Are any key software elements single-threaded? Do the specifications indicate spare CPU capacity that could be used for additional processing?
Schedulability	Is it likely that high priority tasks can be scheduled when needed? What is the clock granularity

Date 5/22/2007

ICSE 2007, Minneapolis, USA

85

## Scenarios – Performance



Initialization	Must perform all initialization activities within 10 minutes.
Latency	Run simulations with no instantaneous lags greater than five seconds, no average lags greater than three seconds.
Capacity	Run-time simulation with debug enabled.
Latency	Finish data collection within 30 seconds of simulation termination.
Throughput	Finish data collection request from three network sensors within x seconds every one minute.

Date 5/22/2007

ICSE 2007, Minneapolis, USA

86

## Quality Framework – Reliability



Fault tolerance	Look for a fault tolerance in terms of: software and hardware redundancy, separation of flight /mission functions, and potential single points of failure, mechanics of notify operators of faults.
Recoverability	Check for time required for cold/warm start ups, for cold start to full performance, amount of data loss on a failure/ restart, any loss of situational information and how to recover/readjust to that.
Maturity	Are existing tested components used? Level of process maturity of the organization.
Survivability	Check for the kind of redundancy for key software/hardware components, communication channels.
Availability	Check for restart/recovery times after failure and supportability.

Date 5/22/2007

ICSE 2007, Minneapolis, USA

87

## Scenarios – Reliability



Availability	Simulation controller initiates simulation execution (a game), starts subordinate processes, loads parameter files, and simulation runs from T=-infinity to T=0 within 10 minutes.
Availability	Runs scenarios without crashes: no hardware and/or software failures. Terminates normally.
Fault tolerance	Ability to change the location and orientation of assets and still meet the performance, reliability, and credibility criteria.
Maturity	Organization has been developing similar systems for the last 20 years and has been certified as CMMI level 4.
Recoverability	System encounters serious data corruption and needs to restart. System should recover in 30 seconds.

Date 5/22/2007

ICSE 2007, Minneapolis, USA

88

## Quality Framework in BRedB



hitergate v2.1

Start Collaborative Tools Contact Management Project Management Content Production Corporate Library Project Collaboration Configuration

Home Knowledge-Based | Project-Based | Search

Home Select Project Select Artefact Report

Home General Scenario Concrete Scenario ASR Quality Factor Architecture Decision Design Option Finding

General Scenario Listing of AMS Project 1

New Delete Accept Reject Home Search Discard Show 200 results

Name	Description	Source	Date Entered	Logs	
Reliability - Availability	An unexpected event internal to the system occurs. The system must record the event, notify the user/other systems and be available within a specified period.	User-Defined	Sun 17 Dec 2006 23:18		<input type="checkbox"/>
Portability - Adaptability	The user wishes to introduce new hardware to the system. The system has to ensure modification without affecting other functionality.	User-Defined	Sun 17 Dec 2006 23:18		<input type="checkbox"/>
Performance - Latency	A periodic event from an independent source arrives at the system under normal conditions. The system has to process the event within a specified latency.	User-Defined	Sun 17 Dec 2006 23:21		<input type="checkbox"/>
Functionality - Security	An unidentified user tries to access the data/services on the system. The system has to block access to the data/services until the user is authenticated.	User-Defined	Sun 17 Dec 2006 23:24		<input type="checkbox"/>
Usability - Operability	The user wants to operate the system efficiently. The system has to provide an interface to support efficient navigation within the screen and allow multiple simultaneous activities.	User-Defined	Sun 17 Dec 2006 23:24		<input type="checkbox"/>
Maintainability - Growth	The developer wishes to upgrade software to add new functionality. The system has to incorporate the change within the specified amount of cost/effort.	User-Defined	Sun 17 Dec 2006 23:26		<input type="checkbox"/>

Accepted General Scenarios

Name	Description	Source	Date Entered	Logs	
------	-------------	--------	--------------	------	--

Rejected General Scenarios

Name	Description	Source	Date Entered	Logs	
------	-------------	--------	--------------	------	--

Date 5/22/2007

ICSE 2007, Minneapolis, USA

89

## Utility Tree of the Quality Framework



hitergate v2.1

File Edit View Favorites Tools Help Address http://local... Go Norton Internet Security

Utility Tree

- Usability
  - Understandability
  - Learnability
  - Operability
- Performance
  - Latency
  - Accuracy
  - Capacity
  - Throughput
  - Resource Usage
  - Scalability
  - Schedulability
- Reliability
  - Fault Tolerance
  - Recoverability
  - Maturity
  - Survivability
  - Availability
- Maintainability
  - Growth
  - Changeability
  - Testability
  - Analysability
- Functionality
  - Security
  - Interoperability
  - Compliance
  - Safety
  - Certification
  - COTS Components
- Portability
  - Adaptability
  - Standard Conformance
  - Hardware Replaceability

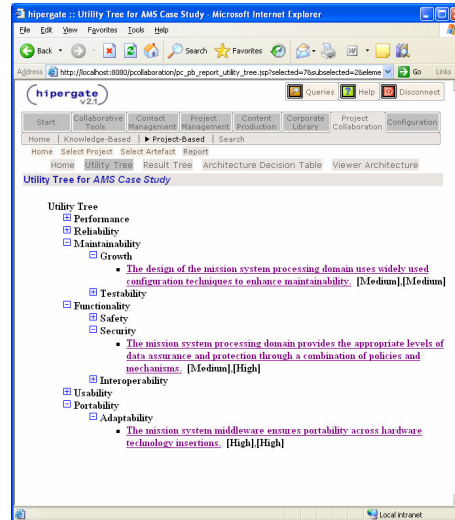
Date 5/22/2007

ICSE 2007, Minneapolis, USA

90

## Case Study – Use of the Project Base

- Concrete scenarios relating to the quality factors were added to the Project Base.
- The architecture design was captured and compared with alternative designs with respect to the scenarios



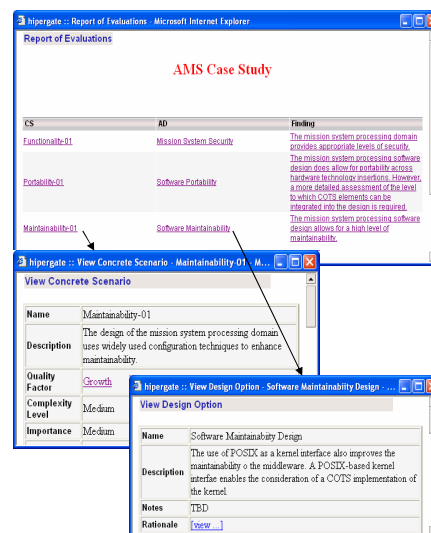
Date 5/22/2007

ICSE 2007, Minneapolis, USA

91

## Case Study – Reporting

- A report of the evaluation shows the findings of an architectural decision that is aimed at satisfying a concrete scenario



Date 5/22/2007

ICSE 2007, Minneapolis, USA

92

## Challenges



- Domain understanding
- Access to the required information/knowledge
- Security modelling and implementation
- Modification of the underlying data model
- Communication/coordination issues

Date 5/22/2007

ICSE 2007, Minneapolis, USA

93

## Initial Findings



- The use of an evaluation framework and knowledge management tool brings added rigour to the evaluation process.
- It is anticipated that access to past project experience stored in BRedB will be valuable in the evaluation process.
- The modified version of BRedB provides AMS with an effective and efficient mechanism to organise and understand large amounts of architecture knowledge

Date 5/22/2007

ICSE 2007, Minneapolis, USA

94

## Tool Demo



- Short Video clips on using BRedB and Wiki based tool

## Summary of Section 4



- Industrial case study – Avionics Architecture
- A framework for avionic architecture evaluation using knowledge management
  - Quality attributes
  - Evaluation framework
  - Risk management
- Demonstration: Using BRedB for avionics architecture evaluation



## Some useful references



- Bass et al. – Software Architecture in Practice
- Clements et al. - Documenting Software Architectures : Views and Beyond
- Ian Gorton – Essential Software Architecture
- IEEE Recommended Practice for Architecture Description of Software-Intensive System (IEEE Std 1471-2000)
- Philippe Kruchten - <http://philippe.kruchten.com/>
- Grady Booch's site

Date 5/22/2007

ICSE 2007, Minneapolis, USA

97

## Acknowledgement



- Tutorial incorporates material provided by Andrew Northway, Paul Heuer, and Thong Nguyen of DSTO. Discussions with our collaborators Antony Tang and Jun Han resulted in many ideas presented in the tutorial.
- Shirley Xu worked on the engineering side of the BRedB. Before that several students of the UNSW helped us build the first version of that tool.
- Most of the reported work was produced while the presenters were working at NICTA

Date 5/22/2007

ICSE 2007, Minneapolis, USA

98

## Collaborative Opportunities



- We welcome collaborative opportunities to deploy our tools and techniques to help design better architectures and effectively and efficiently evaluate and manage them.

### Contacts:

- Ali Babar, Lero, Ireland ([malibaba@lero.ie](mailto:malibaba@lero.ie))
- Ian Gorton, PNNL, USA ([ian.gorton@pnl.gov](mailto:ian.gorton@pnl.gov))

Thank you!