A Structured Legacy to SOA Migration Process and its Evaluation in Practice

Ravi Khadka
Amir Saeidi
Slinger Jansen
Jurriaan Hage
# Agenda

- Background
- Problem Definition
- Objectives
- The structure process
- Evaluation
- Discussion
Caution

- We do not claim COMPREHENSIVENESS

- The conclusions drawn are suggestive, rather than conclusive.
Background

- **Legacy Systems**
  - Inflexible, higher maintenance cost, rigid…
  - Business critical, profitable systems, backbone system …

- **SOA**
  - Claimed benefits
  - Reusability, Maint. Cost reduction, loose coupling, etc

- **Legacy to SOA migration**
  - A plethora of research (>100 academic papers\(^1,2\))
Problem Definition

- General Observation
- 2 categories of solution

Is it feasible???

How ???
Eureka!!! It works 😊

Migration feasibility
Supporting technology
Objectives

1. To develop a structured process that combines feasibility and migration execution
   - 6 phased structured process

2. To identify rationale, current practices and challenges for each phase of the process
   - Rationale
   - Current Practices
   - Challenges
The Structured Process

- Target system understanding
- Migration feasibility determination
- Legacy system understanding
- Candidate service identification
- Implementation & management
- Deployment & provisioning
Legacy System Understanding

• Rationale
  ✓ a deductive process of acquiring knowledge about the “as-is” situation
  ✓ Consequences of lack of resources, (up-to-date) documentation, experts
  ✓ creating an inventory of the existing features
  ✓ maximizing reusability of legacy features

• Challenges
  ✓ Knowledge erosion problem
  ✓ Developing generic reverse engineering tools
  ✓ Maximizing automation of reverse engineering tools
Legacy System Understanding

- Current Practices
Rationale

- representation of the desired architecture of the “to-be” SOA
- major components of the SOA environment
- standards to be used
- quality of services expectations
- interaction patterns between services
- target architecture largely determine the reusability
Target System Understanding

**Current Practices**
- Not sufficient attention
- Guidelines for TSU – SMART
- Service design & orchestration - SOAMIG
- Standards for NFR (Maint., interop.) - Cuadrado et al.

**Challenges**
- Business-IT alignment
- Componentization - deconstruct, analyze and identify buss. Components
- Preserving NFR within TSU
- Infrastructure Engineering
Migration Feasibility Determination

• Rationale
  ✓ Feasibility based on from technical, economical and organizational perspectives
  ✓ technical feasibility – details on how to reuse
  ✓ Economical feasibility – cost and ROI
  ✓ Organizational feasibility- do higher management agree to invest PLUS Business goals
  ✓ “If ain’t broken, don’t fix it”
Migration Feasibility Determination

• Current Practices
  ✔ Cost-Benefit Analysis- Sneed (1995)
  ✔ Extended CBA for integration vs. migration –Umar & zordan (2009)
  ✔ Option Analysis for Re-engineering -SMART

• Challenges
  ✔ Automated toolsets – decision making tools for MFD
  ✔ Toolset integration with LSU and TSU to facilitate automation
Candidate Service Identification

- **Rationale**
  - ✓ Spaghetti code, lack of resources
  - ✓ Enables reusability & leveraging the existing legacy assets
  - ✓ An interesting area - uses various techniques
Candidate Service Identification

• Current Practices
  ✓ Top-down approach- modeling a business process and mapping to legacy functionalities
    ✓ Alahmari et al. (2010), Fuhr et al. (2011), Zillmann et al. (2011)
  ✓ Bottom-up approach- utilizes the legacy code to identify services using various techniques
    ✓ information retrieval, concept analysis business rule recovery, source code visualization

• Challenges
  ✓ Spaghetti code – difficulty to isolate business logic
  ✓ Determining optimal granularity of services
  ✓ service rationalization and service consolidation
**Implementation**

- **Rationale**
  - Execution of the migration of the legacy applications
  - depends- migration strategies, availability of tools/techniques etc.
  - Four realization strategies
Implementation

- Current Practices

- Challenges
  - Selection criteria for appropriate strategy
  - Testing and service versioning after exposing as a service
  - Service commonality
Deployment & Provisioning

• Rationale
  ✓ Management to facilitate life cycle for a service
  ✓ Infrastructure engineering – testing for correctness
  ✓ Service provisioning- publishing, discovering, NFC
  ✓ Objective is to ensure that the SOA environment operates reliably and efficiently

• Current Practices
  ✓ Research in (automated) publication & discovery
  ✓ Service evolution- Papazoglou (2008); Andrikopoulos et al. (2008);
  ✓ Service Versioning- Fang et al. (2007)
Deployment & Provisioning

• Challenges
  ✓ automated service discovery with minimal user involvement
  ✓ Service testing combined with run-time verification
  ✓ Service compatibility with versions of same services
  ✓ Service commonality
Evaluation

- Two simple/experimental case studies
- SrnaCalc & Java calculator suite applications

<table>
<thead>
<tr>
<th>Application</th>
<th>LSU</th>
<th>TSU</th>
<th>MFD</th>
<th>CSI</th>
<th>IMP.</th>
<th>D&amp;P</th>
</tr>
</thead>
<tbody>
<tr>
<td>SrnaCalc</td>
<td>X</td>
<td>X</td>
<td>o</td>
<td>m</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Java Calculator</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>m</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

X: performed
-: partially performed
m: manually done
Evaluation

**Criteria**

- High citation count
- Industrial or preliminary case study
## Overview of the current practices, challenges and the possible solutions

<table>
<thead>
<tr>
<th>Phases</th>
<th>Current Practices</th>
<th>Challenges</th>
<th>Possible Solutions</th>
</tr>
</thead>
</table>
| **Legacy System Understanding** | Feature location  
Software Metrics  
Architecture recovery  
Software visualization  
Soft knowledge                          | Preventing knowledge erosion  
Developing generic tooling for heterogeneous legacy understanding  
Maximizing automation in reverse engineering process | Knowledge transfer programs  
Model-Driven engineering  
Utilizing the human feedback |
| **Target System Understanding** | Specific standards  
Specific technology  
Functional specification                   | Identifying optimal business-IT alignment  
Maintaining non-functional characteristics                       | Componentization  
Use of proper standards & technologies |
| **Migration Feasibility Determination** | Cost-Benefit Analysis  
OAR  
Code complexity  
Reusability assessment | Automating migration feasibility determining toolset                       | Technical, economical & business value information based toolset |
| **Candidate Service Identification** | Modeling legacy process  
Information retrieval  
Concept analysis  
Business rule recovery  
Code visualization | Identifying functional areas in source code                              | Feature location  
Trace visualization  
Source code search |
| **Implementation**          | Slicing  
Code extraction  
Wrapping  
Code transformation  
Refactoring  
Redevelopment  
Graph transformation | Selecting appropriate migration strategies  
Tooling for developing generic toolset | Model-Driven engineering |
| **Deployment & Provisioning** | Discovery  
Testing  
Evolution  
Publication | Automated service discovery  
Testing with run-time verification  
Addressing service versioning  
Addressing service commonality | Use of semantic markup languages  
Techniques to combine testing with run-time verification  
Usage of service compatibility  
Self-adaptive services |
Further

Participate in our legacy modernization survey, if you have not done yet. 😊

www.servicifi.org/legacy-survey
References
