Definition and Evaluation of Geographic Information System Architecture using ADD and ATAM

Ibrahim Habli
Outline

• Motivation
• Architectural Solution
• GIS Introduction
• Analysis of GIS Architectural Requirements
• Design of GIS Architecture using ADD
• Evaluation of GIS Architecture using ATAM
• Lessons Learnt
Motivation: Industrial

- Attention is often paid to GIS functionalities
- Complexity and large volume of geographic information
  - Sensitivity of data integrity
- Criticality of achieving quality attributes such as:
  - Performance
  - Availability
  - Maintainability
  - Accuracy
- Common to find ambiguous quality statements such as:
  - “System shall be portable”
  - “System shall be highly secured”
  - “GIS operations shall be efficient”
Motivation: Academic

• Teaching: clarifying concept for software engineering students
  – Typical question: “how do we move from requirements to architecture?”

• Complete real world example: integration of quality attribute scenarios, ADD and ATAM
  – Architectural Requirements
  – Architectural Design
  – Architectural Evaluation
Proposed Architectural Solution

- Quality Attribute Capture:
  - Scenarios
- Architecture Design
  - Attribute-Driven Design Method (ADD)
- Architecture Evaluation
  - Architecture Tradeoff Analysis Method (ATAM)
GIS Introduction: Overall Concepts

• Computerized systems for creating & depicting digital visualizations of earth’s surface, specifically for:
  – Data Visualization
    • Zoom & Pan
  – Data Analysis
    • Spatial, 3D & Network Analysis
  – Data Management
    • Share & backup
GIS Introduction: Typical Uses

• Utilities
  – Electricity, Water & Gas
• Defence and Security
  – Training, planning & warfare
• Governmental Organizations
  – Counsels and municipalities
• Internet GIS
  – Sharing geographic data & map routing
GIS Domain Analysis: Domain Modelling

- Documented in UML class diagrams
  - No introduction of new notation
- Reveal common and variable elements
  - Supporting large-scale reuse
GIS Functional Requirements: Use Cases

- Functional commonality & variability derived from domain model
- Common use cases
- Variable use cases
GIS Quality Requirements: Scenarios

• 22 quality attribute scenarios covering 6 quality attributes
GIS Quality Requirements: Example Scenarios

• PRF RETRIEVE: Retrieve Data

<table>
<thead>
<tr>
<th>Source</th>
<th>GIS viewer/external system/internal source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stimulus</td>
<td>A request to retrieve data from the data source</td>
</tr>
<tr>
<td>Artefact</td>
<td>GIS system (data source)</td>
</tr>
<tr>
<td>Environment</td>
<td>Normal operation</td>
</tr>
<tr>
<td>Response</td>
<td>The system processes the request, and retrieves the requested data</td>
</tr>
<tr>
<td>Response Measure</td>
<td>0.5 seconds/feature</td>
</tr>
</tbody>
</table>

• MDF_SW: Interface GIS with an External Software System

<table>
<thead>
<tr>
<th>Source</th>
<th>GIS programmer/system administrator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stimulus</td>
<td>Enable GIS to communicate with an external software system</td>
</tr>
<tr>
<td>Artefact</td>
<td>GIS system</td>
</tr>
<tr>
<td>Environment</td>
<td>Compile Time</td>
</tr>
<tr>
<td>Response</td>
<td>The GIS system is interfaced with an external software system</td>
</tr>
<tr>
<td>Response Measure</td>
<td>One day programmer/One day system administrator</td>
</tr>
</tbody>
</table>
ADD Application

• Attribute Driven Design Method (ADD)
  – Recursive
  – Explicit representation of quality attributes
  – Explicit association between architectural decisions and quality attributes

• SEI Module View
  – Reveals modularization: how change to one part of system affects other parts

• SEI Component-and-Connector (C&C) View
  – Reveals aspects of performance, security & availability
Module View: First Decomposition

- <<Subsystem> User-Interface
- <<Subsystem> OperatingSystem
- <<Subsystem> Applications
- <<Subsystem> Communication
- <<Subsystem> Data
Second Level of Decomposition: Data Subsystem

- Architectural Decisions:
  - Façade design pattern
  - Authentication & authorization tactics
  - Data replication tactic

- Scenarios addressed at this stage:
  - Change GIS Data Format
  - Authenticate Access
  - Unauthorized
  - Data Source Failure Detection
Module View Updated
Second Level of Decomposition: Applications Subsystem

- Architectural Decisions:
  - Separation unit operation
  - Mediator design pattern
  - Load balancing tactic

- Scenarios addressed at this stage:
  - Updating Real-time Data
  - Interface with Internal Components
  - Retrieving Data
Module View Updated
Second Level of Decomposition: User-Interface Subsystem

• Architectural Decisions:
  - Generalization style
  - Adapter (or wrapper) design pattern

• Scenarios addressed at this stage:
  - Interface GIS with External Software
  - Interface GIS with External Hardware
  - Add New GIS Client Application
Quality Attribute Scenarios:
- Editing a Geographic Feature
- Retrieve Data
- Change GIS Data Format
- Add GIS Component
- Updating Real-time Data
- Interface GIS with External Software

Architectural Decisions:
- Maintain semantic coherence
- Published interfaces
- Client-Server style
- Minimize clients & servers interaction
- Data accessor design pattern
- Façade design pattern
- Concurrency
- Caching
- Increase hardware resources
- Mediator design pattern
- Geographic information standards
- Separation unit operation
- Generalization style
- Adapter (or wrapper) design
Requirements Analysis

Functional Requirements (Use case)  Quality Attribute Scenarios

Domain Analysis

ADD

Conceptual Architecture

Architectural Design Alternatives

Architectural Design Rationale  Architectural Views  Architectural Design Decisions

ATAM

Risks  Tradeoff Points  Sensitivity Points

Risk Themes
Architecture Tradeoff Analysis Method (ATAM)

Presentation
1. Present the ATAM
2. Present the business drivers
3. Present the architecture

Investigation and Analysis
4. Identify the architectural approaches
5. Generate the quality attribute utility tree
6. Analyze the architectural approaches

Testing
7. Brainstorm and prioritize scenarios
8. Analyze the architectural approaches

Reporting
9. Present the results
### Step 4: Architectural Approaches Identification

<table>
<thead>
<tr>
<th>Design Decision</th>
<th>Scenarios Addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain semantic coherence tactic</td>
<td>MDF_COMPONENT</td>
</tr>
<tr>
<td>Isolate expected change tactic</td>
<td>MDF_CLIENT</td>
</tr>
<tr>
<td>Published interfaces tactic</td>
<td>MDF_FORMAT</td>
</tr>
<tr>
<td>Mediator design pattern</td>
<td>MDF_OPERATION</td>
</tr>
<tr>
<td>Client/Server style</td>
<td>PRE_RETRIEVE,PRE_RT_UPDATE</td>
</tr>
<tr>
<td>Minimize interaction between Clients and Servers tactic</td>
<td>PRE_RETRIEVE,PRE_RT_UPDATE</td>
</tr>
<tr>
<td>Data accessor design pattern</td>
<td>MDF_FORMAT,MDF_SOURCE</td>
</tr>
<tr>
<td>Facade design pattern</td>
<td>MDF_FORMAT,MDF_SOURCE</td>
</tr>
<tr>
<td>Authentication and authorization tactics</td>
<td>SEC_AUTHENTICATE,SEC_AUTHORIZE</td>
</tr>
<tr>
<td>Heartbeat tactic</td>
<td>AVLB_FAILURE</td>
</tr>
<tr>
<td>Data replication tactic</td>
<td>AVLB_FAILURE</td>
</tr>
<tr>
<td>Concurrency tactics</td>
<td>Performance Scenarios</td>
</tr>
<tr>
<td>Coating tactic</td>
<td>PRE_RETRIEVE,PRE_RT_UPDATE</td>
</tr>
<tr>
<td>Increase hardware resources tactic</td>
<td>PRE_RETRIEVE,PRE_RT_UPDATE</td>
</tr>
<tr>
<td>Mediator design pattern</td>
<td>MDF_SW,MDF_HW,MDF_CLIENT</td>
</tr>
<tr>
<td>Use geographic information standards tactic</td>
<td>MDF_FORMAT,MDF_CLIENT</td>
</tr>
<tr>
<td>Separation unit operation</td>
<td>PRE_RT_UPDATE,PRE_RT_REFRESH</td>
</tr>
<tr>
<td>Load balancing tactic</td>
<td>PRE_RETRIEVE,PRE_RT_UPDATE and PRE_RT_UPDATE</td>
</tr>
<tr>
<td>Generalization style</td>
<td>MDF_CLIENT,MDF_SW,MDF_HW</td>
</tr>
<tr>
<td>Adapter (or wrapper) design pattern</td>
<td>MDF_CLIENT,MDF_SW,MDF_HW</td>
</tr>
<tr>
<td>Add firewall tactics</td>
<td>Security Scenarios</td>
</tr>
<tr>
<td>Model-View-Controller design pattern</td>
<td>MDF_COMPONENT,MDF_OPERATION</td>
</tr>
<tr>
<td>Decorator design pattern</td>
<td>USB_CUSTOMIZE</td>
</tr>
<tr>
<td>Proxy design pattern</td>
<td>PRE_RETRIEVE</td>
</tr>
<tr>
<td>Builder design pattern</td>
<td>MDF_FORMAT,MDF_CLIENT</td>
</tr>
<tr>
<td>Pipes and filters style</td>
<td>MDF_FORMAT</td>
</tr>
<tr>
<td>Fixed priority scheduling (FPS) tactic</td>
<td>Real-time Performance Scenarios</td>
</tr>
<tr>
<td>Hide information tactic</td>
<td>MDF_SOURCE</td>
</tr>
<tr>
<td>Maintain existing interface tactic</td>
<td>MDF_SOURCE</td>
</tr>
<tr>
<td>Sharing tactic</td>
<td>Real-time Performance Scenarios</td>
</tr>
<tr>
<td>Master-Slave design pattern</td>
<td>Real-time Performance Scenarios</td>
</tr>
<tr>
<td>Intrusion Detection tactic</td>
<td>SEC_INTRUSION</td>
</tr>
<tr>
<td>Auditing tactic</td>
<td>SEC_INTRUSION,AVLB_FAILURE</td>
</tr>
<tr>
<td>Maintain integrity tactic</td>
<td>SEC_INTEGRITY</td>
</tr>
<tr>
<td>Layered style</td>
<td>MDF_COMPONENT,TEST_COMPONENT</td>
</tr>
<tr>
<td>Aggregation of data architecture tactic</td>
<td>USB_EDITING</td>
</tr>
<tr>
<td>Supporting undo tactic</td>
<td>USB_UNDO</td>
</tr>
</tbody>
</table>
ATAM Steps

Presentation
1. Present the ATAM
2. Present the business drivers
3. Present the architecture

Investigation and Analysis
4. Identify the architectural approaches
5. Generate the quality attribute utility tree
6. Analyze the architectural approaches

Testing
7. Brainstorm and prioritize scenarios
8. Analyze the architectural approaches

Reporting
9. Present the results
Step 5: Quality Attribute Utility Tree

- Refinement of each quality attribute into goals & concrete scenarios
- Prioritisation
  - H (High)
  - M (Medium)
  - L (Low)
ATAM Steps

Presentation
1. Present the ATAM
2. Present the business drivers
3. Present the architecture

Investigation and Analysis
4. Identify the architectural approaches
5. Generate the quality attribute utility tree
6. Analyze the architectural approaches

Testing
7. Brainstorm and prioritize scenarios
8. Analyze the architectural approaches

Reporting
9. Present the results
Step 6: Architectural Approaches

Analysis

- **S3**: Minimum number of backup sources is one; nevertheless, adding more than one backup source improves availability.

- **T2**: The availability improves when active data sources synchronize more frequently with passive sources. Yet, this makes the system less efficient.

- **R4**: The amount of data loss might be large if the active data source synchronizes less frequently with the passive sources.
ATAM Steps

Presentation
1. Present the ATAM
2. Present the business drivers
3. Present the architecture

Investigation and Analysis
4. Identify the architectural approaches
5. Generate the quality attribute utility tree
6. Analyze the architectural approaches

Testing
7. Brainstorm and prioritize scenarios
8. Analyze the architectural approaches

Reporting
9. Present the results
Step 9: Present Results

- Application of ATAM to GIS uncovered:
  - 16 sensitivity points
  - 10 tradeoff points
  - 13 risks

- Mere employment of architectural decisions insufficient to achieve quality attributes if not employed in light of sensitivity points, tradeoff points & risks
Lessons Learnt: Quality Attribute Scenarios

☑ Understandability: unambiguously define factors controlling achievement of quality attributes

☑ Precision: response & response measure offer specific means for assessing GIS architectures

☑ Traceability: decomposing each quality attribute into scenarios enables traceability of how an attribute is addressed during architectural design & evaluation

✗ Difficulty of defining quality attribute response & response measure for controversial qualities such as modifiability
Lessons Learnt: ADD

☑️ Attribute Driven Design Method:
- Simple architectural design process
- Systematic consideration of quality attributes
- Mapping between quality attribute scenarios & architectural decisions

☑️ Design Documentation:
- Well organized architectural documentation
- Record of architectural design decisions applied, resultant architectural views and underlying design rationale

✗ Difficulty of functional decomposition and allocation (use cases) at early architecture design stages
Lessons Learnt: ATAM

- Assesses main architectural artefacts: requirements & design
- Enhances & enriches architectural documentation
- Articulates parameters needed for effective reuse of architectural design decisions

❌ Quantitative evaluation
  ❌ More specialized assessment techniques required to gauge response measures for qualities such as performance & availability
Related Work at York

• Rolls-Royce:
  – Electronic Engine Controllers (EECs): Architectural definition and evaluation for safety

• Airbus:
  – Employment of use case maps for functionality allocation
  – Definition of safety tactics
  – Incorporation of safety analysis in architectural evaluation
Thank you

Questions?