

## SOA Decision Modeling

Lessons learned on a 3-year industrial knowledge harvesting and sharing project

November 2008

Olaf Zimmermann  
Executive IT Architect  
IBM Zurich Research Lab

© 2008 IBM Corporation



## Abstract

In this presentation, we first motivate the need for architectural decision modeling in SOA case studies. Next, we introduce an architectural decision modeling framework and a reusable SOA decision model which was developed with this framework. As examples, we investigate selected SOA decisions such as the selection of message exchange patterns and the configuration of system transaction boundaries. After that, we demonstrate how to capture architectural decisions and alternatives in Architectural Decision Knowledge Wiki, a Web 2.0 collaboration system and application wiki supporting decision maker collaboration. Framework, reusable SOA decision model and collaboration system jointly support usage scenarios such as education, knowledge exchange, architecture design method, as well as facilitation of technical quality assurance reviews and SOA governance.

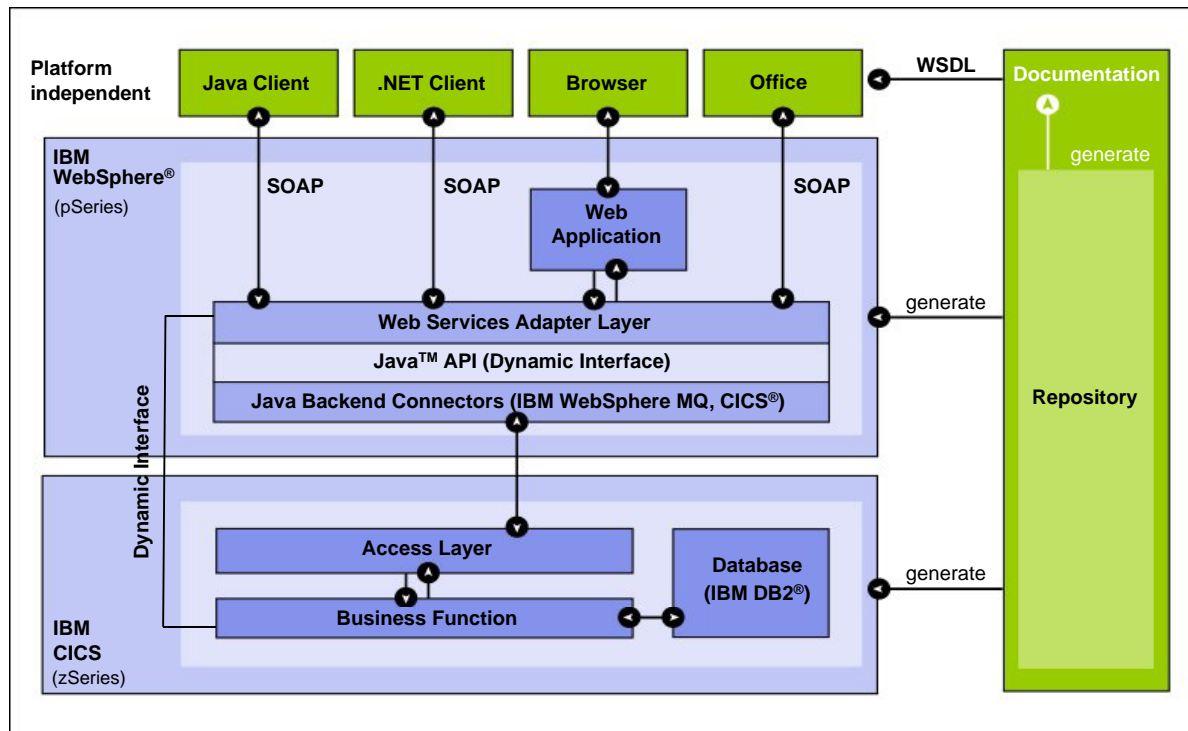
## Agenda

1. State of the practice: case study
  - Broker-based integration of core banking application components
  - Order management in the telecommunications industry
2. Existing work
  - Architectural patterns and decision capturing
3. SOA Decision Modeling (SOAD) contributions
  - Framework and domain meta model
  - Excerpts from Reusable Architectural Decision Model (ADM) for SOA
  - Architectural Decision Knowledge Wiki collaboration system
4. Discussion
5. Conclusion

## Agenda

1. **State of the practice: case study**
  - **Broker-based integration of core banking application components**
  - **Order management in the telecommunications industry**
2. Existing work
  - Architectural patterns and decision capturing
3. SOA Decision Modeling (SOAD) contributions
  - Framework and domain meta model
  - Excerpts from Reusable Architectural Decision Model (ADM) for SOA
  - Architectural Decision Knowledge Wiki collaboration system
4. Discussion
5. Conclusion

## Case Study: Core Banking SOA with Web Services



Zimmermann, O.; Milinski, M.; Craes, M.; Oellermann, F.: **Second Generation Web Services-Oriented Architecture in Production in the Finance Industry**, OOPSLA Conference Companion, 2004

## Some of the Required Architecture Design Activities

Buschmann F., Meunier R., Rohnert H., Sommerlad P., and Stal M., **Pattern-Oriented Software Architecture – a System of Patterns**. Wiley, 1996

- Selection of top-level integration pattern
  - BROKER from “Patterns of Software Architecture” (POSA) is an obvious choice

- POSA suggests six steps to implement the pattern:

“(1) Define an object model. (2) Decide which type of component interoperability the system should offer, binary or Interface Description Language (IDL). (3) Specify the APIs the broker component provides for collaborating with clients and servers. (4) Use proxy objects to hide implementation details from clients and servers. (5) Design the broker component (6) Develop IDL compilers.”

Step (5) has nine sub steps: “(5.1) On-the-wire protocol, (5.2) Local broker, (5.3) Direct communication variant, (5.4) (Un)marshalling, (5.5) Message buffers, (5.6) Directory service, (5.7) Name service, (5.8) Dynamic method invocation, (5.9) The case in which something fails”.

- This is helpful, but *which (design) patterns are suited to implement the steps? What about the required technology choices? Product selections?*

## Multi-Channel Order Management SOA in the Telecommunications Industry (in production since Q1/2005)

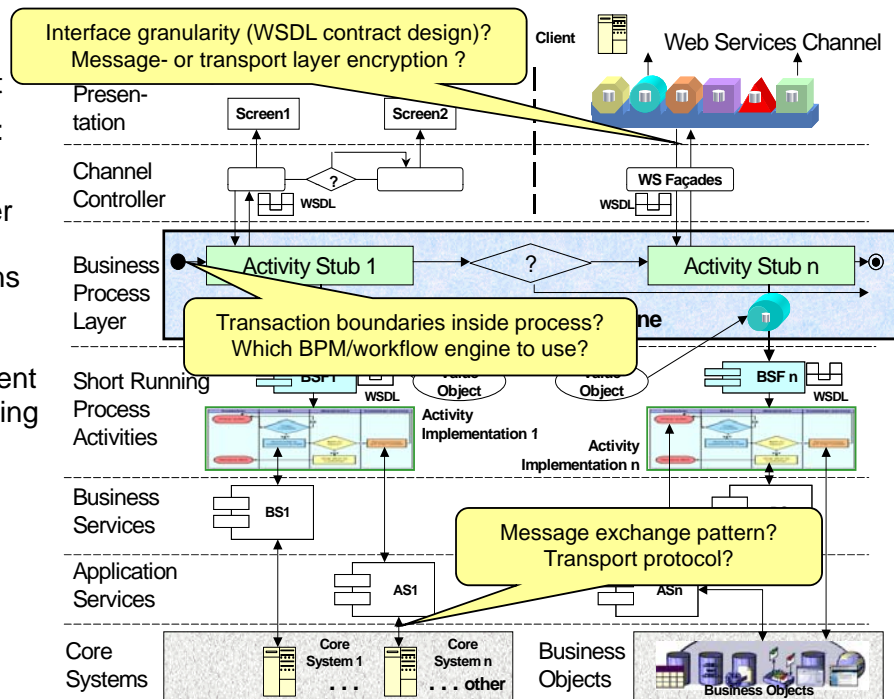
Zimmermann et al, „SOA and BPC in an Order Management Scenario“, <http://soadecisions.org/art.htm#casestudy1>

### Functional domain

- Order entry management
- Two business processes: new customer, relocation
- Main SOA drivers: deeper automation grade, share services between domains

### Service design

- Top-down from requirement and bottom-up from existing wholesaler systems
- Recurring architectural decisions:
  - Protocol choices
  - Transactionality
  - Security policies
  - Interface granularity



## Recurring Service Design Issues and Decisions

1. Selection and adoption of *application and integration patterns*
2. Technology choices – *protocols, containers, operating systems, etc.*
3. Commercial and open source *asset selection and configuration*

### Hundreds, if not thousands, of such decisions per *role, phase, scope*:

- Many decisions already made before project starts
  - By client – previous projects, legacy systems
  - By other parties – software vendor, IT consultants
- Many *forces* that influence decision making, often contradictory [Booch]
- Numerous complex *dependencies* between the decisions [Kruchten]
- Never enough time to analyze all *options* in detail
- Pattern literature is overwhelming

Booch G., **Handbook of Software Architecture**,  
<http://www.booch.com/architecture>

Kruchten P. et al, **Building Up and Reasoning About Architectural Knowledge**, QOSA 2006

## Agenda

1. State of the practice: case study
  - Broker-based integration of core banking application components
  - Order management in the telecommunications industry
2. Existing work
  - **Architectural patterns and decision capturing**
3. SOA Decision Modeling (SOAD) contributions
  - Framework and domain meta model
  - Excerpts from Reusable Architectural Decision Model (ADM) for SOA
  - Architectural Decision Knowledge Wiki (collaboration system)
4. Discussion
5. Conclusion

## Decision Capturing Templates and Ontologies

- IBM Global Services Method a.k.a. Unified Method Framework (UMF)
  - Defines a work product (artifact) “ARC 100 Architectural Decisions”
- Tyree and Akerman created a decision capturing template having worked with an ARC 100 work product for e-business architectures
  - Issue, decision, status, group, assumptions, constraints, positions, argument
  - Implications, related decisions, related artifacts, related principles, notes

J. Tyree, A. Akerman, **Architecture Decisions: Demystifying Architecture**, IEEE Software, vol. 22, no. 2, 2005

- Kruchten et al. define decision and dependency types
  - State model, decision types, decision dependencies

Kruchten, P., Lago, P., van Vliet, H.: **Building up and Reasoning about Architectural Knowledge**. In: Hofmeister, C., Crnkovic, I., Reussner, R. (eds.) QoSA 2006. LNCS, vol. 4214, pp. 39–47. Springer, Heidelberg (2006)

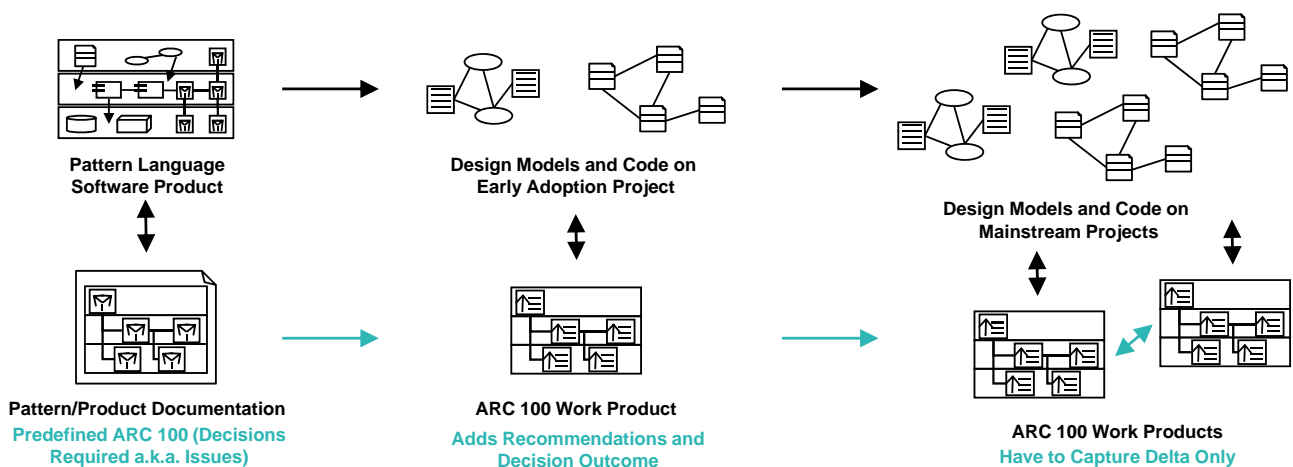
*Retrospective decision capturing for documentation purposes (no reuse)!*

## Agenda

1. State of the practice: case study
  - Broker-based integration of core banking application components
  - Order management in the telecommunications industry
2. Existing work
  - Architectural patterns and decision capturing
3. **SOA Decision Modeling (SOAD) contributions**
  - **Framework and domain meta model**
  - **Excerpts from Reusable Architectural Decision Model (ADM) for SOA**
  - **Architectural Decision Knowledge Wiki (collaboration system)**
4. Discussion
5. Conclusion

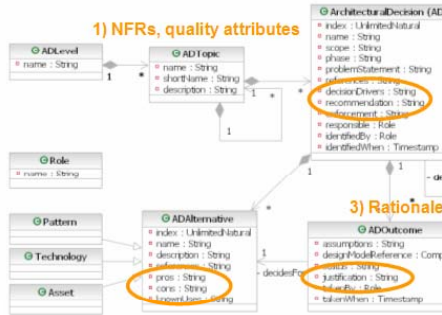
## Retrospective vs. Proactive Decision Capturing/Modeling

- Documenting architectural decisions *should* be state of the practice
  - “For IBM GTS, ARC 100 is most important UMF work product besides Operational Model”
  - Real-world constraints for retrospective decision capturing: motivation, budget, tools?



- Using pre-populated *AD models* in an active, guiding role is a novel approach
  - Give reusable advice regarding decisions still required on project, as opposed to rationale behind decisions already made

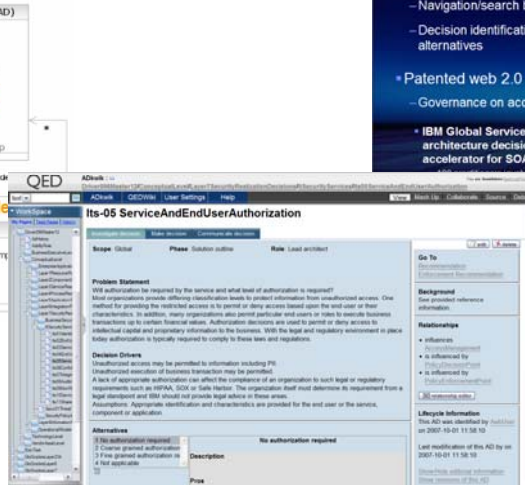
# Modeling Architectural Decisions



2) Advantages and disadvantages of available design options

Part 1: Framework (method/model) to represent architectural decision knowledge and decision outcomes

- Structural representation
- Integrity constraints to ensure consistency of knowledge base and decisions



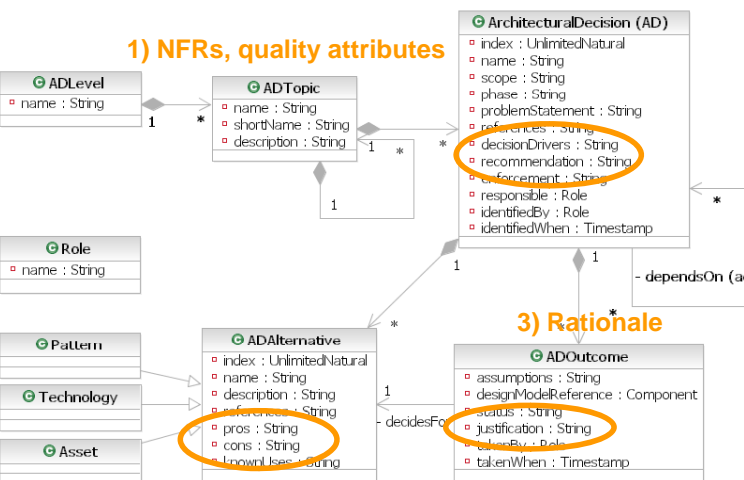
Part 3: Tool that implements the model to view and work with the content

- Evaluated UML (and others) – different modeling paradigm
- Selected QEDWiki for alphaWorks release (an application wiki technology developed by IBM Software Group Emerging Internet Technologies)



Part 2: Content for SOA - Announced by IBM GTS - Proven successful in several customer projects

# IGS Method/UMF ARC 100 template extended into a UML meta model which facilitates reuse and collaboration.

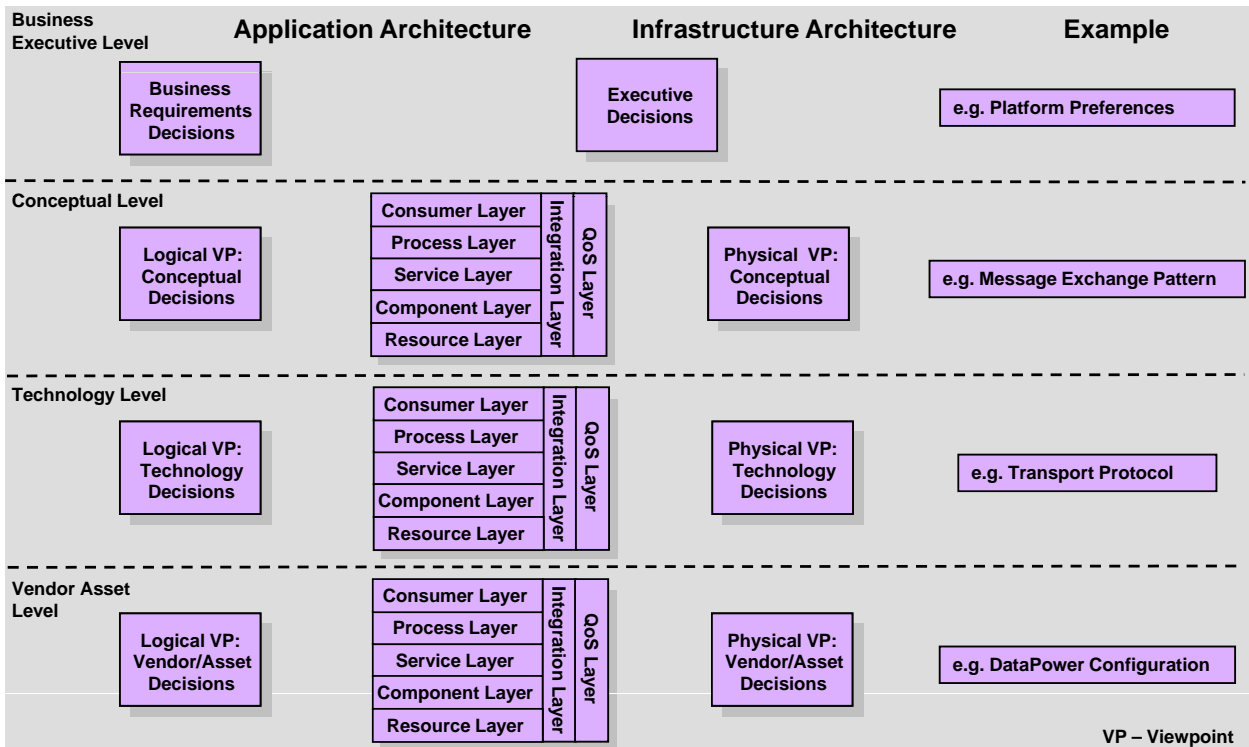


2) Advantages and disadvantages of available design options

- Quality attributes & NFRs:
  - decision drivers and “best practices” recommendation (AD)
  - pros and cons of alternatives (AA)
  - justification (ADOutcome)
- Design model and process alignment:
  - scope attribute (AD)
  - phase attribute (AD)
  - responsibleRole attribute (AD)
- Decision lifecycle:
  - status and owner (ADOutcome)
  - takenWhen (ADOutcome)

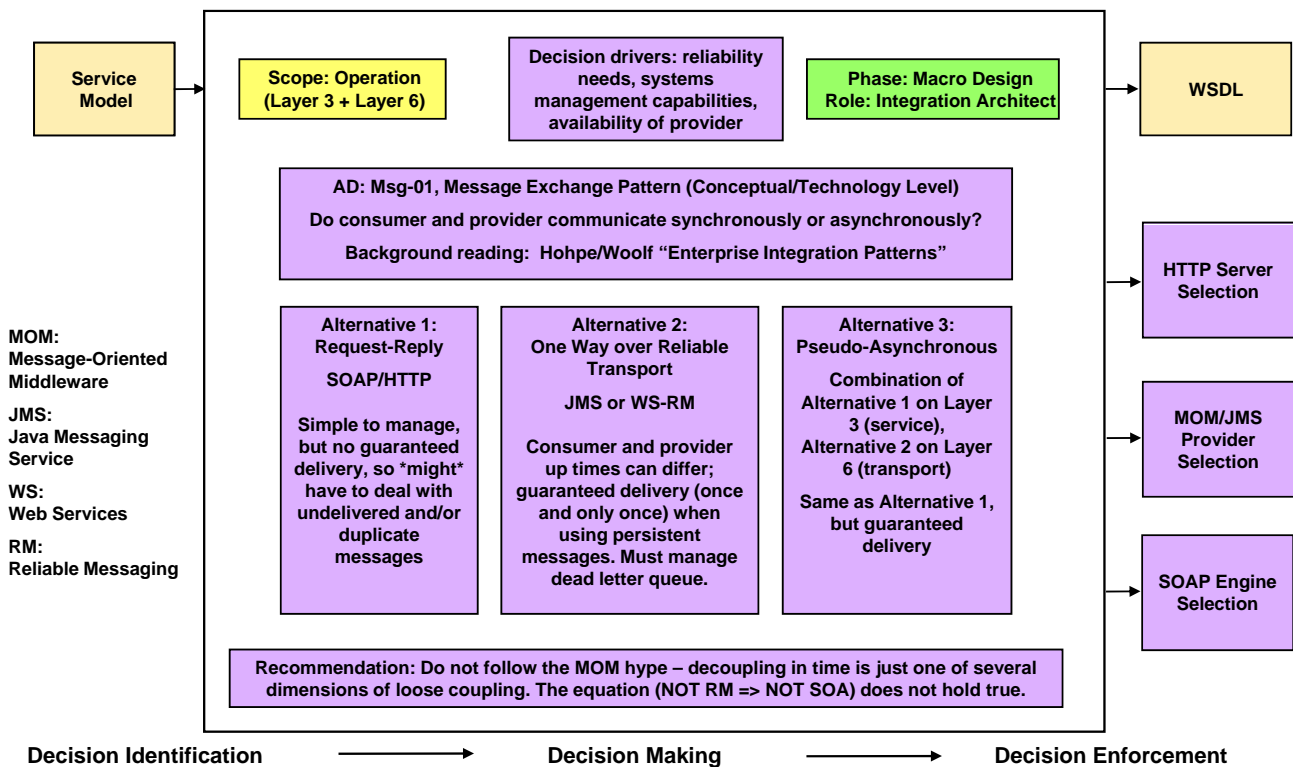
Zimmermann O., Gschwind T., Küster J., Leymann F., Schuster N., Reusable Architectural Decision Models for Enterprise Application Development. In: Third International Conference on the Quality of Software-Architectures (QOSA 2007).

# Decision Model Organized by Layers and Levels



VP – Viewpoint

## Part 2: Sample SOA Decision Content (AD, ADAAlternative)



MOM: Message-Oriented Middleware  
JMS: Java Messaging Service  
WS: Web Services  
RM: Reliable Messaging

## More SOA Decisions

- Selection of a *reference architecture* and *layering schema*
  - IBM SOA Solution Stack, SOA/ODOE/WBI Reference Architecture, other
- Governance decisions
  - *Service ownership* and *lifecycle management*
  - *Registry type*, *publishing*, and *lookup policies* (who, when, how)
- *Information management* directions
  - Master data management yes/no?
  - Virtualization yes/no?
- *Service parameter origin* and *granularity*?
  - Enterprise data model reference/persistent or local copy/transient?
- Decisions and patterns and for *transactional workflows in SOA*

Zimmermann O., Grundler J., Tai S., Leymann F., **Architectural Decisions and Patterns for Transactional Workflows in SOA**. In: Krämer, B., Lin K.-J., Narasimhan, P. (eds.): ICSSOC 2007, LNCS 4749, Springer, Heidelberg (2007)

## Part 3 (Tool): Architectural Decision Knowledge Wiki

The screenshot shows the QED Architectural Decision Knowledge Wiki interface. The main content area displays the details for the decision 'Its-05 ServiceAndEndUserAuthorization'. The interface includes a navigation pane on the left, a search bar at the top, and a main content area with sections for Scope, Problem Statement, Decision Drivers, Alternatives, and Lifecycle Information. The 'Alternatives' section lists four options, with 'No authorization required' selected. The 'Lifecycle Information' section shows the decision was identified by AwbUser on 2007-10-01 11:58:10.

**QED** ADKwik : [Home](#) [User Settings](#) [Help](#) [View](#) [Mash Up](#) [Collaborate](#) [Source](#) [Debug](#)

**Its-05 ServiceAndEndUserAuthorization**

Investigate decision | Make decision | Communicate decision

Scope: Global    Phase: Solution outline    Role: Lead architect

**Problem Statement**  
Will authorization be required by the service and what level of authorization is required?  
Most organizations provide differing classification levels to protect information from unauthorized access. One method for providing the restricted access is to permit or deny access based upon the end-user or their characteristics. In addition, many organizations also permit particular end-users or roles to execute business transactions up to certain financial values. Authorization decisions are used to permit or deny access to intellectual capital and proprietary information to the business. With the legal and regulatory environment in place today authorization is typically required to comply to these laws and regulations.

**Decision Drivers**  
Unauthorized access may be permitted to information including PII.  
Unauthorized execution of business transaction may be permitted.  
A lack of appropriate authorization can affect the compliance of an organization to such legal or regulatory requirements such as HIPAA, SOX or Safe Harbor. The organization itself must determine its requirement from a legal standpoint and IBM should not provide legal advice in these areas.  
Assumptions: Appropriate identification and characteristics are provided for the end user or the service, component or application.

**Alternatives**

Alternative	Description	Pros
1 No authorization required		
2 Coarse grained authorization		
3 Fine grained authorization re		
4 Not applicable		

**Go To**  
[Recommendation](#)  
[Enforcement Recommendation](#)

**Background**  
See provided reference information.

**Relationships**

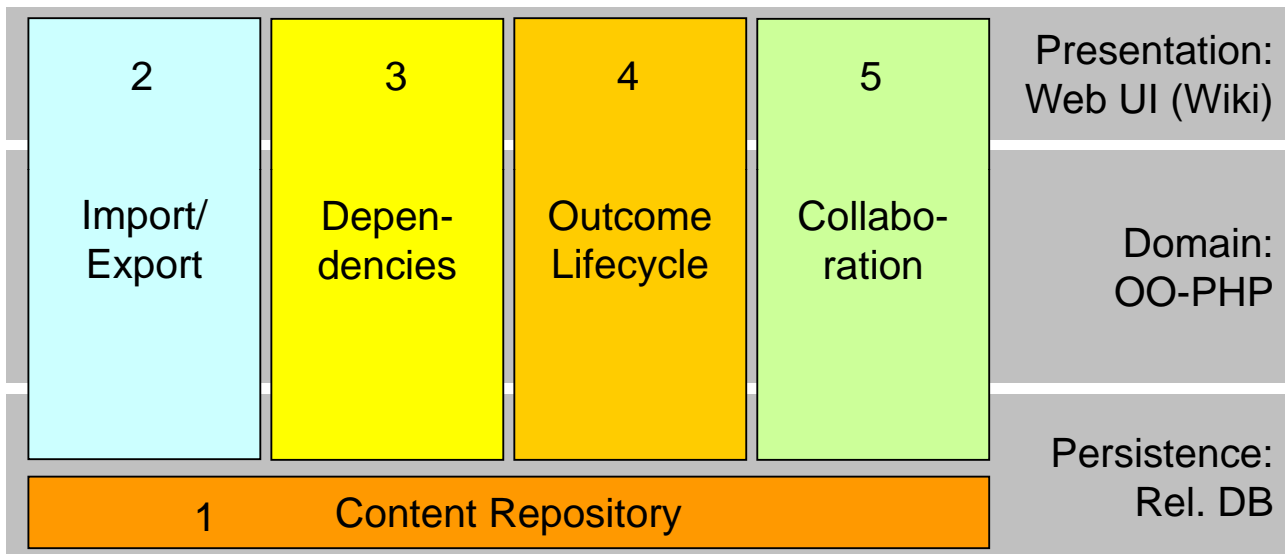
- influences [AccessManagement](#)
- is influenced by [PolicyDecisionPoint](#)
- is influenced by [PolicyEnforcementPoint](#)

[relationship editor](#)

**Lifecycle Information**  
This AD was identified by AwbUser on 2007-10-01 11:58:10  
Last modification of this AD by on 2007-10-01 11:58:10  
[Show/Hide editorial information](#)  
[Show revisions of this AD](#)

Download URL: <http://www.alphaworks.ibm.com/tech/adkwik>

## Architecture of Architectural Decision Knowledge Wiki



Schuster N., Zimmermann O., Pautasso C., **ADkwik: Web 2.0 Collaboration System for Architectural Decision Engineering**. In: Proc. of the Nineteenth International Conference on Software Engineering & Knowledge Engineering (SEKE 2007)

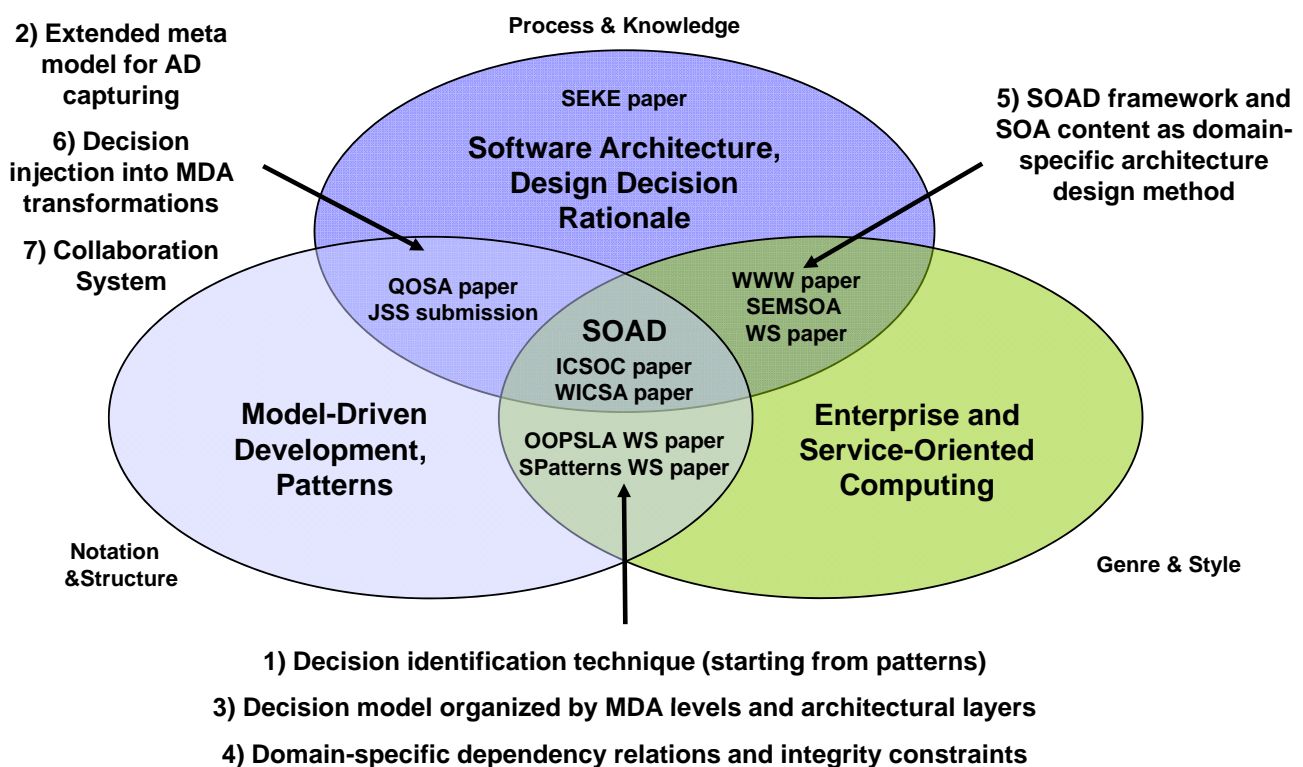
## Agenda

1. State of the practice: case study
  - Broker-based integration of core banking application components
  - Order management in the telecommunications industry
2. Existing work
  - Architectural patterns and decision capturing
3. SOA Decision Modeling (SOAD) contributions
  - Framework and domain meta model
  - Excerpts from Reusable Architectural Decision Model (ADM) for SOA
  - Architectural Decision Knowledge Wiki (collaboration system)
4. Discussion
5. Conclusion

## Summary of Key Concepts

- **Decision models** rather than text templates
  - Three levels of refinement from concepts to technology to assets
  - Population of decision space based on experience from earlier projects
- Decision model as domain-specific guide through the stages, **architecture and design patterns** describe alternatives in detail
  - Architectural decisions are conceptual alternatives
  - Design and technology patterns come in on subsequent stages
- Prescriptive **design method** for domain-specific architecture design
  - Comprehensive, but still comprehensible
  - Works for domains and styles with recurring decisions for which patterns have been mined (new decision alternatives also are a pattern source)
  - For example: enterprise applications and SOA

## Summary of Research Contributions



## Validation via Case Studies and Action Research

- **Gained feedback from practitioners on SOA industry projects**
  - Five client engagements in several industries (telecommunications, government, retail and distribution)
  - Education, review, and governance workshops (professional services)
- **Applied method to own projects retrospectively**
  - Case study in this presentation plus BPEL-based telco order management
  - Revised and integrated book content:
    - “Perspectives on Web services “(26 ADs dealing with Web services design)
    - “Patterns of Enterprise Application Architecture”, “Enterprise Integration Patterns”, Pattern language for distributed computing (POSA 4)
- **Released Architectural Decision Knowledge Wiki**
  - 150 registered users in hosted company-internal instance

## Agenda

1. State of the practice: case study
  - Broker-based integration of core banking application components
  - Order management in the telecommunications industry
2. Existing work
  - Architectural patterns and decision capturing
3. SOA Decision Modeling (SOAD) contributions
  - Framework and domain meta model
  - Excerpts from Reusable Architectural Decision Model (ADM) for SOA
  - Architectural Decision Knowledge Wiki (collaboration system)
4. Discussion
5. **Conclusion**

## Summary – ArchPad (Method), RADM for SOA (Content), Architectural Decision Knowledge Wiki (Tool)

- **Method:** Decision model as domain-specific guide through the stages, **architecture and design patterns** describe alternatives in detail
  - Architectural decisions are conceptual alternatives
  - Design and technology patterns come in on subsequent stages
- **Content:** **Decision models** rather than text templates
  - Four refinement levels: executive, conceptual, technology, and vendor asset decisions
  - Pre-population of RADM from experience gained on earlier SOA projects
- **Tool:** **Application wiki** for decision knowledge sharing and decision maker collaboration
  - Wiki frontend for model-driven domain layer and relational database
  - 50+ use cases: CRUDS on domain model elements, ARC 100 report generation

## Advanced Usage Scenarios and Future Research

- **Project planning and health checking**
  - Work breakdown structures can be created from a decision model, as open decisions correspond to required activities
  - If there are many, frequent changes, or many questions are still unresolved in late project phases, the project is likely to be troubled
- Decision models as **input to software configuration planning**
  - Product selection and operational modeling decisions define which software licenses are required, and on which hardware nodes the required software has to be installed
- **Enterprise architecture** instrument
  - Company-specific instance of the SOA RADM, consisting of a subset of decisions and alternatives to give freedom of choice to individual project teams without sacrificing overall architectural integrity

## Vision: Enterprise-Wide RADM

- *Enterprise architecture group owns and maintains RADM*
  - Input comes from solution architects on development/integration projects
  - Quality assured, aligned with enterprise IT strategy
- *Does not mandate a particular architecture, but frames design work*
  - Recommend certain alternatives:
    - E.g. “always use document/literal SOAP”
  - Ban others:
    - E.g. “no open source assets can be used due to open legal issues”
  - Finding a *balance between freedom-of-choice and freedom-from-choice*
- Allows project teams to share lessons learned and best practices
  - EAM perceived as friend, not foe ☺

## Thank You!

- Questions?
- Comments?
- “Perspectives on Web Services”, Springer-Verlag
  - Captures project experience 2001-2003, incl. 26 reusable architectural decisions
  - Foreword by Grady Booch
  - Website also has case study reports and other referenced papers

<http://www.soadecisions.org>

