

Application Servers G22.3033-011

Session 2 – Sub-Topic 2 Enterprise Architecture Frameworks (EAFs) & Pattern Driven EAFs

Dr. Jean-Claude Franchitti

*New York University
Computer Science Department
Courant Institute of Mathematical Sciences*

1

Definition: Software Architecture

- Software Architecture
 - A set of artifacts (that is: principles, guidelines, policies, models, standards, and processes) and the relationships between these artifacts, that guide the selection, creation, and implementation of solutions aligned with business goals
 - Software architecture is the structure of structures of an information system consisting of entities and their externally visible properties, and the relationships among them
 - A software architecture is a description of the subsystems and components of a software system and the relationships between them
 - Subsystems and components are typically specified in different views to show the relevant functional and non-functional properties of a software system
 - The software system is an artifact. It is the result of the software design activity

2

Definition: Component

- **Component**
 - A component is an encapsulated part of a software system
 - A component has an interface
 - Components serve as the building blocks for the structure of a system
 - At a programming-language level, components may be represented as modules, classes, objects or as a set of related functions

3

Definition: Subsystem

- **Subsystem**
 - A subsystem is a set of collaborating components performing a given task
 - A subsystem is considered a separate entity within a software architecture
 - It performs its designated task by interacting with other subsystems and components...

4

Definition: Architectural Style

- Architectural Style
 - An architectural style is a description of component types and their topology
 - It also includes a description of the pattern of data and control interaction among the components and an informal description of the benefits and drawbacks of using that style
 - Architectural styles are important engineering artifacts because they define classes of designs along with their associated known properties
 - They offer experience-based evidence of how each class has been used historically, along with qualitative reasoning to explain why each class has its specific properties
 - *"An architectural style is a coordinated set of architectural constraints that restricts the roles/features of architectural elements and the allowed relationships among those elements within any architecture that conforms to that style."*

5

Definition: Framework

- Framework
 - A set of assumptions, concepts, values, and practices that constitutes a way of viewing the current environment
 - A software framework is a partially complete software (sub-) system that is intended to be instantiated
 - It defines the architecture for a family of (sub-) systems and provides the basic building blocks to create them
 - It also defines the places where adaptations for specific functionality should be made

6

Definition: ABASs

- **ABASs: Attribute Based Architectural Styles**
 - ABASs build on architectural styles to provide a foundation for more precise reasoning about architectural design by explicitly associating a reasoning framework (whether qualitative or quantitative) with an architectural style
 - These reasoning frameworks are based on quality attribute-specific models, which exist in the various quality attribute communities (such as the performance and reliability communities).

7

Definition: Architectural Pattern

- **Architectural Pattern**
 - An architectural Pattern expresses a fundamental structural organization schema for software systems
 - It provides a set of predefined subsystems, their responsibilities, and includes rules and guidelines for organizing the relationships between them

8

Definition: Design Pattern

- Design Pattern
 - A design pattern provides a scheme for refining the subsystems or components of a software system, or the relationships between them
 - It describes a commonly-recurring structure of communicating components that solves a general design problem within a particular context.

9

Definition: Idioms

- Idioms
 - An Idiom is a low-level pattern specific to a programming language
 - An idiom describes how to implement particular aspects of components or the relationships between them using the features of the given language

10

Enterprise Reference Architectures

- Enterprise Reference Architectures (ERAs) are “self-contained” Architectural Styles that provide all the ingredients required to define or support a business purpose in the Enterprise
 - ERAs are either generic to support custom development or domain/context specific
- Sample Horizontal ERAs
 - Service Management Architecture (SMA)
 - SMA is a reference architecture that encapsulates the SOA, STA, SOI, SOP, and SOM architectural styles
 - Object Management Architecture (OMA)
 - Enterprise Reference Architecture defined by the Object Management Group in the mid-90s
 - <http://www.omg.org/gettingstarted/specintro.htm#OMA>
- Sample Vertical/Domain ERAs
 - Enterprise Content Management (ECM)
 - Customer Relationship Management (CRM)
 - Business Intelligence (BI)
 - Business Process Management (BPM)
 - Enterprise Resource Planning (ERP)
 - Groupware/Collaboration

11

Enterprise Reference Elements

- Reference elements correspond to building blocks (a.k.a., prime citizens) in a given reference architecture
 - For example, application components are building blocks in component-based architectures developed with modern application servers based on the OMA ERA
- The following patterns are part of the SMA Enterprise Reference Elements family:

- **Component Services**

- A componhttppt Sei2ecks2(n)-7a53i2eri2eni2ea53apponhttp app2 i c9Relat sreni2ee) suppor2(n)-7.3(si.9(e ponh(aCus.6(i2e)6.53)17 ponh(a5 i 4(.53)17elati

12

Sample Architectural Styles

- **Service Oriented Architecture (SOA)**
 - SOA, at a basic level, is an architectural style made up of a collection of loosely coupled services regardless of whether they have a technical or a business focus
 - More generally, SOA is a software architecture of services, policies, practices and frameworks in which components can be reused and repurposed rapidly in order to achieve shared and new functionality
 - This enables rapid and economical implementation in response to new requirements thus ensuring that services respond to perceived user needs
 - At the Enterprise level, SOA is typically used to implement business functionality as a set of shared reusable business services
 - In this context, technical services such as underlying heterogeneous systems are exposed purely as business services
 - SOA uses the object-oriented principle of encapsulation in which entities are accessible only through interfaces and where those entities are connected by well-defined interface agreements or contracts

13

Sample Architectural Styles (cont.)

- **Service Trader Architecture (STA)**
 - The part of SMA that enables finding and binding to loosely coupled services in an implementation independent way
- **Service Oriented Integration (SOI)**
 - SOI is an SMA infrastructure component that enables the integration of loosely coupled services
- **Service Oriented Process**
 - In a Service Oriented Process (SOP), a process consists of an orchestrated flow of services, and the process itself is exposed as a service
 - In this manner, the actual details of the process are abstracted from the client application that consumes these services
 - Another benefit of a SOP is that it doesn't specify any particular user interface - the process may be consumed automatically as part of a behind-the-scenes integration activity, or it may be exposed to the user via a portal as part of an interactive, workflow activity
- **Service Oriented Management (SOM)**
 - SOM is an SMA infrastructure component that facilitates the use of business-neutral services for monitoring, auditing, logging, notification, and security purpose
 - SOM is an essential prerequisite for SMA as it enables loose coupling and coarse granularity, enforces the Quality of Service (QoS) of SOI, and enables SOP by managing business services and the processes that link them

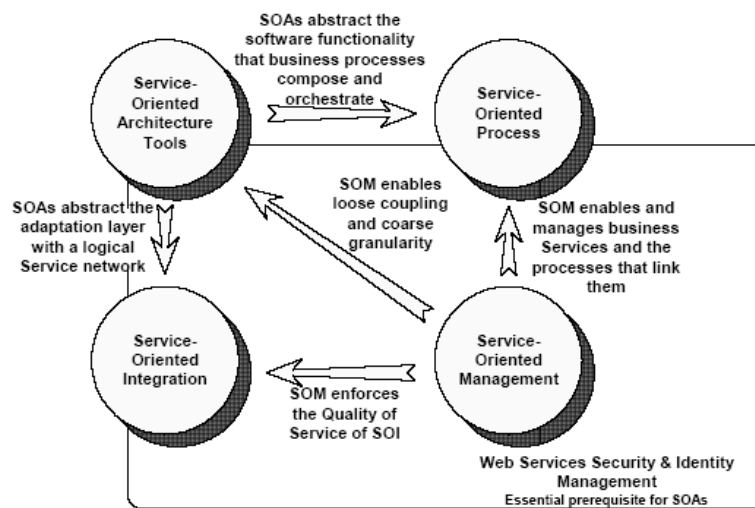
14

Sample Architectural Styles (cont.)

- N-Tier
- EAI (Enterprise Application Integration)
- MOM (Message Oriented Middleware)
- POP (Presentation Oriented Publishing)
- Data Warehouse
- Portal-Centric Architecture
- Pipe and Filters
- Distributed Component
- Product Line
- Product
- Enterprise Application
- Stand Alone Architecture
- Client Server Architecture
- Message Bus
- REST (Representational State Transfer)
- RPC (Remote Procedure Call)
- etc.

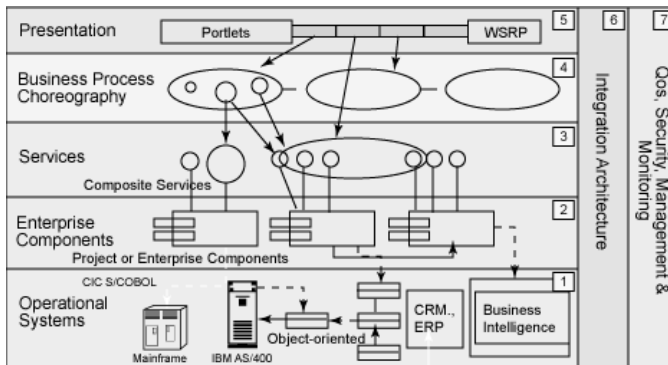
15

Service Management Architecture



16

Service Management Architecture



- STA Support: Layers 1-3 (SMA.STA)
- SOP Support: Layer 4 (SMA.SOMA.SOP)
- SOI Support: Layer 6 (SMA.SOMA.SOI)
- SOM Support: Layer 7 (SMA.SOMA.SOM)
- SOM STA Service Group Support: Layer ->STAS Support: Layer 5 (SMA.SOMA.SOM->STAS)

17

Popular EAFs

- Gartner's EAF
 - MetaGroup's EPAS and AIS Programs
- Zachman's EAF
 - <http://www.zifa.com/>
- Open Group's TOGAF
 - <http://www.opengroup.org/architecture/togaf8/index8.htm>
- Treasury Enterprise Application Framework (TEAF)
 - <http://www.software.org/pub/architecture/teaf.asp>
- Feature Oriented Domain Analysis (FODA)
 - <http://www.sei.cmu.edu/domain-engineering/FODA.html>
- Pattern Driven EAF ...

18

Enterprise Perspective-View Grid

Enterprise Perspectives				
	Business	Information	Application	Technology

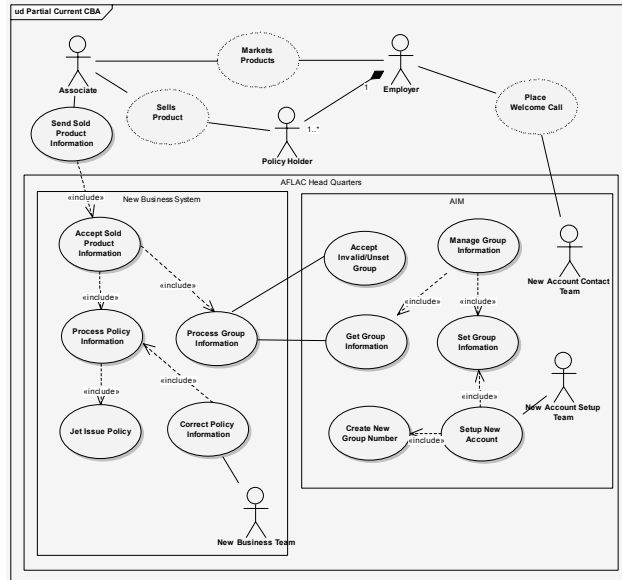
19

Filling up the Business Grid

Architectural Views	Model					
	Impl.					
	Product					
	Deployment					

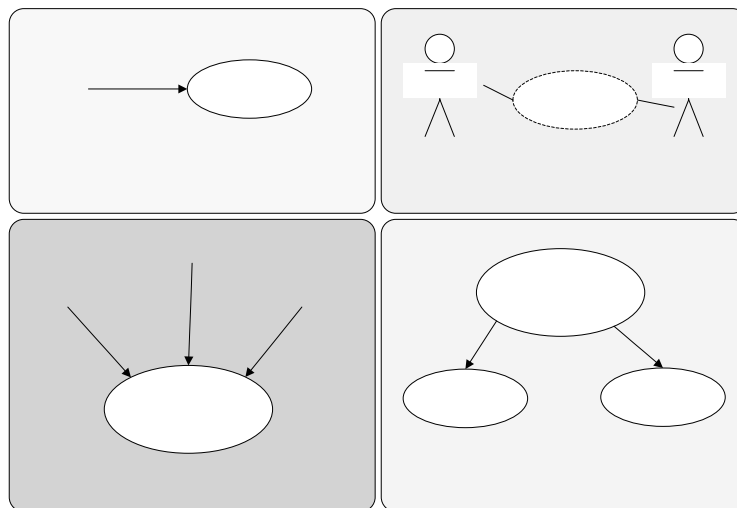
20

Sample CBA Diagram



21

Identifying Possible Reference Architectures



22

Business Model Terminology

- Business Process
 - A long running set of actions or activities performed with specific business goals in mind
 - Business processes typically encompass multiple service invocations
 - Examples of business processes are: *Initiate New Employee*, *Sell Products or Services*, and *Fulfill Order*
 - In SOA terms, a business process consists of a series of operations which are executed in an ordered sequence according to a set of business rules
 - The sequencing, selection, and execution of operations is termed service or process *choreography*
 - Typically, choreographed services are invoked in order to respond to business events.

23

Business Model Terminology

(continued)

- Choreography
 - A choreography is the observed sequence of message exchanged by peer services when performing a unit of work
 - Services do not need to be orchestrated to perform a unit of work (this is a concept that emerged and should have stayed in the last century)
 - This is a very common misconception, actually most units of work are accomplished by a series of "orchestrated services" performing a choreography
 - There are several industry efforts in the area of choreography languages, such as BPML (defined by BPML.org), BPSS (defined by ebXML), IBM's WSFL, Microsoft's XLANG, and IBM/Microsoft/BEA's BPEL4WS and their companion specifications WS-Coordination and WS-Transaction, etc.

24

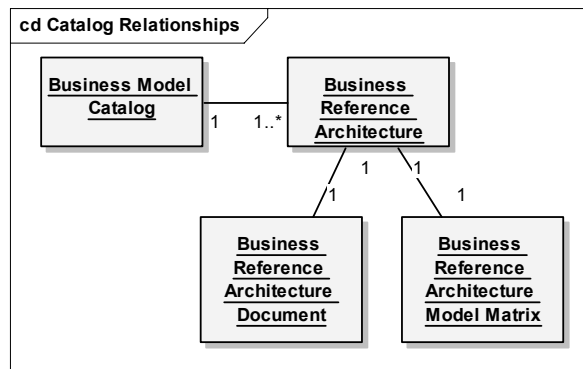
Business Model Terminology

(continued)

- Orchestration
 - An orchestration is a generalization of composition that sequence services and provide additional logic to process data that does not include data presentation
 - The same language can be used to perform a complex unit of work achieved by invoking a series of service operations
 - Any given orchestration is not forced to expose a service interface
 - If it does, it is a composition
 - An orchestration is executed by an orchestration engine
 - BPEL is an orchestration programming language

25

Business Model Catalog



26

Business Reference Architecture Model Matrix

27

Various Types of Model Matrices

- Catalog shows all applicable styles and patterns for a reference architecture
- Standard model matrix instance shows only the patterns that apply to the ideal solution of a problem
- Domain Specific Model Matrix instance shows only the patterns that apply to a solution imposed by the problem constraints and restrictions

28

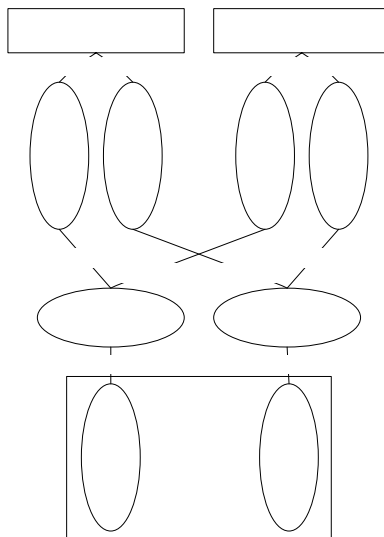
Business
Architectural
Styles

Business

Capability and Requirements Matrix

29

Adding a New Reference Architecture



30

Motivation

Busi
(resulti
Techn

Sample Non-Functional Capabilities

- Project-Based NFCs
 - Accuracy
 - Availability
 - Efficiency
 - Extensibility - Upgradeability – Modifiability – Adaptability- Flexibility
 - Interoperability
 - Portability
 - Recoverability
 - Reliability – Dependability
 - Reusability
 - Scalability – Capacity
 - Security – Accessibility – Anonymity- Vulnerability
 - Usability – Operability
- Organizational NFCs
 - Readability – Simplicity – Understandability
 - Maintainability
 - Testability – Verifiability
 - Traceability
- External NFCs
 - Ethical
 - Legislative (Privacy – Safety)
 - Planning (Cost, development time)

31

Sample Functional Capabilities

- OMA Specific Services
 - Concurrency Service (<http://www.omg.org/docs/formal/00-06-14.pdf>)
 - Externalization Service (<http://www.omg.org/docs/formal/00-06-16.pdf>)
 - Event Service
 - Interface Invocation Service
 - Life Cycle Service
 - Naming and Directory Services (<http://www.omg.org/docs/formal/04-10-03.pdf>)
 - Notification Service
 - Persistence State Service
 - Security Service
 - Trading Object Service
 - Transaction Service
- OMA Specific Facilities
- OMA Application Objects

32

Application Perspective Viewpoints

Microsoft's Enterprise Solution Patterns' Viewpoints

Database
Application

Deployment
Infrastructure

Gartner's Expanded Architecture Framework's Viewpoints

Data
Application
Integration
Point of Access

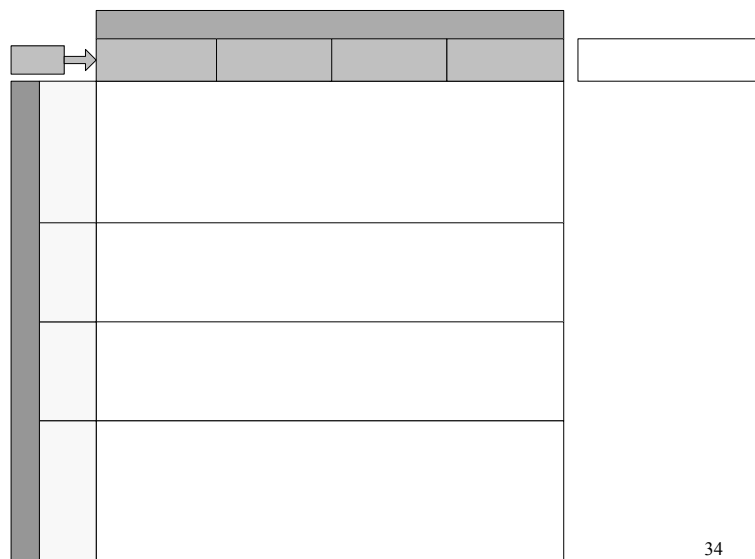
n/a
n/a

PDA EAF Viewpoints

Data
Application
Integration
Point of Access
Moved to Deployment Mapping
Moved to Infrastructure Mapping

33

Filling up the Application Grid



34

Using a PDA EAF

- Gather problem definition – Business Requirements
- Create Conceptual Business Architecture Diagrams
- Create Business Catalogs
 - Business model matrix (BMM) captures reusable business reference architectures, architectural styles and patterns
 - Business implementation matrix (BIM) captures reusable reference implementations, styles and implementation patterns
 - The implementation view is prescriptive and the model view is descriptive
- Run Through Decomposition Process
 - Populate Standard and Domain Specific Business Model Matrices

35

Using a PDA EAF (continued)

- Populate Capabilities and Requirements Matrix (CR-Matrix)
 - Before the architects start creating instances of CR-Matrices several questions must be answered:
 - What is the primary viewpoint for this business problem?
 - Which patterns are related to each other across viewpoints?
 - Which patterns or styles do not contribute to a technology solution pattern?
 - Let us assume that the answers to the above questions are:
 - The primary viewpoint is **Process**.
 - The related patterns across viewpoints are **EBP.Producer.LowVolume** and **C2B.RequestResponse.FastAccess**.
 - The **GroupsOfIndividuals** and **Centralized** styles do not contribute to the business problem solution .
 - Then
 - Create one CR-Matrix instance for each pattern in the primary viewpoint.
 - Create one CR-Matrix instance for each set of related patterns across viewpoints, and do not include non-contributing patterns.
 - As follows:
 - EBP.Producer.LowVolume – C2B.RequestResponse.FastAccess
 - EBP.Transformer.HighVolume.
 - Policies are entered into the CR-Matrix

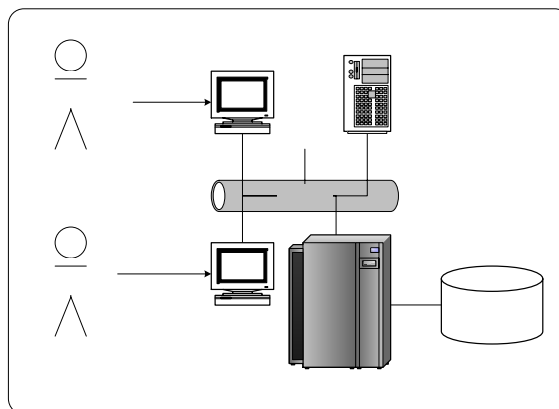
36

Using a PDA EAF (continued)

- Using a CR-Matrix
 - Generation of the Conceptual Technology Architecture Diagram
 - Identification and Confirmation of Appropriate Reference Architecture(s)
 - Generation of a Logical Architecture Analysis Diagram
 - Generation of the Analysis Model
 - Identification of Applicable Pattern(s)
 - Generation of a Logical Architecture Design Diagram
 - Generation of the Design Model
 - Identification of Applicable Reference Implementation(s)
 - Identification of Applicable Implementation Pattern(s)
 - Refinement of the Logical Architecture Design Diagram
 - Refinement of the Design Model
- Product Mapping
- Deployment
- Working with Developer
- Deployment Mapping

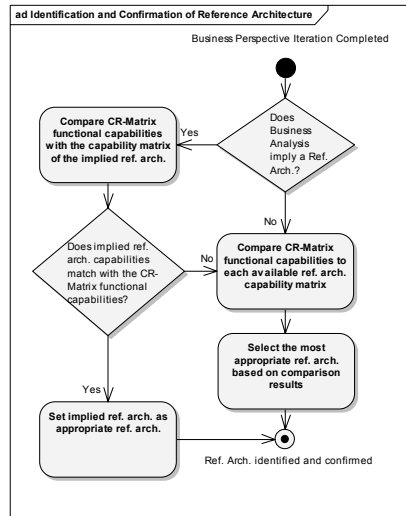
37

Sample CTA Diagram



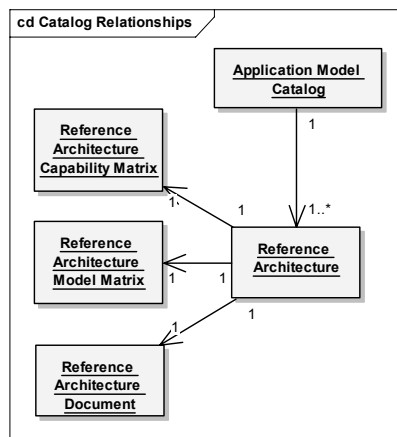
38

Identifying/Confirming RA



39

Application Model Catalog Relationships



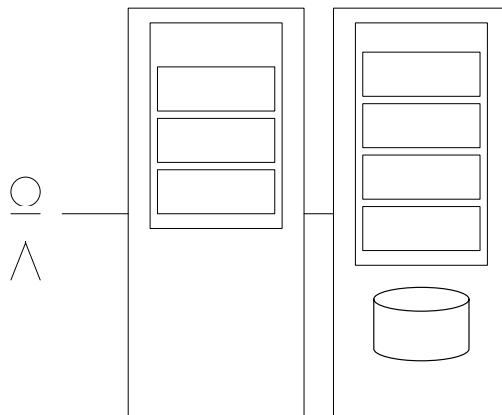
40

Working with Application Model Catalogs

- The application model catalog contains a list of the reference architectures
- For each reference architecture in the application model catalog there is a:
 - Document which explains the reference architecture in detail. The document may contain many links to more detailed explanations, graphics and tables
 - Model matrix which identifies the related patterns for that reference architecture
 - Capability matrix which lists the functional capabilities of the given reference architecture
 - A capability matrix for a reference architecture (Application Server) looks like the following:

41

Sample Logical Architecture Analysis Diagram



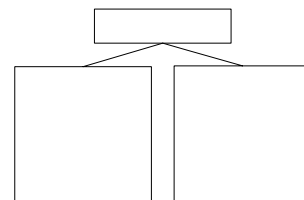
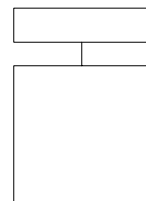
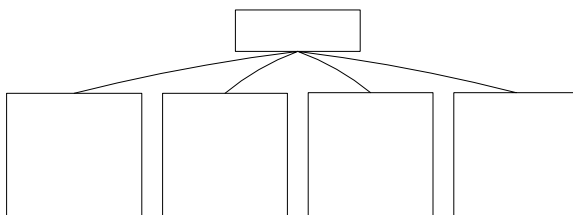
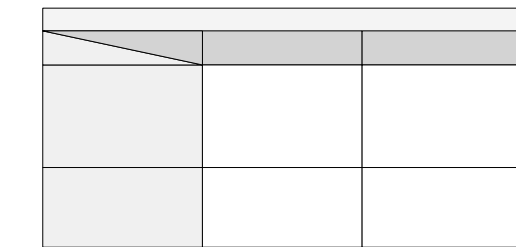
42

Identification of Applicable Patterns

- NF policies identified in the CR-Matrices are used to determine appropriate styles and patterns for the solution
 - Not all non-functional capabilities lead to software design patterns
 - Some non-functional capabilities such as reliability, availability, recoverability and dependability (and possibly others) require hardware deployment patterns along with organizational behavior changes towards quality
- There is no (bullet-proof) defined process that would help an architect to identify appropriate patterns
 - A pattern can be applicable to a certain problem but it may not be appropriate
 - The identification of the applicable and appropriate pattern requires immense working pattern knowledge which is not only knowing and understanding what patterns are but also recognizing when and when not to use certain patterns
 - NFR Framework Based Approach is suggested

43

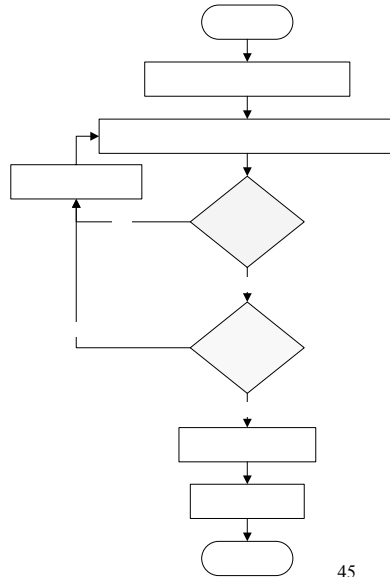
Sample Guidelines



44

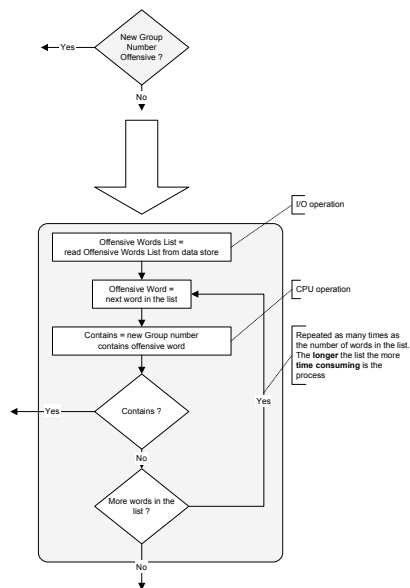
Sample Design Considerations

- Sample Steps: Generation of a new group number
- Steps are:
 - Read the last used group number from the persistence store
 - Calculate the next group number based on an algorithm
 - Validate the calculated group number
 - Check that it does not include offensive words
 - Check that it has never been used before
 - If the newly calculated group number is not valid go to step 2. Otherwise, update the last used group number persistence store with the newly calculated group number
 - Return the newly calculated group number to the client



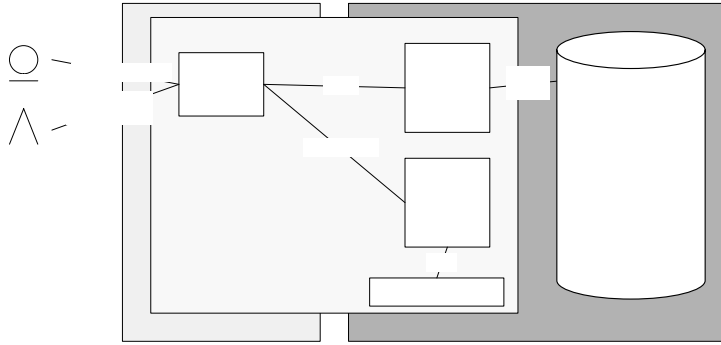
Sample Design Considerations (cont.)

Improvement on Efficiency.Time



Sample Design Considerations (cont.)

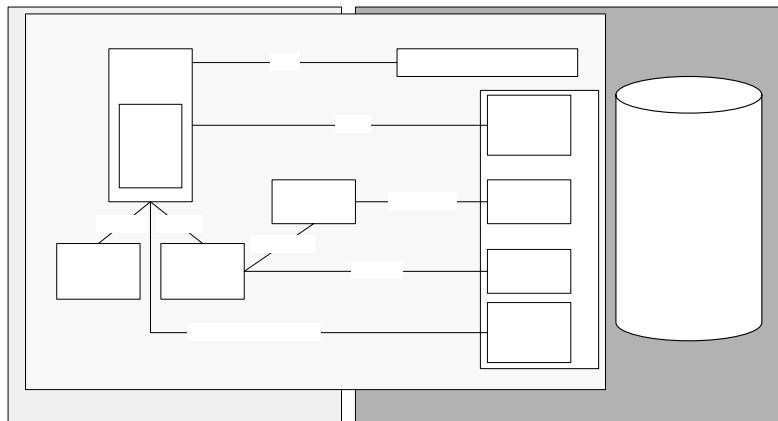
Suggested Synchronous Layer



47

Sample Design Considerations (cont.)

Suggested Asynchronous Layer



48

AIM
Employee>>

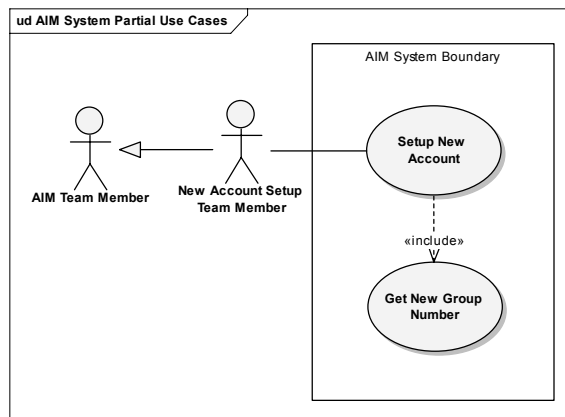
G
G

Sample Design Considerations (cont.)

Standard Application Model Matrix

49

Sample Design Model

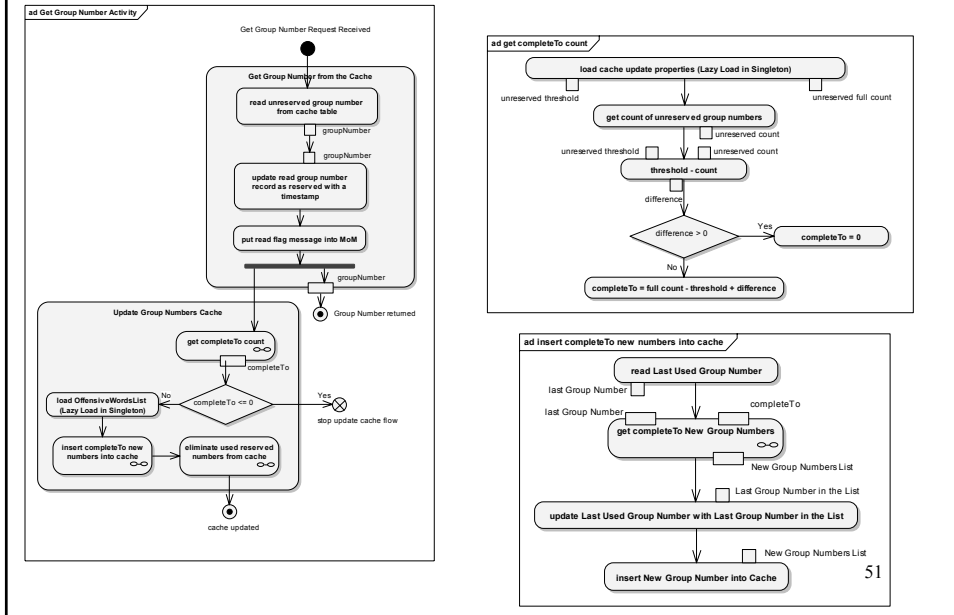


50

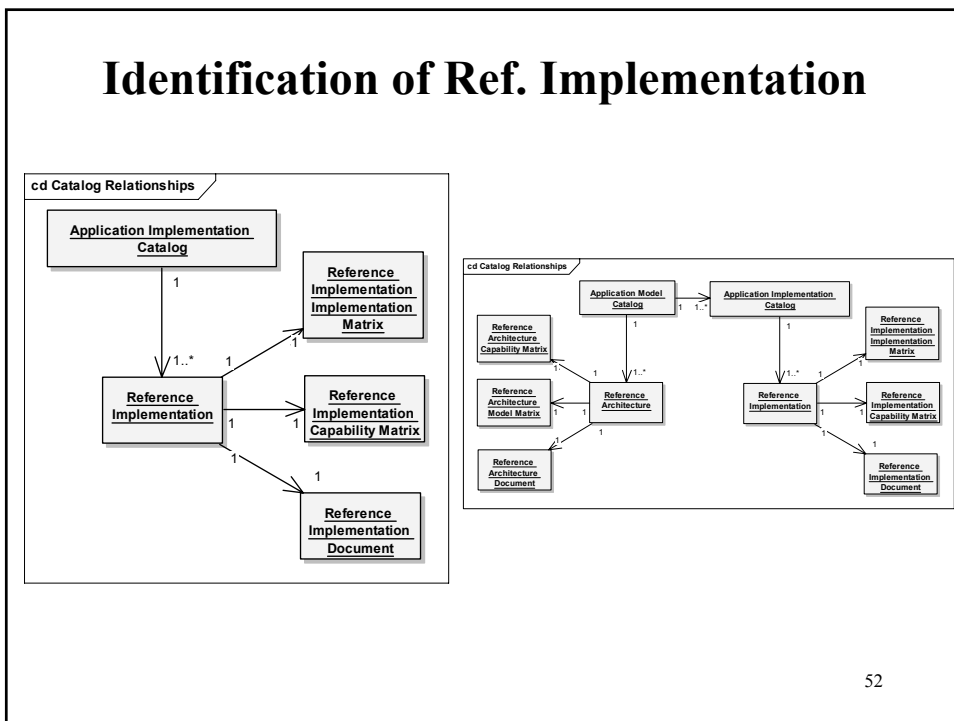
Architectural S
Reference Ele

Architectural P

Sample Design Model (cont.)



Identification of Ref. Implementation



Product Mapping

