
Patterns & Techniques for Separating Business Logic in New SOA System Development

Iyad Jabri
Intelliun Corporation
CEO

Building a Service Oriented Architecture with BPM & MDA

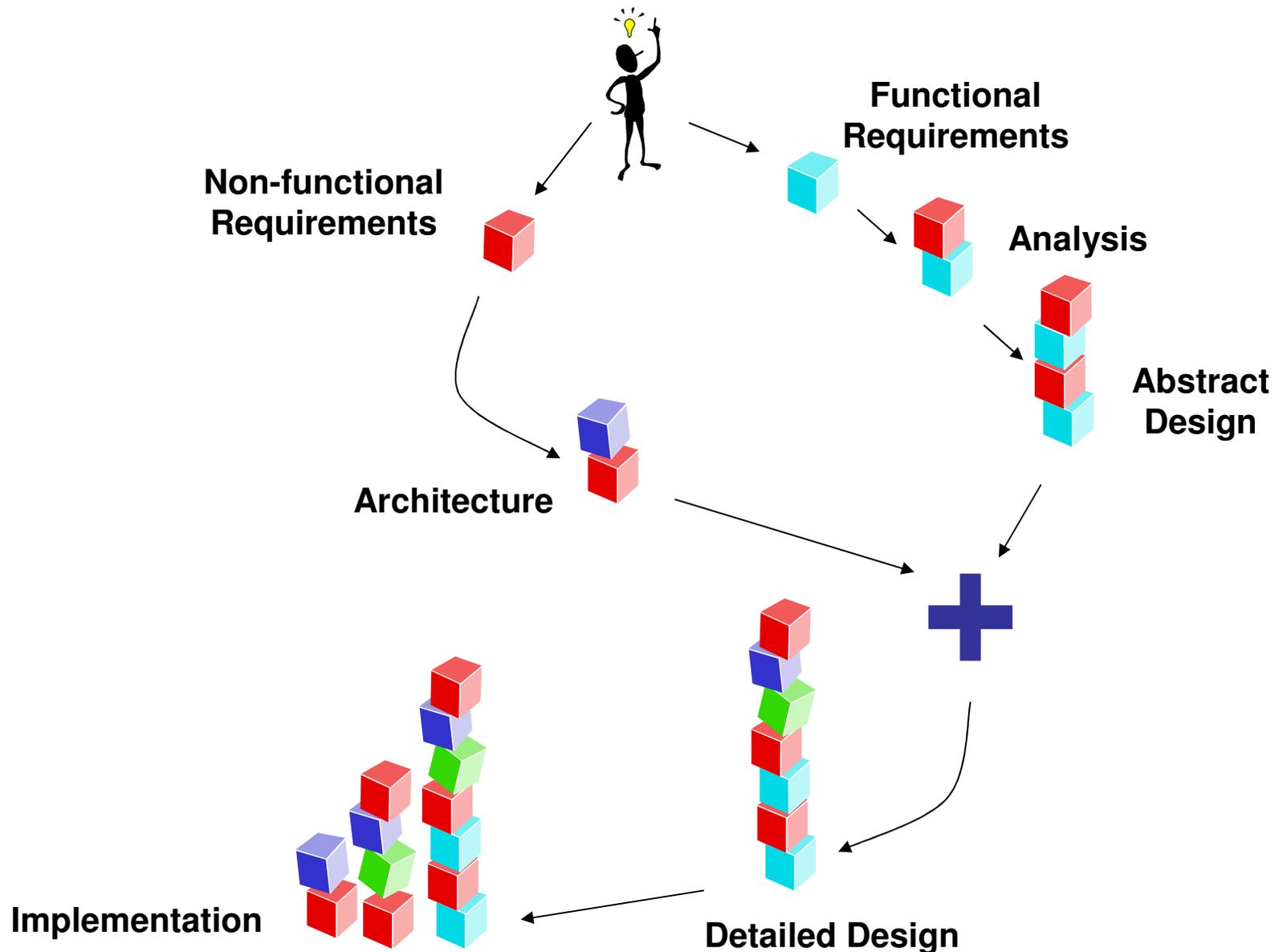
October 16-19, 2006
Burlingame, CA



Agenda

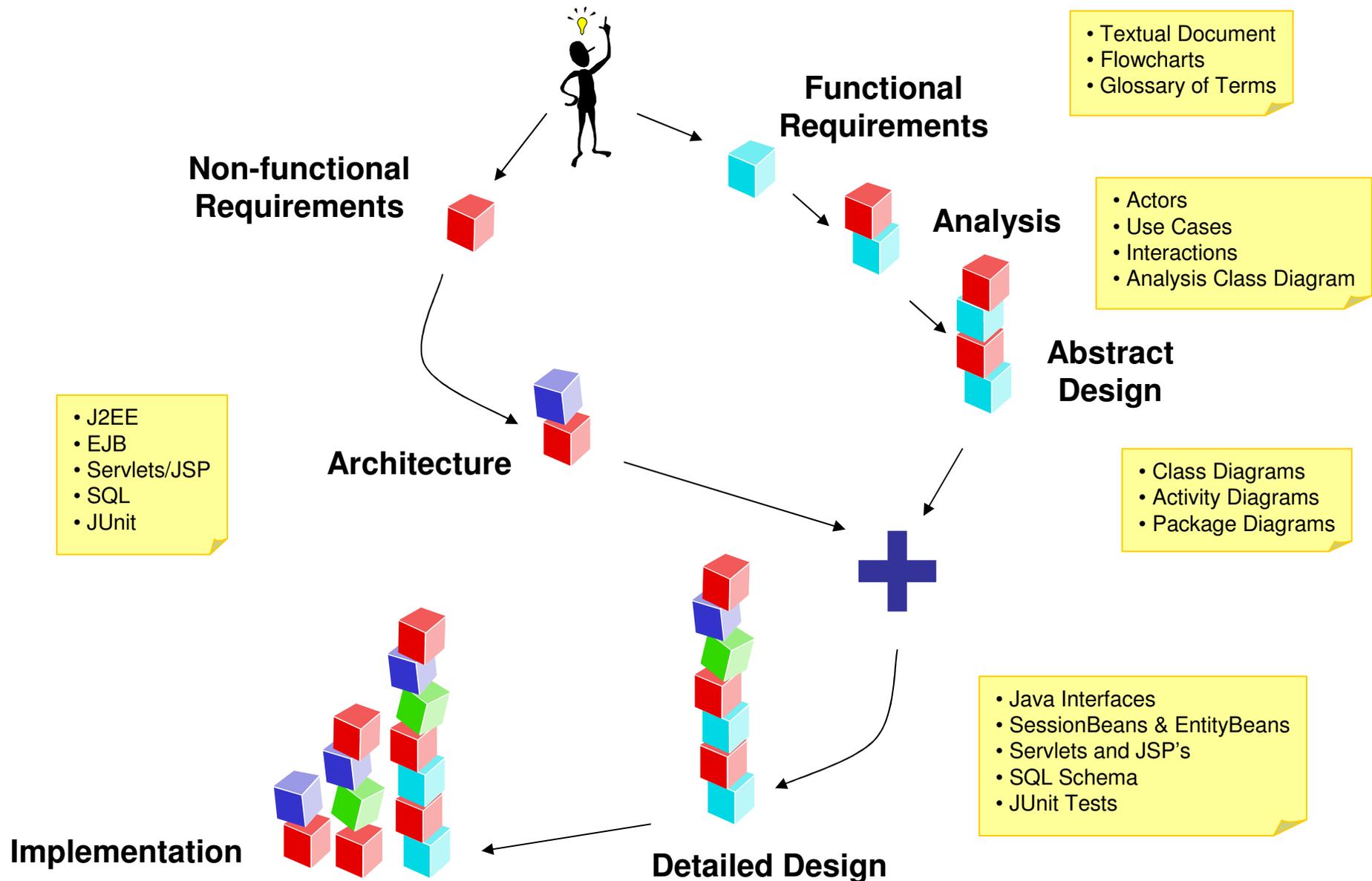
- Object-Oriented Development
- SOA Development
- SOA with Model-Driven Development
- SOA Patterns & Techniques
- Summary
- Q&A

Object-Oriented Development



Object-Oriented Development

Common Scenario



Object-Oriented Development

Example

Problem Statement:

A practice management system is needed to schedule and track patient visits to a doctor's office. The patient should be able to schedule an appointment, check-in at the doctor's office then check-out and pay the visit fee. A nurse needs to capture the patient vitals and record his/her chief complaint. The doctor needs to view the patient's vitals and medical history, examine the patient and record the diagnosis and prescribed treatment.

Object-Oriented Development

Use Case Analysis

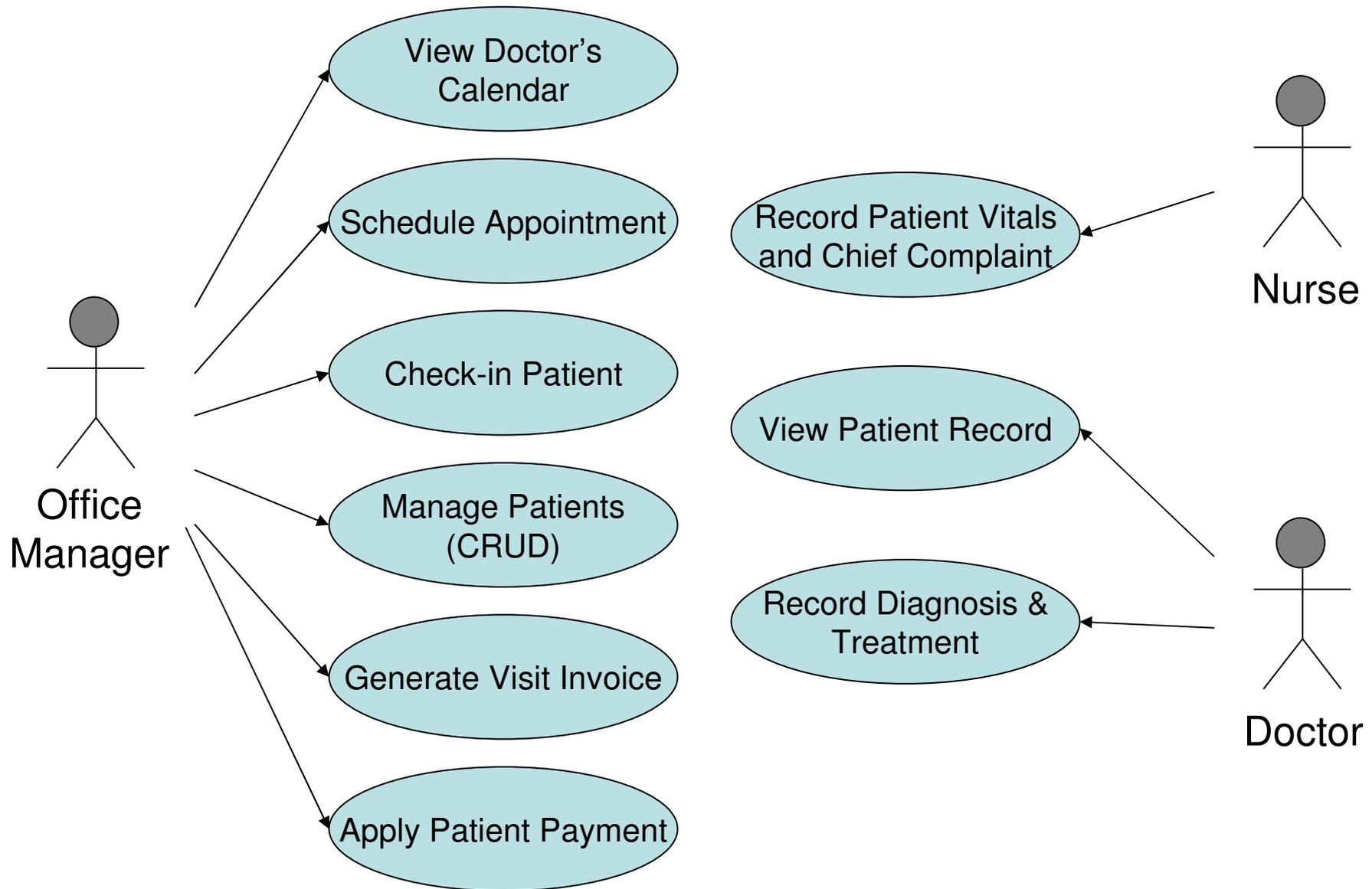
Business Use-Case:

Manage Patient Visits

- A patient calls the doctor office and asks for an appointment
- The office manager reviews the doctor's calendar and schedules an appointment for the patient
- The patient visits the doctor's office, signs in with the office manager, then fills in a patient intake form
- The office manager enters patient intake form into the system and queues the patient for the nurse
- The nurse picks the next patient in the queue and captures his/her vitals and records the chief complaint.
- The doctor visits with the patient, reviews the patient record then establishes the diagnosis and treatment.
- The patient checks out with the office manager and pays the visit fee

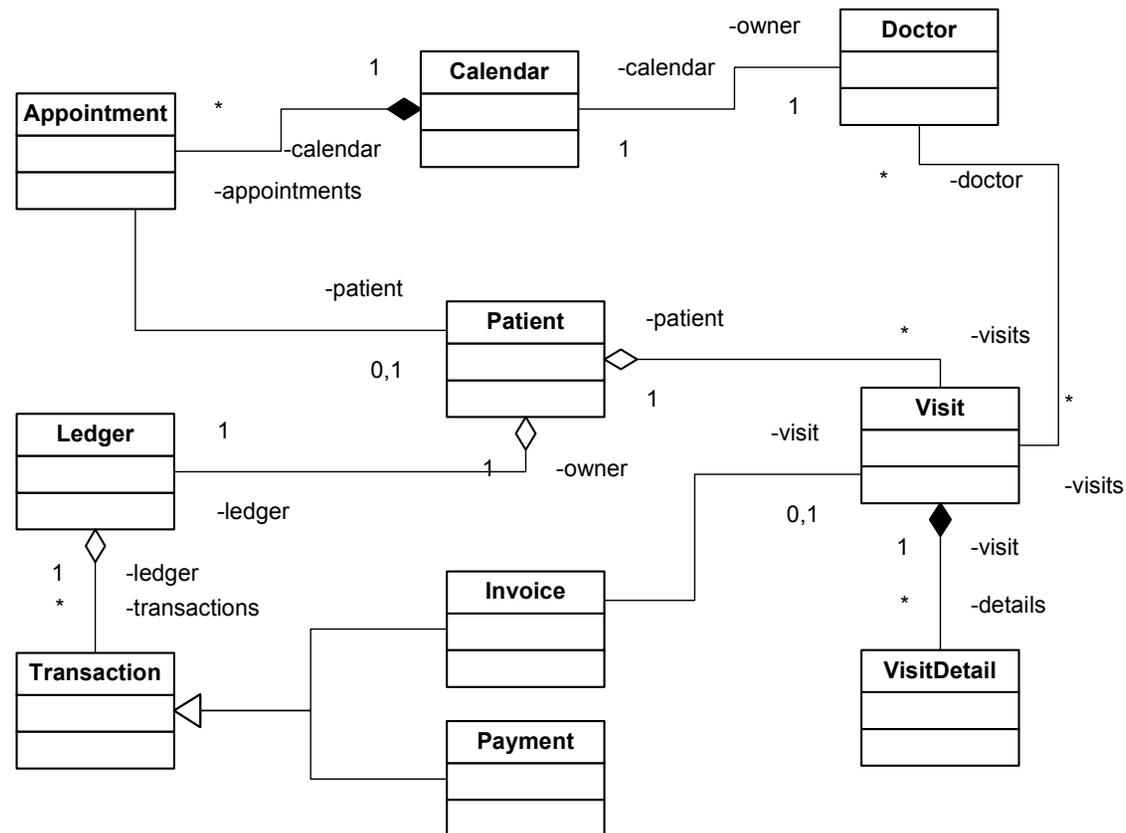
Object-Oriented Development

Use Case Model



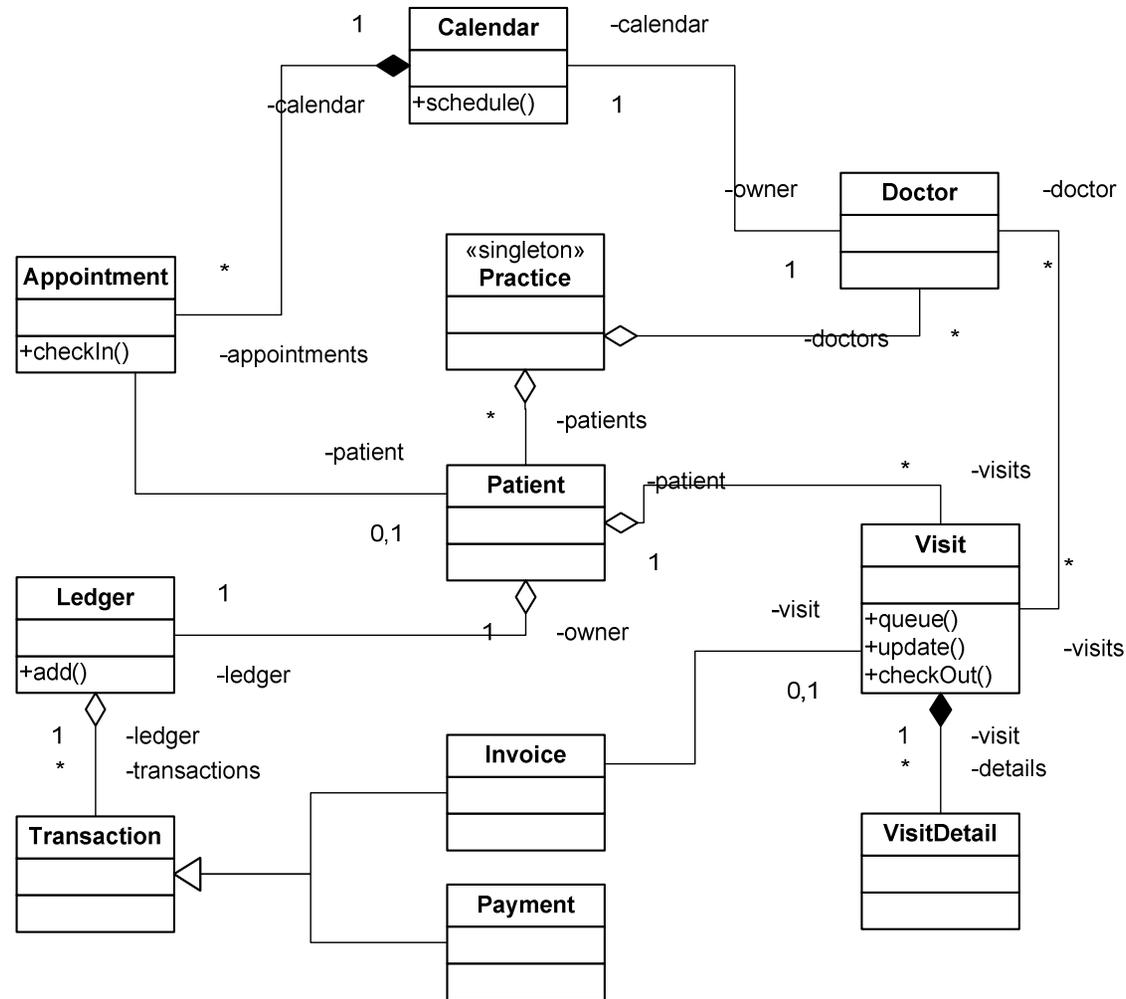
Object-Oriented Development

Analysis Class Diagram



Object-Oriented Development

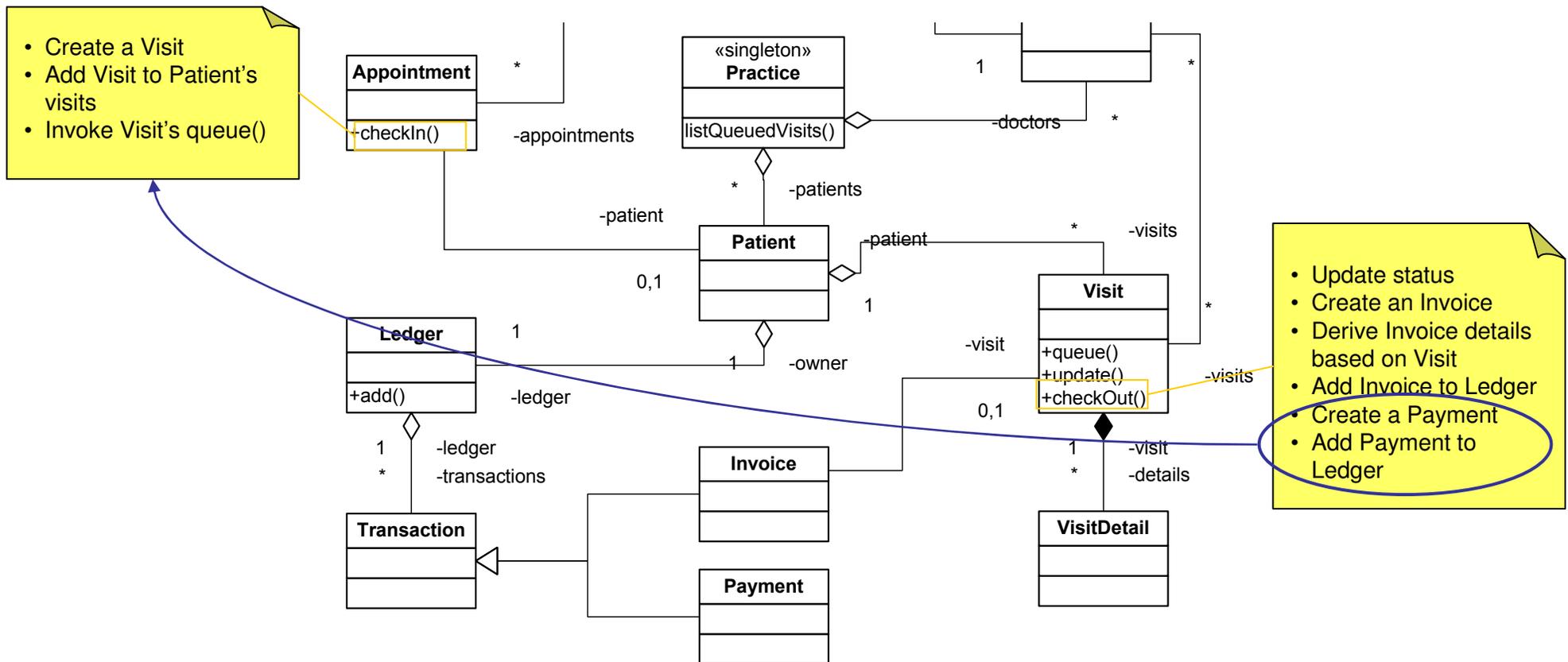
Abstract-Design Class Diagram



Object-Oriented Development

Requirement Change

- Different doctors setup their practices differently:
 - Collect payment on patient check-in

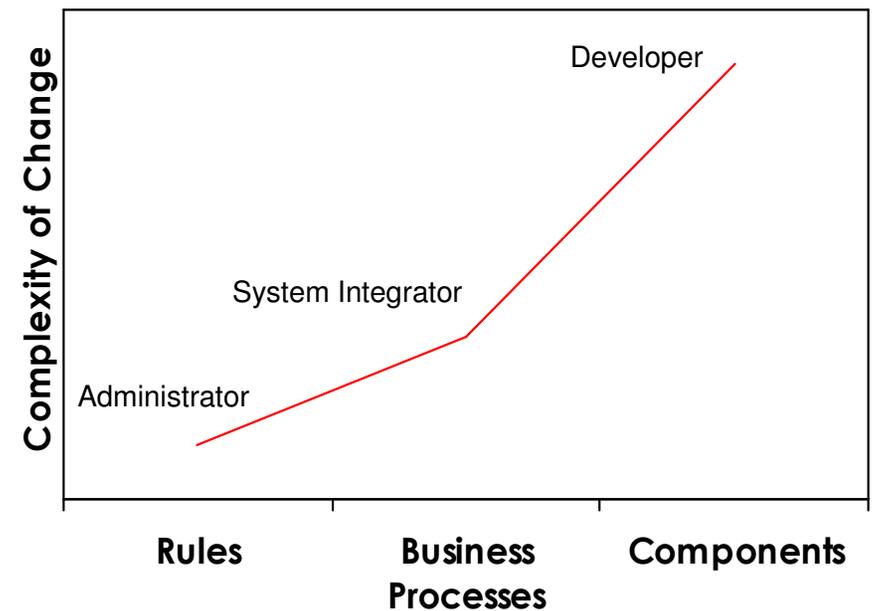
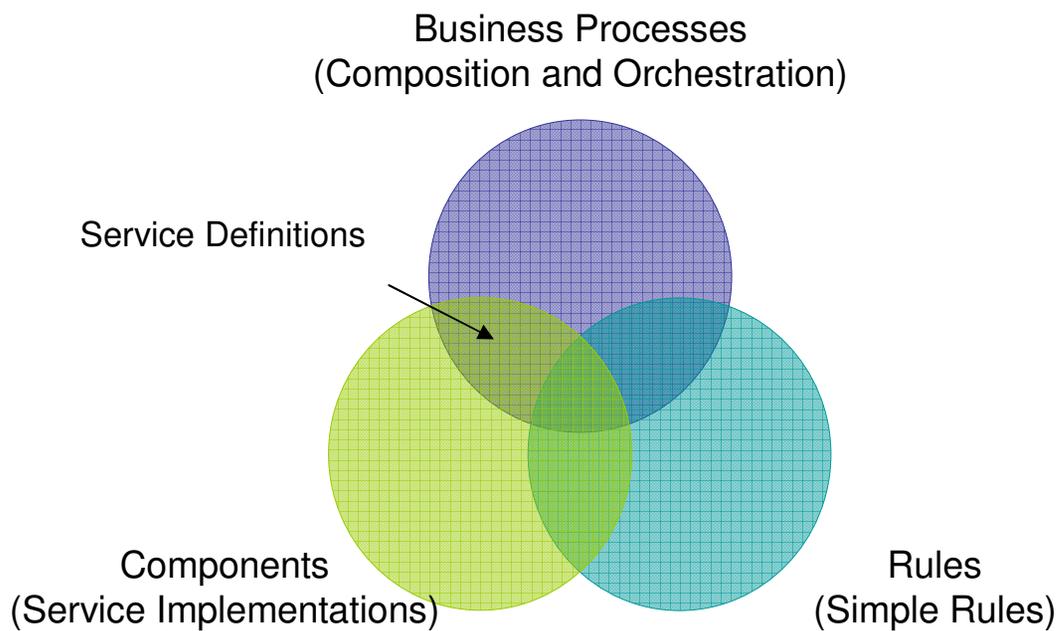


checkIn() and checkOut() has Process Logic

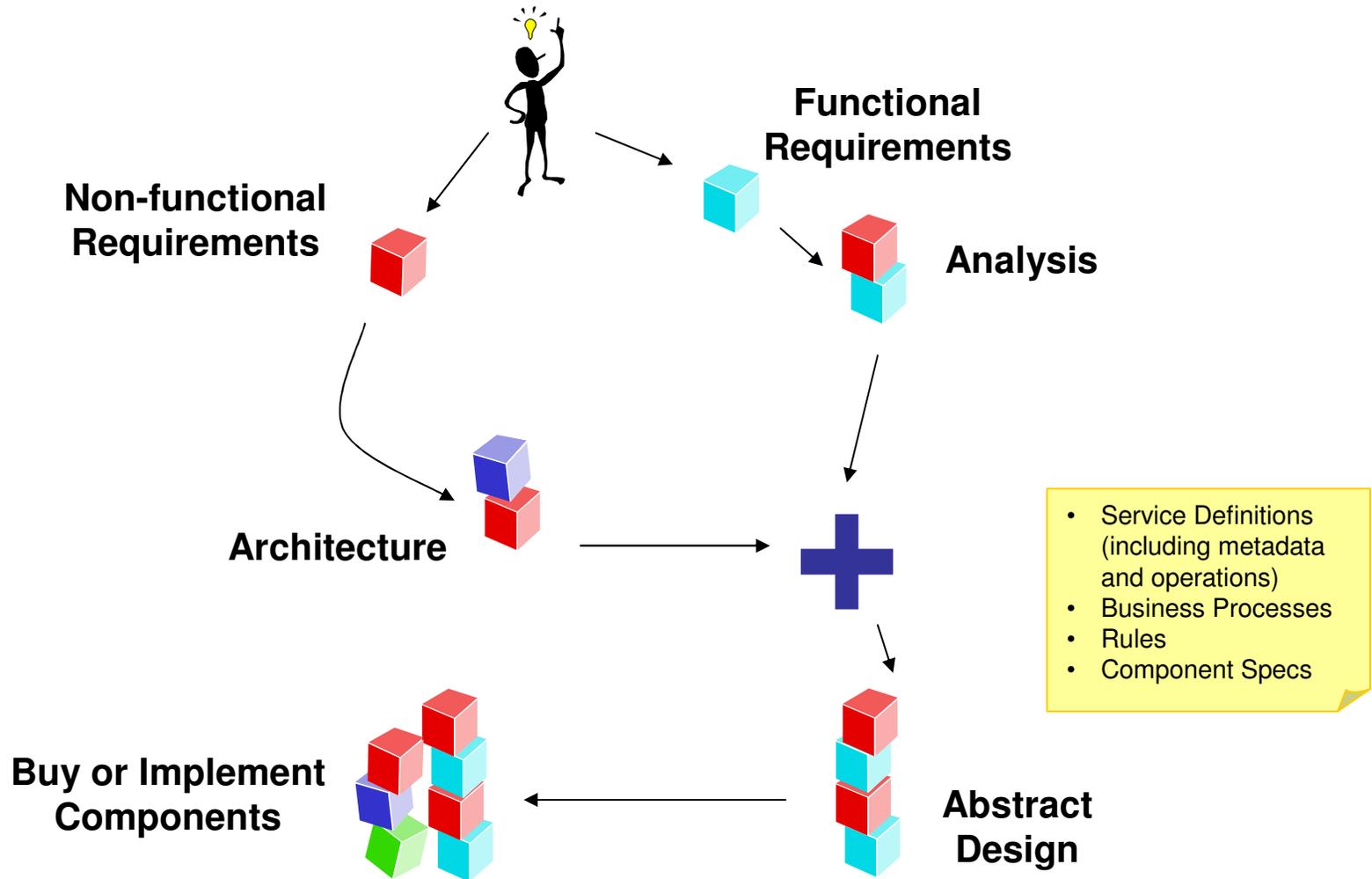
SOA Development

- Principals
 - Agility
 - Business Driven
- Approach
 - Service Orientation
 - Service Encapsulation, Autonomy and Reuse
 - Service Composition
 - Service Configurability
 - Service Loose Coupling

The Architecture in SOA

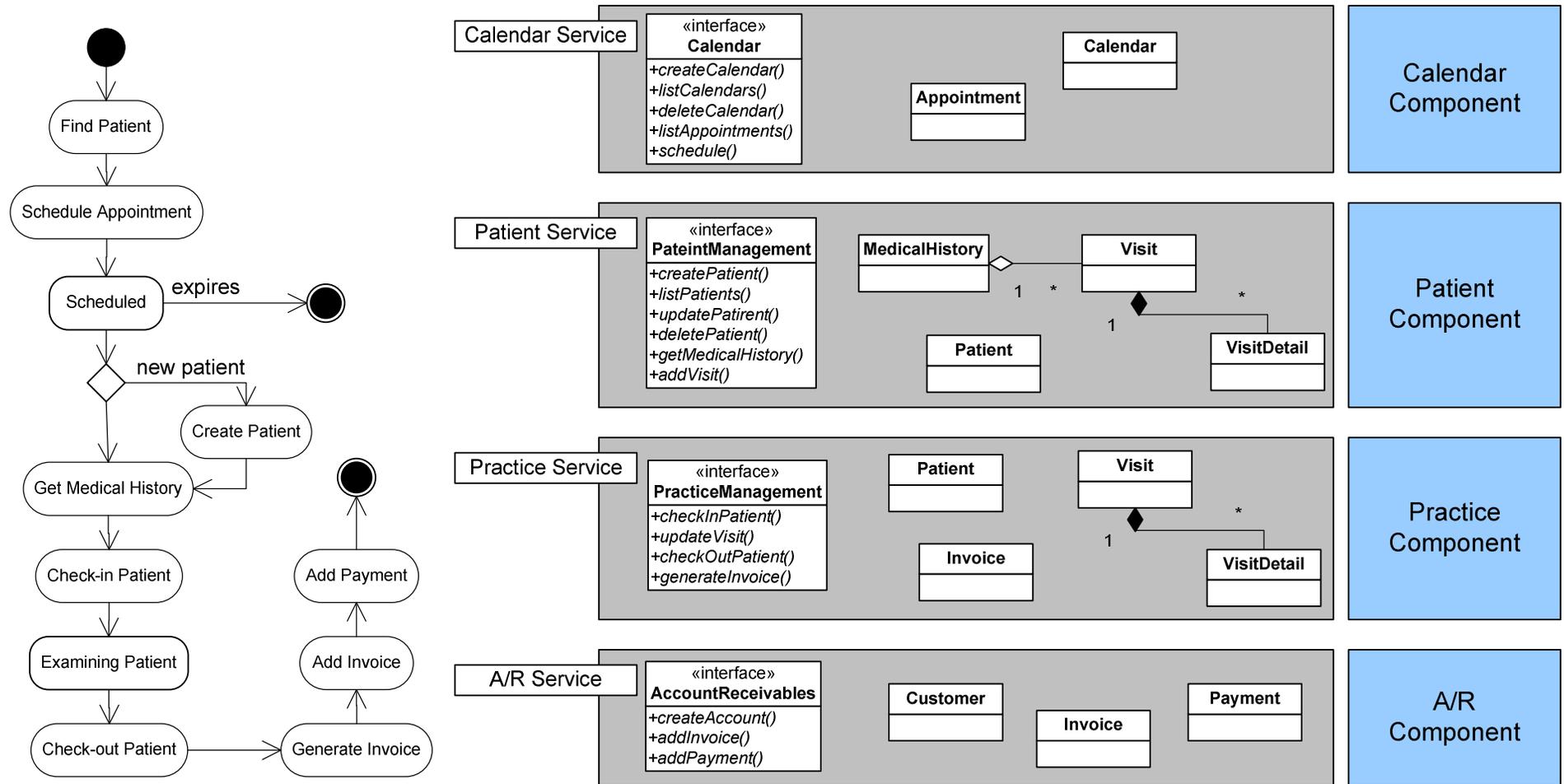


SOA Development



SOA Development

Abstract Design

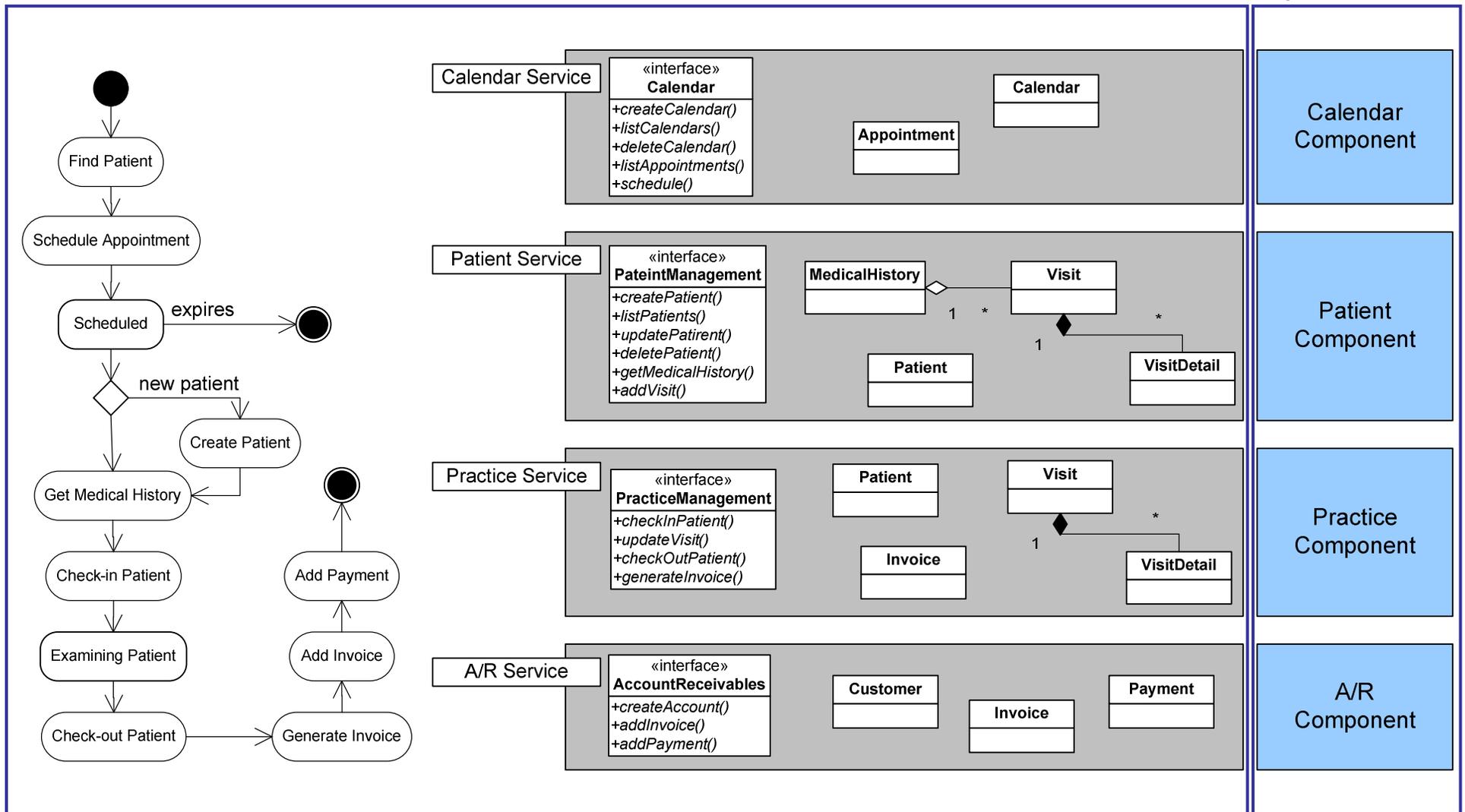


SOA Development

Abstract Design

Service Composition & Orchestration

Service Implementation



SOA Development

Implementation (Today)

- Service Implementation
 - Traditional server-side development
- Service Composition & Orchestration
 - Business process and workflow modeling
 - XML manipulations and transformations
 - User interface development (e.g. web development)
 - Human interaction and usage flow
 - Graphics design and GUI components
 - Usability Logic (bridging the gap between coarse grain Web services and usability needs)

Mixed Development/Mixed Skills

SOA Development

Implementation with MDD?

- Service Implementation
 - Traditional server-side development
- Service Composition & Orchestration
 - Business process and workflow modeling
 - XML manipulations and transformations
 - User interface development (e.g. web development)
 - Human interaction and usage flow
 - Graphics design and GUI components
 - Usability Logic (bridging the gap between usability needs and coarse grain Web services)

MDD

MDD?

Mixed Development/Mixed Skills

SOA Development Implementation with MDD?

Visit - Microsoft Internet Explorer

Address: http://iyad/ve/xv/~ijabri/demos/~pms/ve/tools/personalization/save.v?veWhereAmI=ve_oid:101&wmi=ve_oid:101

Visit
John Dow

[Add CPTs](#) [Add Note](#) [Add Attachment](#)

Details Notes Attachments

CPT Code	Diagnosis	Actions
11603		Delete
11900		Delete
11950		Delete

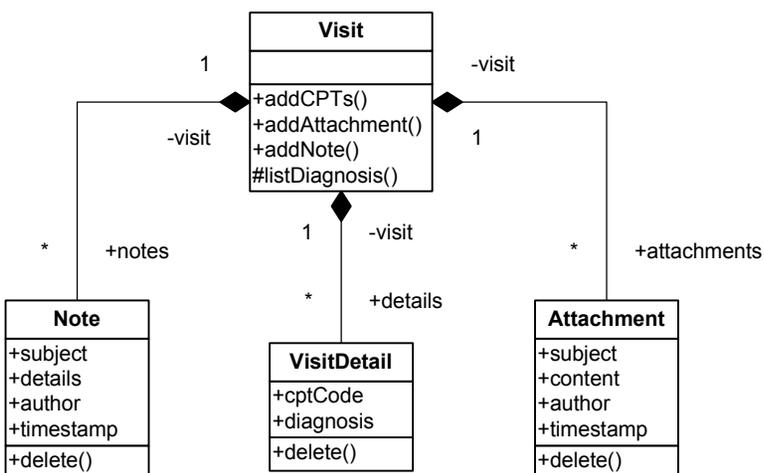
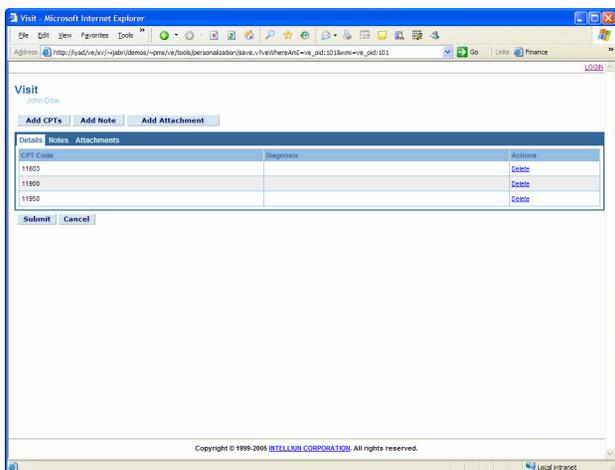
[Submit](#) [Cancel](#)

Copyright © 1999-2005 [INTELLIUN CORPORATION](#). All rights reserved.

Local intranet

SOA Development

Implementation with MDD?



```

public void addCPTs( String[] codes )
{
    for( String code : codes )
    {
        VisitDetail detail = new( VisitDetail );
        detail.cptCode = code;
        this.details.add( detail );
    }
}
  
```

```

public void addNote( String subject, Text details )
{
    Note note = new( Note );
    note.subject = subject;
    note.details = details;
    note.author = User.current().id;
    note.timestamp = Timestamp.now();
    this.notes.add( note );
}
  
```

```

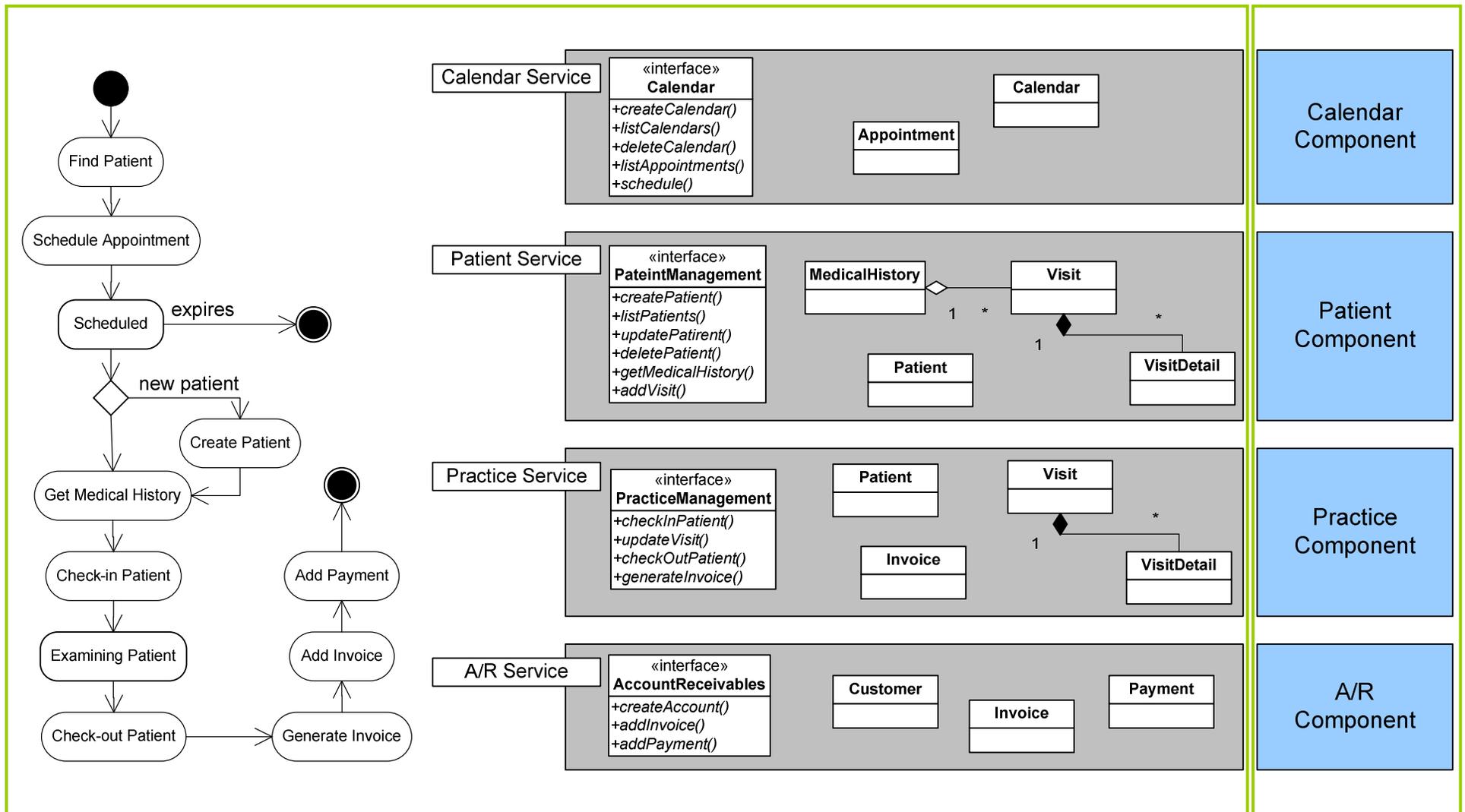
public String listDiagnosis()
{
    return details->diagnosis.distinct().sorted();
}
  
```

SOA Development

SOA and Model-Driven Development (MDD)

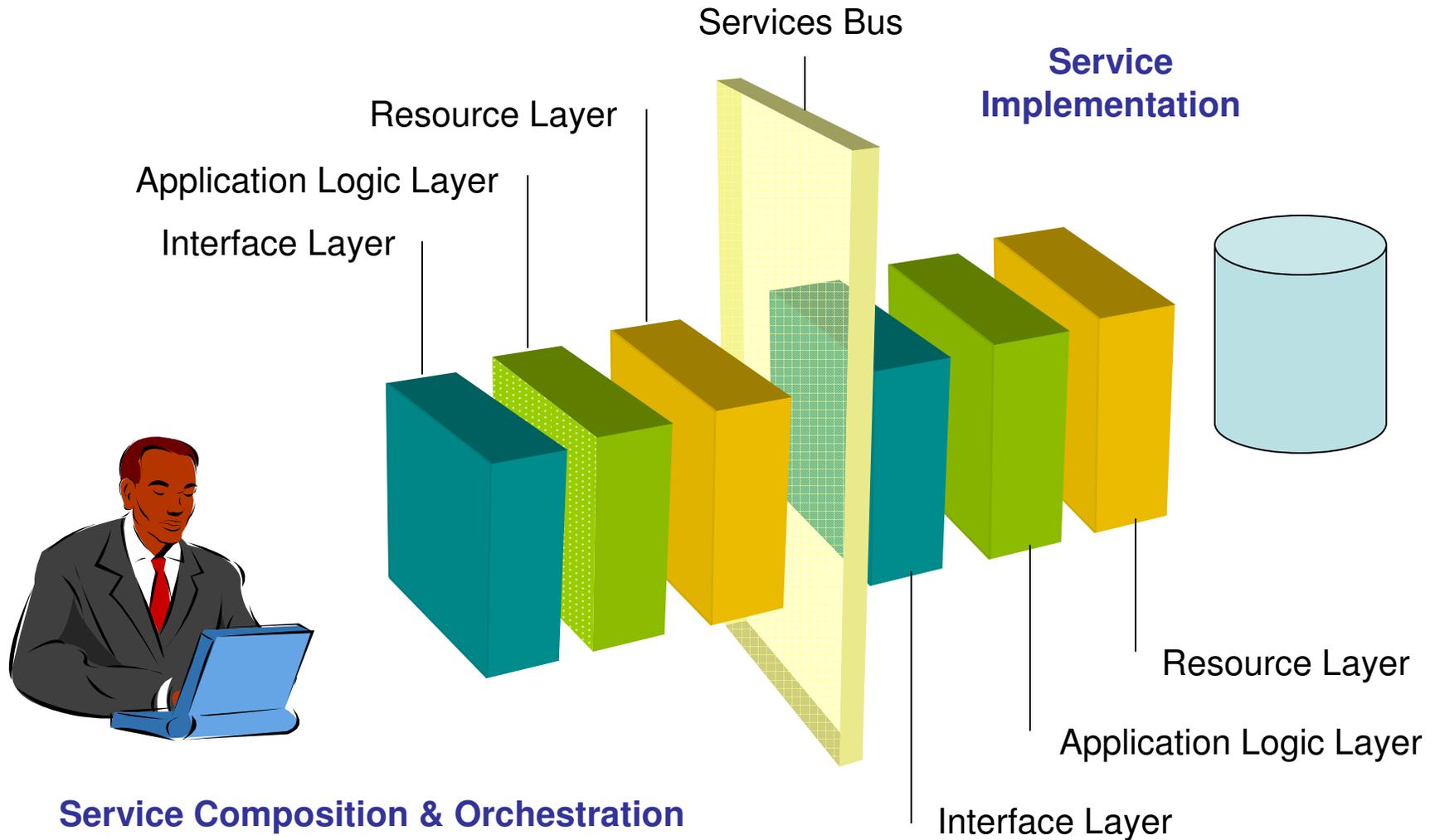
MDD

MDD



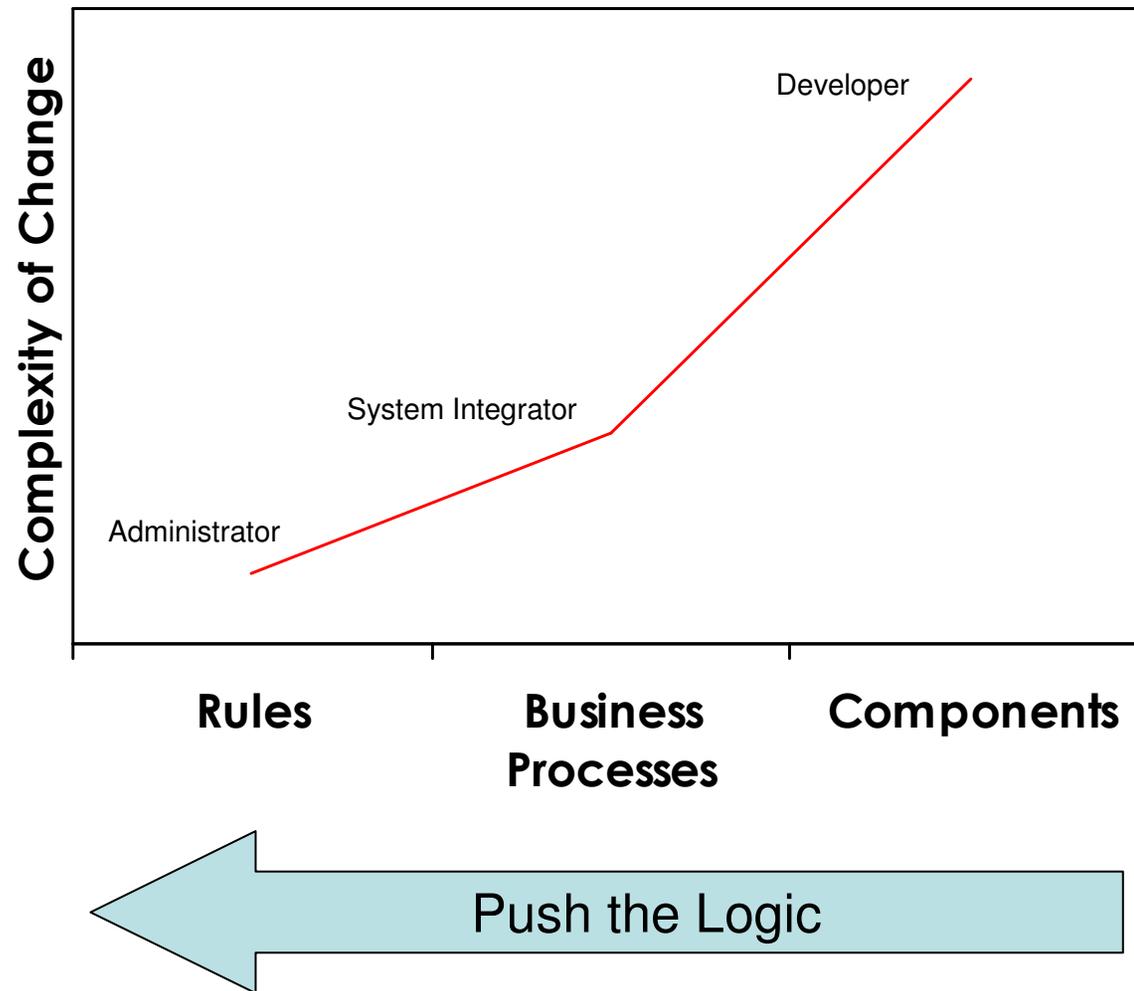
SOA Development

SOA and Model-Driven Development (MDD)



Separating Logic in SOA/BPM/MDD

The Problem



Separating Logic in SOA/BPM/MDD

Patterns & Techniques – Business Processes

- Logic Flow
 - Business Process
 - Minimize business specific flows in the Service implementations
 - Loose Service Coupling
 - Minimize invoking one service from the implementation of another
 - Usability Features
 - Minimize service operations that are geared toward fine grain end-user usability features

Separating Logic in SOA/BPM/MDD

Patterns & Techniques – Business Processes

- Workflow
 - User Tasks Management
 - Extract all task assignment and task list management requirements from Service implementations
 - Notification
 - Emails, paging, etc. should be kept in the workflow
 - Escalation
 - Giving task control to another resource when a certain condition occurs (e.g. after a set time) belong in the workflow
 - Call for Action (e.g. Approvals)
 - Call to review, approve, decline, etc. should be separated from the service implementation and managed in the workflow

Separating Logic in SOA/BPM/MDD

Patterns & Techniques – Rules

- **Parameter Rules**

- **Limit Values**

- For example if a credit score must be greater than 600, or a student GPA higher than 3.0

- **Breakpoint Values**

- For example different number of signatures (approvals) required for loans between \$0-\$4,999; \$5,000-\$9,999; \$10,000-\$99,999; and \$100,000 and above

- **Units and Scales**

- Measurements, currency, etc.

- **Algorithms and strategies**

- For example encryption, search, domain-specific algorithms, etc.

Separating Logic in SOA/BPM/MDD

Patterns & Techniques – Rules

- Complex Rules
 - Decision Rules using a Rule Engine
 - Complex rules can be represented as services

Summary

- SOA brings new ingredients to component architecture that better its chances of success
- MDD offers a unified approach to SOA
- It's important to extract and separate rules and processes from service implementations to maximize reuse and agility
- It's important to maintain separation between interface and application logic

Questions & Answers

Q&A