Developing program systems

Introduction

Vilmos Bilicki PhD
University of Szeged
Department of Software Engineering
Overview

- Requirements for fulfilling the course
- Introduction
- Goal of the course
  - Focus on the
    - Non functional requirements
    - Scabaility
- Overview of the lectures
Requirements

- Mid-term exam (20% of the total score) (during the lecture 29.03.2017)
- The final exam gives 80% of the total score
Trends

UX user experience
- GUI
- touch UI
- next-gen mobile

IT delivery
- data center
- cloud
- browser
- app stores
- social media

info & comm
- e-mail
- CMS/DMS

innovation & usage model
- enterprise
- consumer

data
- databases & BI
- big data
Evolution of the technology

Expectations
- Wireless Power
- Hybrid Cloud Computing
- HTML5
- Gamification
- Big Data
- Crowdsourcing
- Speech-to-Speech Translation
- Silicon Anode Batteries
- Natural-Language Question Answering
- Internet of Things
- Mobile Robots
- Autonomous Vehicles
- 3D Scanners
- Automatic Content Recognition
- Volumetric and Holographic Displays
- 3D Bioprinting
- Quantum Computing
- Human Augmentation

Technology Trigger
- Predictive Analytics
- Speech Recognition
- Consumer Telematics
- Idea Management
- Biometric Authentication Methods
- Consumerization
- Media Tablets
- Mobile OTA Payment

Peak of Inflated Expectations
- Cloud Computing
- Machine-to-Machine Communication Services
- Mesh Networks: Sensor
- Gesture Control

Trough of Disillusionment
- Activity Streams
- NFC Payment
- Audio Mining/Speech Analytics
- Augmented Reality
- In-Memory Database Management Systems

Slope of Enlightenment
- Complex-Event Processing
- Social Analytics
- Private Cloud Computing
- Application Stores
- Big Data

Plateau of Productivity
- 3D Printing
- BYOD

As of July 2012

Plateau will be reached in:
- less than 2 years
- 2 to 5 years
- 5 to 10 years
- more than 10 years
- obsolete

2017. 02. 08. Developing program systems
Paradigm shift: Mobile devices

Facts:

- 6.7 billion mobile subscribers, 1.3 billion smart phone
- 1.3 billion 3G, 1.1 billion WiFi based Internet access
- It is the fastest growing industry during the whole human history
- More than 2/3 of the devices is equipped with camera, more than 72% of the total users use it
- A typical end user check 40 times per day the mobile phone

- 91% of the mobile end users has the mobile phone within 1 m during the day (7x24)
- 88% of the US TV watchers use the mobile as a second screen during
- 49% of the US consumers changed his decision during shopping based on the information received from mobile
- Every third people reads news on the mobile phone
- More than 1 billion people:
  - Downloads applications
  - Plays with the mobile
  - Uses it for social network based applications
Paradigm shifts: Virtualization

- **Targets:**
  - IT Outsourcing
  - IT as service

- **Cloud:**
  - Robust
  - Scalable
  - Location neutral
  - Safe
  - Cheap
Types of virtualization

- **Type 2: Hosted architecture:**
  - It runs above the host operating system
  - Small context handling driver

- **Type 1: Bare metal**
  - The Hypervisor is installed directly on the HW

- **Paravirtualization**
  - The guest operating system knows about the virtualization
Virtualization: Capabilities

High availability

Consolidated backup

Developing program systems
Open issues:
PaaS - Gartner

- PaaS Use-Pattern Suites
  - Application Platform as a Service (aPaaS)
  - Integration Platform as a Service (iPaaS)

- Specialist PaaS Services
  - Integrated Application Servers
  - DBMS: File Systems
  - Application and B2B Integration
  - Data Integration
  - Managed File Transfer
  - Portal/User Experience
  - Business Process Management
  - Management/Governance
  - Messaging Middleware
  - Application Lifecycle Management
  - Service Interface Governance
  - Application Security

- Comprehensive Application Infrastructure Platform as a Service (PaaS)

- 2011-2015 Gartner Roadmap

- Based on "PaaS Road Map: A Continent Emerging" (G00209751)

Developing program systems
Paradigm shift: BYOD/BYOA

- The working hours and the free time are not strictly separated
- There is a productivity gain when the employee uses his own device
- The employees like their own environment
Paradigm shift: Big Data

- The technology for handling big data is given
- Data -> Treasure
Security

- Cloud/BYOD
- New point of view:
  - We should focus on protecting the data instead of focusing on infrastructure
  - Integrated approach is needed (People, Processes, Systems)
  - Deep monitoring
  - Data Mining - > Big Data
Summary

- Using virtualized environments is commonly accepted
- The cloud is there
- Mobile is the default platform
Goals

- High level overview of:
  - D&D of modern software systems
- Issues to be solved
- Java - JEE based solutions
  - What layering strategy should I apply?
  - How can we develop scalable applications?
- JavaScript/TypeScript based solutions
  - GUI vs. Backend
  - Multiplatform
Topic of the course

- Trends
- Cloud, virtualisation
- Microservices
- Web vs- Mobil
- TypeScript
- CAP
- REST/WS
- Node.JS
- NoSQL
- MongoDB
- HTML5/CSS/AJAX
- AngularJS + IONIC
- JPA + Hibernate
- EJB + CDI
Application layers

- Separating the components based on focus
  - **Presentation layer**
    - Focuses on request/response objects
    - Main focus is on rendering the UI
    - The validation is atomic taking care only single data
    - It handles the exception thrown by lower layers (e.g.: paralell modification)
  - **Persistence layer**
    - Communicates with backend storage solutions
    - It provides a query language
    - ORM capability
    - JDBC, Hibernate, iBATIS, JDO, Entity Beans, …
  - **Domain layer**
    - It defines and handles the business objects
    - It handles the relation among the business objects
    - Rich business logic
    - ORM
Service layer?

- How to position the tightly coupled business logic?
- What is the service logic?
- Who can we implement container level services?
- How can we handle the POJO based transactions?
- How can we communicate between the presentation and persistence layer?
Layers of an application

- **Service layer**
  - A gateway which provides an interface to the business logic for the outside word
  - Handles the container level services (transaction, security, …)
Application architectures

Diagram:
- Presentation
- Service
- Persistence

Presentation
- Service
- Persistence

Presentation
- Service
- Persistence

Presentation
- Service
- Persistence
Application architectures

Web architectures

- Model 1: classic Web application
- Model 2: AJAX Web application
- Model 3: client-side MV* Web application


Developing program systems
Lect 2.: TypeScript

- ES6, Typescript
- Environment (GIT, GRUNT, LINT, BABEL)
- REPL
- Runtime
- Variables, Constants, Data Types
- Objects
- Collections
- Functions
- Function expressions
- Scopes
- Closures
- IIFE
- Asynchronous programming (callback, promise)
Lect 3: Distributed systems

- Internet scale systems
- Non functional requirements
- Availability (SLA)
- States of a system, MTTF, ...
- Behind the system unavailability
- Solutions
- Redundancy
- Examples
- Scalability
- Cloud IAAS, PAAS
- CAP
- Non CA, AP, CP
- Types of consistency
- Partition types
- PACELC
### Example: Web scale system (Inktomi)

<table>
<thead>
<tr>
<th>Event</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crashes/HDD errors</td>
<td>Hourly</td>
</tr>
<tr>
<td>Database schema modifications</td>
<td>Daily</td>
</tr>
<tr>
<td>Software updates</td>
<td>Weekly/Monthly</td>
</tr>
<tr>
<td>OS updates</td>
<td>Twice a year</td>
</tr>
<tr>
<td>Power supply errors</td>
<td>&lt;10 per year</td>
</tr>
<tr>
<td>Network errors</td>
<td>&lt;10 per year</td>
</tr>
<tr>
<td>Migrating the whole infrastructure</td>
<td>2 per year</td>
</tr>
</tbody>
</table>
Scalability

- Data storage/Processing
  - Vertical (Scale up)
    - It has physical barriers (number of processors, JVM garbage collector, …)
    - Shared memory as communication medium
  - Horizontal (Scale out)
    - Distributed system
    - Theoretically unlimited due of the distributed nature it can not provide ACID
    - Paradigm change with remote accessible objects (reference vs value)
CAP Theorem

- **Consistency**: All nodes should see the same data at the same time
- **Availability**: Node failures do not prevent survivors from continuing to operate
- **Partition-tolerance**: The system continues to operate despite network partitions

A distributed system can satisfy any two of these guarantees at the same time but **not all three**
CAP Theorem
CAP Theorem

- A simple example:

**Hotel Booking**: are we double-booking the same room?
CAP Theorem

- A simple example:

**Hotel Booking**: are we double-booking the same room?

Bob

Dong
CAP Theorem

- A simple example:

Hotel Booking: are we double-booking the same room?

Developing program systems
Lect 4: Web Services/REST

- HTTP
- RMI...
- REST vs WS
- REST
- WS
- SOAP
- WSDL
- WS profilok
Rest/Web Service

WS-* - Industry Adoption

Developing program systems
REST architecture
Basic elements of SOA

- XML
- SOAP
- WSDL
- WSIL
- UDDI
Lect 5: Backend: Node.JS

- Web backend
- Little’s law, scalability
- 3rd generation backend
- Eventloop
- Threads vs Events
- NodeJS
- Event oriented programming
- Blocking/Non blocking
- Modules, variables, global objects
- Basic modules
- NPM
- REPL
- Express, Routing
- RESTFull API
Lect 6: Data handling: NoSQL

- Datatypes
- History
- Scaling methods
- Replication
- Consistency- Commit protocols
- RDBMS - sharding
- Amdahl’s law
- NoSQL
- Dynamo - Bigtable
- CAP again
- Good enough consistency
- BASE
- NoSQL types
- Document db
- Graph database
- Key-Value store
- Columnar databases
Lect 7: MongoDB

- Motivation
- Data model
- JSON, BSON
- Collections
- CRUD
- Limitations
- Scheme
- Indexing
- Aggregation, pipes, map-reduce
- Replication, sharding

```javascript
db.users.insert ( {
  name: "sue",
  age: 26,
  status: "A"
})
```
Lect 8: User Interaction

- Trends: RIA, SPA, ...
- HTML5
- AJAX
- DOM
- CSS
- Responsive web application
- Bootstrap
Lect. 8. User interaction

- Mobile vs. Web
- Mobile and Web
- Javascript based frameworks
- Java based frameworks
AJAX

Classic web application model (synchronous):
- User activity
- Time
- Server
- System processing

Ajax web application model (asynchronous):
- User activity
- Time
- Server
- Server-side processing

Developing program systems
Lect 9: Angular2 + IONIC2

- SPA
- Angular approach
- Modules
- Components
- Templates
- Metadata
- Decoarations
- Databinding
- Directives
- Services
- Injection
- Base directive set
- Change detection
- Routing
- HTTP
- Async programming
- Reactive programing
- IONIC
Lect. 10. Data Handling

- Persistence
- Object serialization API
- ORM
- Hibernate
  - Architecture
  - Collections, Polymorphic, etc.
  - Data manipulation
  - Optimisation (fetching and caching)
Lect. 11. Backend

Java Enterprise Edition
- EJB specification (3.0)
- EJB components

Injection

Scope

Transaction

EJB
- Session Bean
  - Stateless
  - Statefull
- Entity Bean
  - BMP
  - CMP
- Message Driven Bean
  - Durable
  - Non Durable
Developing program systems