JICSIT2011 / ITAIC 2011 Keynote
http://www.jicsit.org/

A Dream of Software Engineers
-- Service Orientation and Cloud Computing

Yinong Chen
Arizona State University, Tempe, Arizona, U.S.A.
Outline

➢ Introduction
  ▪ Programming Paradigms and Software Engineering
  ▪ Service Orientation vs. Object Orientation

➢ A Dream of Software Engineers: Service-Oriented Computing and Workflow-Based Software Development

➢ Cloud Computing, we could not even dream
  ▪ Cyber-Physical Device and Robot as a Service
  ▪ ASU Service Repository
Programming Paradigms

Granularity Level

1st Generation Software Engineering
- Procedural Programming
  - Assembly, Algo, FORTRAN

2nd Generation Software Engineering
- Object-Oriented Programming
  - C++, Java, C#
  - Functional Programming
    - e.g. LISP/Scheme
  - Logic Programming
    - e.g. Prolog

3rd Generation Software Engineering
- Service-Oriented Programming
  - Java, C#, Workflow
- Component-Based Programming
  - Distributed Objects
  - C++, Java, C#

Time
- 50s
- 60s
- 70s
- 80s
- 90s
- 00s
- 10s
The First Generation Software Engineering

❖ The features of the first generation
   ▪ Waterfall model
   ▪ Structured programming and design
   ▪ Structured analysis
   ▪ Compilers and interpreters advancement
   ▪ Abstract data types
   ▪ Layered architecture

❖ From machine and assembly programming to high-level programming. Significant productivity gain.

❖ Main technologies include compilers and OS, software development models, and programming languages

❖ Programming languages are the key.
The Second Generation Software Engineering

- The features of the second generation
  - Object-oriented analysis, design, and programming
  - UML (Unified Modeling Language), Agile processes
  - Software architecture patterns and design patterns
  - CMM (Capability Maturity Model) and CMMI (CMM Integration)
  - Model checking

- Modeling (such as object-oriented modeling) rather than programming is the key technology, and also classification and cataloguing (patterns) best software practices, and refinement of processes. Not just coding.

- Further productivity gain due to availability of tools, techniques, and documentation.

- Development process and techniques are the key.
The Third Generation Software Engineering

❖ The features of the third generation
  ▪ Service-oriented computing (development + execution combination)
  ▪ Cloud computing and SaaS (Software as a Service) with applications
  ▪ SaaS: development + execution + automated runtime management, including resource (sharing) and security (privacy) management. It introduces many scientific research questions into software engineering, such as data mining, control theory, and statistics.

❖ Expect very rapid software customization and deployment.
❖ Platform is the key.
Imperative Software Development

Software Development

- **Requirement analysis**
  - What is the need of users? e.g., sorting numbers

- **Specification**
  - What is the formal requirement?
    - Input $x_i$, $i = 1, \ldots, n$;
    - Output: $x_i \leq x_j$, if $i \leq j$

- **Designs**
  - How to do it?
    - Algorithms: Bubble, Merge, …

- **Implementation (coding)**
  - How to code it?
    - (Programming in C, C++, Java, C#)
  - Run it in a virtual environment, simulation in the last two pages.

- **Testing / Evaluation**
  - Put it in the real environment, e.g., the grade book application.

- **Deployment**
Component-Based Software Development

Bricks and Tiles

imperative

Component-based
Object-Oriented and Service-Oriented Software Development

1. Requirement analysis
2. Problem decomposition
3. Services development
4. Services testing
5. Service repository
6. Object-oriented development
7. Class/Object library
8. Object testing
9. Application building
10. Application builder
11. Testing
12. Deployment
Object-Oriented Software Development

Organization X: Component library

Organization Y: Component library
Service-Oriented Software Development

Application

Organization X:
Component library

Organization Y:
Component library

Organization Z:
Component library

Service broker

Auto-searchable

Registration

Standard Interface

Found
Outline

- Introduction
- A Dream of Software Engineering: Service-Oriented Computing and Workflow-Based Software Development
- Cloud Computing, we could not even dream
  - Cyber-Physical Device and Robot as a Service
  - ASU Service Repository
Clearer Tiered Architecture

- Presentation Layer (GUI)
- Application Processing Layer, with Low-Level Code
- Service and Component Layer
- Data Management Layer
- Workflow Layer with High-Level Composition
- Service and Component Layer
- WF Activities with Low-Level Code
- Data Management Layer
Using Flowchart as Code

Movie Service:
http://www.ignyte.com/webservices/ignyte.whats.showing.webservice/moviefunctions.asmx

Zip Code Service:
http://www.webserviceex.net/uszip.asmx
Flowchart and Workflow Code

An Online Ordering Process

User submits request

Submitted

Assigned

Approved

Rejected

Ordered

Completed

approve

reassign

cancel
Define the States of a Finite State Machine

User submits request

- Submitted
  - SubmittedInitialization
  - SubmittedFinalization

- Assigned
  - AssignedInitialization
  - AssignedFinalization

- Approved
- Rejected
  - Reassign
  - Drop StateActivity, EventDrivenActivity, StateInitializationActivity or StateFinalizationActivity here

- Ordered
  - Drop StateActivity, EventDrivenActivity, StateInitializationActivity or StateFinalizationActivity here

- Completed
  - Drop StateActivity, EventDrivenActivity, StateInitializationActivity or StateFinalizationActivity here
Define the Transitions between the States

User submits request

- Submitted
- Assigned
  - Approved
  - Rejected
    - Ordered
    - Completed

- Assigned
  - AssignedInitialization
  - OnAssigned
  - SubmittedFinalization

- Approved
  - ApprovedInitialization
  - OnApproved
  - ApprovedFinalization

- Rejected
  - RejectedInitialization
  - OnReassigned
  - OnCanceled
  - RejectedFinalization

- Ordered
  - OrderedInitialization
  - OnOrderReceived
  - OrderedFinalization

- Completed
Flowchart of a Mortgage Application Site

Executable Workflow
Open the “Initial Screening” Flowchart
Outline

- Introduction
- A Dream of Software Engineering: Service-Oriented Computing and Workflow-Based Software Development
  - Cloud Computing, we could not even dream
    - Cyber-Physical Device and Robot as a Service
    - ASU Service Repository
The U.S. FEDERAL CLOUD COMPUTING STRATEGY


FEDERAL CLOUD COMPUTING STRATEGY

Vivek Kundra
U.S. Chief Information Officer

FEBRUARY 8, 2011
Cloud computing headed for $20B market

Administration strategy calls for data center reduction to pay for plan

By Kathleen Hickey  •  Feb 18, 2011

The market for cloud services is about to explode in the government space if Federal CIO Vivek Kundra has his way. His recently released Federal Cloud Computing Strategy calls for about a quarter of federal IT spending, or $20 billion, to be committed to cloud systems.

Additionally, under the Cloud First program, agencies will be required to move three services to the cloud within 18 months, adopt a cloud model wherever feasible and evaluate cloud options before making investments.

An estimated $20 billion of the federal government’s $80 billion in IT spending could be used for cloud computing, Kundra said in the report. The agencies expected to spend the most on cloud technology are the Homeland Security and Treasury departments, at approximately $2.4 billion apiece, followed by the Defense, Veterans Affairs and Transportation departments. The top contractors at those agencies include companies such as Hewlett-Packard, Computer Sciences Corp., IBM, and Lockheed Martin.
Vivek Kundra’s “Cloud First” Policy


• Government agencies have been asked to consider a cloud computing option first when they planned to launch a new IT project; and they are required to identify three systems they would like to move to the cloud.

• Kundra believes Cloud Computing is the next “Internet” that has changed the world, not just computing!
Essential Characteristics of Cloud Computing

http://csrc.nist.gov/groups/SNS/cloud-computing/

- On-demand services,
- Broad network access,
- Resource pooling,
- Rapid elasticity, and
- Measured services
- Minimal management effort
Web 2.0, Web 3.0, and Cloud Computing

Static

WWW (Web 1.0)
URI, HTML, HTTP

Dynamic

SOC-based Web 2.0
UDDI, WSDL, SOAP

Semantics-based Web 3.0
RDF, RDFS, OWL

Syntax

Semantics

Cloud Computing
Components of Cloud Computing

- Software as a Service
- Platform as a Service
- Infrastructure as a Service
- X as a Service
  - Test as a Service
  - Cyber Physical Devices
  - Device as a Service
  - Robot as a Service
X as a Service: What is X?

- X is unknown
- X is a variable
- X is a dream
- X is what we could not even dream
- X is everything
  - Social networking: We can hide nothing
  - Ontology: Everything can be reasoned of
  - Virtual and reality

**Everything is under the cloud!**
Outline

- Introduction
- A Dream of Software Engineering: Service-Oriented Computing and Workflow-Based Software Development
- Cloud Computing, we could not even dream
  - Cyber-Physical Device and Robot as a Service
  - ASU Service Repository
As a Part of Cloud Computing

Software as a Service
Platform as a Service
Infrastructure as a Service
X as a Service

Cyber Physical Devices
Device as a Service
Robot as a Service

Service Interface in
HTTP, URI, REST. WSDL, SOAP, etc.
Current Efforts in Device Integration: Augmented Reality (1)

- **Pachube**
  - Data *infrastructure* for users to build their Internet of Things
  - Manage real-time data from sensors, devices, and environments

- **Wikitude** World Browser:
  - Organize and display information about users' surroundings in a mobile camera view.
  - Similar to Pachube, but focus on photos and videos
Current Efforts: Device as a Service (2)

- **Devices Profile for Web Services (DPWS)** defines implementation constraints to enable secure Web Service messaging, discovery, description, and eventing on resource-constrained devices;
- DPWS specification was initially published in 2004 and was submitted for standardization to OASIS in 2008. DPWS 1.1 was approved as OASIS Standard together with WS-Discovery 1.1 and SOAP-over-UDP 1.1 2009;
- Microsoft .Net Framework Class Library defined classes for supporting DPWS device programming

Reference: http://en.wikipedia.org/wiki/Devices_Profile_for_Web_Services
Current Efforts: Device as a Service (2)

• Device with Built-in Service Interface, for example:
• Netduino Plus: Works with .Net Micro Framework to facilitate service to device communication

http://www.amazon.com
Current Efforts: Robot as a Service (3)


- ASU Implementation of Robot as a Service
- Web service wraps the device drivers
- Web Application access the Web services

Video: http://vimeo.com/9740048
Join the Cloud and Develop RaaS

Add Partner

Add Service

Add Application

ASU

Tsinghua University

RaaS

RaaS

RaaS

RaaS

RaaS

RaaS
Outline

- Programming Paradigms and Software Engineering
- Service Orientation vs. Object Orientation
- Service-Oriented Computing and Workflow-Based Software Development
- Cloud Computing, we could not even dream
  - Cyber-Physical Device and Robot as a Service
  - ASU Service Repository
http://venus.eas.asu.edu/WSRepository/repository.html

Textbook
Third Edition

Service-Oriented Computing and Web Software Integration
From Principles to Development

Yinong Chen and Wei-Tek Tsai

ASU Repository of Web Services and Web Applications
ASU Service Repository
http://venus.eas.asu.edu/WSRepository/repository.html

- SOAP/WSDL Services
- RESTful Services
- Workflow services
- Web applications
- Robot as Service:
<table>
<thead>
<tr>
<th>Crypto service</th>
<th>ASP.Net Encryption and decryption string(string)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data caching</td>
<td>Caching disk file contents in browser</td>
</tr>
<tr>
<td>Dynamic graphics</td>
<td>Vending machine, generate graphics without using user control</td>
</tr>
<tr>
<td>Dynamic graphics</td>
<td>Vending machine, generate graphics in user control</td>
</tr>
<tr>
<td>Forms security</td>
<td>Authentication and authorization application</td>
</tr>
<tr>
<td>Image Verifier</td>
<td>Application that tests the RESTful ImageVerifer service</td>
</tr>
<tr>
<td>Image Verifier</td>
<td>Application that tests the WSDL-SOAP ImageVerifer service</td>
</tr>
<tr>
<td>Random String</td>
<td>Application that tests the RandomString service</td>
</tr>
<tr>
<td>Shopping cart</td>
<td>Enter items to catalogue, add to cart, remove from cart</td>
</tr>
<tr>
<td>XML file read write</td>
<td>Save book information into XML file in server</td>
</tr>
<tr>
<td>Service Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Crypto service in SVC</td>
<td>WCF-based WSDL-SOAP service with two operations: string Encrypt(string); and string Decrypt(string);</td>
</tr>
<tr>
<td>Image Verifier in RESTful</td>
<td>WCF RESTful service with GetImage/3Nt$@ operation</td>
</tr>
<tr>
<td>Image verifier in workflow</td>
<td>Workflow-based service</td>
</tr>
<tr>
<td>Messenger service</td>
<td>WCF service with two operations: bool SendMessage(string Username, string Message); and string[] ReceiveMessage(string UserID);</td>
</tr>
<tr>
<td>Mortgage Service in Workflow</td>
<td>Microsoft MSDN Magazine mortgage service example in workflow:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Number Guess in RESTful</td>
<td>WCF RESTful service with two operations: int secretNumber(int lower, int upper); and string checkNumber(int userName, int secretNum);</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Document type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Document type</td>
<td>Definition example</td>
</tr>
<tr>
<td></td>
<td><a href="http://venus.eas.asu.edu/WSRepository/xml/instructor.dtd">http://venus.eas.asu.edu/WSRepository/xml/instructor.dtd</a></td>
</tr>
<tr>
<td>RDF file</td>
<td>RDF schema definition file</td>
</tr>
<tr>
<td></td>
<td><a href="http://venus.eas.asu.edu/WSRepository/xml/Courses.rdf">http://venus.eas.asu.edu/WSRepository/xml/Courses.rdf</a></td>
</tr>
<tr>
<td>Robot as a Service</td>
<td>A Web application that accesses a Web service implemented in on cyber-</td>
</tr>
<tr>
<td></td>
<td>physical device, a Parallax Hex Crawler controlled by Atom</td>
</tr>
<tr>
<td>Robot in simulation</td>
<td>Simulated robot with laser sensor in a maze</td>
</tr>
<tr>
<td>Smart home</td>
<td>A smarthome using simulated cyber-physical devices</td>
</tr>
<tr>
<td></td>
<td><a href="http://venus.eas.asu.edu/WSRepository/SmartHome/Smarthome.html">http://venus.eas.asu.edu/WSRepository/SmartHome/Smarthome.html</a></td>
</tr>
<tr>
<td>XML file</td>
<td>Books stored in XML file</td>
</tr>
<tr>
<td></td>
<td><a href="http://venus.eas.asu.edu/WSRepository/xml/Courses.xml">http://venus.eas.asu.edu/WSRepository/xml/Courses.xml</a></td>
</tr>
<tr>
<td>XML schema file</td>
<td>Schema of the XML book file</td>
</tr>
<tr>
<td></td>
<td><a href="http://venus.eas.asu.edu/WSRepository/xml/Course.xsd">http://venus.eas.asu.edu/WSRepository/xml/Course.xsd</a></td>
</tr>
<tr>
<td>XML style sheet</td>
<td>Style sheet for the XML book file</td>
</tr>
<tr>
<td></td>
<td><a href="http://venus.eas.asu.edu/WSRepository/xml/Courses.xs">http://venus.eas.asu.edu/WSRepository/xml/Courses.xs</a></td>
</tr>
</tbody>
</table>
Where to Find the Information?

Yinong Chen

About 74,600 results (0.33 seconds)

Yinong Chen and Yoshiaki Kakuda, Autonomous decentralised systems in web computing environment, Int. J. Critical Computer-Based Systems, Vol. 2, No. ...
www.public.asu.edu/~ychen10/ - Cached - Similar

ASU Directory Profile: Yinong Chen
Yinong Chen received Ph.D. from the University of Karlsruhe, Germany, in ...
https://webapp4.asu.edu/directory/person/328180 - Cached - Similar

Yinong Chen - Ira A. Fulton Schools of Engineering
Yinong Chen joined ASU in 2001. From 1994 to 2000, he was a lecturer and ...
engineering.asu.edu/people/328180 - Cached

Yinong Chen - Arizona State University -