Internetware
A Software Paradigm for Internet as a Computer

Gang HUANG
Peking University
2011-03-23, Beijing
• Internet As A Computer
• Software of this NEW Computer
• Challenges Addressed by Internetware
• Internetware Research and Practice
Internet as a Computer (Internet Computer)

- Internet is evolving to a Global Ubiquitous Computer
  - Many big and hot trends in IT research and business try to study such evolution from different perspectives

Technical Trend
- Semantic Web
- Social Computing
- Service Computing
- System of Systems
- Pervasive Computing
- Grid/Cloud Computing
- Internet of Things

Big Trend
- Internet as a Computer

Business Trend
- Digital Economy
- E-government
- Internet Culture
- Social Network
- Modern Service
- Virtual World
- Smarter Planet

- Grid/Cloud computing proposes a new model of networked applications from the perspective of resource sharing and management.
- Pervasive computing discusses a new situation of networked applications from the perspective of human computer interaction.
- Service Oriented Computing focuses on a new form of software with emphasis on collaboration and dynamism from the philosophy of software as a service.

...
Cooperation On Demand (emergent)

- more and more goals are unclear, unexpected or un-deterministic before the cooperation finally happens (e.g. real-world emergency)
- cooperators may have no or little relations before and after

1. Contact CDC when suspected H1N1 infectors are detected

2. CDC acquires all customers in the same flight

3. CDC notifies all customers go to the given hotels

4. CDC sends customer info to the hotels

5. CDC notifies the hospitals go to the hotels and take case of the customers

H1N1 Control and Prevention
“Internet Computer” requires substantial improvements in software characteristics for implementing new business naturally with new technology.

- Before “Internet as a Computer”, the goal of cooperation is predefined while the partners and interactions can be fixed or not.
- Such emergent co-operations are uneasy, labor-based and un-trusted today.

Labor-based emergent cooperations can be observed in such as mashups and end user programming.
Situational: software is capable of perceiving its runtime context and scenarios

Cooperative: software can interact with others in static, dynamic and even on demand manners

Emergent: software may have un-designed behaviors or unexpected effects on its runtime instances or interactions with others

Autonomous: software is relatively independent of others; it can perform operations as it will and adapt itself when necessary

Evolvable: software is easy to add, remove and change its functionalities on-the-fly and just-in-time

Trustworthy: software should promise some kind of tradeoff among process quality, internal system quality, external system quality and usage quality.
Existing paradigms cannot well support those new characteristics of software on Internet computer, and then Internetware comes to being.

Software paradigm always evolves itself for evolving application domains and runtime environments.
3C Modes of Internetware

On demand Collaboration

Behavior vs. Organization

Micro vs. Macro

Self-organizing Community (scope/domain)

reuse-based, knowledge-driven, bottom-up, online Construction

Asset and Knowledge

direct output

macro output
Challenges to Internetware

**Programming Paradigm (what to be)**
- abstracts the elements and their relationships of a software system
- Internetware model should
- leverage legacy software and new characteristics
- Enable open collaboration between components
- Adapt itself for emergent contexts and situations

**Engineering approach (how to produce)**
- Systematically control the software development, deployment, maintenance and evolution
- Internetware engineering should
  - Identify the self-organized communities and domains or facilitate the self-organizations
  - Satisfy requirements via collaborating existing and/or emergent components
  - Involve all stakeholders, especially the actual end users

**Programming Language & System (how to be)**
- incarnates the elements and their relationships of a software model
- Internetware middleware should
- Provide a container for instantiating and operating Internetware components
- Provide collaboration mechanisms.
- Equip legacy software systems with Internetware characteristics
- Enable context-awareness and reflection

**Quality assurance (how to be good enough)**
- Focal points of software quality change from system-centric to usage-centric
- Internetware quality assurance should
- Define quantitative and qualitative evaluation framework for quality
- Assure the quality via engineering approach at development time as well as middleware at runtime
Internetware Research in China

• “Theory and Methodology of Agent-based Middleware on Internet Platform”
  – The first national basic research program (973) project on software
  – From 2002~2008; > 80 faculty; 17 post-doc; 88 Ph.D; 364 Master

• “High Confidence of Internetware”
  – IBM joined as the first foreign company in 973 program (GTO 2010)
  – From 2009~2013; > 100 faculty
Internetware R&D Outputs

Internetware Engineering Approach

Internetware Middleware Framework

Internetware Software Model

Internetware Platforms and tools

Internetware Demos and Case Studies

Characteristics

Cooperative, Situational, Emergent, Evolvable, Autonomous, Trustworthy

On demand Construction/cooperation

Bottom-up Knowledge-Driven engineering

Architecture centric

Self-organizing Domain/community

Middleware for Runtime

Agent-like Entity
Internetware Testbed:

Synergy of computers, individuals and things on Internet

Synergy of Computers:
Grid Computing, Service Computing, Cloud Computing, System of Systems ...

Synergy of Computers and Individuals:
Web 2.0, Virtual World, Social Network, Web Science, Social Computer ...

Synergy of Computers and Things:
Pervasive/Ubiquitous Computing, Internet of Things, Cyber-Physical Systems, Smart Planet ...

SM@RT (Supporting Models at Runtime)
Towards Model-based Programming Languages, Systems and Applications
Licensing challenges multi-tenancy (then new architecture)
Vendor-centric challenges delivery models (then cloud)
Customer-friendly challenges ... (then ...) ?

• Software as a service (SaaS, typically pronounced [sæs]) is software that is deployed over the internet. With SaaS, a provider licenses an application to customers as a service on demand, through a subscription or a “pay-as-you-go” model. SaaS is also called “software on demand.” SaaS vendors develop, host, and operate software for customer use. Rather than install software on site, customers access the application over the Internet. The SaaS vendor may run all or part of the application on their hardware, or may download executable code to client machines as needed—disabling it when the customer contract expires. The software can be licensed for a single user, or group of users.
When Rich Client meets SaaS ...

Rather than install software on site, customers access the application over the Internet. The SaaS vendor may run all or part of the application on their hardware, or may download executable code to client machines as needed—disabling it when the customer contract expires.
Wide Spectrum of SaaS Rich Client

- SaaS Rich Client is that supports service discovery, subscription, composition and even evolution in web browsers with rich user experiences

<table>
<thead>
<tr>
<th>Service Subscription</th>
<th>Service Discovery &amp; Subscription</th>
<th>Service Composition</th>
<th>Service Discovery, Subscription &amp; Composition</th>
</tr>
</thead>
</table>

Web APIs: Google, Amazon, Flickr, Twitter
SaaS: iGoogle, eXo, Zoho
Mashup Apps: WeatherBonk, HouseMap, Google Wave
Mashup Environment & Apps: iMashup, Palm WebOS

How to synthesize services with rich user experiences

How to handle diverse client devices and platforms

Simple, Easy-to-use

Powerful, Professional
Synthesize Services with Rich User Experiences

Rich Client Component

On-the-fly Qualification of Component & Composition

Rich Client Component Composition

new client-side programming model is needed
Handle Diverse Client Devices and Platforms

Adapt to heterogeneous local storage supports

<table>
<thead>
<tr>
<th></th>
<th>IE</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>IE (Mobile)</th>
<th>Safari (iPhone)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE userData</td>
<td>5.5+</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Flash LSO</td>
<td>Plug-in</td>
<td>Plug-in</td>
<td>Plug-in</td>
<td>Plug-in</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>HTML5 Local</td>
<td>8.0+</td>
<td>3.5+</td>
<td>3.0+</td>
<td>3.1+</td>
<td>N/A</td>
<td>3.1+ (OS 2+)</td>
</tr>
<tr>
<td>HTML5 DB</td>
<td>N/A</td>
<td>N/A</td>
<td>3.0+</td>
<td>3.1+</td>
<td>N/A</td>
<td>3.1+ (OS 2+)</td>
</tr>
<tr>
<td>Google Gears</td>
<td>6.0+, Plug-in</td>
<td>1.5+, Plug-in</td>
<td>1.0+</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Select the best-of-the-breed local storage

Model Data from Local Storage

IE 8

Firefox 3.6

HTML5 Performance in Safari Desktop & Mobile

Create, Read, Update, Delete Performances
Handle Diverse Client Devices and Platforms

Model Data from Remote Service

Transmit data between Rich Client and (remote) Business Process

Working days report in intern payroll

Data Cache of Remote Service (online/offline)

- No Cache
- Fixed Cache
- Adaptive Cache (wrt. hit ratio and timeout)
**Handle Diverse Client Devices and Platforms**

**Implementation**

- View
- Model
- Controller

**Controller adapt to Client Devices**

- All at the client device
- Code back to server from client
- Code at both server and client

*Language dynamism (Javascript) is challenging*

*Chrome V8 at server can drastically speed up IE, Firefox and other browsers at client (even on PC)*
SM@RT Clients:
An MVC-Oriented Component-Based Programming Framework (in Javascript)

On-the-fly Composition:
@MVC Component: create instances at the time when a component is selected
@MVC Composition: propagate messages at the time when a connection is specified
@Quality Assurance: design at runtime; model checking at runtime

Adaptation for Heterogeneity:
@Model:
- local/remote/mobile storage;
- online/offline (consistency)
@Controller:
- phone/tablet/PC computing power
@View: different screen size
SM@RT Clouds:
A Model Driven Framework for Autonomic Management

- **Synchronizer**: causally connecting MOF-compliant models to runtime systems so that changes on one side will cause corresponding changes on the other side just-in-time
- **Synthesizer**: integrating, orchestrating and applying multiple user-defined model analysis methods on runtime models

**MOF and QVT compliant, Non-intrusive, Automated**

Trivial, scattered and tangled management-related data is accessible via system APIs and files (scripts & configurations)

- Gather data just-in-time
- Organize to MOF-compliant models
- Feed is-be models to analysis & decision tools
- Change data according to to-be models
- Change data
- Query, transform & synthesize
- Visualize
- Effectuate & integrate analysis results
SM@RT Sensors:
An Object/Sensor Mapping Framework in Java *(just like ORM)*

Transform (in a model-driven way) sensor-specific data and functions to application-specific objects

```
public void OnRecvTag(object sender, RecvTagEventArgs e) {
    if (controlPane.IsStopRecvTagID) return;

    lock (lstRecvData)
    {
        lstRecvData.Add(e);
    } //lock

    ShowTagIDHandle d = new ShowTagIDHandle(ShowTagID);
    this.BeginInvoke(d);
}
```

```
public class HouseImpl extends EObjectImpl implements House {
    @Override
    protected static final int ID_EDEFAULT = 0;

    @Override
    protected int id = ID_EDEFAULT;

    @Override
    protected static final String TEMPERATURE_EDEFAULT = null;

    @Override
    protected String temperature = TEMPERATURE_EDEFAULT;
```

<table>
<thead>
<tr>
<th>Sensor ID</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS846</td>
<td>10131</td>
</tr>
<tr>
<td>SS419</td>
<td>10131</td>
</tr>
<tr>
<td>SS070</td>
<td>10131</td>
</tr>
</tbody>
</table>
Internetware

A Software Paradigm for Internet as a Computer

Internetware Testbed: Demonstrate the Synergy of computers, individuals and things on Internet