Design by Units
Abstractions for Human and Compute Resources for Elastic Systems

Schahram Dustdar
Distributed Systems Group
TU Wien
Structure

1. Understanding advanced Services Systems

2. Finding Abstractions/Programming model for Human-, Compute-, and Storage Resources -> Design by Units
Autonomic Nervous System
More than 7 billion devices and sensors exist for M2M application.

Galaxy and Gbots forms enterprise city centric cloud architecture to enable smart services ecosystem and collaboration opportunities.

Managed services
- Portfolio management
- Event management
- Analytics

Provisioning
- Services
  - SIM profile configuration
  - Network configuration

Controls
- Activation
- Deactivation
- Privacy
- Security

Transaction Mgmt.
- Visibility
- Billing
- Reporting

Ubiquitous Managed Services Solution Across Business Verticals

Numerous Forms Of Smart Services..

Managed City Governance Service Oriented Architecture
ICT enabled Security Services

Command Control Center

- SMEs
- Dashboards
- User interfaces
- Reports
- Carbon footprint measurement
- Benchmarking
- Remote monitoring
- Engineers

- Hospitals
- Shopping malls
- Factories
- Schools
- Airports

© Copyright 2010 System Control Systems All Rights Reserved
ICT enabled Telematics

Command Control Center

Vehicle tracking system

Logistics Management
ICT enabled services for food storage and delivery

Command Control Center

- SMEs
- Dashboards
- User interfaces
- Reports
- Carbon footprint measurement
- Benchmarking
- Remote monitoring
- Engineers

Cold storage system

Freezer rooms

Food display cabinets
ICT enabled services for health care

Command Control Center

Hospital operations management

Hospital security systems

Hospital equipments monitoring
ICT enabled smart education systems

Command Control Center

- SMEs
- Dashboards
- User interfaces
- Reports
- Carbon footprint measurement
- Benchmarking
- Remote monitoring
- Engineers

Smart classrooms

Smart Universities

Campus infrastructure
HVAC (Heating, Ventilation, Air Conditioning) Ecosystem
Water Ecosystem
Air Ecosystem

Outside air intake

Cooling water

Ducts

Diffuser

Terminal Unit

Air handler

Energy management system

Boiler

Chillers

© Copyright 2010 Pacific Control Systems. All Rights Reserved.
Command Control Center for Managed Services
Advanced software to analyze data; make it actionable.
Big Data – Your Data in action

IP Backbone

Web portal
+ Integrated billing & payment
+ Internet/intranet login access
+ Personalization
+ Trouble ticketing
+ Service tracking
+ SLA management

IP services
+ Digital signage
+ Multimedia / IP TV
+ Visitor management system
  Hand-held Wi-Fi devices, barcode, card printing, admission/exit control, visitor & vehicle screening

Intelligent Building Management System (IBMS)

Master clock
Public addressing system

Traffic management system
RFID, user-identity based services

Computerized controls (for gates, flower-beds, ramps)

Fire detection alarm system
Access control system
Irrigation control system
Intruder detection system
CCTV security system
Power monitoring system
Digital audio-video management

HVAC cooling system
Lighting control system
Part 2 – Finding Abstractions for Human and Compute Resources for Elastic Systems
## Computing Models

<table>
<thead>
<tr>
<th>Machine-based Computing</th>
<th>Human-based Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processing Unit</strong></td>
<td></td>
</tr>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td><strong>Architecture</strong></td>
<td></td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td><strong>Grid</strong></td>
<td></td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td><strong>Comm.</strong></td>
<td></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
</tbody>
</table>

From *Design by Contract*...

Extends conventional software component definitions with

- pre- and post-conditions
- invariants

these specifications are referred to as *contracts*. 
To Design by Units

• Extends software service definitions with a resource model to better address human and compute resource requirements in system design.

• *Units* are abstractions to model diverse resources in the cloud that are required to operate a system and to guarantee nontrivial system requirements, such as *elasticity*. 
Elasticity of Resources
Elasticity in computing – a broad view

1. Elastic demands from consumers

2. Multiple outputs with different price and quality (output elasticity)

3. Elastic data inputs, e.g., deal with opportunistic data

4. Elastic pricing and quality models associated resources
Elasticity

Resource elasticity
Software / human-based computing elements, multiple clouds

Non-functional parameters
performance, quality of data, service availability, human trust

Pricing/Rewarding /Incentive elasticity
Unit Types - Examples

**Social Resources**
- (Teams/Masses of) People
- Using majority votes
- Units as crowds’ structuring mechanism

**Middleware Platform Resources**
- Units as application containers
- Units as user workspaces

**Legal Framework Resources**
- Units might model compliance policies
- Units might model contracted license keys

**Financial resources**
- Units to represent funding sources within an organization
- Units to represent funding sources from external sources
**Impact**

- Each unit provides a resource to the application (functional)
- E.g. impact on availability of the call center, storage units on data management, etc.

**Measurability**

- Measure the impact (e.g. on metric scale) for each unit.
- Measure call center availability via average response times

**Cost**

- Units consumption costs
- Unit usage costs must be measurable
- Preemptively reason over the benefits and value of unit configurations

**Dynamicity**

- Acquire and release new units in a timely, on-demand fashion
- Model runtime reactions to changes in the application’s environment
- Resource management might be automated or involve human interactions
Design of Units

• Units provide a virtual representation of resources that can be monitored, reserved, bought, sold, and used

• They have current state
  – cost, utilization, quality, correctness, efficiency, location ...

• And they provide operations
  – store in memory unit, compute in computational unit, send in network unit, ...
Origins Model (current work)

- Origins provide a programming model for large-scale heterogeneous systems

- An origin
  - represents a single property of a unit
  - which can be composed with other origins to provide higher-level information about the state of a system
Origins Model (current work)

• Thus, we can use origins to
  – build a system
  – that can reason about its current state and adapt to sudden changes
  – by gathering information about its units

• This in turn allows us to guarantee non-trivial requirements in the system
  – e.g. by adapting the system to use more or less units (i.e., provide elasticity)
Units & Origins

System

* is supported by *

Units

* is described by *

Origin Origin Origin Origin Origin

e.g. free space, location, availability, cost
// composition
origin.create("memory/utilized") {
  val memoryUnits: Array[Unit] =
    system.unit("memory")
  val available: Long =
    memoryUnits map {x => x.get("size")} sum
  val used: Long =
    memoryUnits map {x => x.get("use")} sum

  return used / available.toDouble
}

// adaptation
origin.access("memory/utilized" when {_ > .95}) {
  system.unit += market.buy("unit/memory")
}
Some papers...


Thanks for your attention

Prof. Dr. Schahram Dustdar

Distributed Systems Group
TU Wien