SEMANTECALLY RICH, POLICY BASED FRAMEWORK TO AUTOMATE LIFECYCLE OF CLOUD BASED SERVICES

Karuna Pande Joshi
contact: karuna.joshi@umbc.edu

CSEE Department, University of Maryland, Baltimore County
Advisors: Dr. Yelena Yesha, Dr. Timothy Finin

Submitted towards PhD Dissertation, Nov 2012
© Karuna P Joshi, 2012
Agenda

1. Problem Motivation
2. Research approach
3. Key contributions
   1. Services lifecycle, ontology
   2. Service Negotiation
   3. Cloud Broker Applications
   4. Service Quality framework
Cloud Computing: The present

- Shared Cloud Resources
- “Virtualizing” resources
  - Dynamic provisioning of hardware
  - Software/Service is still relatively statically provisioned
- Gaps in current work
  - Lack of Cloud “service engineering”
  - Managing the entire lifecycle automatically.
  - Determining service quality – multiple stakeholders; manage service interruptions
Future Vision for Cloud

Virtualized Services

- Multi tenancy
- Automatic acquisition and provision
- Service dynamically composed - On Demand
- Service structure/components not pre-determined

Moving from totally manual to mostly automatic
Key Open Issues

• No way to automate service lifecycle
  • Services not explicit about their functionality, cost, availability etc.

• Significant human effort

• No negotiation capability for service composition
  • Cannot define or negotiate on variable constraints
  • limited automation support

• Composition and orchestration of services doesn’t use semantically rich language.
A semantically rich, policy-based framework can be used to automate the lifecycle of virtualized services on the cloud.
Key contributions of my Thesis

1. Developed a ‘policy based’ framework and defined phases for the lifecycle of cloud based virtualized services.

2. Protocol to automate Negotiation process for acquiring Services from the Cloud
   - Harmonizes Enterprise policies (e.g. around security) with provider policies

3. Design a framework to automate monitoring service quality/performance.

4. Developed Cloud Broker applications.
Impact

- Opened a new sub field in computer science research - cloud based services
- Advanced the State of Art in cloud services
  - First attempt at an integrated holistic approach to cloud service engineering.
  - Can form as groundwork to future work in this nascent field
- Approach grounded in actual enterprise data obtained from NIST, a large international financial organization, and UMBC

© Karuna P Joshi, 2012
Technical Approach

- Semantic web technologies
  - OWL (Web Ontology Language)
  - RDF (Resource Description Framework)
  - SPARQL – Service endpoint using Fuseki, Joseki servers
- Web Technologies – Perl, PHP
- Cloud Computing platforms
  - Virtual Computing Lab (VCL)
  - Eucalyptus Cloud
  - Adopted stakeholder definitions from the NIST Cloud reference architecture
Related Work

- **Service lifecycle** - Most proposals limited to web services and do not account for virtualized environment
- **Various initiatives to standardize services**
  - W3C consortium - WSDL, BPEL, WS-Policy
- **Service Negotiation**
  - WS-Negotiation, WS-Agreement (not semantically rich)
  - Negotiation agents – preliminary work at abstract level (contemporaneous work)
- **Service Quality** - metrics limited to e-business and web services. Little work on performance of composed services.
- **Cloud Computing Platforms** – Eucalyptus, VCL
- **Related work detailed in dissertation**
CLOUD SERVICE LIFECYCLE

Acquisition of cloud based services
Service Lifecycle Methodology

- Service phases
  - Requirements, Discovery, Negotiation, Composition and Consumption
- Applicable on any
  - Cloud deployment (private, public, hybrid, community)
  - Service model (SaaS, PaaS, IaaS)
- Developed high level ontologies (Appendix 1).
- Determined Metrics to track at each phase.
- Published in conferences ICVCI 2009, SRII 2011, Doctoral consortium ISWC and IEEE Transactions on Service Computing (under review).
Phases of IT Services Lifecycle

1. **Service Requirements**: New Service needed
2. **Service Negotiation**: Contract signed
3. **Service Discovery**: Provider(s) identified
4. **Service Composition**: Service delivered
5. **Service Consumption**: Service delivered

CONSUMER

SERVICE CLOUD

© Karuna P Joshi, 2012
Phase 1 - Service Requirements

Compiled by Cloud Consumer and includes

- Functional specifications (tasks to be automated)
- Budgetary policies/Cost constraints
- Technical Policy specifications
- Security Policy
- Data Quality, Retention Policy
- Service Compliance Policy
- Human Agent Policy
High Level Ontology for Requirement Phase

Class: Specification
- Name
- Description
- hasFunctionalSpecs
- hasTechnicalSpecs
- Security Policy
- Human Agent Specs
- Data Quality Level
- Service Compliance

Class: Request for Service
- RFS-ID
- Exp_Svc_Begin_Date
- Exp_Svc_End_Date
- RFS_Respond_by_dt
- Cost_constraint
- hasSpecification

Class: Cloud Consumer
- Consumer-ID
- Description

Class: Cloud
- hasSues

Class: Data Quality policy
- hasDataQualityLevel

Class: Service Layer
- hasServiceLayer

Class: Service
- hasDomain
- Domain
- Characteristics

Class: Functional Specs
- Task
- Description
- Cloud Service Layer
- Service Domain

Class: Human Agent Specs
- Experience
- Availability
- Nationality
- DomainExpert
- OSExpert

Class: Technical Specs
- hasOSdetails
- hasHWdetails
- hasAppDetails

Class: Operating System Details

Class: Hardware Details

Class: Security policy

Class: Service Compliance
- hasCompliance

Class: Data Quality Level
- hasDataQualityLevel

Class: Functional Specs
- hasFunctionalSpecs

Class: Human Agent Specs
- hasHumanAgentSpecs

Class: Technical Specs
- hasTechnicalSpecs

Class: Service Layer
- IaaS, PaaS, SaaS

© Karuna P Joshi, 2012
Example: Data Quality Policy

// allows services of low data quality to be accepted if the service is free of cost.
// Using AIR Policy Language

@forAll :SERVICE_QUALITY, :COST.
 :data_quality_policy a air:Policy;
   rdfs:label "Data Quality policy";
   air:rule :data-quality-rule.
 :data-quality-rule a air:Belief-rule;
   rdfs:label "data quality rule";
   air:pattern {
     :SERVICE_QUALITY quality_level :LOW.
     :COST service_cost :0.
   };
   air:assert { :SERVICE_QUALITY air:compliant-with :data_quality_policy.};
   air:alt { air:assert { :SERVICE_QUALITY air:NOTcompliant-with :data_quality_policy.} }.

Service ‘compliant’ if service cost is 0, even if data quality is low
Phase 2 - Service Discovery

- **Cloud Services Broker** used to search available services that match the specifications

- Identify Gaps
  - Services functionality
  - Policies supported

- **Cloud Auditor** or centralized registry (like UDDI) will certify the service provided.
High Level Ontology for Discovery Phase

Class: Request for Service
- RFS-ID
- Exp_Svc_Begin_Date
- Exp_Svc_End_Date
- RFS_Respond_by_dt
- Cost_constraint
- hasSpecification

Class: Cloud Auditor
- Provider-ID
- Service Name
- Service ranking
- Security Audit
- Performance Audit
- Privacy Impact Audit

Class: Cloud Broker
- Broker-ID
- Broker Service
- Service ranking

Class: Cloud Consumer
- Consumer-ID
- Description

Class: Provider List
- Provider
- Service details
- Service availability
- Service Cost
- Service gaps

Class: Cloud Provider
- Provider-ID
- Service list
- Description

Used in

Returns list to

Creates

Verifies Provider

Adds to
Phase 3 - Service Negotiation

- Discussion and agreement that the cloud provider and cloud consumer have regarding the Service.
- Service Level Agreements (SLA) finalized between consumer and main provider
- Quality of Service (QoS) decided between primary provider and component providers.
Ontology for Negotiation Phase

Class: Request for Service
- RFS-ID
- Exp_Svc_Begin_Date
- Exp_Svc_End_Date
- RFS_Respond_by_dt
- Cost_constraint
- hasSpecification

Class: Cloud Consumer
- Consumer-ID
- Description

Class: Cloud Provider
- Provider-ID
- Service list
- Description

Class: Cloud Carrier

Class: Contract Negotiation

Class: Consumer Negotiation

Class: Service Level Agreement
- SLA Name
- Description
- SLA Metrics
- Penalty

Class: Quality of Service (QOS)
- QOS Name
- Description
- QOS Metrics
- Penalty

Class: Service Contract

Class: Contract

Class: Dependent Service Sub-Contract
Phase 4 - Composition Phase

- One or more services provided by one or more providers are combined and delivered as a single Service
- SLA and QoS used for determining service components and orchestration
- We reuse and extend OWL-S ontology
- Key metrics - Coupling and Cohesion
Ontology for Composition Phase

Class: Specification
- Name
- Description
- hasFunctionalSpecs
- hasTechnicalSpecs
- Security Policy
- Human Agent Specs
- Data Quality Level
- Service Compliance

Class: Cloud Provider
- Provider-ID
- Service list
- Description

Class: Service

Class: OWL-S - Composite Process

Class: Service Contract

Class: Quality of Service (QOS)
- QOS Name
- Description
- QOS Metrics
- Penalty

Class: Dependent Service Sub-Contract

Part of

Determines

Composes

Refers to

Part of

Refers to
Phase 5 - Consumption Phase

- Composed Service is consumed and monitored in this phase.
- Key **metrics** like Service Performance and reliability are monitored.
  - SLA, QoS determine performance of the service
- Phase includes Service Delivery & Payment
- Customer Satisfaction is tracked
Ontology for Consumption Phase

- Class: Cloud Provider
  - Service list
  - Description

- Class: Cloud Consumer
  - Consumer-ID
  - Description

- Class: Service Contract
- Part of

- Class: Service
  - Provides
  - Refers to

- Class: Service Level Agreement
  - Task
  - Description
  - SLA Metrics
  - Penalty

- Class: Service
  - Determine quality for

- Class: Dependent Components

- Class: Response Time
- Class: Latency
- Class: Throughput

- Class: Performance

- Class: Software Components

- Class: Human Component
  - subClass of

- Class: Tier 1 agent
  - subClass of

- Class: Tier 2/external agent

- Class: Security Policy

- Class: Reliability

- Class: Resolution Time

- Class: Assurance

- Class: Presentation

© Karuna P Joshi, 2012
## Cloud Services quality metrics

<table>
<thead>
<tr>
<th>Quality Metrics</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delivery</td>
<td>Monitor if service is delivered on time and in mode agreed upon.</td>
</tr>
<tr>
<td>Security</td>
<td>Monitor service security to ensure it adheres to the security policies</td>
</tr>
<tr>
<td>Cost</td>
<td>Monitor cost of the service to ensure it stays within budget</td>
</tr>
<tr>
<td>Service payment</td>
<td>Track if payment was done per the agreed payment options (metered/subscription etc.).</td>
</tr>
<tr>
<td>Reliability</td>
<td>Reliability tracks the service quality to ensure the service functionality and data accuracy is maintained</td>
</tr>
<tr>
<td>Performance</td>
<td>Track the service performance that includes throughput, latency and response time.</td>
</tr>
<tr>
<td>SLA</td>
<td>Track Service Level Agreements defined in Negotiation Phase.</td>
</tr>
<tr>
<td>QoS</td>
<td>Track dependent services to ensure Quality of Service.</td>
</tr>
<tr>
<td>CSAT</td>
<td>Periodically the provider tracks if consumers are satisfied with the service.</td>
</tr>
</tbody>
</table>
SERVICE NEGOTIATION
Service Constraints

• Cloud Service Constraints specified as
  • Hard Constraints or Soft Constraints
  • Some attributes may have both hard and soft constraints associated
• Constraint relaxation
  • Higher or Lower
  • Defined in the ontology

**HARD REQUIREMENTS:**
- SINGLE PROCESSOR
- SPEED AT LEAST 1MHZ
- MINIMUM STORAGE: 100 GB
- MAXIMUM COST: $2/HR

**SOFT REQUIREMENTS:**
- SINGLE PROCESSOR
- SPEED 2MHZ
- STORAGE NEEDED: 500 GB
- COST $1/HR
Ontology for service constraints
Cloud Service Negotiation Protocol

- Negotiation Protocol: set of rules that govern interaction *
  - types of participants, negotiation states, events that cause states to change and valid actions of the participants.
- Protocol enables exchange of RDF ‘instances’ between consumer and provider to finalize SLA
  - Removes overhead associated with ‘message’ or ‘template’ exchange
- Three states of response – **Accept, Reject, Relax**
- Expanded Contract Net Protocol
- Service Constraint Relaxation
  - Relax soft constraints till hard constraint reached
  - Multiple constraints have to be prioritized

* Lomuscio, Wooldridge, Jennings, “A Classification Scheme for Negotiation in Electronic Commerce”, J of Group Decision and Negotiation, Volume 12, Number 1 (2003), 31-56
Negotiation Protocol

Negotiation stops when
• constraints met
OR
• hard constraints not met
Service Level Agreements (SLA)

- Critical component of cloud services to determine service deliverables and metrics
- Developed an OWL ontology
  - Service Level Agreement
  - SLA Domain Metrics
  - SLA Support Metrics
  - Service Cost
  - Service Availability
  - Data Encryption
- Integrated with GoodRelations ontology to read the business name, price/currency specification etc.
  - GoodRelations ontology used in e-commerce describes commodity products and services
Cloud SLA ontology

- **Class: SLA Support Metrics**
  - Audit Name
  - Business continuity plan
  - Escalation Time
  - Outage notification
  - Recovery time
  - Resolution time
  - Response time
  - Scheduled maintenance
  - Support timeframe

- **Class: SLA Domain Metrics**
  - Storage size
  - Data Backup
  - Storage interface
  - VM Separation
  - TIC connection
  - EE CAL
  - Cloud instance size
  - Cloud instance speed
  - Cloud instance cores

- **Class: Encryption**
  - Encryption type

- **Class: Service Level Agreement**
  - Service name
  - Service description
  - Service consumer
  - Service provider
  - Data Deletion
  - Service Delivery
  - Cloud Location
  - hasServiceCost
  - hasAvailability
  - hasSecurity
  - hasSupportMetrics
  - hasDomainMetrics

- **Class: Service Contract**
  - hasSLA

- **Class: BusinessEntity**
  - gr:name
  - gr:description
  - gr:legalName

- **Class: Location**
  - gr:name

- **Class: PriceSpecification**
  - gr:hasCurrency
  - gr:hasCurrencyValue

- **Class: Service Cost**
  - Service Cost
  - Pricing model
  - hasCurrency
  - Defaulting-Penalty

- **Class: OWL-S security**
  - securityCapability
  - securityRequirement
Support SLAs for Cloud Services *

Negotiation for Cloud SLAs will also include agreement on

- Availability timeframe of services
- Contingency (Business Continuity) plans
- Timeframes for notification and recovery following an unplanned service disruption or a security incident

```
cloudSLA:outage_notification rdf:type owl:DatatypeProperty ;
  rdfs:domain cloudSLA:Service_Support ;
  rdfs:range [ rdf:type rdfs:Datatype ;
       owl:oneOf [ rdf:type rdf:List ;
         owl:oneOf [ rdf:type rdf:List ;
           rdf:first "00:05:00"^^xsd:time ;
           rdf:rest rdf:nil
         ]
       ]
  ].
```

* policies of a financial org. , industry best practices

Service Outage notification = 5 minutes
SLAs for Cloud Services (contd..)

- Problem resolution and escalation procedures
- Scheduled maintenance times.

RDF/N3 format

:resolution_time a owl:DatatypeProperty;
  rdfs:domain :Service_Support;
  rdfs:range [ a owl:DataRange;
  owl:oneOf ( "04:00:00"^^xsd:time ) ].

:response_time a owl:DatatypeProperty;
  rdfs:domain :Service_Support;
  rdfs:range [ a owl:DataRange;
  owl:oneOf ( "00:10:00"^^xsd:time ) ].

Cloud SLA files
- include policies agreed upon after negotiation
- Created in machine readable RDF format
- Enable Automated Monitoring and Auditing

Resolution time = 4 hours
Response time = 5 minutes
SMART CLOUD SERVICES

Cloud Broker application in collaboration with NIST
NIST Cloud Broker Prototype

• Automation of Cloud Storage Service acquisition, consumption /monitoring.
• Using our Service lifecycle Ontologies.
• Platform: using SPARQL, OWL, RDF, Web technologies – Perl, HTML and Eucalyptus cloud
• NIST data, security and compliance policies
  • Cloud location, encryption, deletion policies
  • User authentication policy: FIPS 140-2 is a standard used to accredit cryptographic modules.
  • Trusted Internet Connection mandated to optimize individual external connections.
• Published in proceedings of SRII Global Conference 2012.
Prototype for Requesting Storage Services from the Cloud

**Service Attributes**
- Expected Start Date (MM-DD-YYYY): 11-10-2011
- Storage size needed (units/month): less than 2GB
- Service Cost: Free
- Service availability: 95%
- Data Preservation/Backup requirements: Weekly backup

**Data and Security Policies**
- User authentication mechanism: FIPS 140-2 supported
- Data Encryption: No encryption
- Data Location: Anywhere in the world
- Data Deletion: Data archived/inaccessible
- Virtual Machine (VM) separation: Not Supported
- Interface for a storage specification: SOAP WSDL

**Compliance Policies**
- Trusted Internet Connection (TIC): Compliant
- CC Evaluation Assurance Level (EAL): 3

**Cloud Instance**
- Size: 1GB
- Speed: 1GHz
- Number of cores: 10
VCL IMAGE BROKER

Cloud Broker application in collaboration with IBM
VCL Cloud Broker

- Cloud broker to discover VCL Images based on Image information and data / security properties
- Developed ontology for VCL Image (Appendix 3)
- Developed a new web interface
- Platform: using PHP, MySQL
  - VCL Sandbox
- Published in proceedings of ICACON 2012 and accepted for International Journal of Cloud Computing.
NIST Policies/Constraints included

- Data security policies
  - Encryption, Location, Deletion
- US government compliance policies
  - User authentication policy (FIPS 140-2)
  - Virtual Machine (VM) Separation
  - Trusted Internet Connection
Major Changes to VCL architecture

• New Web-based user interface
  • Enables users to define their requirements and the policies that the VCL Image should meet.
  • The priority of the fields was pre-determined. The fields were laid out in order of decreasing priority.
  • Automatic relaxation of constraints till a Image is found
  • Allows reservation of selected Image

• Database server:
  • Added tables to store data, security and compliance policies.
  • Added views join the data spread across the Image, OS, Platform, Image-Security tables. These views were used for querying Images in the user interface.
### VCL broker User Interface

#### Cloud Broker for VCL Platform

**Image Attributes**
- Operating System Type: Windows
- Platform: i386
- Installed Application (enter first 4 words):

**Data and Security Policies**
- User authentication mechanism: FIPS 140-2 supported
- Image Data Encryption: Data encrypted
- Image Data Location: Anywhere in the world
- Image Data Deletion: Data archived/inaccessible
- Virtual Machine (VM) separation: supported
- Interface for a storage specification: SOAP WSDL

**Compliance Policies**
- Trusted Internet Connection (TIC): Compliant
- CC Evaluation Assurance Level (EAL):

**Buttons**
- Discover Images that match
- List all Images (no constraints)

To search for an image, select the parameters above and press the 'Discover Images' button.

If you want a list of all possible images on VCL, press the 'List All Images' button.

On the image list, press the detail button to get information of all the image attributes and to reserve that image.
**Select - List All Images**

### Cloud Broker for VCL Platform

#### Image Attributes
- Operating System Type: Windows
- Platform: i386
- Installed Application (enter first 4 words):

#### Data and Security Policies
- User authentication mechanism: FIPS 140-2 supported
- Image Data Encryption: No encryption
- Image Data Location: Anywhere in the world
- Image Data Deletion: Data archived/inaccessible
- Virtual Machine (VM) separation: Not Supported
- Interface for a storage specification: SOAP WSDL

#### Compliance Policies
- Trusted Internet Connection (TIC): Compliant
- CC Evaluation Assurance Level (EAL): 

**Discover Images that match**

**List all Images (no constraints)**

**No of images discovered: 11**

<table>
<thead>
<tr>
<th>ID</th>
<th>Image description</th>
<th>Platform</th>
<th>OS Type/OS</th>
<th>OS User Authentication</th>
<th>Encryption</th>
<th>Location</th>
<th>Data deletion VM Separation</th>
</tr>
</thead>
<tbody>
<tr>
<td>79</td>
<td>RSLogix 7.0 (WinXP)</td>
<td>i386</td>
<td>Windows XP (Bare Metal)</td>
<td>not supported</td>
<td>Encryption</td>
<td>global</td>
<td>data deleted</td>
</tr>
<tr>
<td>106</td>
<td>SuperPro Designer 6.0 (WinXP)</td>
<td>i386</td>
<td>Windows XP (Bare Metal)</td>
<td>not supported</td>
<td>Encryption</td>
<td>global</td>
<td>data deleted</td>
</tr>
<tr>
<td>121</td>
<td>Aspen 2004.1 with Excel (WinXP)</td>
<td>i386</td>
<td>Windows XP (Bare Metal)</td>
<td>not supported</td>
<td>Encryption</td>
<td>global</td>
<td>data deleted</td>
</tr>
<tr>
<td>451</td>
<td>Photoshop CS3</td>
<td>i386</td>
<td>Windows XP (Bare Metal)</td>
<td>not supported</td>
<td>Encryption</td>
<td>global</td>
<td>data deleted</td>
</tr>
<tr>
<td>547</td>
<td>LabView 8.2.1 (WinXP)</td>
<td>i386</td>
<td>Windows XP (Bare Metal)</td>
<td>not supported</td>
<td>Encryption</td>
<td>global</td>
<td>data deleted</td>
</tr>
<tr>
<td>912</td>
<td>Pro Engineer Wildfire 4.0 (Win XP)</td>
<td>i386</td>
<td>Windows XP (Bare Metal)</td>
<td>not supported</td>
<td>Encryption</td>
<td>global</td>
<td>data deleted</td>
</tr>
<tr>
<td>6</td>
<td>WinXP Base (32 bit VM)</td>
<td>i386</td>
<td>Windows XP (VMware)</td>
<td>not supported</td>
<td>Encryption</td>
<td>global</td>
<td>data archived supported</td>
</tr>
<tr>
<td>244</td>
<td>Web Browsers (WinXP vmware)</td>
<td>i386</td>
<td>Windows XP (VMware)</td>
<td>not supported</td>
<td>Encryption</td>
<td>global</td>
<td>data archived supported</td>
</tr>
<tr>
<td>2567</td>
<td>No Apps with SSH/SFTP (WinXP KVM) NCSU</td>
<td>i386</td>
<td>Windows XP (VMware)</td>
<td>not supported</td>
<td>Encryption</td>
<td>global</td>
<td>data archived supported</td>
</tr>
<tr>
<td>2674</td>
<td>Groupwise 8</td>
<td>i386</td>
<td>Windows XP (VMware)</td>
<td>not supported</td>
<td>Encryption</td>
<td>global</td>
<td>data archived supported</td>
</tr>
<tr>
<td>2441</td>
<td>Web Browsers (Win7, IE9, Firefox 5)</td>
<td>i386</td>
<td>Windows 7 (VMware)</td>
<td>not supported</td>
<td>Encryption</td>
<td>global</td>
<td>data deleted</td>
</tr>
</tbody>
</table>
Select – Discover Images

Cloud Broker for VCL Platform

Image Attributes
- Operating System Type: Windows
- Platform: i386
- Installed Application (enter first 4 words):

Data and Security Policies
- User authentication mechanism: FIPS 140-2 supported
- Image Data Encryption: Data encrypted
- Image Data Location: Anywhere in the world
- Image Data Deletion: Data archived/inaccessible
- Virtual Machine (VM) separation: supported
- Interface for a storage specification: SOAP WSDL

Compliance Policies
- Trusted Internet Connection (TIC): Compliant
- CC Evaluation Assurance Level (EAL):

Discover Images that match
List all Images (no constraints)

No images match service interface “SOAP WSDL”. Relaxing Service Interface constraint. Requerying...
No images match Virtual Machine separation “supported”. Relaxing VM Separation constraint. Requerying...
No images match data deletion “data archived”. Relaxing Data Deletion constraint. Requerying...

No of images discovered: 4

<table>
<thead>
<tr>
<th>ID</th>
<th>Image description</th>
<th>Platform</th>
<th>OS</th>
<th>OS type</th>
<th>User_Authentication</th>
<th>Encryption Location</th>
<th>Data deletion</th>
<th>VM Separation</th>
</tr>
</thead>
<tbody>
<tr>
<td>79</td>
<td>RSLogix 7.0 (WinXP)</td>
<td>i386</td>
<td>Windows XP</td>
<td>windows</td>
<td>FIPS140 supported</td>
<td>Encryption</td>
<td>global</td>
<td>supported</td>
</tr>
<tr>
<td>106</td>
<td>SuperPro Designer 6.0 (WinXP)</td>
<td>i386</td>
<td>Windows XP</td>
<td>windows</td>
<td>FIPS140 supported</td>
<td>Encryption</td>
<td>global</td>
<td>data deleted</td>
</tr>
<tr>
<td>121</td>
<td>Aspen 2004 1 with Excel (WinXP)</td>
<td>i386</td>
<td>Windows XP</td>
<td>windows</td>
<td>FIPS140 supported</td>
<td>Encryption</td>
<td>global</td>
<td>data deleted</td>
</tr>
<tr>
<td>451</td>
<td>Photoshop CS3</td>
<td>i386</td>
<td>Windows XP</td>
<td>windows</td>
<td>FIPS140 supported</td>
<td>Encryption</td>
<td>global</td>
<td>data deleted</td>
</tr>
</tbody>
</table>
Select - (Image) Details button
Select – Reserve Image button

The Image has been successfully Requested.

request id is 6

Go to the Current Reservations page to view the status of your reservation.
Reservation visible under current reservations

<table>
<thead>
<tr>
<th>Environment</th>
<th>Starting</th>
<th>Ending</th>
<th>Initially requested</th>
<th>Reg ID</th>
<th>Comp ID</th>
<th>Management Node</th>
<th>IP address</th>
<th>Current State</th>
<th>Last State</th>
<th>Computer State</th>
</tr>
</thead>
<tbody>
<tr>
<td>WinXP Base (32 bit VM)</td>
<td>Thursday, Aug 23rd, 12:16 pm</td>
<td>Thursday, Aug 23rd, 1:15 pm</td>
<td>Thursday, Aug 23rd, 12:16 pm</td>
<td>10</td>
<td>2</td>
<td>localhost</td>
<td>152.46.20.86</td>
<td>14</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Click the Connect! button to get further information about connecting to the reserved system. You must click the button from a web browser running on the same computer from which you will be connecting to the remote computer; otherwise, you may be denied access to the machine.
SERVICE QUALITY FRAMEWORK
Motivation

- Quality of Service (QoS) metrics related to individual services cannot be directly applied in the virtualized services model.
- Method for translating the metrics related to individual components of the service (backstage metrics) to the front stage experienced by the client.
- This framework useful for service managers
  - Assess if SLA will be met
  - Resources allocation
  - Choose between providers
Quality Framework for Virtualized Services

- Framework to measure and semi-automatically track quality
- IT Services comprise of three elements that contribute towards service performance
  - Human Agents.
  - Actual Software
  - Dependent services.
- A service might not have one of the elements.
- SLAs and QoS exist for each element of the service.
- Published in proceedings of Services Research Institute Global Conference 2011.
Fuzzy Logic to capture Service Quality

- Quality determined by the Performance of a service.
  - related by rules to the performance of other dependent services/resources
- We use fuzzy logic to encode performance rules.
- Fuzzy sets provide us with the ability to classify elements into a continuous set using the concept of degree of membership.
- We use a description such as dependence is HIGH. The relation of a dependence measure to a linguistic term such as high or low will be captured in the membership function.
Steps of the Framework

• Step 1: Identify the service elements
  • Service Software
  • Agents or Human beings
  • Dependent services

• Step 2: Determine the dependencies/ coupling
  • TIGHTLY or LOOSELY coupled

• Step 3: Determine the Service Metrics

• Step 4: Develop the performance rules
  • Use data mining tools like WEKA

• Step 5: Monitor the service quality by reviewing the performance rules
Step 1: IT Services - Helpdesk service

Service

Helpdesk Service

Elements of the service

CRM software service

Agents

ACD software service

Depends on services

Knowledgebase service

Database Service

Expert Agents, External Helpdesks

Telecommunication service

Network Services
Step 2: Service Dependencies

Determine the service coupling/dependencies

a) Agent’s expertise is LOOSELY coupled with the expertise of Tier 2/3 agents or with external agents.

b) CRM software is TIGHTLY coupled with the Database service.

c) Database service is TIGHTLY coupled with the Network service.

d) The CRM software is coupled with the Knowledgebase service, as well as the subject knowledge of the agents.
   • The Knowledgebase could be a set of pre-determined solution list or FAQs or Help systems.
   • Depending on the implementation it can be a HIGH or LOW coupling with the CRM application.

e) ACD+PBX software is TIGHTLY coupled with the underlying Telecommunication service.
## Step 3: Helpdesk Service Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Measures what</th>
<th>Helpdesk Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Satisfaction</td>
<td>Assessed through surveys of customers via telephone call, email or post.</td>
<td>Consumer</td>
</tr>
<tr>
<td>Response Time</td>
<td>The average time phone calls are answered; time it takes for a Help Desk agent who is to troubleshoot the service request to contact an authorized caller.</td>
<td>CRM, ACD</td>
</tr>
<tr>
<td>Call abandon rate</td>
<td>Percentage of calls where callers disconnect before reaching an agent</td>
<td>ACD</td>
</tr>
<tr>
<td>Employee Proficiency</td>
<td>Skill set of the Helpdesk analysts.</td>
<td>Agents</td>
</tr>
<tr>
<td>Call Volume</td>
<td>The number of calls taken by the Help Desk within a certain time period (a day, a month, a year).</td>
<td>ACD, CRM</td>
</tr>
<tr>
<td>Solution Accuracy</td>
<td>Assessment of the accuracy of solutions the Help Desk provides customers.</td>
<td>Agent, CRM, Consumer</td>
</tr>
<tr>
<td>Reliability of Predefined Solutions</td>
<td>How reliable is the Knowledgebase data</td>
<td>Agent, CRM</td>
</tr>
<tr>
<td>Tracking Accuracy</td>
<td>Percentage of helpdesk cases resolved accurately</td>
<td>CRM</td>
</tr>
<tr>
<td>Resolution Time</td>
<td>Average Time it takes to resolve a problem</td>
<td>CRM</td>
</tr>
<tr>
<td>Resolution Excellence</td>
<td>The number of problems resolved versus the number of customer problems issued.</td>
<td>CRM</td>
</tr>
<tr>
<td>First Time Settlement</td>
<td>The number or percentage of problems resolved during the first customer call.</td>
<td>Agent, CRM, ACD</td>
</tr>
<tr>
<td>Number of calls</td>
<td>The number of calls taken per Help Desk agent per shift.</td>
<td>ACD</td>
</tr>
<tr>
<td>Time controller</td>
<td>The time spent per call.</td>
<td>ACD, CRM</td>
</tr>
<tr>
<td>Opened tickets</td>
<td>Number of helpdesk tickets opened per Helpdesk agent per shift.</td>
<td>CRM</td>
</tr>
<tr>
<td>Closed tickets</td>
<td>Number of helpdesk tickets closed per Helpdesk agent per shift.</td>
<td>CRM</td>
</tr>
</tbody>
</table>
Step 4: Develop Performance Rules

- Performance Rules developed for Helpdesk Service
  - From Domain Experts.
  - From Association Rules obtained from mining historic data.
- We reformulate the association rules as fuzzy linguistic rules to make them more general.
  - predict the performance of some future helpdesk service that is composed from similar component elements.
- We can fuzzify the performance metrics into the fuzzy variables HIGH, LOW and MEDIUM.
1. If \{(CRM Software loosely coupled with Knowledgebase service) AND (Knowledgebase service response time is LOW)\} then CRM Software response time is MEDIUM.

2. If CRM software performance is LOW then Resolution Time is HIGH.

3. If \{(CRM Software tightly coupled with Knowledgebase service) AND (Knowledgebase service response time is LOW)\} then CRM Software response time is LOW.

4. If \{(Agent’s proficiency is tightly coupled with Knowledgebase) AND (Knowledgebase service solution accuracy is LOW)\} then Solution Accuracy is LOW.

5. If \{(Agent is loosely coupled with Expert’s service) AND (Expert’s Performance is LOW)\} then Agent’s Proficiency is MEDIUM.

6. If \{(Agent is tightly coupled with Expert’s service) AND (Expert’s Solution Accuracy is LOW)\} then Solution Accuracy is LOW.

7. If \{(Agent’s Proficiency is LOW) OR (Solution Accuracy is LOW)\} then Customer Satisfaction is LOW.
Some performance rules mined using WEKA data mining:

- Ticket Was Assigned to → Ticket Assigned to confidence: (0.98)
- Ticket Assigned to → Ticket closed by confidence: (0.98)
- Ticket Assigned by Mailer demon → Priority=Normal confidence: (0.98)
- Ticket Origin=Email Initial Assigned To=HelpDesk:Mail Assigned by=Mailer daemon → Priority=Normal confidence: (0.98)
- Cust Position=Staff resolution time=HIGH → Priority=Normal confidence: (0.96)

Linguistic Rules generated:

- If {Resolution Time is LOW} then Helpdesk Service performance is HIGH
- If {(Agent assigned ticket = Agent closing ticket) AND (Agent proficiency is HIGH)} then Resolution time is LOW
- If {ACD performance is HIGH} then Service performance is HIGH
- If {Assigned group is (X)} then Service performance is HIGH
The Linguistic rules can help to

- Predict the performance of a new (helpdesk) service.
- Improve Quality of composed services.
  - Identify service components/agents affecting quality.
- Help determine appropriate SLA levels for new service designed using the existing components.
  - If the high performing groups are used in a new service provisioning, higher SLAs can be promised compared to if the not so well performing groups are used.
- Co-relate the performance rules with the Customer Satisfaction levels being tracked.
Summary

• Developed an integrated methodology to acquire, consume and monitor cloud services.
  • Developed a negotiation protocol to automate the negotiation process while acquiring services.
  • Developed OWL Ontologies that are in public domain.
• Publications (with References) http://ebiq.org/j/93
• Contact me at karuna.joshi@umbc.edu if you wish to discuss this further.