Virtual Organizations By the Rules

Carl Kesselman
Industrial and Systems Engineering
University of Southern California

Ian Foster
Computation Institute
Argonne National Lab & University of Chicago

Quan Pham
Computer Science
University of Chicago
“With the establishment of large scale multidisciplinary production Grid infrastructures such as the EGEE, OSG, DEISA, TeraGrid, or NAREGI, the concept of Virtual Organizations (VO) has been constantly refined and efficient management of VOs and their policies is becoming one of the central topics for these infrastructures.”
The problem that underlies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations. The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a virtual organization (VO).
What is an Organization?

- A organization has an **identity** and a **purpose**, which it seeks to fulfill within its **environment**

- The organization’s purpose influences its **participants**, **structure**, **activities**, and **deliverables**, whether products or services

- The organization’s **performance** can be evaluated with respect to various metrics

Is a virtual organization any different?
From the Organizational Behavior and Management Community

“[A] group of **people** who interact through interdependent tasks guided by common purpose [that] works across space, time, and organizational boundaries with links strengthened by webs of **communication technologies**”

— Lipnack & Stamps, 1997

- Yes—but adding cyber-infrastructure:
  - People → computational agents & services
  - Communication technologies → IT infrastructure

Collaboration based on rich data & computing capabilities
Enterprise Architecture

- Model structure and operation of business from perspective of achieving business objectives
  - Codify in terms of business rules and processes
  - Many tools exist to capture this (e.g. UML, BPMN)
- Processes and rules
  - Business processes capture business objectives
  - Business rules determine when to apply processes
- Identify which functions map into IT
  - Model core business functions as services (SOA)
  - Compose services into business processes
    - WS-CDL (choreography), BPEL (orchestration)
<table>
<thead>
<tr>
<th><strong>Identity</strong></th>
<th>Legal aspects. Credentials.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Purpose</strong></td>
<td>Anything legal ...</td>
</tr>
<tr>
<td><strong>Environment</strong></td>
<td>Available service &amp; resource providers. Legal &amp; organizational constraints</td>
</tr>
<tr>
<td><strong>Participants</strong></td>
<td>Identity-based or attribute-based. People, services, resources, sensors.</td>
</tr>
<tr>
<td><strong>Structure</strong></td>
<td>Centralized, decentralized, ...</td>
</tr>
<tr>
<td><strong>Activities</strong></td>
<td>Business processes. Workflows.</td>
</tr>
<tr>
<td><strong>Deliverables</strong></td>
<td>Data products. Services. Instrument operations. ...</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>Throughput, responsiveness, growth, happiness, security, ...</td>
</tr>
</tbody>
</table>
VO as a Service (VOaaS)

- Virtual organizations integrate participants and resource providers
  - Participants are selected or self assemble
  - Select “best of breed” providers for VO services
- Much of this process can be automated
  - Provisioning of enabling services, at least
GT4 authorization and delegation services provide first implementations
Policy, Revisited

- Traditionally policy is enforced at end points, integrated with application
  - E.g., PDP call-out in GT container
- We can also apply policy at the VO level
  - Define interactions between services at the organizational level
  - Factor policy out of service implementations
- Policy is broader than access control
Policy-Driven Service Oriented Architecture

- Need stand-alone policy engine to coordinate at VO level
- Connection between application policy and infrastructure policy (dynamic provisioning)
- Policy extension points designed into services allow
  - Coordination at VO level
  - Dynamic policy enforcement across services and service oriented infrastructure

Web Services 2.0: Policy-driven Service Oriented Architectures
Thomas B Winans and John Seely Brown
Establishing VO-Wide Policy

ATA: Attribute Authority
AZA: Authorization Authority
Policy Driven VO?

- Question: can we use a “generic” rules engine to organize science based VO?
- Advantage would be
  - Better adaptability to address VO lifecycle evolution
  - More sophisticated policy
    - E.G. composibility with local participant polices, for example with regard to SLA
  - Less special built software
- Disadvantages
  - Complexity of writing and maintaining rules
  - Performance of rules engine
Data Replication In LIGO

- Pull “missing” files to a storage system

Data Movement

Data Replication Service

List of required Files

Reliable File Transfer Service

GridFTP

Data Location

Local Replica Catalog

Replica Location Index

GridFTP

Local Replica Catalog

Replica Location Index

"Design and Implementation of a Data Replication Service Based on the Lightweight Data Replicator System,” Chervenak et al., 2005
Data Replication In LIGO

- Pull “missing” files to a storage system

Data Movement

Rules Engine

Data Replication Policy

GridFTP

Local Replica Catalog

Replica Location Index

Data Location
Rules Engines

- **Use DROOLS**
  - Forward chaining (if conditions then actions)
  - Preconditions on current state called working memory
  - Actions can update state or initiate business process (i.e. make Java method call)
  - Timed rules
- **Implemented in Java**
  - Can be wrapped into a service itself
  - Implements JSR 94 rules engine interface
Functionality - Operation

- Add new replication site.
- Remove existing replication site.
- Add new directories for replication monitoring
  - once a directory is added for monitoring, any file changes in the directory (and its subdirectories, recursively) will be updated to the replicas of that file.
- Remove directories from monitoring pool
Functionality - Query

- **Query for file replication status**
  - Number of replications
  - Location of replications
- **Query for replication site status**
  - Site availability
  - Number of files replicated on that sites
- **Stored in working memory of rules engine**
# System Rule – Sample Rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rule &quot;New Replication Site&quot;</strong></td>
<td></td>
</tr>
<tr>
<td>• new site</td>
<td>• add new site to the session</td>
</tr>
<tr>
<td></td>
<td>• add this site to a site selector (currently a RoundRobin object)</td>
</tr>
<tr>
<td><strong>Rule &quot;New DataCatalog&quot;</strong></td>
<td></td>
</tr>
<tr>
<td>• data STATUS_AVAILABLE</td>
<td>• create a DataTransfer (data, site)</td>
</tr>
<tr>
<td>• the site selector selects a site</td>
<td>• inform the site selector of this usage</td>
</tr>
<tr>
<td>• no DataTransfer for this site and data</td>
<td></td>
</tr>
<tr>
<td>• number of DataTransfer is less than required</td>
<td></td>
</tr>
<tr>
<td><strong>Rule &quot;Site Became Error&quot;</strong></td>
<td></td>
</tr>
<tr>
<td>• site has STATUS_ERROR</td>
<td>• change data as needed</td>
</tr>
<tr>
<td>• there is a DataTransfer to this site (finished or not)</td>
<td>• remove the DataTransfer</td>
</tr>
<tr>
<td><strong>Rule &quot;DataCatalog Updated&quot;</strong></td>
<td></td>
</tr>
<tr>
<td>• data has STATUS_MODIFIED</td>
<td>• clean up and delete that DataTransfer</td>
</tr>
<tr>
<td>• exists DataTransfer for this data</td>
<td></td>
</tr>
</tbody>
</table>
Replication Rule

rule "New DataCatalog"
dialect "java"
when

    # total number of replicas does not meet requirement
    $data : DataCatalog(
        status == DataCatalog.STATUS_AVAILABLE,
        requiredReplicaCount > replicaCount 
    )

    # the round robin controller
    $roundRobin : RoundRobin(site == $site)

    # site still has free resources
    $site : ReplicationSite ( available == ReplicationSite.STATUS_AVAILABLE )

    # site does not have this replica yet
    not DataTransfer( data == $data && site == $site )
then

    insert ( new DataTransfer ( $data, $site, $session ) );
    modify ( $data ) { addReplicationSite ($site) };
    modify ( $site ) { addDataCatalog ($data) };
    modify ( $roundRobin ) { use( $site ) };
end
Summary

● Have created a prototype
  ◆ Performance studies
  ◆ Reliability studies

● Interesting questions
  ◆ Complexity of building reusable policy
  ◆ Composition of different types of policy, e.g. replication and site availability
  ◆ Smooth coupling of traditional VO security policy with business rules
  ◆ How to build a scalable and robust VO wide policy cloud
Beyond Being There: A Blueprint for Advancing the Design, Development, and Evaluation of Virtual Organizations

Final Report from Workshops on Building Effective Virtual Organizations

May 2008

This work was supported by the National Science Foundation under Award Nos. 0751539 and 0816932. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[Search “BEVO”]