Policy-Based Governance in Service-Oriented Computing

Munindar P. Singh
(Joint work with Matthew Arrott, Emilia Farcas, Michael Meisinger, Kartik Tadanki, Amit Chopra, Nirmi Desai, Yathi Udupi)
singh@ncsu.edu

North Carolina State University

April 2, 2010
Outline

Introduction

Service Engagements
  Commitments
  Modeling Engagements

Governance
  Policy
  Contracts

Case Study

Conclusions and Directions
The Evolution of IT

▶ **Applications:** Control of computations hidden in code; integration a nightmare

▶ **Workflows:** Control abstracted out; integration still difficult

▶ **Standards-driven orchestration:** Integration improved; limited support for autonomy

▶ **Messaging:** Integration simplified by MoM and transformations; limited support for autonomy

▶ **Choreography:** Model conversations over messages; limited support for autonomy

▶ **Governance:** Administer resources via interactions among autonomous parties
What is a Service?

- **Traditional, as in WS**: Abstraction of a computational object
- **Improved conventional**: Abstraction of a “capability”
- **Real life**: Participant in a service engagement
  - Independent parties
  - Symmetric relationships: complementary capabilities and goals
Service Engagements
Crucial to the modern economy; major trend in computing

- Business interactions characterized by
  - Autonomous parties
  - Coproduction
  - Contractually constrained
  - Symmetric relationships: complementary capabilities and goals

- Contrast with web services, which merely abstract a computational object
Challenges for Policies and Decision Making

- No unique locus: separate policies for each autonomous participant
- Dependence on business relationships
- Complexity of modeling
  - Specifying vocabulary pertinent to service engagements
  - Determining where policy decisions apply
- Idea: Architecture for governance centered on conversations
  - Domain-specific policies: Incorporate monitoring and responding to events
  - Generic policies: Altering business relationships
Understanding Governance

Philosophy

Governance is about how stakeholders administer their resources

- Focus on stakeholders
- Focus on interactions among stakeholders, framed as normative relationships
- Focus on policies
- Focus on where the policies apply
- Focus on perspicuous specification of policies
Outline

Introduction

Service Engagements
  Commitments
  Modeling Engagements

Governance
  Policy
  Contracts

Case Study

Conclusions and Directions
Elements of a Service Engagement

- **Transactional**: main purpose and enactment of the engagement
- **Structural**: partnerships and contracts
- **Contextual**: setting of the engagement
Traditional Approaches
Quite Unlike a Real-Life Service Engagement

- Take participants flows (e.g., in BPEL) as units of abstraction
  - Mix private policies and public interactions
  - Proprietary: may not be available for reuse
  - Context-laden: even when available, cannot be readily reused
- Focus on low-level (e.g., WS-CDL) or data-level meanings (e.g., OWL)
  - Ignore business-level significance of messages
  - Ambiguous; not verifiable
A Real-Life Service Engagement

Expressed as Interacting Flows

AGFIL (Insurance Company)

Notify
Lee CS

Obtain
claim form

Check
claim form

Amend
estimate

Reconcile
info

Finalize
claim

Lee CS (Claim Handler)

Obtain
details

Estimate
< 500

Assign
adjustor

Agree
repair

Check
invoice

Receive
car

Estimate
repair cost

Inspect
car

Repair
car

Invoice

Gather
t

Verify
info

Assign
garage

Contact
garage

(Call Center)
Introducing Protocols and Policies

Centered on Interaction

- Interaction protocols are modules of abstraction
  - Separate from *policies*, which are inherently private
  - Help identify *policy points*: where policies apply
  - Modular, reusable

- Express interaction meanings chiefly via *commitments*, which
  - Are atoms of contractual relationships
  - Enable correctness checking of contracts
  - Yield precise meanings and verifiability
What is a Protocol?
A Description of Business-Level Interactions

A reusable unit of interaction

- Analogous to an abstract class or interface for objects
- Specifies well-defined roles
- Specifies messages among the roles and how they affect interaction state
  - Capturing commitments on an endpoint (business partner playing a role)
  - Setting local policies while complying with a protocol
- Stored in a repository, i.e., as an asset or resource in its own right
- Refined and composed for implementation
Commitment Life Cycle (and Patterns)
CC(debtor, creditor, antecedent, consequent)

(a) Commit
- **commitment**
  - null
  - active
    - create
    - expire
  - conditional
    - detach
  - base
    - discharge
    - cancel
    - satisfied
    - violated

(b) Relieve
- **commitment**
  - active
  - release
  - null
Commitment Operations

- $create(CC(x, y, p, q))$ establishes the commitment
- $detach(CC(x, y, p, q))$ turns it into a base commitment
- $discharge(CC(x, y, p, q))$ satisfies the commitment
- $cancel(CC(x, y, p, q))$ cancels the commitment
- $release(CC(x, y, p, q))$ releases the debtor from the commitment
- $delegate(z, CC(x, y, p, q))$ replaces $x$ by $z$ as the debtor
  - $x$ remains ultimately responsible (in our work)
- $assign(w, CC(x, y, p, q))$ replaces $y$ by $w$ as the creditor
## Patterns for Delegate

(a) Transfer responsibility

(b) Retain responsibility

(c) Escalate

(d) Withdraw delegation
**Contextual Patterns: Penalize and Revert**

- **Penalize**:
  - **Original**: \( C(\text{debtor, creditor, context, true, original-condition}) \)
  - **Context**: \( C(\text{context, creditor, context, cancel(Original), create(Penalty)}) \)
  - **Penalty**: \( C(\text{debtor, creditor, context, true, penalty-condition}) \)

- **Revert**:
  - **Context**: \( C(\text{context, debtor, context, undo(precondition), released(Progress) or active(Revert)}) \)
  - **Revert**: \( C(\text{creditor, debtor, context, true, undo(condition)}) \)
  - **Progress**: \( C(\text{debtor, creditor, context, precondition, condition}) \)

(a) Penalize

(b) Revert offer
A Purchase Service Engagement

(a) Pair of conditional commitments describing purchase

(b) Introducing bank and shipper via delegations of commitments

(c) Allowing buyer to skip payment or get a refund upon returning goods.
A Real-Life Service Engagement (Repeated)

AGFIL (Insurance Company)
- Notify Lee CS
- Obtain claim form
- Check claim form
- Amend estimate
- Reconcile info
- Finalize claim

Lee CS (Claim Handler)
- Obtain details
- Estimate < 500
- Assign adjustor
- Agree repair
- Check invoice

Gather info
- Validate info
- Assign garage

Notify AGFIL
- Assign garage

Receive car
- Estimate repair cost
- Inspect car
- Repair car
- Invoice

(Call Center)

Mechanic

Singh (NCSU)
Example Contractual Relationships (at Outset)

Europ Assist
C0=CC(AG, EA, reqAuth, authResponse)
C1=CC(AG, EA, claimResponse, payForResponse)

AGFIL
C4=CC(AG, LC, consultingService, payForService)

Inspector
C2=CC(I, LC, inspectReq, inspectRes)
C3=CC(LC, I, inspectRes, payForInspection)

Lee CS

John Doe
Mechanic
Outline

Introduction

Service Engagements
  Commitments
  Modeling Engagements

Governance
  Policy
  Contracts

Case Study

Conclusions and Directions
Governance Understood
Broadly, Administering Service Engagements

- *IT Governance*: How IT resources are administered
- *SOA Governance*: How services are created, deployed, removed, ...
- Currently, humans achieve governance manually
  - Low productivity
  - Poor scalability to fine-grained, real time governance decisions
  - Hidden, implicit considerations yield low confidence in correctness and poor maintainability
Importance of Governance
Stakeholders Using Resources to Best Serve Each Other’s Needs

- Share resources in a controlled manner
- Configure and reconfigure
- Enable unanticipated uses for resources
- Administer respecting human organizational needs

In particular, stakeholders administer themselves
Governance versus Management
Alternative Approaches to Administration

- Management: by superiors of subordinates
  - Control over managed resources
  - Necessary but not sufficient
- Governance: by autonomous equals of themselves
  - Collaborative decision-making among stakeholders
  - Share resources flexibly, enabling unanticipated uses
  - Administer respecting human organizational needs

Currently, governance is manual via out-of-band communications
Difficulty of Governance

Independence of Stakeholders

- **Autonomy**: Stakeholders behave independently, constrained only by their agreements
- **Heterogeneity**: Stakeholders are independently constructed, constrained only by interface descriptions
- **Dynamism**: The set of stakeholders and their mutual relationships may change continually
Achieving Governance: Agents and Orgs
Put collaboration center stage

- Agents represent the stakeholders: people and organizations
  - Provide a locus for interaction
- Orgs are like *institutions*: have an identity and life time distinct from their members; also modeled as agents
  - Examples: NCSU, UNC System, ...
  - Provide a locus for roles and authorizations
  - Enforce behavioral constraints on members
    - Their main hold over their members is the threat of expulsion
Approach
Based on a conceptual model for governance

- Determine what attributes are subject to Identity Management
- Specify an execution architecture
- Specify interactive aspects building on the execution architecture
- Determine a core language for expressing governance structures, policies, and interactions
- Understand policy authoring needs
Governance Operationally
Policy Model: Types

The policy interactions need to go beyond traditional access control

- Each policy can be understood in terms of its cause and its effect
- **Cause**
  - Reactive: triggered by a request from another stakeholder
  - Proactive: triggered by local observations
- **Effect**
  - Authorization of action to be taken on behalf of requester
  - Enablement of action, which would otherwise not be taken
  - Obligation of action, which would now be performed
Policy Model: Information

Each policy relies upon certain information in order to produce a decision

- Attributes of the parties involved
  - Qualifications, affiliations
- Attributes of the capabilities involved
  - Interactions to be carried out upon resources
  - Collated as interaction types and resource types
- Attributes of the relationships among the parties involved
  - Participations in different Orgs
  - Arrangements among institutions (captured as participations)
  - Ongoing conversations
Contracts Lifecycle
Outline

Introduction

Service Engagements
  Commitments
  Modeling Engagements

Governance
  Policy
  Contracts

Case Study

Conclusions and Directions
Ongoing Studies
Ocean Observatories Initiative (OOI)

- Primary: Operational Activity Model (OV5) document describing the entire life cycle via several use cases
  - Resources being created
  - Resources being registered and published
  - Resources being commissioned and decommissioned
  - Several more ...

- Secondary: OOI Concept of Operations document
The OV5 Register Activity Diagram

Developed by others

Define all the particulars of this product or service (e.g., location, content, function, authorities, permissions).

Registramt

- Characterize
- Submit

Infrastructure

- Certify
- Accepted?
  - No
  - Yes

Registrar

- Document
- Index & Cross-Reference
- Advertise?
  - No
  - Yes
  - Register

Registration Catalog

External Catalog

Publish
What We Extract from the OV5 Register Activity

- **Roles**
  - Registrar (e.g., facility administrator)
  - Registrant (e.g., a researcher)

- **Main interactions**
  - Registrant registers a new resource (e.g., a data stream) to make it available to others
  - Registrar advertises a registered resource

- **Policy points for the registrar**
  - Whether to accept the registrant’s request
  - Whether to advertise a registered resource
The OV5 Commission Activity Diagram

Developed by others
What We Extract from the OV5 Commission Activity

- **Roles**
  - Operator (e.g., a test facility or deployment engineer)
  - Provider (e.g., a researcher)

- **Main interactions**
  - Provider requests the operator to certify a data stream from a sensor
  - Operator completes verification of deployment of a sensor that has been requested for commissioning

- **Policy point for the operator**
  - Whether to accept the provider’s request

- **Policy point for the provider**
  - Whether to proceed to validate the deployment
Governance of AMQP Exchange Space
Highlighting the business relationships

- Consumer Application (as Communicator)
- Publishing Application (as Communicator)
- Exchange Point (as Distributor)
- Exchange Space (as Org Singleton)

1. Enroll as communicator
2. Enroll as distributor
3. Find Distributor
4. Allocate PubSub
5. Publish
Case Study

Vocabulary Example for a Resource Sharing Community

// The following are the generic properties of our formal governance
// model, and may be used in any specification.

// The following are the signatures of the various properties that we
// use. These are introduced in the governance models (see
governance-models.vsd).

// The prefixes of the property names ("C_" and such) are introduced
// in the governance models vocabulary.

Capability:Communicative C_Request (?Who, ?Whom, ?What);
Capability:Normative N_Grant(?Who, ?Whom, ?What);
Capability:Normative N_Revoke(?Who, ?fromWhom, ?What);

Capability:Participation P_Admit(?Who, ?Org, ?Role, ?Whom);
Capability:Participation P_Eject(?Who, ?Org, ?Role, ?Whom);

Capability:Resource R_Contribute(?Owner, ?anOrg, ?aResource, ?aCapability);
Capability:Resource R_Withdraw(?Owner, ?anOrg, ?aResource, ?aCapability);

// A S_Member is any principal playing any role in an Org
Predicate:Participation S_Member(?anOrg, ?aPrincipal, ?aRole);

// A S_Registrand (note that the last letter is "d") is a resource
// that has been contributed (and not yet withdrawn) to an org; the
// contributor is the "registrant"
Predicate:Participation S_Registrand (?anOrg, ?aRegistrant, ?aResource, ?aCapability);

// S_Owns simply reflects the idea that a principal owns a resource.
// In some cases, we could instead apply an alternative relationship
// such as "controls" or "represents" but then we would need to
// describe how such an alternative relationship arises. Mostly, it
// would be rooted in the owner transferring its powers to another
// principal (in the sense of a power of attorney). In some cases,
// it could involve stewardship of a resource wherein the owner of a
// resource may be divested of all authority over it, and such
// authority invested in another party.

---

Governance-Vocabulary.txt 15% (31,0) (C++/1 Abbrev)------------------------
Loading cc-mode...done
Governance of Community Affiliation Scenario

- User (as Member)
  - Enroll as member
  - Discover service in affiliate community
  - Negotiate usage
  - Engage service

- User (as Member)
  - Enroll as member

- Affiliate Community
  - Negotiate affiliation

- Affiliate Community
Outline

Introduction

Service Engagements
  Commitments
  Modeling Engagements

Governance
  Policy
  Contracts

Case Study

Conclusions and Directions
Research Challenges

Bridging the gap between current IT architectures and user needs

- Capture and generalize important scenarios
- Develop a repository of validated protocols for governance
Principles of Governance

Administration that is intelligent and intelligible

- Vividness of Modeling
  - Grounded in applications; modeled entities are real
- Autonomy of Participants
  - Stating rules of encounter; omitting policies from specifications
- Centrality of Organizations
  - Modeling communities, facilities, the OOI; specifying rules of encounter; monitoring contracts; sanctioning violators
- Minimality of Operational Specifications
  - Leaving restrictions unstated except where essential to correctness
- Institutional Actions
  - Creation and manipulation of commitments; granting or denying powers, authorizations; effecting sanctions
  - Separation of concerns from those of operational interactions
- Reification of Representations
  - Explicit: hence, inspectable, sharable, and manipulable
Important Themes for Further Study

- **Conceptual models**
  - Organization theory
  - Norms and institutions
- **Operational models**
  - Such as those based on the pi calculus
  - And how to map conceptual models to operational models
- **Policy languages and architectures**
  - Policies for virtual organizations (Foster; Feeney; . . . )
  - Policy languages (Ponder; Datalog; Rei; . . . )
- **Standardization efforts**
  - IETF Policy Framework
  - XACML
  - DMTF’s Common Information Model
  - WS-Agreement
  - WS-Policy
Thanks!

http://www.csc.ncsu.edu/faculty/mpsingh/