Challenges in Service Modeling, Composition, and Analysis

Jianwen Su
University of California, Santa Barbara
Web Services: The Big Questions

Simplify and/or automate web service

- Discovery
  - What properties should be described?
  - How to efficiently query against them?

- Composition
  - Specifying goals of a composition
  - Specifying constraints on a composition
  - Building a composition
  - Analysis of compositions

- Invocation
  - Keeping enactments separated
  - Providing transactional guarantees

- Monitoring
  - How to track enactments
  - Recovering from failed enactments

Primary focus of this tutorial

An old slide from SIGMOD tutorial [Hull-S. 04 SIGMOD Rec 05]
Data for Services: A New Frontier

Jianwen Su
University of California, Santa Barbara
Outline

- Application Needs
- “Legacy” Services
- “Programmable” Services
- Data Encapsulating Services
- Research Challenges
- Conclusions
A Housing Management Bureau

500 workflow models
300,000 cases/year

200,000+ for

[Jin et al CoopIS 2011]

ICSOC 2012
Permit for Selling an Unbuilt Apartment

**Obtaining a Permit**

1. Application
2. Preliminary review
3. Secondary review
4. Approval
5. Delivery
6. Certificate
7. Lic. fee payment
A Housing Management System

- Ad hoc design, developed over time, patches, multiple technologies, ... a typical legacy system

-Problems:
  - Embedded business logic, hard to learn
  - hard to maintain, costly to add new functionality
  - hard to change/evolve
- Services encapsulate system details and reflect business logic, easier to learn
- Easier to manage even if not technically
- New functions on top of services

---

**ICSOC 2012**

**Tax Calculation**

**Reassessment**

**Appraisal**

**Determine tax base**

**Inheritance**

**Title Change**

**Sales-transaction**

services
Scene 2: A World of “PAL”s, ...

- Organize into collections of services that may be offered to other cities.
Towards a goal of

- Business Process as a Service (BPaaS)
Scene 3: Virtual Enterprise Systems

Towards a goal of
- Business Process as a Service (BPaaS)
- Enterprise may run virtual IT systems
Technical Issues

How to query?
Warn if #applications for title change involving tax reassessment reach 5

How to compose?
Is it “correct”?

How to do transactions?

Add new edu tax

How to change & evoled?
Services have states, but only finitely many.

Can be modeled with:
- finite state machines, process algebras, workflow nets (Petri nets), activity diagrams, state charts, ...

Have been used in studying problems related to service composition:
- Automated design, e.g., Roman services
- Verification
- Optimization (e.g., QoS based)

Wealth of knowledge, rich literature

[WWW, ICSOC, ICWS, SCC, SOCA, ...]
A service consists of activities with a finite state control.

Transitions labeled by activities.

For a given state, the out-edges represent the set of options that will be presented to the user.
The Composition Problem

Online Music Store

Web store

Juke

Bank

available services

desired service

init
search
listen
cart
buy

init
search
listen
cart
buy

search

search

search

init

search

search

listen
cart
buy

buy

2012/11/15
Composition As a Delegator

Delegator: a service annotated with delegations

Available services

Web store

Juke

Bank
Roman Service Composition

- Problem:
  Given a target Roman service and a set of Roman services, find a delegator if exists
  [Berardi-Calvanese-DeGiacomo-Lenzerini-Mecella ICSOC 03]

- Deterministic [Berardi-Calvanese-DeGiacomo-Lenzerini-Mecella ICSOC 03]

- Nondeterministic & Lookahead (batched)
  [Gerede-Hull-Ibarra-S. ICSOC 04]

- May delegate more than once [Berardi et al ICSOC 04]

- With messages [Berardi-Calvanese-De Giacomo-Hull-Mecella VLDB 05]

- Online delegation [Gerede-Ibarra-Ravikumar-S. TCS 08]

- Use only a subset of services [Hassen-Nourine-Toumani ICSOC 08]
Legacy Services in Practice: Limited Applicability

How to query?
Warn if #applications for title change involving tax reassessment reach 5

How to compose?
Is it “correct”?

new service
Certificate

age>55 & …

How to do transactions?

Add new edu tax

How to change & evolve?

Services

Reassessment
Tax Calculation
Appraisal
Determine tax base
Inheritance
Title Change
Sales transaction
Sales transaction

Outline

- Application Needs
- “Legacy” Services
- “Programmable” Services
- Data Encapsulating Services
- Research Challenges
- Conclusions
Service Programming: Data as Variables

- Roughly: finite states plus variables
  - Pragmatic approach
- Examples: BPEL, jBPM, YAWL, ...

- Design:
  Programming with services

- Analysis/checking “correctness”:
  well, undecidable
BPEL is Turing Complete

- BPEL can express all Turing computations
- Checking properties for standalone BPEL is not solvable
- Take a step back: finite state variables

Finite State Control

Before: $b \ a$
Current: $b$
After: $c \ a \ ...$
State: $q$

Before: $a \ b \ a$
Current: $c$
After: $a \ ...$
State: $p$
“Finite State” Variables

- BPEL control structure → a finite state machine
- XML Schema typed variables:
  - primitive types limited to a finite set of values
  - Structured: finitely many “repeats”

```xml
<invoke operation="approve", invar="request", outvar="aprvInfo" >
  <catch faultname="loanfault">
    ? approve_In:
    [approve_In := request]
    ![approve_In]
    handler1
  </catch>
</invoke>
```

Can be improved with the “hyperplane” technique

- [Fu-Bultan-S. WWW 04]
- [Fu-Bultan-S. ISSTA 04]
- [Gerede-S. ICSOC 07]
Helps, But Services Are Connected via Data

How to query?
Warn if applications for title change involving tax reassessment reach 5

How to compose?
Is it “correct”?

new service

Certificate

Reassessment
Appraisal
Determine tax base
Inheritance
Title Change
Sales-transaction

age>55 & …

Add new edu tax

Tax Calculation

How to do transactions?

Service

How to change & evolve?
Two finite state machines with input queues can simulate Turing machines [Brand-Zafiropulo JACM 83]

Model checking BPEL services with queues and finite state variables is not solvable

Collaborating BPEL Services

Finite State Control

\[
\begin{array}{c|c|c|c|c|c}
\hline
a & q & a & r & L \\
\hline
b & q & a & p & R \\
\hline
... \\
\hline
\end{array}
\]

M1

queue

a b # b # q # c a ...

queue

... a # p # c # a c a

M2
Service Programming is an Art

How to query?
Warn if #applications for title change involving tax reassessment reach 5

How to compose?
Is it “correct”?

new service

Certificate

age>55 & …

How to do transactions?

How to change & evolve?

Reassessment

Appraisal

Determine tax base

Sales transaction

Sale tax base tax

Title Change

Inheritance

services

Tax Calculation

Add new edu tax

How to query?

 Warn if #applications for title change involving tax reassessment reach 5
Current Practice

- Data and services are separately modeled, designed, managed
- The separation adds difficulties in design, execution, maintenance, changes
- In addition, many issues can’t be addressed
  - Workflow transaction remains an art
  - Data consistency is a concern of DBMS even though violations are caused by service execution
  - Biz analytics is an after thought
  - Long tail phenomenon is a “holy grail”
Big Data—A Growing Torrent

- McKinsey Global Institute, June 2011: Big data: The next frontier for innovation, competition, and productivity

- Availability of “big data” brings opportunities for improving productivity

$600 to buy a disk drive that can store all of the world’s music
5 billion mobile phones in use in 2010
30 billion pieces of content shared on Facebook every month
40% projected growth in global data generated per year vs. 5% growth in global IT spending
235 terabytes data collected by the US Library of Congress by April 2011

15 out of 17 sectors in the United States have more data stored per company than the US Library of Congress
Big Data + Biz Processes → Big Potential

US health care
- $300 billion value per year
- ~0.7 percent annual productivity growth

Manufacturing
- Up to 50 percent decrease in product development, assembly costs
- Up to 7 percent reduction in working capital

US retail
- 60+% increase in net margin possible
- 0.5–1.0 percent annual productivity growth

Europe public sector administration
- €250 billion value per year
- ~0.5 percent annual productivity growth

Two observations
- A significant portion of big data generated by biz processes
- Productivity growth only obtainable via more efficient/effective biz processes

Global personal location data
- $100 billion+ revenue for service providers
- Up to $700 billion value to end users

Source: MGI Analysis
Service Programming is an Art

How to query?
Warn if applications for title change involving tax reassessment reach 5

How to compose?
Is it “correct”?

The real world is not too kind

How to do transactions?

HELP NEEDED
Needed: Enterprise Data Management

- HRPal
- AccountingPal
- AssessorPal
  - Reassessment
- AppraisalPal
- TaxPal
  - Tax Calculation
  - Determine tax base
- TitlePal
  - Inheritance
  - Title Change

- Business Process as a Service (BPaaS)
Possibility 1: Monopoly Model

- Served by an alliance of PALs lead by a major player
- The major player defines data management, protocols, etc.
Possibility 2: Open Market Model

- No major players
- Enterprise makes its own data management plan
- Bring Your Own Data (BYOD)
- Could also use a “dataPal”

Enterprise system

Data needed for services + services states
An Application Challenge

What are appropriate models for both:

- Enterprise data and management
- Enterprise services inter-related through persistent data

Must support

- Composition design and analysis
- Runtime service execution management
- Transactions
- Process evolution
Outline

- Application Needs
- “Legacy” Services
- “Programmable” Services
- Data Encapsulating Services
- Research Challenges
- Conclusions
First attempt: services + persistent data + mappings
Verbatim copy of the reality, not much help
Services and data are not intimately related

How to query?

Is it “correct”? 

How to do transactions?

How to change & evole?
Four Kinds of Data

- **Business data** essential for business logic
  - Examples: *items, shipping addresses*

- **Enactment status**: the current execution snapshot
  - Examples: *order sent, shipping request made*

- **Resource usage and state** needed for service execution
  - Examples: *cargo space reserved, truck schedule to be determined*

- **Correlation** between processes instances
  - Example: *3 warehouse fulfillment process instances for Jane’s order*

- Need models that include both control and data
Process Models & Data

Four classes of process models:

- **Data abstraction models**: data mostly absent
  - WF (Petri) nets, BPMN, UML Activity Diagrams, …

- **Data-aware models**: data present (as variables), but storage and management hidden
  - BPEL, YAWL, …

- **Storage-aware models**: schemas for persistent stores, mappings to/from data in BPs defined and managed manually
  - jBPM, …

- **Data encapsulating models**: logical data modeling, automated modeling other 3 types, data-storage mapping
  - Business artifact-centric models
Business Artifacts

- A business artifact is a key conceptual business entity that is used in guiding the operation of the business
  - *fedex package delivery, patient visit, application form, insurance claim, order, financial deal, registration, ...*
  - both “information carrier” and “road-maps”

- Technically, it includes two parts:
  - **Information model:**
    - data needed to move through workflow
  - **Lifecycle:**
    - possible ways to evolve

✓ Very natural to business managers and BP modelers
Example: Restaurant Processes

**Artifacts**
- Guest Check
- Kitchen Order
- Receipt
- Cash Balance

**Activity**
- Create Guest Check
- Add Item
- Pending KOs
- Prepare & Test Quality
- Ready KOs
- Deliver
- Prepare Receipt
- Update Cash Balance
- Open GCs
- Closed GCs
- Pending GCs
- Recalculate Receipt
- Disagreed Receipts
- Payment
- Paid Receipts
- Archived Receipts
- Archived GCs
- Archived KOs
- Cash Balance

**Example:**
Restaurant Processes

2012/11/15
Artifact-Centric Biz Process Models

- Informal model [Nigam-Caswell IBM Sys J 03]
- Systems: BELA (IBM 2005), Siena (IBM 2007), EZ-Flow (ArtiFlow) (Fudan-UCSB 2010), Barcelona (IBM 2010)
- Formal models
  - State machines [Gereede-Bhattacharya-S. SOCA 07][Gereede-S. ICSOC 07]
SeGA: A Service Wrapper/Mediator

- SeGA separates data from execution engine
- Serves as a mediator

SeGA receives incoming events
1. SeGA fetches the correlated BP instances according to the type of the incoming event
2. The dispatcher sends the incoming event, the BP instances, and their schemas to the corresponding engine
3. The engine processes the incoming event, updates the BP instances, and emits outgoing events
4. The dispatcher retrieves the updated BP instances from the engine and stores them back to the repository

Incoming event to SeGA

SeGA

Dispatcher

Event Queue

Repository

Send the event, the process instances, & their schemas to the engine

Fetch correlated instances of the event

Barcelona Engine 1

EZ-Flow Engine n

Barcelona Engine n

Outgoing event

Process the event, update the instances, & emit outgoing events
Data Encapsulating Services

- **Data package between SeGA & services:**
  - Business data, enactment data, resource data, correlation data

- **Data encapsulating services:** Stateful services but the engine need not maintain state

- **Independence of data and service management**
Independence of D-S Management

- Freedom to change service and execution without altering data management
- Freedom to change data management without altering services
- The independence hides the differences between the worlds of *services* and *PALs*
  - Great start for some fascinating research!
Towards a “Flat World” of Services

- SeGA is a first step but an ad hoc prototype more structured methodology needed

- Conceptual level:
  - More types of data? Resource models? Transaction issues?
  - Technical problems—four areas

- System level:
  - Architecture for data encapsulating services?
  - APIs for (non-)functional properties?

- Goal: a unified technical framework for services (biz processes and otherwise)
Outline

- Application Needs
- “Legacy” Services
- “Programmable” Services
- Data Encapsulating Services
- Research Challenges
- Conclusions
Research Challenges

Runtime

Certificate

Composition / design

Add new edu tax

Transactions

Evolution

Reassessment

Tax Calculation

Appraisal

Determine tax base

Inheritance

Title Change

Runtime

age>55 & …
Research Challenges

- **Design**: What are appropriate service designs? Choreography vs orchestration (Part II)? Design aid (analysis/model checking tool), interoperation

- **Runtime**: Enforcement of process/data constraints, KPI/monitoring techniques, resource planning and management

- **Transactions**: What is the notion of workflow transaction?

- **Change/evolution**: Process vs instance changes, long lasting vs temporary, longtail

- **Big data**: monitoring to analytics to change
Choreography For Artifacts

- Participants are processes represented by biz artifacts
  - Partial information model visible
- Correlations between process instances (not just models)
- Data from artifacts used in specifying sequencing constraints
- A fragment of first-order linear time logic

- Detailed in the afternoon session [Sun-Xu-S. ICSOC 12]
Execution Semantics

- Formal model (semantics) for task execution based on Petri nets

- Represents data (input/output) requirements and carries enactments

[Xu-S.-Yan-Yang-Zhang  CoopIS 2011]
Artifact-Centric BPs are Easier to Change

- Biz process = biz artifacts = state machine lifecycle + BP change rules
- BP change rules conservatively extend workflow
  - Could be temporary, non-schematic
- Rules allow biz processes to respond to situations with many more options
- Estimated labor savings:
  - 9% for Hangzhou HMB (preliminary study) or 38 out of 400 FTEs

[Xu-S.-Yan-Yang-Zhang  CoopIS 2011]
Workflow and Data Management

**Integrity constraints (ICs): key in data management**

Many possible ways, our approach: Guard injection

[Liu-S.-Yang CoopIS 11]
Incremental (Runtime) Enforcement

- Logical properties: first order + linear time logic
- Execution snapshot: relational database

[De Masellis-S. 2012]
Research Challenges

- **Design**: What are appropriate service designs? Choreography vs orchestration (Part II)? Design aid (analysis/model checking tool), interoperation

- **Runtime**: Enforcement of process/data constraints, KPI/monitoring techniques, resource planning and management

- **Transactions**: What is the notion of workflow transaction?

- **Change/evolution**: Process vs instance changes, long lasting vs temporary, longtail

- **Big data**: monitoring to analytics to change
Outline

- Application Needs
- “Legacy” Services
- “Programmable” Services
- Data Encapsulating Services
- Research Challenges
- Conclusions
Conclusions

- Inclusion of data is critical to capture business logics into services
  - Data are not just variables; important to remember "persistence"
- Separation of data and service management is a promising approach (e.g., BYOD)
  - DSM independence
- Problems are more difficult, demand creative solutions!
  - Do we have alternatives?

- Scientific principles can and should guide engineering practice, but they don’t have to speak the same buzz words
Acknowledgements

Helpful comments from:

- Richard Hull (IBM)
- Yutian Sun (UCSB)
- Jian Yang (Macquarie U)
References