**Contract-Based Quality of Service (QoS) Monitoring and Control of XML Web Services**

Dr. Vladimir Tasic  
NICTA &  
Univ. of New South Wales  
Sydney, NSW, Australia  
E-mail:  
vladat (server: computer.org)  

Prof. Patrick C.K. Hung  
Univ. of Ontario Institute of Technology - UOIT  
Oshawa, Ontario, Canada  
E-mail:  
Patrick.Hung (server: uoit.ca)  
http://www.cs.ust.hk/~cschk/

**Presentation Outline**

I. Introduction: Importance of QoS for WS  
II. Approaches to specification of QoS for WS  
III. Languages for specification of QoS for WS  
IV. Approaches to management (monitoring and control) of QoS for WS  
V. Research and industrial tools for management of QoS for WS  
VI. Summary: Past results and open issues  
**Answers to questions and discussion**

**Module I: Definition of Terms and Importance of Web Service Quality of Service (QoS)**

**Functionality/service = “WHAT” operations does the system execute?”**  
Example: Returns current price for a stock symbol  
**Quality of service (QoS) = “HOW WELL” the system performs its operations?”**  
Examples: Average response time is 2 seconds, availability in the last 24 hours is 99%, ...  
Synonyms: non-functional, extra-functional, ‘ilities’  
Price and security information sometimes included  
QoS exists even when not specified or measured

**The Need for QoS Information in the Service-Oriented Architecture (SOA)**

1. Publish  
   It differentiates from competitors  
   Service Provider (Supplier, Server)

2. Find  
   It determines the best match  
   Service Registry (Broker, Directory)

3. Bind  
   It is used for management (monitoring and control)  
   Service Consumer (Requester, Client)

Many WSes with similar functionality

**Definition of Management - Monitoring**

**Management = monitoring and control**  
Run-time (and some deployment-time) activities  
**Monitoring** determines state of the system:  
Measurement or calculation of QoS metrics (measures of QoS): response time, availability, ...  
Evaluation of conditions (requirements or guarantees): response time < 2 seconds, ...  
Accounting of invoked operations, consumed resources, measured/calculated QoS metrics, evaluated conditions, taken control actions, billed prices/penalties, ...
**Definition of Management - Control**

- Control tries to ensure that the managed system is always in its desired state:
  - Starting/Stopping the system or its components
  - (Re-)Configuration of the system: setting thread priorities, re-composition of Web services, ...
  - (Re-)Allocation of resources: assigning processing time to requests from different consumers, ...
  - Billing of prices or penalties: penalty for not meeting guaranteed response time is US$1.00, ...
  - Modification of requirements or guarantees
  - Notification of human administrators

**Benefits of QoS Management**

- 5 functional areas of system/network management (FCAPS): Fault, Configuration, Accounting, Performance, and Security
- QoS (performance) management helps to:
  - ensure correct operation,
  - attain or surpass guaranteed QoS,
  - discover and fix problems,
  - accommodate change,
  - balance price/performance ratios,
  - maximize profits, ...

**A Motivating Example**

The Place of QoS in the Stack of Web Service Technologies

- Many additional technologies appeared
- There are disagreements about the contents of the stack of Web service technologies
- Basic Web service technologies (SOAP, WSDL, UDDI, WS-BPEL) do not address QoS specification and management
- Two approaches to adding QoS specification and management to the stack
  1. Crosscutting different layers
  2. Adding a new layer

**A Stack of Basic Web Service Technologies**

- Composition – WSBPEL
  (Business Process Execution Language)
- Discovery – UDDI
  (Universal Description, Discovery, and Integration)
- Description – WSDL
  (Web Services Description Language)
- Messaging – SOAP
- Communication
  (HTTP – Hypertext Transport Protocol, ...)

**An Extended Stack of Web Service Technologies**

- Composition – WSBPEL
- Discovery – UDDI
- QoS Description
- Messaging – SOAP
- Communication
- Management
- Security
**Module I: What Has to Be Developed for WS QoS Specification and Management?**

1. Well-defined (ideally: standardized) formats for specification of WS QoS information
2. Many WS QoS-related algorithms & protocols
   - Very diverse: selection of WSeis using QoS info, negotiation of QoS of a WS, monitoring of QoS, exchange of run-time QoS info, control to achieve QoS guarantees, adaptation to changes in QoS, ...
   - We will discuss them in the context of specification formats and/or management tools
3. WS QoS management infrastructures/tools

**Module II: Overview of Approaches to WS QoS Specification**

- QoS specification = description of what/where/when/how to monitor & control
- QoS info = descriptions & monitored values

**Classification of QoS specification approaches:**

1. Implicit – built into the implementation (not flexible)
2. Contracts – formal agreements (for QoS, billing, ...)
   - Service Level Agreements (SLAs)
   - Classes of service – a special type of SLAs
3. Policies – high-level operation & management goals and/or rules (for security, QoS, billing, ...)

---

**Module I: Contract**

- Contract = binding and enforceable formal agreement between two or more parties
- Defines requirements & guarantees of parties
  - Can be used in monitoring and control
- Contracts enable not only QoS description, but also QoS differentiation
  - Different consumers can have different contracts
- Apart from QoS info, a contract can contain other information (e.g., prices/penalties)
  - A WSDL file is a contract

**Module II: Specification of QoS in Extended WSDL, UDDI, or WS-BPEL Files**

- Strengths:
  - The extensions can be relatively simple
  - QoS discovery related to Web service discovery
- Weaknesses:
  - QoS specification language tied to WSDL (UDDI, WS-BPEL) in terms of tools, evolution, ...
  - Extension mechanisms are limited
  - Run-time change of QoS information requires updates of all affected copies of WSDL (UDDI, WS-BPEL) files, which is complicated

---

**Module I: Service Level Agreement (SLA)**

- A special type of contract for QoS (and often price/penalty) requirements & guarantees
- Many different formats, one of them is:
  - Parties (including supporting management parties)
  - Service description
    - Service operations – describe available operations
    - SLA parameters – define monitoring of QoS metrics
  - Obligations
    - Service Level Objectives (SLOs) - QoS guarantees
    - Action guarantees - specify what happens if SLOs are met or not met

**Module II: A Simple Example of an SLA**

- Parties: consumer C and provider P
- Service operations: P has one operation (OP1) float getStockPrice(String stockName)
- SLA parameters: (RT-OP1-C) Response time of operation OP1 measured at consumer C by consumer C
- SLOs: (SLO1) For every OP1 invocation by C, RT-OP1-C will be less than or equal to 2 seconds
- Action guarantees: (AG1) If SLO1 was met, C pays P price of US$0.20 per invocation;
  - (AG2) If SLO1 was not met, P pays C penalty of US$0.10 per invocation
Service Level Agreement (SLA) - Strength and Weaknesses

- **Strengths:**
  - Formal contract specification of QoS and related management aspects
  - Widely used in computing and communications systems (and now also for WSes)

- **Weaknesses:**
  - Negotiation of custom-made SLAs can require complex analysis of offers and generation of counter-offers (can be alleviated by using templates)
  - Management of many concurrent custom-made SLAs can be complex & with high run-time overhead
  - Cannot be used for QoS-enabled WS selection

Class of Service

- A special type of SLA that is not custom-made, but predefined & reusable (anonymous)
  - 1 provider can offer many classes of service that refer to the same functionality, but differ in QoS
  - 1 class of service can be used by many consumers
  - Simple selection instead of complex negotiation
  - Classes of service already checked for consistency
  - **Strengths:** Usable for QoS-enabled WS selection, no complex negotiation, simpler management, lower run-time overhead, faster adaptation
  - **Weakness:** Discrete differentiation - limited choice

Module II (Approaches to QoS Specification) - Summary & Discussion

- **Contract** = binding and enforceable formal agreement between two or more parties
  - SLA is a special type of contract; many SLA formats
  - Class of service is a type of a light, predefined SLA

- **Which one to use depends on circumstances**
  - For comprehensiveness: general contracts
  - For flexibility of QoS specification: custom-made SLAs
  - For low overhead: classes of service

- **Contracts vs. policies:** similar information, different management architectures
  - Contract pluses: negotiation, use in WS selection, ...

Module III: Overview of XML Languages for WS QoS Specification

- Classified based on the main concept:
  1. SLA: WSLA, [SahaiEtal2002] (WSML), SLAng
  2. Class of service: WSOL, WS-QoS, DAML-QoS
  3. General contract: WS-Agreement, OWL-S
  5. Policy: WS-Policy, WS-CoL, WS-Policy4MASC
  6. Manageability capability: WSDM

- There are many other languages and formats, but these are probably best representatives

Why Not Reusing QoS Specification Languages from Other Areas?

- Many existing QoS specification languages in multimedia (HQML, ...), distributed objects (QML, QDL, QIDL, ...), and other areas

- Can not be directly re-used because of:
  - Incompatibility with WS standards (e.g., WSDL)
  - Heterogeneity of WS implementations and interactions styles used (asynchronous & synchronous; document-based & RPC; ...)

  Characteristics of WS compositions: business-to-business, Internet scale, dynamism, automatism, ...

Web Service Level Agreement (WSLA) - Overview

from IBM Research: H. Ludwig, A. Keller, A. Dan, ...

- QoS language & management infrastructure
  - Compatible with, but not restricted to WSes

- Custom-made SLAs that have 3 parts:
  1. Parties (signatory & supporting) and their management operations
  2. Service definitions – service objects (e.g., WSDL operations) and their monitored properties (QoS metrics, SLA parameters, schedules, triggers, ...)
  3. Obligations – SLOs and action guarantees
Web Service Level Agreement (WSLA) - Language Details

- **SLA parameter** - monitored property; contains 1 QoS metric & extra info for exchange of values
- **QoS metric** - defines where & how to measure or calculate; can be reused across SLA parameters
- An **SLO contains** evaluation Boolean expression (limits values of SLA parameters), obliged party, validity periods, evaluation event or schedule
- An **action guarantee** contains: precondition expression, evaluation event or schedule, action to be taken, obliged party, execution modality
- **Reusability**: SLA templates, metric macros, ...

---

Web Service Level Agreement (WSLA) - Strengths and Weaknesses

- **Strengths** (significantly outweigh weaknesses):
  - Detailed and precise description of QoS monitoring and control
  - Several tools for SLA creation, deployment, and compliance monitoring – were distributed with IBM's Emerging Technologies Toolkit (ETTK)
  - Most influential WS QoS specification language
- **Weaknesses** (mainly due to custom-made SLAs):
  - QoS metrics defined within SLAs
  - Can not be used for QoS-enabled WS selection
  - High overhead of supporting custom-made SLAs

---

Presentation Progress

- I. Introduction: Importance of QoS for WS
- II. Approaches to specification of QoS for WS
- III. Languages for specification of QoS for WS
- IV. => Approaches to management (monitoring and control) of QoS for WS
- V. Research and industrial tools for management of QoS for WS
- VI. Summary: Past results and open issues
- => Answers to questions and discussion

---

Module III (Languages for QoS Specification) - Summary & Discussion

- There are many languages; very different
  - Most are based on contracts, particularly SLAs
  - We presented several important ones, but emphasize:
    - WS-Agreement & WS-Policy as possible future general frameworks for WS QoS specification
    - Have industry support, but the "meat" is missing
    - WSLA as precise and detailed contract-based QoS specification language used in practice
    - Its solutions could be used (along with some ideas from other languages, e.g., WSOL) for the "meat"
Using Monitoring Instrumentation Only on the Provider Side

- **Invasive vs. non-invasive instrumentation**
  - Invasive: within business logic code of a WS vs. within WS hosting tools (e.g., within SOAP engine)
- **All monitoring on the provider side**
- **Strengths:** Realistic & consumer-specific measures; independent from network location of measurement
- **Weaknesses:** Provider must have capabilities and willingness; consumers must trust the provider

Some Instrumentation Technologies

- **Industry standards:**
  - Simple Network Management Protocol (SNMP)
  - Application Response Measurement (ARM)
  - Java Management Extensions (JMX)
  - Windows Management Instrumentation (WMI)
- **Other instrumentation approaches:**
  - Addition of composable SOAP message processing filters to WS hosting tools (e.g., a SOAP engine)
  - Aspect-oriented code weaving (mostly research)
  - Mobile agents (mostly research)
- **Most can be used not only on providers, but also on consumers and third parties**

Using SOAP Message Intermediaries

- **Exchange of monitored values:** a) in SOAP headers; b) using special push or pull operations
- **Strengths:** Realistic & consumer-specific measures
- **Weaknesses:** High run-time overhead (can be reduced with periodic/occasional monitoring); results depend on network location of measurement

Using Probes (Probing)

- **Strengths:** Run-time overhead can be lower
- **Weaknesses:** Results not consumer-specific, provider can treat probes in a special way; not possible to re-use SOAP headers to send monitored values; results depend on network location of probes

Using Sniffers (Sniffing)

- **Strengths:** Very low run-time overhead; measures can be realistic & consumer-specific
- **Weaknesses:** Unknown SOAP message’s Internet route; WS security technologies can be a problem; not possible to use SOAP headers to send monitored values; results depend on network location of sniffers

Several Control Approaches That Try to Meet QoS Guarantees

1. Manipulate which request is processed first
   - Provider has several different request queues, e.g., one for each class of service
   - Scheduler within the provider decides from which queue to process a request, depending on QoS guarantees, current load, queue lengths, ...
2. Manipulate thread priorities for different requests and/or OS scheduling discipline
3. General approach: Manipulate allocation of resources for various requests
4. Load balancing between replicas
**Re-composition of Web Services vs. Re-negotiation of Contracts**

- Run-time adaptation of WS compositions
  - a) Re-composition of Web services – more powerful
    - Special case: Switching between WSes (only 1 change)
  - b) Re-negotiation of contracts – faster, simpler, lighter
    - Special case: Switching between classes of service

Legend: C – consumer; P – provider; CS – class of service

![Diagram showing re-composition and re-negotiation of contracts]

**Using Historical QoS Information for WS Selection - Possible Approaches**

- From the same consumer
  - Problem: When consumer did not previously invoke this operation of the provider Web service
- From probes
  - Problem: Easy for providers to give excellent QoS to probes, while bad QoS to real consumers
- From all consumers
  - Problem: Consumers have different characteristics (e.g., could be located on different continents)
  - Problem: Other consumers’ reports can be fake

**Using Historical QoS Information for WS Selection - Discussion**

- General problem: Circumstances of different invocations are different!
  - Example: When the number of provider’s concurrent consumers grows, it is likely that QoS perceived by individual consumers will drop
- General problem: Absence of targets/goals to guide control activities (including billing)
- Conclusion: Historical QoS information can be useful, but it provides no guarantees (and can even be misleading) => contracts are needed

**Approaches to QoS Discovery and Selection**

1. Provider as only source of QoS specifications
2. UDDI extensions with QoS information
3. Additional QoS information registry

- The main issues:
  - QoS publication in a registry enables QoS-based selection (selection is difficult with option 1)
  - QoS changes much more often than WSDL, so QoS updates have to be propagated to consumers (difficult with option 2)
  - Complexity (highest with option 3)

**QoS of WS Compositions**

- Given a set of WSes with known QoS, what is the QoS of their composition?
  - Very difficult question, without a general answer!
  - It is not a set of simple mathematical operations, because distribution of probability varies
  - Queuing methods can be used for analysis
  - How to select QoS of an individual WS to satisfy overall QoS of a known composition?

**Monitoring and Control of Business QoS**

- WS compositions realize business processes
  - Technical QoS (e.g., response time, availability)
  - Business QoS (e.g., profit, customer satisfaction)
- Financial business metrics have been successfully monitored through accounting
  - E.g., prices and penalties in SLAs
- Non-financial business metrics are difficult to monetize and difficult to monitor/control
  - Business strategy determines which business QoS metric to maximise (it is not always profit)
Module IV: Approaches to QoS Management - Summary & Discussion

- Several different approaches to QoS monitoring, selection, and control
  - None is best for all circumstances – knowing their advantages/disadvantages will help you choose
  - In B2B scenarios, QoS monitoring with SOAP message intermediaries seems most flexible
  - QoS control by (re-)allocating resources to meet QoS guarantees is necessary, but hard
  - For QoS-aware WS selection, contracts have significant advantages over historical QoS data

Module V: Research and Industrial Tools for WS QoS Management

- Published as research work in refereed papers
  - Contract-based QoS management: WSLA, WSMF, WSOI, WS-QoS, Cremona, ...
  - Policy-based QoS management: Dynamo, MASC, ...
  - QoS-based WS discovery: UDDI, Ux, ...
  - Other: Smartware, ... and many others
- Commercial products (non-refereed literature)
  - WS QoS management products from specialized companies
  - Large systems management suites
  - Related management products

Web Service Level Agreement (WSLA) Framework

- From IBM Research; uses WSLA language (slides 27-29)
  - Prototype: SLA Compliance Monitor – module 1 is simple, 2 is implemented, 3 & 4 are general purpose, 5 & 6 missing
  - Service Deployment Information (SDI) is a subset of WSLA
  - Special management port types (e.g., for value exchange)
- Strengths: Comprehensive approach to QoS management; support for management third parties; was a widely used research infrastructure
- Weaknesses: Run-time overhead

Smartware

- From Infosys: A. Sharma, H. Adarkar, S. Sengupta
  - QoS control: Differentiated scheduling of requests based on context priorities
    - Context = info about provider application, user, and client device; sent by consumer in request SOAP header
  - Based on Apache Axis SOAP engine, adds:
    - Interceptor – reads context info and determines priority
    - Scheduler – puts request into a queue for its priority; based on scheduling policy fetches a request from a queue
    - Dispatcher – forwards request to the provider
  - Strengths: Rare work that performs QoS control
  - Weaknesses: Scheduling uses limited information

Web Service Level Agreement (WSLA) Framework – Run-Time Use

Some Observations about Industrial Products for WS QoS Management

- They address many practical problems
  - Academic researchers should be aware of these works and their accomplishments
  - Some works contain advanced solutions that show how SLAs and/or policies can be used in practice
- Many products have significant limitations:
  - Crucial role of human administrators (i.e., not completely automated)
  - Limited/predefined choice of used QoS metrics
  - Lack of flexible formal machine-understandable QoS specification (instead, forms are used)
Some Products for WS QoS Management from Specialized Companies
- Often products (1 or more) addressing several management areas, incl. performance (QoS)
- Actional SOA Management (including SOAPStation Web Services Broker) – policies
- AmberPoint (including Service Level Manager) – custom-made SLAs
- SOA Software (incl. Service Manager) – policies
- Web Methods (incl. Infravio) – custom-made SLAs
- WestGlobal mScape (including Performance Management Module - PMM) – custom-made SLAs

Large System Management Suites
- Contain many different management products
  - some related to WSes (or "business services")
  - some related to performance (QoS) management of applications, computing systems, networks
- HP: OpenView (includes SOA Manager)
- IBM: Tivoli (includes Business Systems Manager)
- Computer Associates (CA): Unicenter (includes Web Services Distributed Management – WSDM)
- BMC Software: Patrol (includes MAINVIEW)
- Microsoft (includes Application Center)

Module V (Tools for QoS Management) - Summary & Discussion
- There are many tools; very different in power
  - Most are based on SLAs, some are based on policies
- No current tool or a set of tools addresses all QoS monitoring, selection, and control needs!
- Industrial products address many basic issues, but have limitations (e.g., in QoS control)
  - Powerful, but expensive: system management suites
- Research tools tackle advanced problems, but commercial use requires additional features
- Upcoming: SOA business process management (BPM)

Module VI: Summary of QoS Specification and Management for Web Services
- QoS & manageability (also price, security, trust, ...) will be differentiators in the WS market
- Determine WS to be chosen among similar ones
- QoS specification is the basis for management
- Management (monitoring and control) is necessary to meet QoS guarantees, discover and fix problems, accommodate change, ...
- The vision of service-oriented computing can not be achieved without (QoS) management
- But, the basic Web service technologies do not address QoS specification and management

The Main Achieved Results
- Several ...
  - ... languages for precise and detailed specification of contracts, SLAs, classes of service for Web services
  - ... general frameworks that can be extended for QoS specification (WS-Agreement, WS-Policy, OWL-S)
  - ... industrial standards (WSQm, WSDM, WS-Reliability, ...)
  - ... prototyped solutions for QoS-enabled Web service discovery/selection
  - ... research infrastructures that enable WS QoS management (predominantly monitoring)
  - ... industrial products used in practice

Some Open Research Topics
- Control of Web Services to meet guarantees
  - The main area for near-future research (on the contrary, monitoring is mostly well-researched)
  - Resource capacity planning and management
  - Building complex control plans
- Solutions for adaptation to various changes
- Integrated management of business operations, Web services, and underlying computing/communication infrastructure
  - Standard models of operation & issues at different levels and mappings between them are needed