

The role of Semantics in Web service Composition

Monika Solanki

`monika@dmu.ac.uk`

Software Technology Research Laboratory
De Montfort University, U.K.

September 6, 2006

Broad Talk Outline

- Service Oriented Systems
- Service Composition
- Semantics: Requirements, Value Additions and Issues
- Semantic Web Service Composition: State of the art
- Concluding Questions → Future Work

Broad Talk Outline

- Service Oriented Systems
- Service Composition
- Semantics: Requirements, Value Additions and Issues
- Semantic Web Service Composition: State of the art
- Concluding Questions → Future Work

Good News:

- *No “< ***** >” tags in this talk!!!!*
- *No “Tutorial” on any of the “standards”*

The "Grand" Vision: Industry Driven

The Gartner
Group predicts

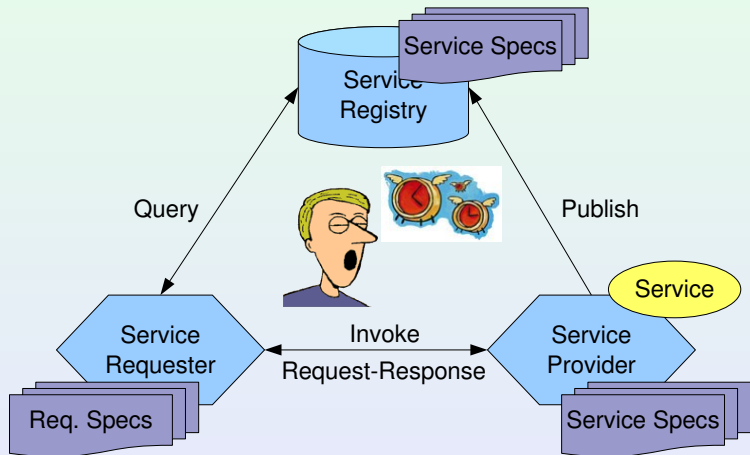
By 2008, SOA will be a prevailing software engineering practice, ending the 40-year domination of monolithic software architecture.

The "Grand" Vision: Industry Driven

Real world business models of arbitrary complexity

- Pay-as-you-go-Computing.
- Software as a Service.
- Replace / encapsulate proprietary solutions.
- Fully Automated: Service Discovery and Invocation.
- On-the-fly **Service Integration**.
- Does not overly constrain service implementation.
- **Loosely coupled** Distributed Applications
- Interoperable within and across enterprises.
- Keen interest shown by Industry (MS, Google, Sun..), Academia, Press, Standards organisation (W3C, OASIS...)
- Application Domains: e-commerce, health care, travel industry, e-government, e-learning, telecom, GRID...

The (Done to Death!!) Architecture



...

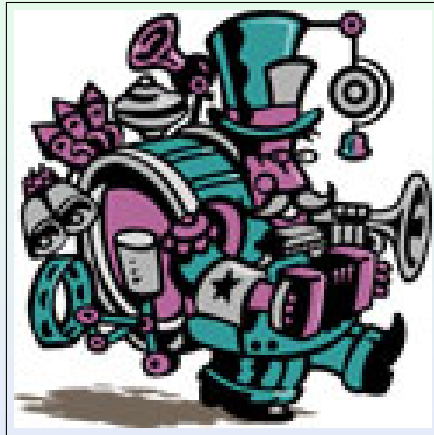
One Realisation of SOA: Web services

With influence from the W3C definition:

- A **service** is an abstract notion realised by a concrete implementation (**agent**)
- Identified by a URI
- Provides a **programmatic** accessible Interface
- **Published, Discovered, Invoked, Composed** over a variety of networks, heterogeneous platforms...blah blah...
- Standard Web/Internet Protocols
- XML+WSDL+SOAP+UDDI



Service Composition



What is Composition Really

Interestingly the W3C Web service Glossary does not define “Composition”!!!

Atleast three aspects

- Putting together Lego pieces: Combining (smaller) services into a (larger) service
- Maintaining the order in which services need to be invoked (Synthesis)
- Ensuring the correctness of the composite service with respect to predefined goals

Our Working Definition

Technique of bringing together services in an orderly and provably correct manner to create new services that satisfy desired goals.

What is Composition Really

Interestingly the W3C Web service Glossary does not define “Composition”!!!

At least three aspects

- Putting together Lego pieces: Combining (smaller) services into a (larger) service
- Maintaining the order in which services need to be invoked (Synthesis)
- Ensuring the correctness of the composite service with respect to predefined goals

Our Working Definition

Technique of bringing together services in an orderly and provably correct manner to create new services that satisfy desired goals.

Realising a Composition

- Specification: Goals, Capabilities, Constraints
- Synthesis: Schema, Plan
- Analysis: Verification, Testing
- Invocation: isolate enactments, provide transactional guarantees
- Monitoring: Runtime Validation, tracing enactments, recovery from failure

Mediation may be required at any/all level(s)

Realising a Composition

- Specification: Goals, Capabilities, Constraints
- Synthesis: Schema, Plan
- Analysis: Verification, Testing
- Invocation: isolate enactments, provide transactional guarantees
- Monitoring: Runtime Validation, tracing enactments, recovery from failure

Mediation may be required at any/all level(s)

- Provider perspective: Bottom-up approach (composing capabilities)
- Requester perspective: Top-down approach (decomposing goals)
- Two topologies: peer-to-peer / Hub-and-spoke

Realising a Composition

- Specification: Goals, Capabilities, Constraints
- Synthesis: Schema, Plan
- Analysis: Verification, Testing
- Invocation: isolate enactments, provide transactional guarantees
- Monitoring: Runtime Validation, tracing enactments, recovery from failure

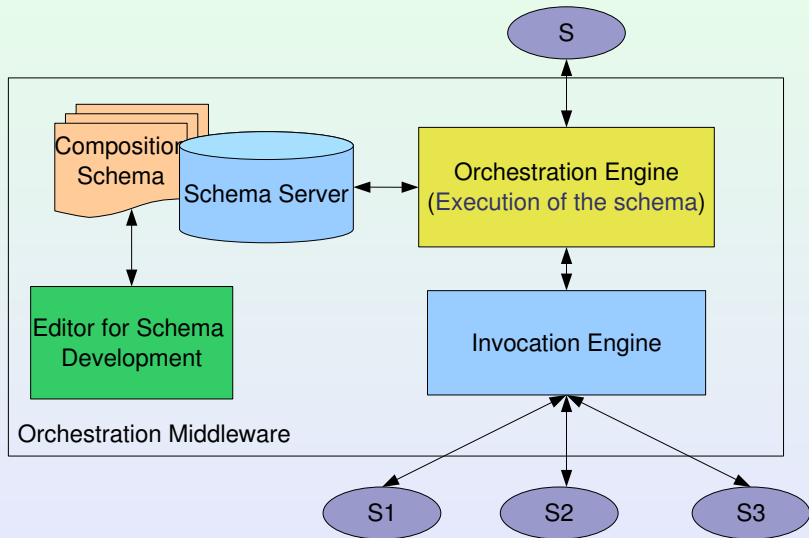
Mediation may be required at any/all level(s)

- Provider perspective: Bottom-up approach (composing capabilities)
- Requester perspective: Top-down approach (decomposing goals)
- Two topologies: peer-to-peer / Hub-and-spoke

Orchestration

- Works at the **Agent** Level (recall W3C definition!!)
- Orchestration \supset Composition (for most people !!!)
- Creation (**Implementation**) of a new service from component services
- Follows a pre-defined Composition schema
- Procedural flavour
- Central controller (Hub and Spoke model)
- Invocation, Scheduling
- Monitoring the execution
- Schema: Workflow languages (executable BPEL4WS), Transition systems

Orchestration

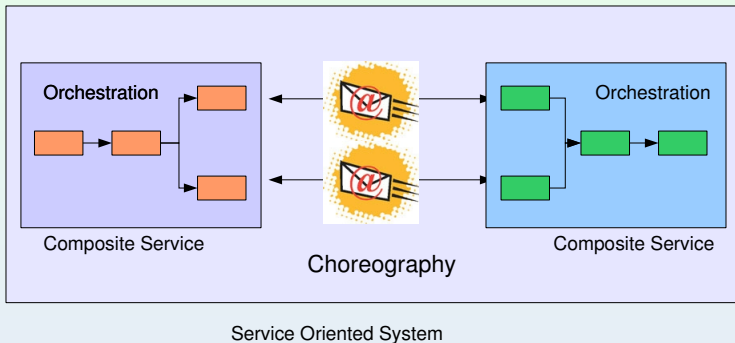


Choreography

- Works at the **Service** level (recall W3C definition!!)
- Choreography \supset Multi party Collaboration
- Global specification of conversation and constraints on them
- Declarative flavour
- Observable message exchange
- No central controller
- Implementation independent
- Monitoring of Conversation

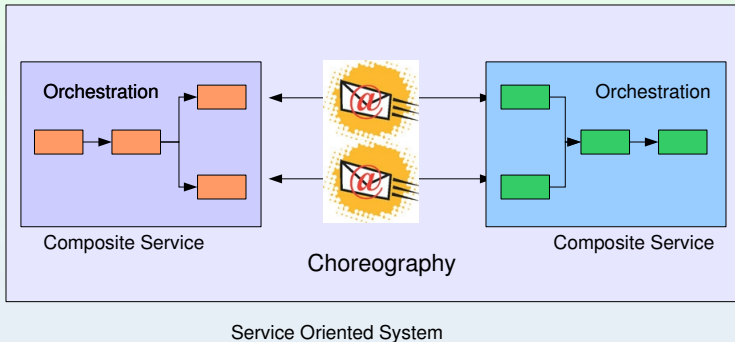
BPEL4WS (abstract), WSCDL (**W3C**)

The Global Picture



Or the other way around???

The Global Picture



Or the other way around???

Automated Composition and Analysis

Analysis: Verification(Model Checking), Runtime Validation

- Crucial for Stateful services (long running processes and their composition)
- Verify service commitments, check for safety and liveness properties (deadlock, livelock, starvation, reachability.....)
- Automata-theoretic approaches, Workflow, AI techniques

Automated Composition and Analysis

Analysis: Verification(Model Checking), Runtime Validation

- Crucial for Stateful services (long running processes and their composition)
- Verify service commitments, check for safety and liveness properties (deadlock, livelock, starvation, reachability.....)
- Automata-theoretic approaches, Workflow, AI techniques

Automated Composition:

Mealy machines	FSM	Hull's group
Activity based	FSM based	Berardi's group
Compositional specification	Temporal logic (ITL)	Solanki's group

Automated Composition and Analysis

Analysis: Verification(Model Checking), Runtime Validation

- Crucial for Stateful services (long running processes and their composition)
- Verify service commitments, check for safety and liveness properties (deadlock, livelock, starvation, reachability.....)
- Automata-theoretic approaches, Workflow, AI techniques

Analysis

Static Model Checking	FSMs Message Sequence Charts Process Algebra	Bultan's group Foster's group Breugel's group
Runtime monitoring	Tempura	Solanki's group

Why is Composition Challenging ?

- **Service granularity:** Coarse vs. Fine grained
- **Visibility:** Black box, Grey box, Glass box
 - No explicit specification
 - Most approaches assume a Glass Box view

- **Synthesis**

Given -

- a set of services S_1, S_2, \dots, S_n
- a goal G

Automatically (manually) build a composition schema (plan) from S_1, S_2, \dots, S_n that satisfies G

- **Ensuring Compositionality:** System is the sum of its component parts
- **Avoiding/Detecting Emergent Behaviour:** System is more than the sum of its component parts

...more to come !!!!!...



Why is Composition Challenging ?

- non-determinism of services, partial observability of the internal states and complex goals.
- **degree of automation of the process model**: manual, interactive, semi-automated, automated
- **binding to concrete services**: static, semi-dynamic, dynamic, adaptive, autonomic.....????
- **Dynamism**: Is there any consensus on the term ?
- **incomplete specification/knowledge**: Web services have only partially kept their promise of being self contained and self describing applications.
- **constantly changing information**
- schema mapping to facilitate interoperability

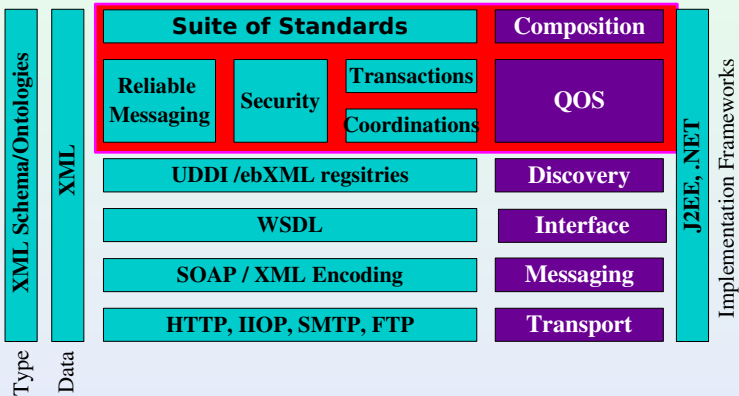
...not done yet!!!!.....

Why is Composition Challenging ?

- message level heterogeneity
- data mediation /process mediation
- data integration/process integration
- Specification of **temporal** requirements
- Support Interactions amongst services
 - Coordinate the asynchronous communication between services
 - Synchronise parallel and independent activities
 - Transactional support for long running interactions
- conformance and consistency rules to ensure that no run-time errors result from the combination of services.
- QoS, Security, scalability, reliability, re-usability.....???

...list is incomplete!!!!....

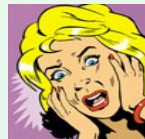
The (infamous) Technology Stack !!!



Standards: How many do we need?



XML
WSDL SOAP
UDDI



BPEL4WS
WS-Coordination
WS-Policy
WS-Interoperability
OWL-S
WSM**
SAWSDL
WS-Security Policy
WSCDL
WSRF
WS-Transaction

Semantic (Web services)

Or

Semantic Web (Services)

Chalk and Cheese

Semantic (Web services)

Or

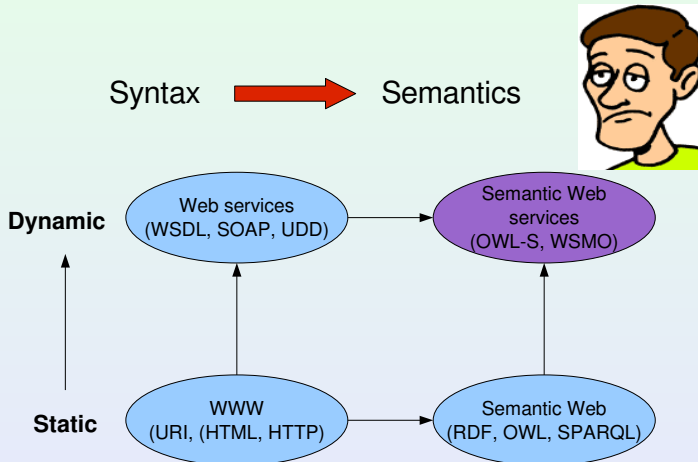
Semantic Web (Services)

The “Grand” Vision: Academia Driven

The W3C
predicts

The Semantic Web will provide an infrastructure that enables not just web pages, but databases, services, programs, sensors, personal devices, and even household appliances to both consume and produce data on the web.

The “Grand” Vision: Academia Driven



Semantics: Requirements / Value Additions

- Automation of Web service tasks: discovery, selection, composition, negotiation and contracting, invocation, monitoring of progress, and recovery from failure
- Semantic Annotation → Automation and ad hoc computing for services
- “Intelligent” Web services
- Mediation: **Data, Ontology, Process, Protocol**
- Sound underlying Formal models at two levels
 - Automated reasoning (e.g. subsumption)
 - Behavioural Semantics

Semantics: Requirements/Value Addition

- Ontologies as data models: Basic Building Blocks
- An ontology provides semantic grounding. It includes a vocabulary of terms, and some specification of their meaning.
- Provides a formal description of the meaning of data exchanged and processes in services
- Enables reasoning tasks to be carried out by agents
- Mapping concepts in a Web service description to ontological concepts

Semantics: Issues

- Ontologies
 - Expressiveness, granularity
 - Domain model: large or narrow
 - mapping, alignment: same language, across languages
 - mediation
- Representation of Rules, Policies, Constraints: Precondition, Effect, Assumption, Commitment
- Are the Semantics compositional?
- What part does Annotation play in Analysis and Monitoring?
- Is it just an encoding or does it represent true semantics ?

Semantic Web services: Standards and Initiatives

- OWL-S: Web Ontology Language for Services (W3C)
- WSM*: Web service Modelling ((O)ntology, (L)anguage, E(X)ecution)(W3C)
- SWSI: Semantic Web services Initiative (SWSL, SWSO)
- SAWSDL: Semantic Annotations for WSDL (W3C)

Comparative Analysis of SWS Initiatives

(Limited) set of Criteria for Comparison: **Composition**

- Top Level Concepts
- Interface Properties and their specification
- Process/Service Modelling
- Behavioural Semantics
- Constraint specification
- Reasoning support
- Mediation

Comparative Analysis: OWL-S

Leverages on OWL

- **Top Level Concept:** Process Model, Grounding (Invocation)
- **Interface Properties:** Inputs, Outputs, Preconditions, Results (Condition, Constraints on output, Effects)
- **Process Modelling:** Atomic, **Composite**, Simple
 - Control Flow, Data Flow
 - **Orchestration**
- **Behavioural Semantics:** Situation calculus, Petri Nets, Concurrent Execution Semantics
- **Constraint Specification:** SWRL, SWRL-FOL, KIF, DRS
- **Reasoning support:** DL reasoning
- **Mediation:** No “explicit” support

WSM* Approach

Leverages on the Web service Modelling Framework (WSMF).
WSML is the underlying ontology language

- **Top Level Concepts:** Ontologies, Goals, Mediators, Web services
- **Interface Properties/Capabilities:** NonFunctional, Preconditions, Postconditions, Effects, Assumptions, Shared Variables
- **Process Modelling:** Support for both Orchestration, Choreography
- **Behavioural Semantics:** ASMs
- **Constraint Specification:** one of the variants of WSML (WSML-core, WSML-DL, WSML-Flight)
- **Reasoning support:** DL reasoning, Logic programming
- **Mediation:** “explicit” support

Comparative Analysis: SAWSDL

Leverages on WSDL

- **Top Level Concept**: types, interfaces, operations
- **Interface Properties**: interfaces, operations
- **Process Modelling**: extension attribute “modelReference” to specify the association between a WSDL component and a concept in some semantic model operations (mapping operations to processes in OWL-S or Choreo/Orc in WSMO)
- **Behavioural Semantics**: defined by ‘modelReference’
- **Constraint Specification**: No “explicit” support (Annotate operations with PE from OWL-S/WSMO)
- **Reasoning support**: No “explicit” support, depends on the “modelReference”
- **Mediation**: No “explicit” support

AI Planning: Synthesis of Complex behaviours

Classical Planning Problem

“Given a set of actions, their preconditions and positive and negative effects, a complete description of the initial state and a user goal find a sequence of actions achieving the goal”

Planning Domain can be described in terms of a five-tuple,

$$\{S, S_i, S_g, A, T\}$$

- S is the set of all possible states of the world.
- $S_i \subseteq S$ is the set of all initial states.
- $S_g \subseteq S$ is the set of all goal states.
- A is the set of actions.
- T is the transition function.

AI Planning

Approach	Research Group
STRIPS-style	Finin's group (US)
Golog, Situation calculus	McIlraith's group
HTN Planning (SHOP2)	Hendler's group (US)
Estimated Regression Planning (Optop)	McDermott's group (US)
Planning as Model checking	Pistore's group (Europe)

Other techniques

- Workflow based approaches
- Logic based approaches
- Case based reasoning

AI Planning

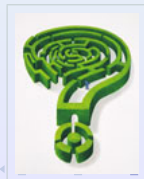
Approach	Research Group
STRIPS-style	Finin's group (US)
Golog, Situation calculus	McIlraith's group
HTN Planning (SHOP2)	Hendler's group (US)
Estimated Regression	McDermott's group (US)
Planning (Optop)	
Planning as Model checking	Pistore's group (Europe)

Other techniques

- Workflow based approaches
- Logic based approaches
- Case based reasoning

Concluding Questions

- Is Composition a means for managing complexity or introducing complexity ?
- No common definition, each approach to composition has its own definition and assumptions
- Can you compare different approaches?
- Can you unify them ?
- What about interoperability (Are we back to square one??)
- Semantics: To what extent do we need Automation (completely unuser-friendly process models)
- Do we need a standard for the development of all these standards?



Concluding Questions

- Is Composition a means for managing complexity or introducing complexity ?
- No common definition, each approach to composition has its own definition and assumptions
- Can you compare different approaches?
- Can you unify them ?
- What about interoperability (Are we back to square one??)
- Semantics: To what extent do we need Automation (completely unuser-friendly process models)
- Do we need a standard for the development of all these standards?

