ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

Εισαγωγή στα Δίκτυα Υπηρεσιών

Διάλεξη 18η: Web Service Composition-A Survey On Existing Approaches

Χρήστος Νικολάου
Τμήμα Επιστήμης Υπολογιστών

Baryannis George
Filiopoulou Irene
Kartsonakis Efthimis
Tesseris George

Editing: C. Nikolaou

University of
Crete, CS 593

Introduction (1/3)

- Problem Statement
  - how to effectively and efficiently compose existing services with minimum user intervention, in order to address needs that cannot be satisfied by a sole (atomic) service
- Generally referred to in the literature as Web service composition when examined with focus on the World Wide Web

Introduction (3/3)

- Composition approaches differ on **when** and **how** the composition schema is created
- **When**
  - **Static or design-time** composition: services are chosen, linked and compiled before runtime, schema does not change
  - **Dynamic or run-time** composition: adapt the composition schema to reflect unpredictable changes and effects at runtime
- **How**
  - **Manual vs. automated** composition

Introduction (2/3)

- **Aims** to combine the functionalities offered by single Web services to develop a value-added composite service
- **Allows** for the definition of increasingly complex applications by progressively combining services at increasing levels of abstraction
- **Accelerates** rapid application development, service reuse and complex service consummation

Service Composition Models

- Orchestration
- Choreography
- Coordination
- Service Component Architecture

Composition Models: Orchestration (1/2)

- Organizes participating services into a **process flow**
- Describes the **interaction** between the participants as well as the **control** and **data flow** between them
- Services can be orchestrated recursively
- WS-BPEL has been generally adopted as a standard orchestration language

Composition Models: Orchestration (2/2)

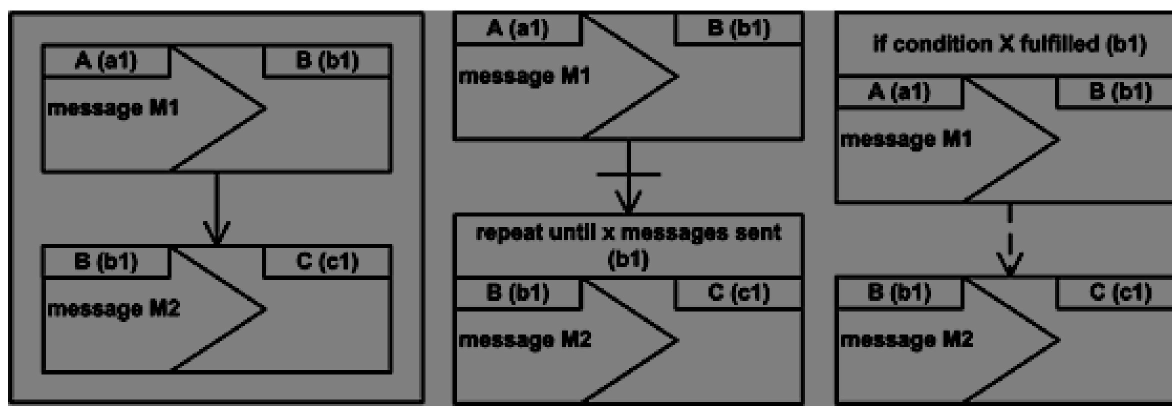
- BPEL
  - XML language to create Web service flows, specifying the control and data flow between participating services
  - Provides constructs for
    - Web service invocation and messaging
    - data manipulation
    - basic control constructs,
    - fault handling, exceptions and compensation.
  - Not necessary to identify an exact endpoint for each service
    - An abstract interface describing the necessary inputs and outputs is enough.

Composition Models: Choreography (1/2)

- **Describes** collaborations between multiple partners of the same organization or of multiple organizations
- **Focuses** only on the conversational aspect of the interaction
  - what messages are exchanged between partners
- **Does not aim** to expose what's underneath, i.e. the internal processes of the partners
- Choreography can be combined with orchestration
  - orchestrations used for the description of the internal business processes of partners
  - choreography used to describe the global external interactions between partners.

Composition Models: Choreography (2/2)

- Let's Dance
  - visual modeling language
  - uses constructs that describe atomic message exchanges to represent the conversations between partners
- 3 conversation types (capitals are Roles, lower case are actors:
  - Precedes (after the
    - receipt of message M1 by
    - B, B sends M2)
  - Inhibits (after the
    - receipt of message M1 by
    - B, B cannot send M2, x times)
  - Weak-precedes (B
    - cannot send M2 until
    - A sends M1 or A
    - is inhibited by condition guarded by b1)



Composition Models: Coordination (1/2)

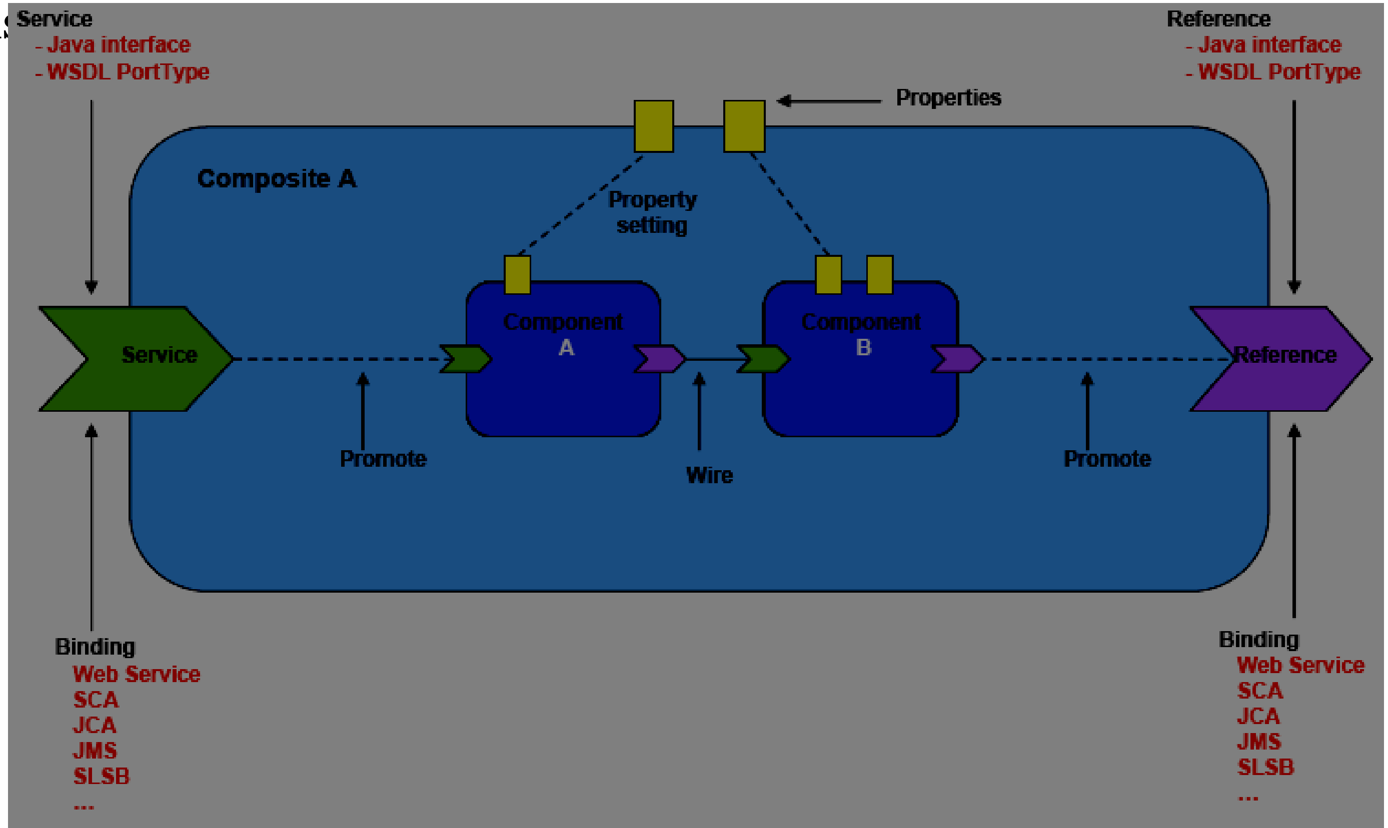
- A group of participating services interacts following a **coordination protocol**
- Special type of participant: **coordinator**
  - makes sure the coordination protocol is followed
  - decides on the outcome of the interaction after it is finished
- Participants don't have to communicate with each other (as in choreography)
  - the coordinator is used as a communication nexus instead

Composition Models: Coordination (2/2)

- WS-Coordination (IBM/Microsoft) vs. WS-CF (Sun/Oracle)
  - Context creation and management
    - WS-CF: built on top of WS-CTX, which defines a Context Service, responsible for creating and managing contexts and interfaces to access them
    - WS-Coordination
      - defines both generic coordination mechanisms and context creation and management together
      - The Coordinator is one entity, responsible for all of the above, not separate entities
  - Context querying
    - WS-CF: message exchanges to query the content of a context or the state of the coordination defined through WS-CTX
    - WS-Coordination: Each coordination protocol must define them separately
- WS-CAF (WS-CTX + WS-CF) a superset of WS-Coordination

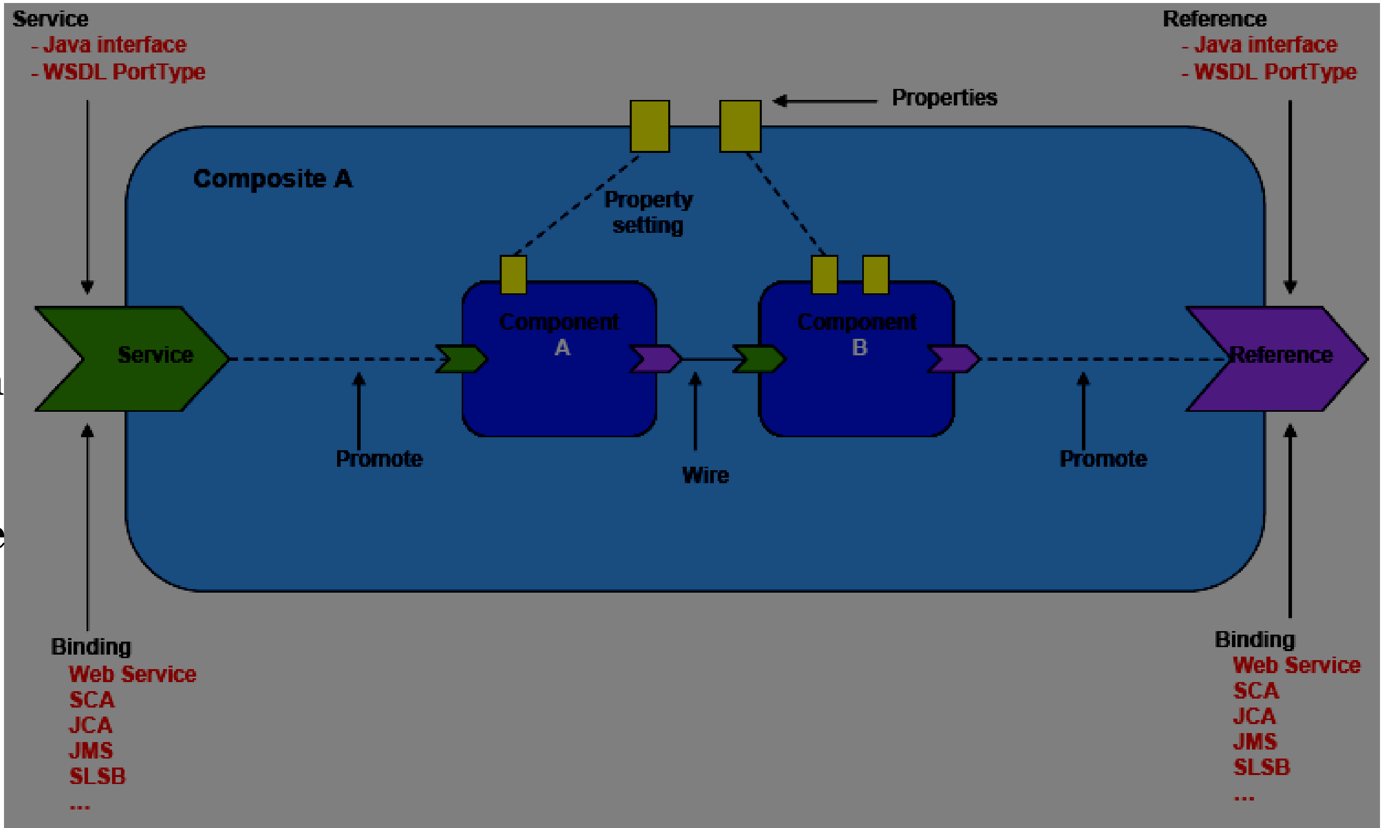
Composition Models: Service Component Architecture (1/2)

- **Components** are configured instances of service implementations which provide business functions
- Implementations can have settable **properties**, i.e. data values which influence the operation of the business function
- SCA created by OSOA Consortium



Composition Models: Service Component Architecture (2/2)

- The business function is offered for use by other components as **services**
- Implementations may depend on services provided by other components – dependencies called **references**
- **Composites** can contain all of the above plus the wiring that describes the connections between these elements

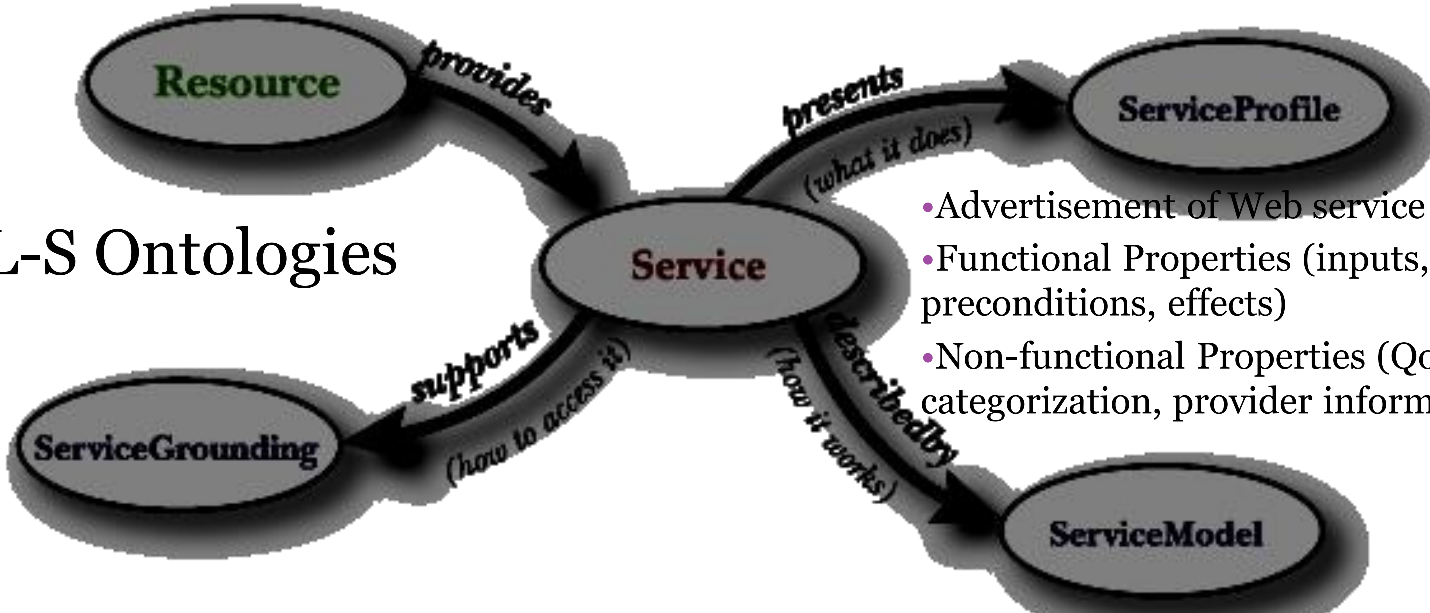


Service Composition in the Semantic Web (1/5)

- WSDL offers only **syntactic-level** descriptions of service functionalities
- The Semantic Web attempts to overcome this problem by introducing rich formal **semantic** descriptions of Web services that aim to facilitate
  - automated composition
  - automated discovery
  - dynamic binding
  - dynamic invocation
- Languages and ontologies such as OWL-S, WSMO and BPEL4SWS are examples of efforts related to Service Composition in the Semantic Web

Service Composition in the Semantic Web (2/5)

OWL-S Ontologies



- Advertisement of Web service capabilities
- Functional Properties (inputs, outputs, preconditions, effects)
- Non-functional Properties (QoS, categorization, provider information)

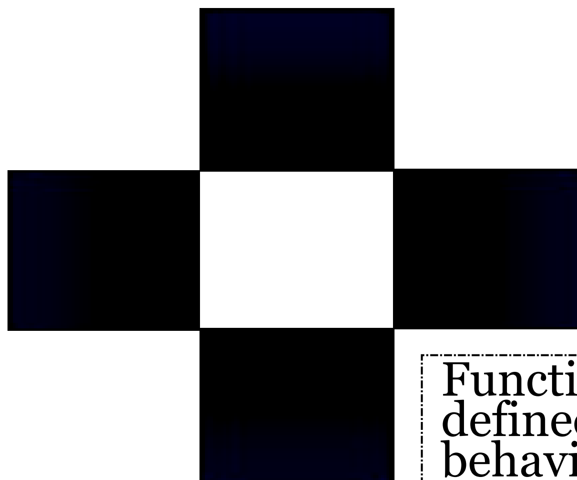
- Mapping from OWL-S to existing description languages such as WSDL
- e.g. inputs/outputs to messages
- e.g. atomic processes to operations

- Service viewed as a process
- Atomic: single interaction, directly invokable
- Composite: consists of other processes
- Simple: abstraction of atomic or simplified view of composite process

Service Composition in the Semantic Web (3/5)

WSMO Ontologies

Objectives that a client wants to achieve by using Web Services



Provide the formally specified terminology of the information used by all other components

Functional and non-functional characteristics as well as their behavioral aspects

Functional characteristics are defined as capabilities while behavior is described using interfaces (using orchestration and/or choreography models)

Connectors between components with mediation facilities for handling heterogeneities. Four cases of mediation: between goals, between ontologies, between a Web service and a goal and between Web services

WSMO (http://www.wsmo.org/)

- "It is the mission of the ESSI WSMO working group to, through alignment between key European research projects in the Semantic Web Service area, further the development of Semantic Web Services and works toward further standardization in the area of Semantic Web Service languages and to work toward a common architecture and platform for Semantic Web Services."

WSMO Ontologies

- Date and Time Ontology - defines a general model for specifying time and dates and relationships of them.
- Location Ontology - describes locations (such as continents, countries and cities and their interrelation).
- Purchase Ontology - describes generic elements of purchasing a product between a buyer and a seller.
- Train Connection Ontology - describes the domain of train tickets.
- Amazon ECS - describes the Amazon E-commerce Service.

Service Composition in the Semantic Web (4/5)

- BPEL4SWS (Nitzsche et. Al 2007)
  - augments BPEL in order to support semantic descriptions of Web services.
  - uses BPEL^{light} as its base, a WSDL-less BPEL
    - leaves out the interface description while keeping the BPEL process model
  - Semantic Web service descriptions (e.g. OWL-S/WSMO) are then attached to BPEL^{light} processes to enable semantic service discovery
    - WSMO is currently used, as OWL-S doesn't support asynchronous communication
  - Allows for the use of both traditional and Semantic Web services (SWS) in a BPEL process model and expose this process as both a Semantic Web service as well as a conventional WSDL service.

Service Composition in the Semantic Web (5/5)

Comparison between Semantic Web languages

SWS Languages	SWS Discovery Support	Process Model	Inter-Process Communication
OWL-S	Yes	Limited Orchestration	Synchronous
WSMO	Yes	Orchestration/ Choreography	Synchronous/ Asynchronous
BPEL4SWS	Through WSMO	Orchestration/ Choreography	Synchronous/ Asynchronous

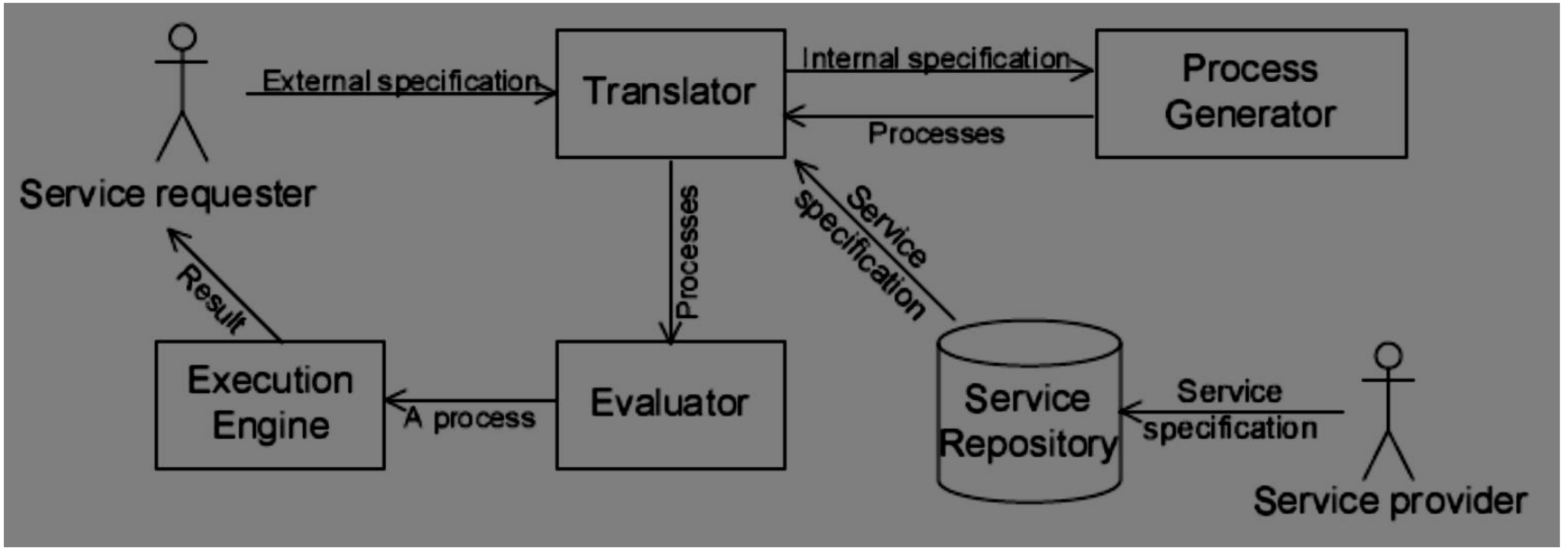
Service Composition Approaches

- Automated Service Composition
  - Workflow Techniques
  - AI Planning Techniques
- Other Approaches
  - Model-driven Service Composition
  - QoS-aware Service Composition

Automated Service Composition (1/3)

- Automated service composition attempts to decrease the level of human intervention in the composition process
- A proposed generic automated composition framework consists of five phases (Rao 2003)
  - Service providers **advertise** Web services that will be used in the composition, possibly with semantic descriptions
  - External descriptions are then **translated** to internal ones that describe the processes behind each Web service
  - A **process model** is created combining the existing service descriptions to satisfy a user request
  - The resulting process models are **evaluated** based on non-functional attributes
  - The one process model chosen in the previous phase is **executed**

Automated Service Composition (2/3)



Automated Service Composition (3/3)

- Approaches can be grouped into two different categories
  - Workflow techniques
    - Exploit the accumulated knowledge of the workflow community and apply it in the Web service domain
    - Views services as flows
  - AI planning techniques
    - Generate a plan containing the series of actions required to reach the goal state set by the service

Automated Service Composition: Workflow Techniques

- Majithia et al. (2004)
  - Automatically generate an abstract workflow based on a high-level goal
  - A Reasoning Service aims to decompose the goal and try to satisfy it with a combination of tasks
  - The abstract workflow tasks are then matched with concrete, already deployed services, or compositions of those
  - The result is a concrete workflow which is executed

Automated Service Composition: AI Planning Techniques (1/3)

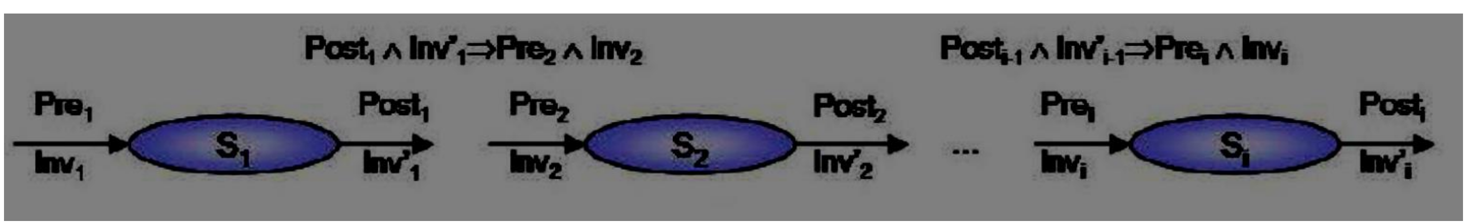
- AI Planning problem definition
  - A quintuple {S, s_o, G, A, Γ} where
  - S is a set of states, with s_o the initial state
  - G is the set of goal states to reach
  - A is the set of actions that can be performed
  - Γ describes the transition caused by an action
- Adapting to the Web service domain
  - A is the set of available services
  - G is the requester goal
  - S, s_o and Γ form a state model applied to the available services

Automated Service Composition: AI Planning Techniques (2/3)

- A planning problem is expressed using a domain theory, such as
  - Classical logic
  - Extra-logic domains, e.g. Planning Domain Definition Language (PDDL), graphs
  - Hierarchical Task Networks
  - Models e.g. Finite State Machines

Automated Service Composition: AI Planning Techniques (3/3)

- Alevizou et al. (2006)
  - Augment OWL-S descriptions with invariants
    - Assertions satisfied in both the initial and the final state
  - Define composition rules for basic constructs such for sequence, parallel composition, conditional composition
    - Rules defined based on inputs, outputs, preconditions, postconditions and invariants
  - Use a backward planner to create a composition plan according to the target goal beginning from the target goal

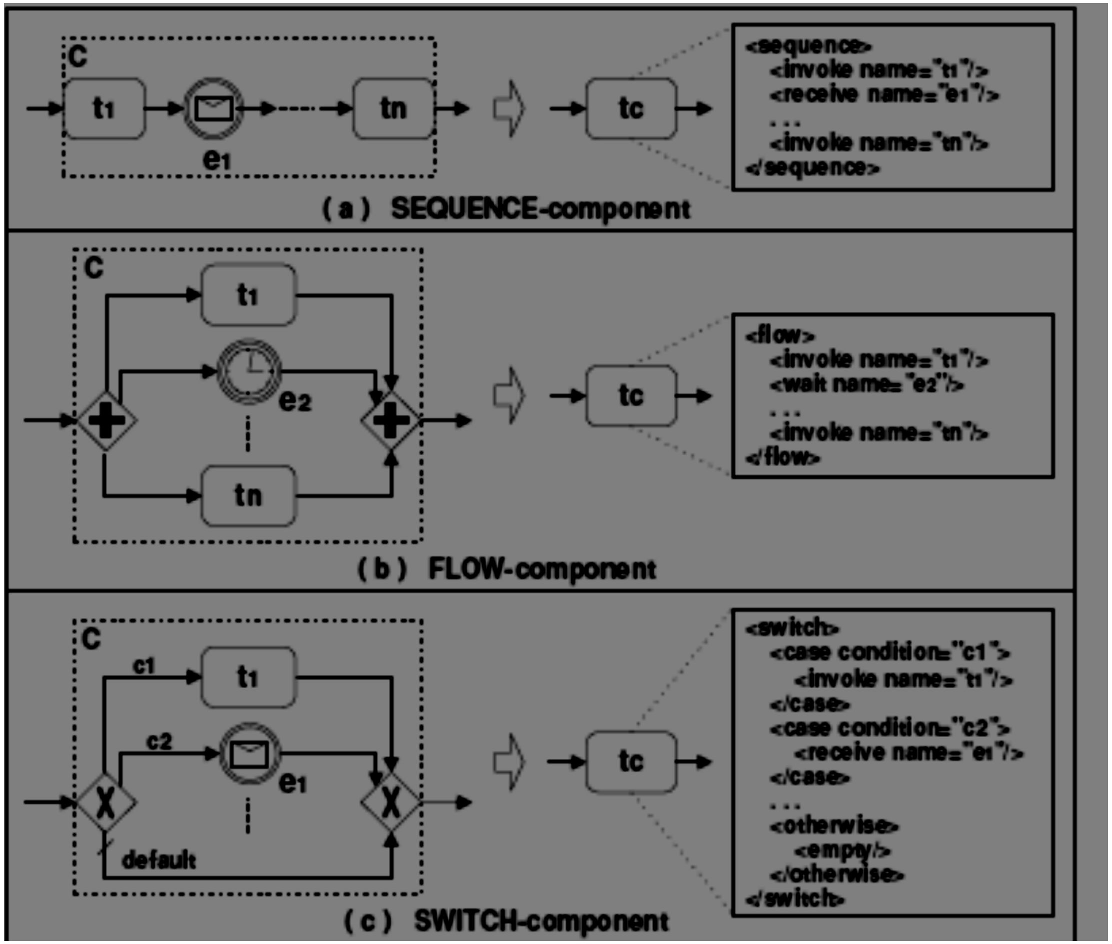


Model-driven Service Composition (1/2)

- Deals with the service composition problem using concepts from Model-Driven Development in software engineering
- Model-Driven Architecture is an example of MDD
  - All software functionality is specified using platform-independent models (PIMs)
  - Then the PIMs are translated to various platform-specific models (PSMs)
  - PSMs can either be compiled into executable code or run directly.

Model-driven Service Composition (2/2)

- Ouyang et al. (2006)
- use BPMN as a PIM and BPEL as a PSM
- Map from BPMN constructs to BPEL code
- BPEL is automatically generated but still human-readable



QoS-aware Service Composition (1/2)

- Augments service selection by taking QoS attributes into account, in addition to functional attributes
- Two phases for QoS calculation
  - **Local** optimization: QoS is calculated for each participating service individually
  - **Global** optimization: Finds a globally optimum set of services rather than a set of individually optimum services, to satisfy global QoS constraints

QoS-aware Service Composition (2/2)

- Most approaches define their own QoS model since a standard one hasn't been adopted
- Rosenberg et al. (2006) QoS model: focuses on performance and dependability attributes
  - **Performance** consists of
    - Various **time** calculations such as execution time, response time, round trip time etc.
    - **Throughput**: # of requests that can be processed within a given period of time
    - **Scalability**: the ability to support multiple parallel requests
  - **Dependability** consists of
    - **Availability**: the probability that a service is up and running
    - **Accuracy**: the ratio of total versus failed requests
    - **Robustness**: the probability that the system reacts successfully given invalid, incomplete or conflicting input

Conclusions (1/5)

- Web service composition is a multifaceted and very complex problem
  - A complete solution taking into account all differing aspects does not exist and may be irrational
  - Solution approaches should focus on combining the most successful aspects of each approach
  - The general target should be to create a standard composition framework to be adopted universally

Conclusions (2/5)

- Viewing service composition as a workflow problem is a reasonable basis
  - given the existence of standards such as BPEL
- However, conventional BPEL only allows for manual creation of composition schemas and static service binding
  - BPEL should be used as an excellent orchestration model that can be adapted to support automated and dynamic service composition

Conclusions (3/5)

- Automated creation of service composition schemas is closely related to an AI planning problem
- However, adopting one of the many preexisting planning techniques and applying it for the case of service composition may not be the solution
  - Design a planning technique specifically adapted to Web service composition requirements
  - Combine planning with workflows by using it to create an abstract BPEL flow

Conclusions (4/5)

- Service selection and matchmaking cannot be based solely on syntactic functional attributes
  - The emergence of the Semantic Web introduces semantics in service descriptions
  - Semantics characterize a service not based solely on its interface but based on its functionality as well
- However, semantics are domain-independent and an attempt to create a global domain-independent ontology will most certainly result in conflicts, misinterpretation and confusion

Conclusions (5/5)

- QoS attributes should play a part not only in service selection but also in the evaluation of composition schemas
  - A standard QoS model has not yet been adopted for classifying services based on their quality aspects

Τέλος Ενότητας

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σημειώματα

Σημείωμα αδειοδότησης

Σημείωμα Αναφοράς