Transactional and QoS-aware Internet Service Management

Maude Manouvrier

PSL Research University, Université Paris-Dauphine, LAMSADE UMR CNRS 7243, France

Mémoire présenté en vue de l'obtention de l'Habilitation à Diriger des Recherches

Outline

Outline

Substantial part of the digital content exposed via Web services (WS):

- ~ 16,000 API on `Programmableweb`
- Several domain-specific catalogues:
  - Various databases, retrieval and analysis services provided by the European Bioinformatics Institute[1] (EBI)
  - Biodiversity Web services of the BiodiversityCatalogue[2]
  - Collection of Web Services offering data and applications in Renewable Energy and Environment[3]
  - Web Services for Image Processing [4]
  - REST API for accessing data of "Mairie de Paris"[5]

1 http://www.ebi.ac.uk/Tools/webservices/
2 https://www.biodiversitycatalogue.org/
3 http://www.webservice-energy.org/
4 https://www.leadtools.com/sdk/image-processing/web-services
5 https://api.paris.fr/

Internet service:

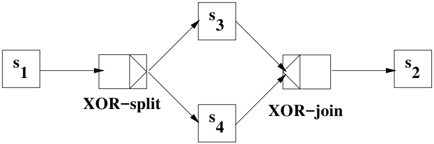
A service s is syntactically described by its set of inputs, $in(s)$, and its set of outputs, $out(s)$ and has a functionality.

Examples of image processing services[6]

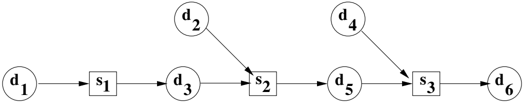
Service	Inputs	Outputs	Functionality
s_1	A coloured image	A gray-scale image	Converts input image to a gray-scale one
s_2	An image and a threshold	A transformed image	Applies a binary thresholding method

[6] From Jungmann, A., and Mohr, F. (2015). An approach towards adaptive service composition in markets of composed services. *Journal of Internet Services and Applications*, 6(1), 1.

Composite service

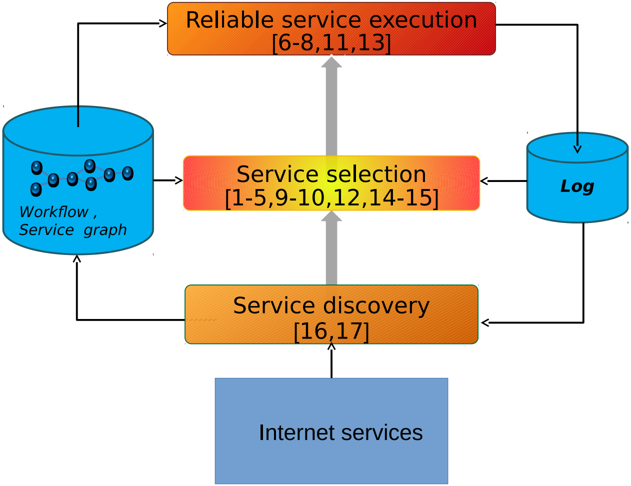


A composite service represented by a concrete workflow



A composite service represented by a ServiceData graph

Our vision of a service management framework



Outline

Outline

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
     SELECT ?person ?book
     WHERE {
     ?person rdf:type db-owl:Writer ;
             db-owl:birthPlace dbpedia:Paris.
     ?book db-owl:author ?person ;
           db-owl:isbn ?isbn .}
```

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

- and a registry of RDF service descriptions

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

- and a registry of RDF service descriptions

  Lightweight Semantic Service Descriptions:

  - SAWSDL (Semantic Annotations for WSDL and XML Schema),
  - WSMO-Lite (Lightweight Semantic Descriptions for Services on the Web),
  - OWL-S (Semantic Markup for Web Services) ...

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

- and a registry of RDF service descriptions*

```
<profile:serviceName xml:lang="en"> BookPrice
    </profile:serviceName><profile:textDescription xml:lang="en">
      Uses the ISBN to return the purchase price
      of a given book title.
    </profile:textDescription>
    <profile:hasInput rdf:resource="#_BOOK"/>
    <profile:hasOutput rdf:resource="#_PRICE"/>
```

* from http://projects.semwebcentral.org/projects/owls-tc/

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

- and a registry of RDF service descriptions*

```
<profile:serviceName xml:lang="en"> BookPrice
    </profile:serviceName><profile:textDescription xml:lang="en">
      Uses the ISBN to return the purchase price
      of a given book title.
    </profile:textDescription>
    <profile:hasInput rdf:resource="#_BOOK"/>
    <profile:hasOutput rdf:resource="#_PRICE"/>
```

* from http://projects.semwebcentral.org/projects/owls-tc/

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

- and a registry of RDF service descriptions[7]

```
<profile:serviceName xml:lang="en"> BookPrice
    </profile:serviceName><profile:textDescription xml:lang="en">
      Uses the ISBN to return the purchase price
      of a given book title.
    </profile:textDescription>
    <profile:hasInput rdf:resource="#_BOOK"/>
    <profile:hasOutput rdf:resource="#_PRICE"/>
```

* from http://projects.semwebcentral.org/projects/owls-tc/

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

Outputs = O_Q (`?person ?book`)

Inputs = I_Q (the WHERE clause triple patterns)

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book              Outputs = O_Q
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.   Inputs = I_Q
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

- and a registry of RDF service descriptions

```
<profile:serviceName xml:lang="en"> BookPrice
    </profile:serviceName><profile:textDescription xml:lang="en">
      Uses the ISBN to return the purchase price
      of a given book title.
    </profile:textDescription>
    <profile:hasInput rdf:resource="#_BOOK"/>  = in(s)
    <profile:hasOutput rdf:resource="#_PRICE"/> = out(s)
```

Service discovery

As part of M.A Mouhoub's PhD work ICSOC'2014 [16], Demo ESWC'2014 [17]

Given:

- A user SPARQL query on linked data

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book                         Outputs = O_Q
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.     Inputs = I_Q
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

- and a registry of RDF service descriptions

```
<profile:serviceName xml:lang="en"> BookPrice
    </profile:serviceName><profile:textDescription xml:lang="en">
      Uses the ISBN to return the purchase price
      of a given book title.
    </profile:textDescription>
    <profile:hasInput rdf:resource="#_BOOK"/>  = in(s)
    <profile:hasOutput rdf:resource="#_PRICE"/> = out(s)
```

Discover services that complete the user query:

1. $\{s \mid I_Q \subset in(s) \text{ and } O_Q \subset out(s)\}$
2. $\{s \mid O_Q \cap out(s) \neq \emptyset \text{ and } O_Q \not\subset out(s)\}$
3. $\{s \mid O_Q \cap in(s) \neq \emptyset\}$
4. $\{s \mid I_Q \cap in(s) \neq \emptyset\}$

Outline

Service discovery

1. Service Request Extraction :
   Extracting elements that can be used as service's I/O

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

Outputs: ?person ?book

Inputs: the WHERE clause triple patterns

Service discovery

1. Service Request Extraction :
   Extracting elements that can be used as service's I/O

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book   Outputs
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.   Inputs
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

2. Semantics Lookup :
   Finding semantic concepts similar to or associated with the ones extracted at step 1, using SPARQL queries on the ontology or on LOD

```
prefix db-owl: <http://dbpedia.org/ontology/>
    SELECT ?person ?book
    WHERE {
    ?person rdf:type db-owl:Writer ;
            db-owl:birthPlace dbpedia:Paris.
    ?book db-owl:author ?person ;
          db-owl:isbn ?isbn .}
```

Outline

Service discovery

4. Service Query Generation using for example strategy 3: $\{s \mid O_Q \cap in(s) \neq \emptyset\}$

```
SELECT DISTINCT ?service WHERE {
?service a service:Service ; service:presents ?profile .
?profile profile:hasInput ?input1 .
?input1 process:parameterType dbpedia-owl:Book .
}
```

Service discovery

4. Service Query Generation using for example strategy 3: $\{s \mid O_Q \cap in(s) \neq \emptyset\}$

```
SELECT DISTINCT ?service WHERE {
?service a service:Service ; service:presents ?profile .
?profile profile:hasInput ?input1 .
?input1 process:parameterType dbpedia-owl:Book .
}
```

```
<profile:serviceName xml:lang="en"> BookPrice
    </profile:serviceName><profile:textDescription xml:lang="en">
      Uses the ISBN to return the purchase price
      of a given book title.
    </profile:textDescription>
    <profile:hasInput rdf:resource="#_BOOK"/>
    <profile:hasOutput rdf:resource="#_PRICE"/>
```

Service discovery

4. Service Query Generation using for example strategy 3: $\{s \mid O_Q \cap in(s) \neq \emptyset\}$

```
SELECT DISTINCT ?service WHERE {
?service a service:Service ; service:presents ?profile .
?profile profile:hasInput ?input1 .
?input1 process:parameterType dbpedia-owl:Book .
}
```

Problem:

When no service answers the user query, several services should be composed to provide the needed outputs from the provided inputs

Outline

Transactional and QoS-aware service selection

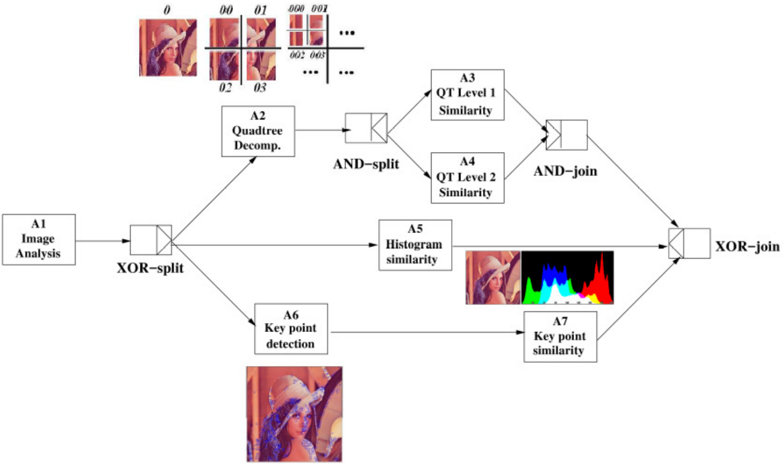
Service selection

Given a user query Q that can not be satisfied by a single service: find a composite service that answers the user query

Two kinds of service selection:

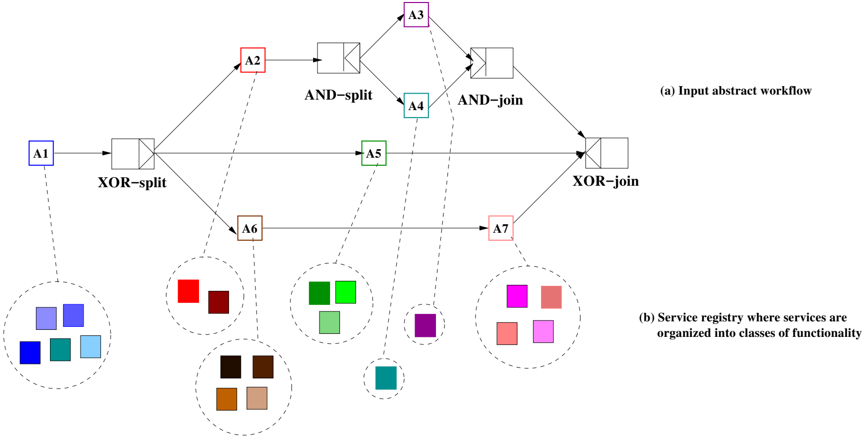
- Top-down: Query Q is a workflow and services are grouped by functionality
- Bottom-up: Query Q is a set of I/O and services linked by their I/O

Outline



An abstract content-based image retrieval workflow

Top-down service selection

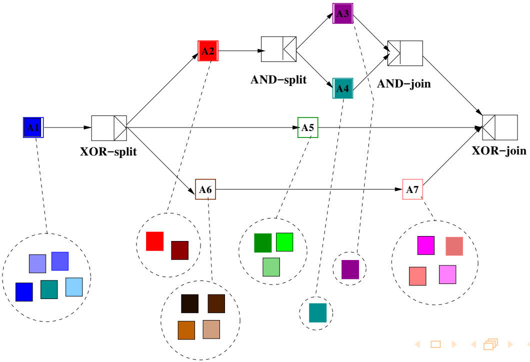


The top-down service selection process

Top-down service selection

Top-down service selection:

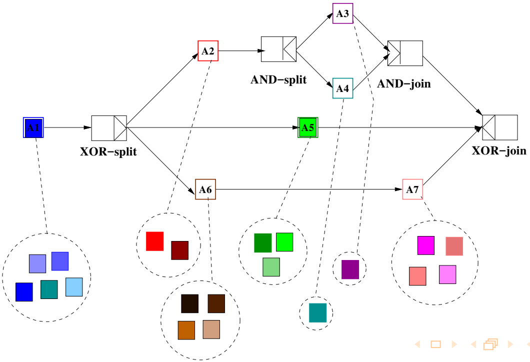
Selecting one service per activity of the workflow for sequence and AND patterns, and one service per activity for one branch of the XOR patterns



Top-down service selection

Top-down service selection:

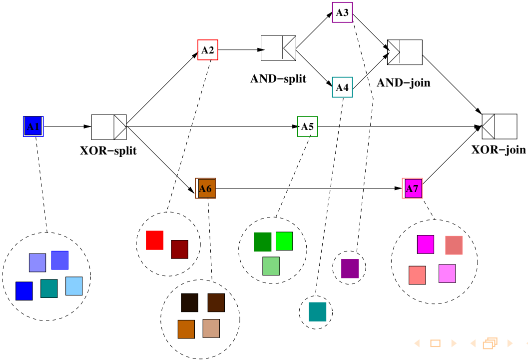
Selecting one service per activity of the workflow for sequence and AND patterns, and one service per activity for one branch of the XOR patterns



Top-down service selection

Top-down service selection:

Selecting one service per activity of the workflow for sequence and AND patterns, and one service per activity for one branch of the XOR patterns

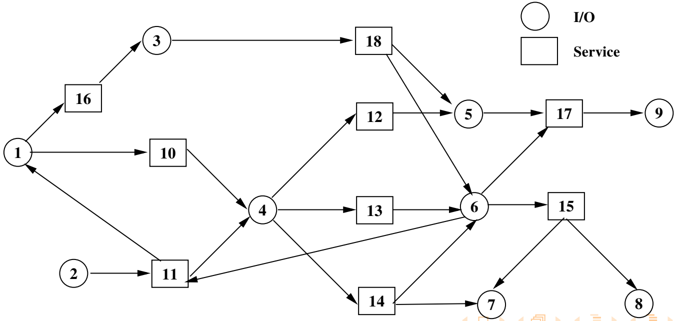


Outline

Bottom-up service selection

Given :

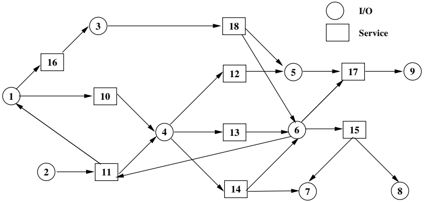
- A query $\mathcal{Q} = (I_\mathcal{Q}, O_\mathcal{Q})$, with $I_\mathcal{Q}$ the set of inputs data provided by the user and $O_\mathcal{Q}$ the set of outputs needed by the user, and
- A service registry organised by a ServiceData Graph where services are linked by their I/O based on an ontology



Bottom-up service selection

Given :

- A query $\mathcal{Q} = (I_{\mathcal{Q}}, O_{\mathcal{Q}})$, with $I_{\mathcal{Q}}$ the set of inputs data provided by the user and $O_{\mathcal{Q}}$ the set of outputs needed by the user, and
- A service registry organised by a ServiceData Graph where services are linked by their I/O based on an ontology



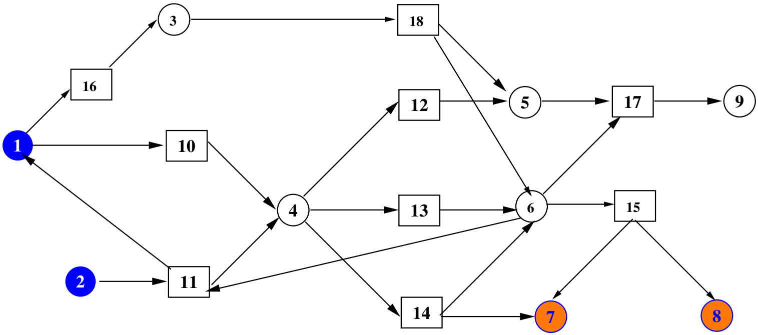
Can be built from WS-Challenge[7] or from real services[8]

[7] S. Kona et al. WSC-2009: A Quality of Service-Oriented Web Services Challenge. In *IEEE CEC* , 2009

[8] Y.-J. Lee. Algorithm for Automatic Web API Composition. In *WEB*, 2013

Bottom-up service selection:

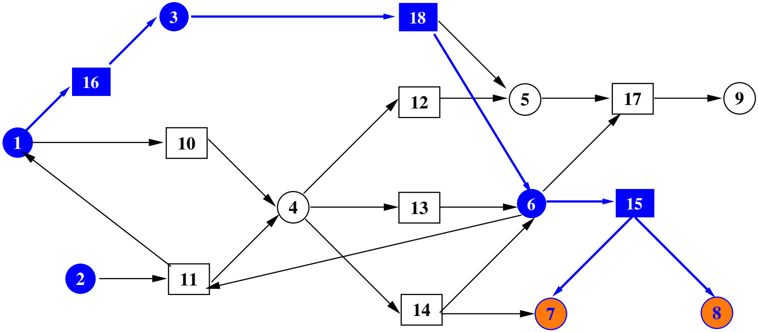
Selecting a sub-graph in the ServiceData Graph



A possible composite service with $I_Q = \{1, 2\}$ and $O_Q = \{7, 8\}$

Bottom-up service selection:

Selecting a sub-graph in the ServiceData Graph



A possible composite service with $I_Q = \{1, 2\}$ and $O_Q = \{7, 8\}$

Outline

QoS-aware service selection

Selecting component services that optimize the Quality of Service

QoS criteria associated with a service:

- Execution time (criterion to minimize):
  Time needed to execute the service
- Throughput (criterion to maximize):
  Average rate of successful service executions
- Cost (criterion to minimize):
  Fee the user has to pay to invoke the service
- Reliability (criterion to maximize):
  Probability of the service successful executions

Examples of QoS dataset:

- QoS values extracted from real web services

Name	Response Time (ms)	Throughput (hits/sec)
SiteSearchService	100	12.3
GoogleSearchService	104.67	8.5
SearchService	234.5	13.3
AmazonSearchService	85.3	5.1
SearchService	343.66	3

QWS Dataset[9]: QoS values extracted from 5,000 web services (Al-Masri et al., 2007)

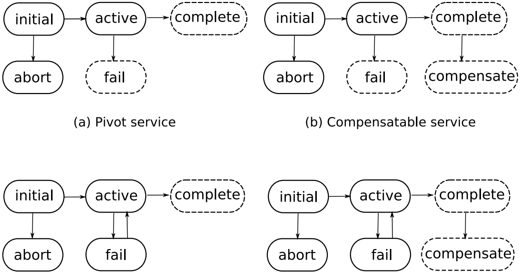
- Synthetic benchmark: Execution time and throughput introduced in WS-Challenge 2009 (Kona et al. 2009)

[9] http://www.uoguelph.ca/~qmahmoud/qws/

Outline

Transactional-aware service selection

Selecting component services to obtain a transactional composite service satisfying the transactional property needed by the user

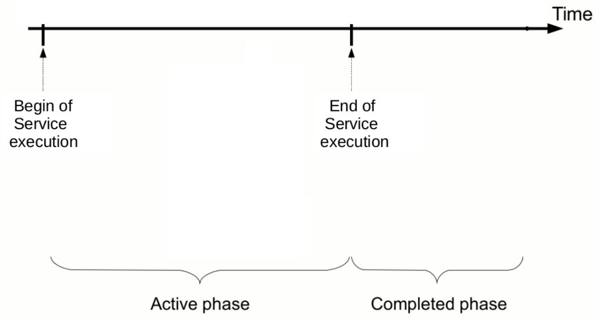


(a) Pivot service
(b) Compensatable service
(c) Pivot and retriable service
(d) Compensatable and retriable service

Transactional properties for non-composite services[10]

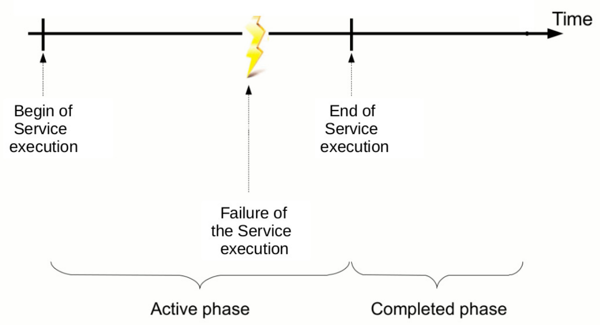
[10] inspired from S. Mehrotra et al. A Transaction Model for Multidatabase Systems. In *ICDC*, 1992

Links between transactional properties and recovery techniques



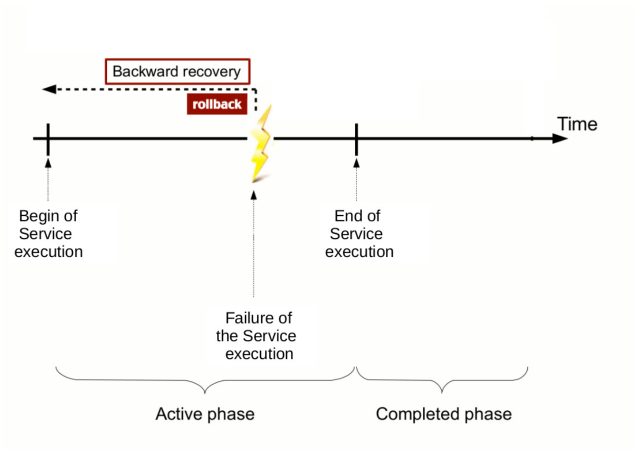
Recovery techniques and life cycle of a non-composite service

Links between transactional properties and recovery techniques



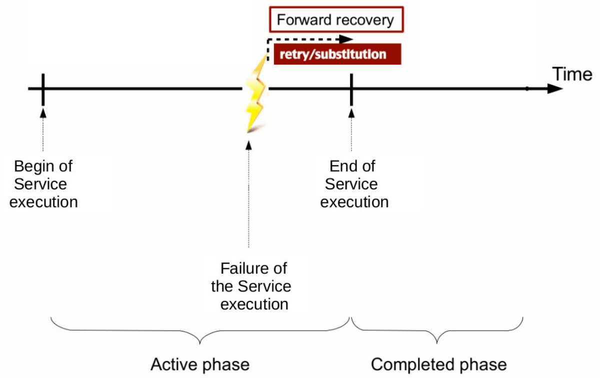
Recovery techniques and life cycle of a non-composite service

Links between transactional properties and recovery techniques



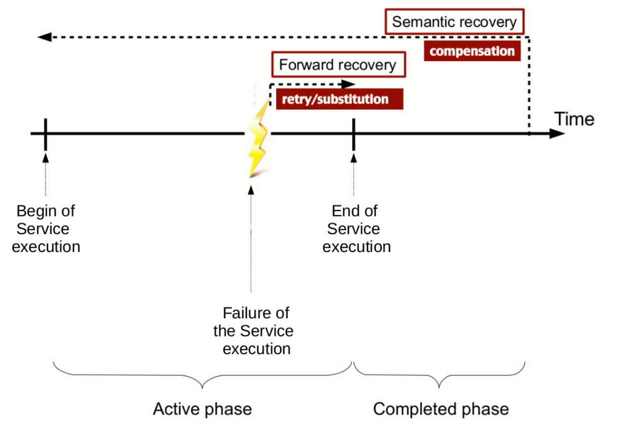
Recovery techniques and life cycle of a non-composite service

Links between transactional properties and recovery techniques



Recovery techniques and life cycle of a non-composite service

Links between transactional properties and recovery techniques



Recovery techniques and life cycle of a non-composite service

Transactional properties extended to composite services in [3] (IEEE TSC 2010) :

A composite service is atomic (*a*):

- if once all its component services complete successfully, they cannot be semantically undone
- if one component service does not complete successfully, all previously successful component services have to be compensated

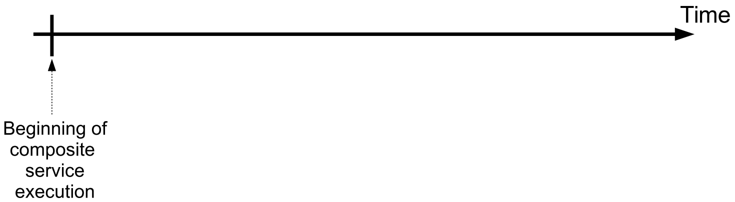
A composite service is compensatable (*c*) if:

all its component services are compensatable

A composite service is retriable (*r*) if:

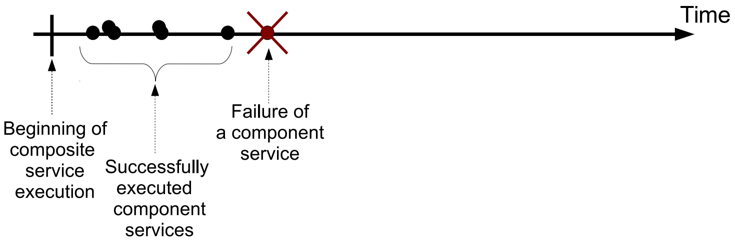
all its component services are retriable

Recovery techniques for a composite service

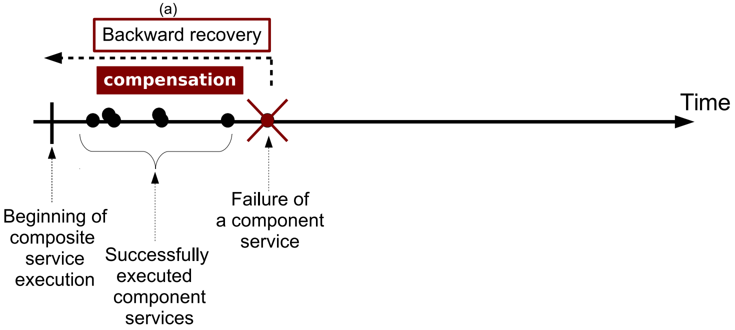


Recovery techniques for a composite service

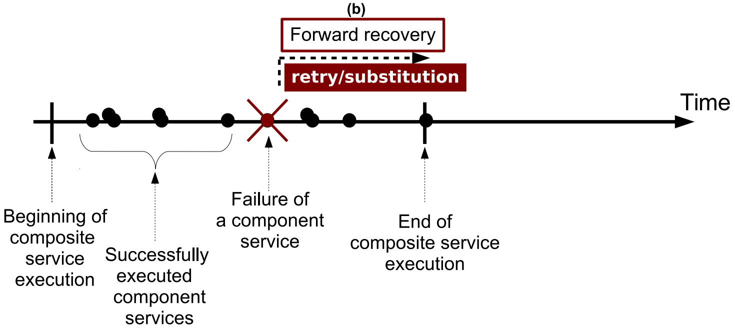
Recovery techniques for a composite service



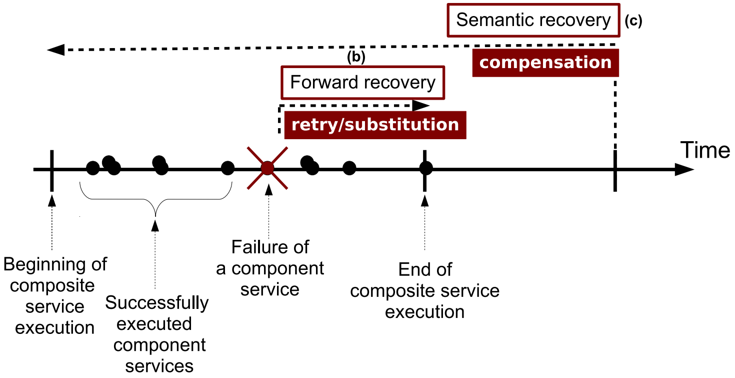
Recovery techniques for a composite service



Recovery techniques for a composite service



Recovery techniques for a composite service



Transactional properties for composite services

Transactional rules defined in [3]

Transactional property of a Service	**Sequential compatibility**	**Parallel compatibility**
p, a	$\{pr, ar, cr\}$ (rule 1)	cr (rule 2)
pr, ar	$\{pr, ar, cr\}$ (rule 3)	$\{pr, ar, cr\}$ (rule 4)
c	$\{p, pr, a, ar, c, cr\}$ (rule 5)	$\{c, cr\}$ (rule 6)
cr	$\{p, pr, a, ar, c, cr\}$ (rule 7)	$\{p, pr, a, ar, c, cr\}$(rule 8)

Transactional properties for composite services

Transactional rules defined in [3]

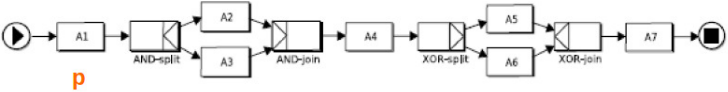
Transactional property of a Service	**Sequential compatibility**	**Parallel compatibility**
p, *a*	{*pr*, *ar*, *cr*} (rule 1)	*cr* (rule 2)
pr, *ar*	{*pr*, *ar*, *cr*} (rule 3)	{*pr*, *ar*, *cr*} (rule 4)
c	{*p*, *pr*, *a*, *ar*, *c*, *cr*} (rule 5)	{*c*, *cr*} (rule 6)
cr	{*p*, *pr*, *a*, *ar*, *c*, *cr*} (rule 7)	{*p*, *pr*, *a*, *ar*, *c*, *cr*}(rule 8)

Example of a transactional composite service

Transactional properties for composite services

Transactional rules defined in [3]

Transactional property of a Service	**Sequential compatibility**	**Parallel compatibility**
p, *a*	{*pr*, *ar*, *cr*} **(rule 1)**	*cr* (rule 2)
pr, *ar*	{*pr*, *ar*, *cr*} (rule 3)	{*pr*, *ar*, *cr*} (rule 4)
c	{*p*, *pr*, *a*, *ar*, *c*, *cr*} (rule 5)	{*c*, *cr*} (rule 6)
cr	{*p*, *pr*, *a*, *ar*, *c*, *cr*} (rule 7)	{*p*, *pr*, *a*, *ar*, *c*, *cr*}(rule 8)

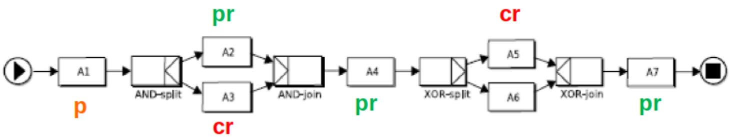


Example of a transactional composite service

Transactional properties for composite services

Transactional rules defined in [3]

Transactional property of a Service	**Sequential compatibility**	**Parallel compatibility**
p, *a*	{*pr*, *ar*, *cr*} **(rule 1)**	*cr* (rule 2)
pr, *ar*	{*pr*, *ar*, *cr*} (rule 3)	{*pr*, *ar*, *cr*} (rule 4)
c	{*p*, *pr*, *a*, *ar*, *c*, *cr*} (rule 5)	{*c*, *cr*} (rule 6)
cr	{*p*, *pr*, *a*, *ar*, *c*, *cr*} (rule 7)	{*p*, *pr*, *a*, *ar*, *c*, *cr*}(rule 8)

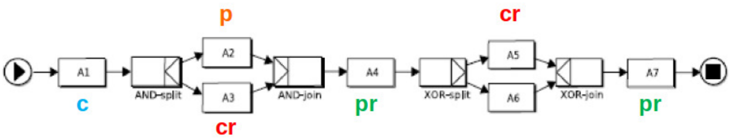


Example of a transactional composite service

Transactional properties for composite services

Transactional rules defined in [3]

Transactional property of a Service	**Sequential compatibility**	**Parallel compatibility**
p, *a*	{*pr*, *ar*, *cr*} (rule 1)	*cr* **(rule 2)**
pr, *ar*	{*pr*, *ar*, *cr*} (rule 3)	{*pr*, *ar*, *cr*} (rule 4)
c	{*p*, *pr*, *a*, *ar*, *c*, *cr*} (rule 5)	{*c*, *cr*} (rule 6)
cr	{*p*, *pr*, *a*, *ar*, *c*, *cr*} (rule 7)	{*p*, *pr*, *a*, *ar*, *c*, *cr*}(rule 8)



Example of a transactional composite service

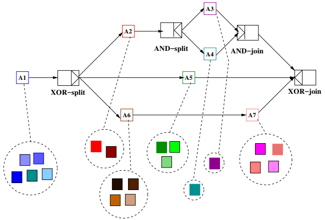
Outline

Transactional and QoS-aware service selection

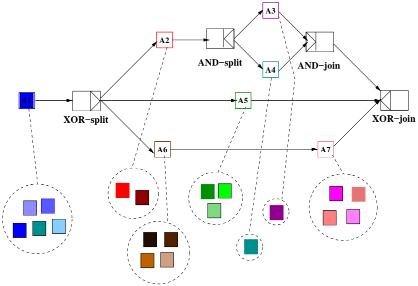
Selecting services by simultaneously considering transactional property support and QoS optimization

Our contributions:

- The first Transactional and *local* QoS-aware service selection approaches for:
  - Top-down selection, presented in ICSW'2008 [12] and IEEE TSC'2010 [3]



User query and service functionality classes



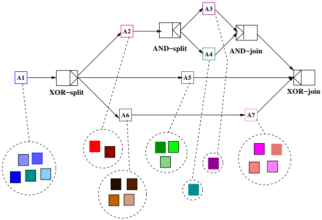
Selected composite service

Transactional and QoS-aware service selection

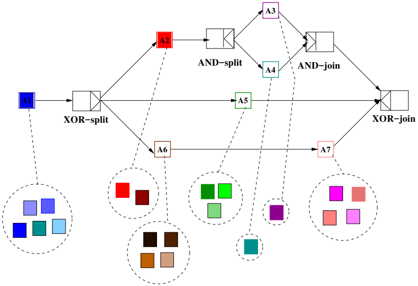
Selecting services by simultaneously considering transactional property support and QoS optimization

Our contributions:

- The first Transactional and *local* QoS-aware service selection approaches for:
  - Top-down selection, presented in ICSW'2008 [12] and IEEE TSC'2010 [3]



User query and service functionality classes



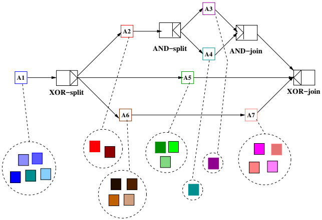
Selected composite service

Transactional and QoS-aware service selection

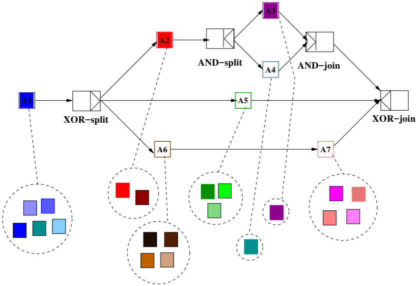
Selecting services by simultaneously considering transactional property support and QoS optimization

Our contributions:

- The first Transactional and *local* QoS-aware service selection approaches for:
  - Top-down selection, presented in ICSW’2008 [12] and IEEE TSC’2010 [3]



User query and service functionality classes



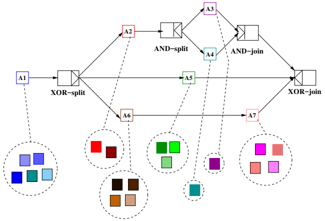
Selected composite service

Transactional and QoS-aware service selection

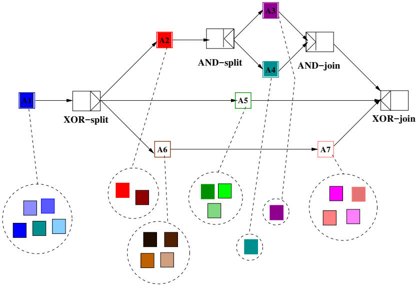
Selecting services by simultaneously considering transactional property support and QoS optimization

Our contributions:

- The first Transactional and *local* QoS-aware service selection approaches for:
  - Top-down selection, presented in ICSW'2008 [12] and IEEE TSC'2010 [3]



User query and service functionality classes



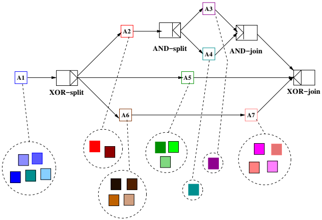
Selected composite service

Transactional and QoS-aware service selection

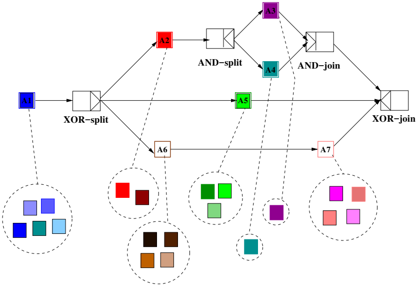
Selecting services by simultaneously considering transactional property support and QoS optimization

Our contributions:

- The first Transactional and *local* QoS-aware service selection approaches for:
  - Top-down selection, presented in ICSW'2008 [12] and IEEE TSC'2010 [3]



User query and service functionality classes



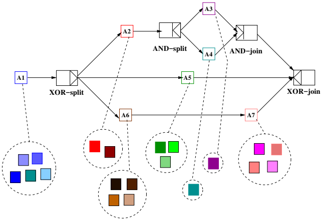
Selected composite service

Transactional and QoS-aware service selection

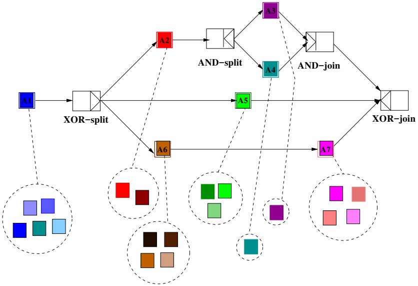
Selecting services by simultaneously considering transactional property support and QoS optimization

Our contributions:

- The first Transactional and *local* QoS-aware service selection approaches for:
  - Top-down selection, presented in ICSW'2008 [12] and IEEE TSC'2010 [3]



User query and service functionality classes



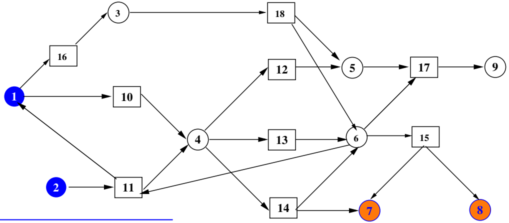
Selected composite service

Transactional and QoS-aware service selection

Selecting services by simultaneously considering transactional property support and QoS optimization

Our contributions:

- The first Transactional and *local* QoS-aware service selection approaches for:
  - Top-down selection, presented in ICSW'2008 [12] and IEEE TSC'2010 [3]
  - Bottom-up selection[11], presented in ICCS'2010 [10], RED'2012 [9] and IJWGS'2011 [2]



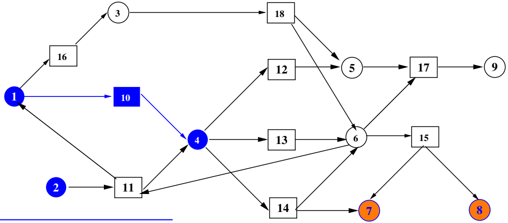
[11] supported by the Franco-Venezuelan CNRS-FONACIT project

Transactional and QoS-aware service selection

Selecting services by simultaneously considering transactional property support and QoS optimization

Our contributions:

- The first Transactional and *local* QoS-aware service selection approaches for:
  - Top-down selection, presented in ICSW'2008 [12] and IEEE TSC'2010 [3]
  - Bottom-up selection[11], presented in ICCS'2010 [10], RED'2012 [9] and IJWGS'2011 [2]



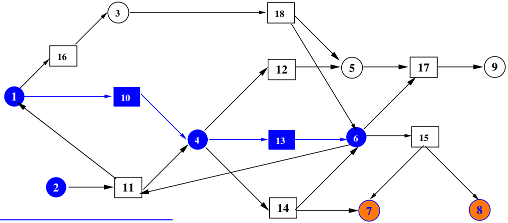
[11] supported by the Franco-Venezuelan CNRS-FONACIT project

Transactional and QoS-aware service selection

Selecting services by simultaneously considering transactional property support and QoS optimization

Our contributions:

- The first Transactional and *local* QoS-aware service selection approaches for:
  - Top-down selection, presented in ICSW'2008 [12] and IEEE TSC'2010 [3]
  - Bottom-up selection[11], presented in ICCS'2010 [10], RED'2012 [9] and IJWGS'2011 [2]



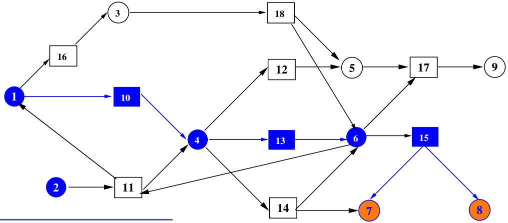
[11] supported by the Franco-Venezuelan CNRS-FONACIT project

Transactional and QoS-aware service selection

Selecting services by simultaneously considering transactional property support and QoS optimization

Our contributions:

- The first Transactional and *local* QoS-aware service selection approaches for:
  - Top-down selection, presented in ICSW'2008 [12] and IEEE TSC'2010 [3]
  - Bottom-up selection[11], presented in ICCS'2010 [10], RED'2012 [9] and IJWGS'2011 [2]



[11] supported by the Franco-Venezuelan CNRS-FONACIT project

Transactional and QoS-aware service selection

Selecting services by simultaneously considering transactional property support and QoS optimization

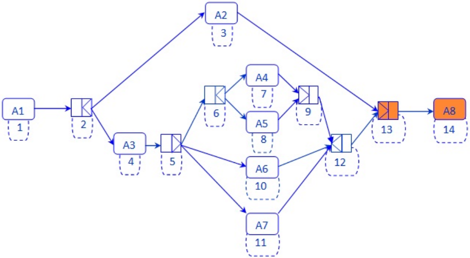
Our contributions:

- The first Transactional and *local* QoS-aware service selection approaches for:
  - Top-down selection, presented in ICSW'2008 [12] and IEEE TSC'2010 [3]
  - Bottom-up selection, presented in ICCS'2010 [10], RED'2012 [9] and IJWGS'2011 [2]
- First linear programs modelling Transactional and *global* QoS-aware service selection problem, for:
  - Top-down selection, presented in ISCC'2012 [14] and in DAM'2015 [4]
  - Bottom-up selection, presented in ICSOC'2014 [15]

Linear program modelling the service selection

For Top-down selection[12]:

- Decision variables for selecting activities and services
- Constraints:
  - induced by the workflow,



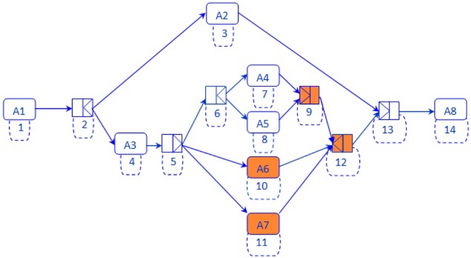
- ⋆ A_8 (N°14) should be executed : $x_{14} = 1$
- ⋆ To execute A_8, AND-JOIN (N°13) should be executed : $x_{14} = x_{13}$

[12] supported by the Action de Recherche Fondamentale en Recherche Opérationnelle (ARFRO) of the French Operational Research Group (GDR RO)

Linear program modelling the service selection

For Top-down selection[12]:

- Decision variables for selecting activities and services
- Constraints:
  - induced by the workflow,



To executed XOR-JOIN (N°12): A_6 or A_7 (N°10 and 11) or AND-JOIN N°9, should be executed : $x_{12} = x_9 + x_{10} + x_{11}$

[12] supported by the Action de Recherche Fondamentale en Recherche Opérationnelle (ARFRO) of the French Operational Research Group (GDR RO)

Linear program modelling the service selection

For Top-down selection[12]:

- Decision variables for selecting activities and services
- Constraints:
  - induced by the workflow,
  - modelling the execution time (limited to T) and the reliability (guarantee to be $\geq P$) and,
  - induced by transactional properties
- Linear program minimizing the cost, with or without transactional properties
- Experimentations with CPLEX on randomly generated test sets
  - to compare with related works and,
  - to analyse the impact of QoS constraints, number of activities or services on the computational time

[12] supported by the Action de Recherche Fondamentale en Recherche Opérationnelle (ARFRO) of the French Operational Research Group (GDR RO)

Linear program modelling the service selection

For Top-down selection[12]:

- Experimentations with CPLEX on randomly generated test sets, for instances of complex 100-activities workflows

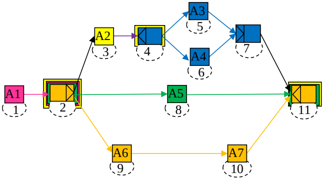
Number of WS per trans. prop.	Average number of variables	Average number of constraints	Average running time (s)
5	2109	16902	2,5
10	4111	24684	3,4
15	6112	17950	3,4
20	8108	20975	4,6
25	10111	16670	5,5
30	12110	19302	6,2
40	16109	13578	5,8
50	20111	19960	10,0

[12] supported by the Action de Recherche Fondamentale en Recherche Opérationnelle (ARFRO) of the French Operational Research Group (GDR RO)

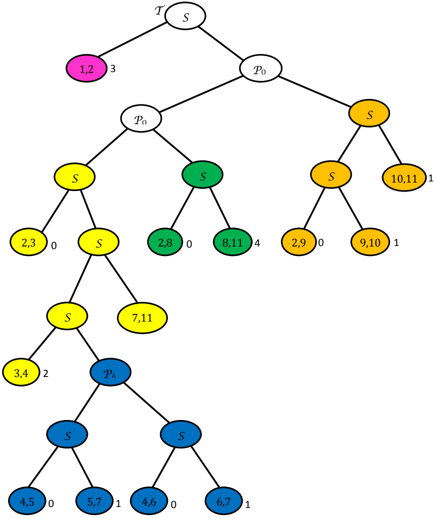
As shown in [4] (Discrete Applied Mathematics 2015)

Top-down service selection problem can be solved:

- In polynomial time when optimizing sum-type QoS criterion (e.g. cost) or a product-type QoS criterion (e.g. reliability)
- In $O(m)$ for min/max-type QoS criterion (e.g. execution time), with m the number of leaves in the decomposition tree associated with the serie-parallel graph representing the workflow



A workflow



Its decomposition tree

Linear program modelling the service selection

For Bottom-up selection:

- Decision variables for component services, the resulting sub-graph and the topological order
- Constraints:
  - modelling the I/O of services,
  - implied by the user query,
  - induced by transactional properties
- Linear program minimizing the execution time or maximizing the throughput, with or without transactional properties
- Experimentations with CPLEX on randomly generated test sets and on WS-Challenge benchmark
  - to compare with related works, and
  - to analyse the impact induced by transactional properties

Linear program modelling the service selection

For Bottom-up selection:

- Linear program minimizing a cost or QoS score, with or without transactional properties
- Experimentations with CPLEX on randomly generated test sets and on WS-Challenge benchmark

Experiments of our model on the WS-Challenge 2009 (WSC) test sets

WSC test set	1 (500WS)	2 (4000 WS)	3 (8000 WS)	4 (8000 WS)	5 (15000 WS)
To find optimal sol.	0.35s	3.46s	4.23s	22s	27s
To prove optimality	6.65s	5.8s	6.7s	$>$ 300s	$>$ 300s

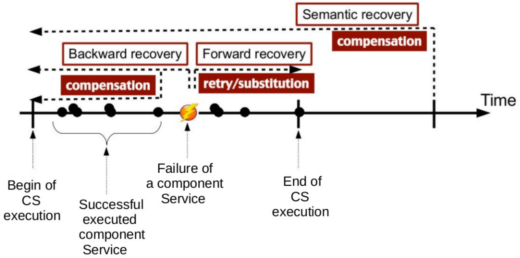
Outline

Self-healing transactional composite service

As part of R.Angarita's PhD work ICSOC'2014 [8] and MEDES'2016 [7]

Self-healing transactional composite service

Non intrusive dynamic fault-tolerant approach for composite service providing a reliable service execution

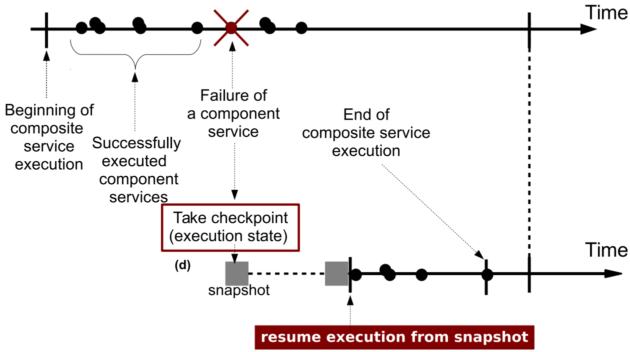


Recovery techniques and life cycle of a composite service

Outline

Self-healing transactional composite service

As part of R.Angarita's PhD work ICSOC'2014 [8] and MEDES'2016 [7]

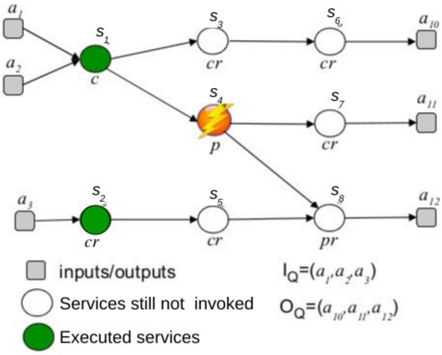


Checkpointing[13] as recovery technique

[13] M. Rukoz et al. Faceta*: Checkpointing for transactional composite web service execution based on petri-nets. *Procedia Computer Science*, 2012

Self-healing transactional composite service

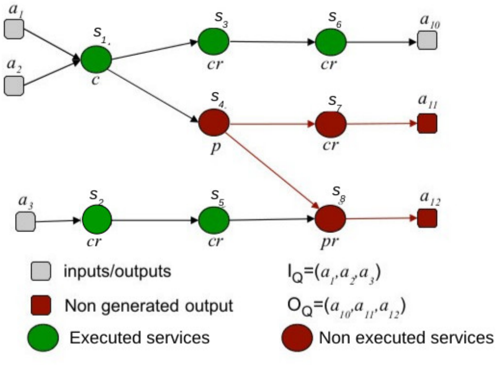
As part of R.Angarita's PhD work ICSOC'2014 [8] and MEDES'2016 [7]



An atomic composite service after a component failure

Self-healing transactional composite service

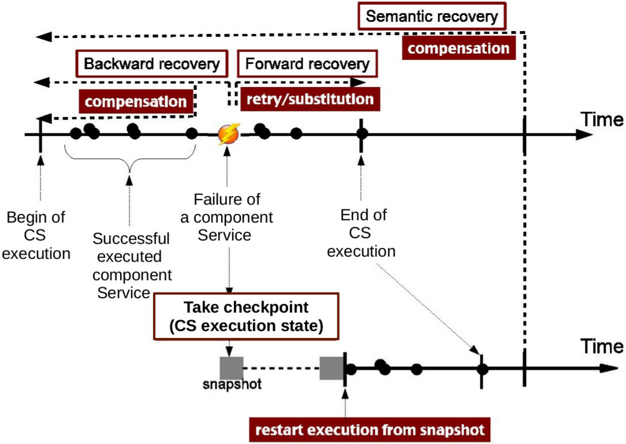
As part of R.Angarita's PhD work ICSOC'2014 [8] and MEDES'2016 [7]



A checkpointed composite service

Self-healing transactional composite service

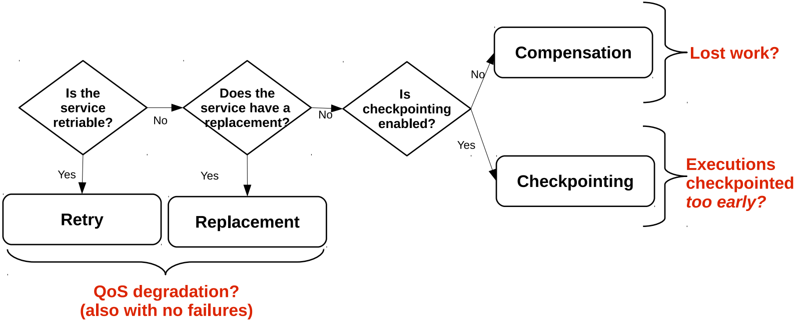
As part of R.Angarita's PhD work ICSOC'2014 [8] and MEDES'2016 [7]



Recovery techniques for composite service

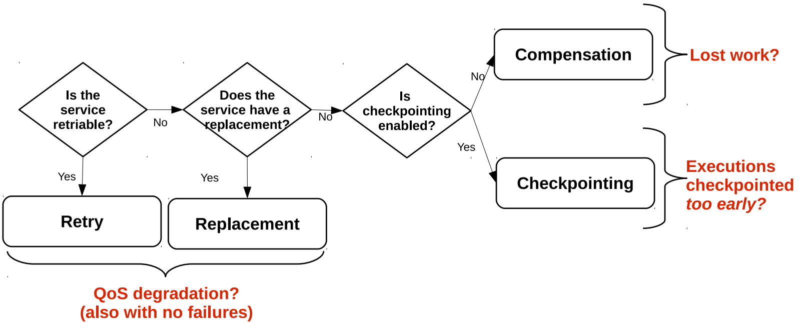
Outline

Which mechanism is the best one when a failure occurs?



Problems of the Static Selection of Fault Tolerance Strategies

Which mechanism is the best one when a failure occurs?



$\Longrightarrow$ A self-healing composite service approach ensuring QoS criteria, based on: (i) transactional properties, (ii) checkpointing, (iii) knowledge, and (iv) user QoS preferences

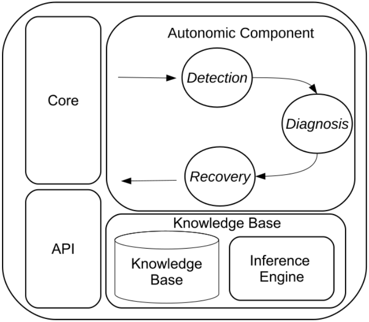
Outline

Our system:

- A Coordinator Agent in charge of analysing the composite service to be executed
- A set of self-healing service agents (software components) participating in the same composite service execution:
  - in charge of the execution control of a component service s_i in a composition
  - having some knowledge about its component service (estimated QoS, transactional property and list of functionally equivalent services)
  - making decisions based on:
    - **what it is expected to happen**,
    - **what really happened** before and during the invocation of its service s_i, and
    - **what it remains to happen** after the invocation of its service

Service Agent self-healing performs:

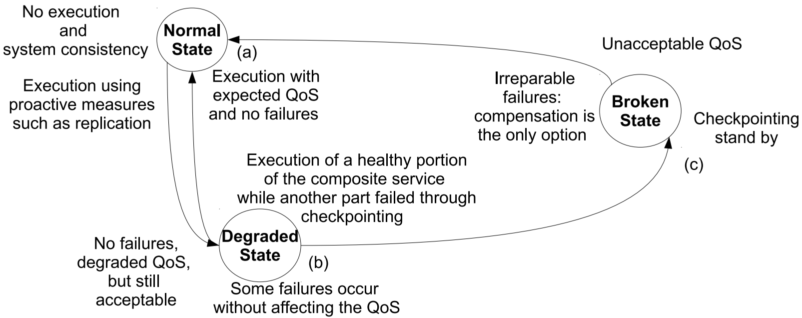
- **detection** based on QoS degradation or service failure;
- **diagnosis** based on CS state, the available recovery mechanism, and user preferences; and
- **recovery** by applying selected actions



Service Agent Architecture

Service Agent self-healing performs:

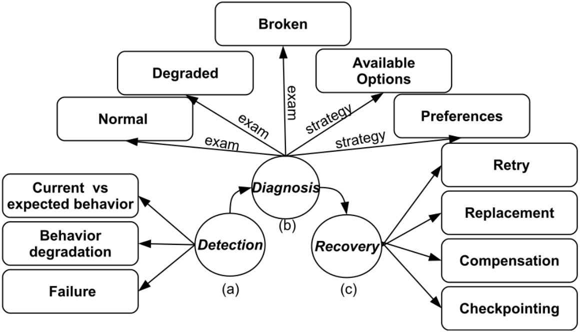
- **detection** based on QoS degradation or service failure;
- **diagnosis** based on CS state, the available recovery mechanism, and user preferences; and
- **recovery** by applying selected actions



Self-healing States for Composite Services

Service Agent self-healing performs:

- **detection** based on QoS degradation or service failure;
- **diagnosis** based on CS state, the available recovery mechanism, and user preferences; and
- **recovery** by applying selected actions



Service Agent Self-healing Loop

Service Agent self-healing performs:

- **detection** based on QoS degradation or service failure;
- **diagnosis** based on CS state, the available recovery mechanism, and user preferences; and
- **recovery** by applying selected actions

In IoTBD'2016 [6]:

Application of our approach to Web of Things (WoT), where WoT agents represent physical objects, Web services, or humans in WoT applications

Outline

1. Composite service selection:
   - Top-down [3,4,12]:
     - User query expressed by a workflow
     - Selection of a service for each activity of the workflow
     - No I/O link considered between services

1. Composite service selection:
   - Top-down [3,4,12]:
     - User query expressed by a workflow
     - Selection of a service for each activity of the workflow
     - Links between services not considered
   - Bottom-up [2,9,10,14,15]:
     - User query expressed by a pair of I/O sets
     - Selection a subgraph in the graph representing services linked by their I/O
     - No expression of the business process wanted by the user

1. Composite service selection:
   - Top-down [3,4,12]:
     - User query expressed by a workflow
     - Selection of a service for each activity of the workflow
     - Links between services not considered
   - Bottom-up [2,9,10,14,15]:
     - User query expressed by a pair of I/O sets
     - Selection a subgraph in the graph representing services linked by their I/O
     - No expression of the business process wanted by the user

   $\Longrightarrow$ Combining both approaches

1. Composite service selection: combining top-down and bottom-up approaches
2. Robust dynamic service composition:
   - Two separate steps:
     - Selection process [2,3,49,10,12,14,15]
     - Self-healing execution of the selected composite service [5-8]

   ⟹ Taking into account QoS degradation or component failures during the selection process

1. Composite service selection: combining top-down and bottom-up approaches
2. Robust dynamic service composition: integrating recovery strategies into the selection process
3. Transactional properties issue:
   - Definition of a formal model of transactional composite services [3]
   - Extended by several authors [Ding et al., 2015; FanJiang et al., 2014; Rajaram et al., 2014; Wu et al. 2013]

---

[14] Z. Ding, et al. A transaction and qos-aware service selection approach based on genetic algorithm. *IEEE Trans. on SMC* 45(7), 2015

[15] Y.-Y. FanJiang et al. Semantic-based automatic service composition with functional and non-functional requirements in design time: A genetic algorithm approach. *Information and Software Tech.*, 56(3) 2014

[16] K. Rajaram et al. Deriving reliable compositions using cancelable web services. *ACM SIGSOFT Soft. Eng. Notes*, 39(1), 2014

[17] Q. Wu aet al. Transactional and QoS-aware dynamic service composition based on ant colony optimization. *Future Generation Computer Sys.*, 29(5), 2013

1. Composite service selection: combining top-down and bottom-up approaches
2. Robust dynamic service composition: integrating recovery strategies into the selection process
3. Transactional properties issue:
   - ? How providing real services with such transactional properties?
   - ? How implementing retriable property?
   - ? How measuring the atomicity of a composite service based on transactional properties?

1. Composite service selection: combining top-down and bottom-up approaches
2. Robust dynamic service composition: integrating recovery strategies into the selection process
3. Transactional properties issue:
   - ? How providing real services with such transactional properties?
   - ? How implementing retriable property?
   - ? How measuring the atomicity of a composite service based on transactional properties?

$\Longrightarrow$ First step to answer such questions in [11]: Measuring Fuzzy Atomicity for Composite Service Execution

1. Composite service selection: combining top-down and bottom-up approaches
2. Robust dynamic service composition: integrating recovery strategies into the selection process
3. Transactional properties issue: Measuring Fuzzy Atomicity for Composite Service Execution
4. A full transactional service management must be able :
   a) to discover/identify services achieving a specific goal,
   b) to compose services when no single service reach the needed goal,
   c) to ensure a reliable composite service execution

$\Longrightarrow$ Challenges: integrating all approaches and providing/finding a registry of real services

1. Composite service selection: combining top-down and bottom-up approaches
2. Robust dynamic service composition: integrating recovery strategies into the selection process
3. Transactional properties issue: Measuring Fuzzy Atomicity for Composite Service Execution
4. Toward a full transactional service management

Thank you! Questions ?

References

[1] Y. Cardinale, J. El Haddad, M. Manouvrier, and M. Rukoz. Web Service Composition Based on Petri Nets: Review and Contribution. In *RED* – LNSC 8194, 2013 – *Volume containing extended papers of the works presented at 5th Int. Workshop RED 2012, through a two-step peer-review process*

[2] Y. Cardinale, J. E. Haddad, M. Manouvrier, and M. Rukoz. CPN-TWS: a coloured petri-net approach for transactional-QoS driven Web Service composition. *IJWGS* 7(1), 2011 – *Impact Factor: 1.919*

[3] J. El Haddad, M. Manouvrier, and M. Rukoz. TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition. *IEEE TSC* 3(1), 2010 – - *Selectivity rate: 21%*

[5] V. Gabrel, M. Manouvrier, and C. Murat. Web services composition: Complexity and models. *Discrete Applied Mathematics* 196, 2015

[6] R. Angarita, M. Manouvrier, and M. Rukoz. Service Selection Based on Crowdsourcing. In *MobiWis* - LNCS 9228, 2015

[7] R. Angarita, M. Manouvrier, and M. Rukoz. An Agent Architecture to Enable Self-healing and Application-context-aware Web of Things Applications. In *IoTBD*, 2016

[8] R. Angarita, M. Rukoz, and M. Manouvrier. Dynamic Composite Web Service. In *ICSOC*, 2014 – *CORE 2014 A-ranked conf., PhD Symposium*

[9] E. Blanco, Y. Cardinale, M.-E. Vidal, J. El Haddad, M. Manouvrier, and M. Rukoz. A transactional-QoS driven approach for web service composition. In *RED* - LNCS 6799, 2012

[10] Y. Cardinale, J. El Haddad, M. Manouvrier, and M. Rukoz. Web service selection for transactional composition. In *ICCS*, 2010 – *ERA 2010 A-ranked conf.*

[11] Y. Cardinale, J. El Haddad, M. Manouvrier, and M. Rukoz. Measuring Fuzzy Atomicity for Composite Service Execution. In *OBD*, 2016 – *Invited paper*

[12] J. El Haddad, M. Manouvrier, G. Ramirez, and M. Rukoz. Qos-driven selection of web services for transactional composition. In *IEEE ICWS*, 2008 – *Selectivity rate 16%*

References

[13] J. El Haddad, M. Manouvrier, and M. Rukoz. A Hierarchical Model for Transactional Web Service Composition in P2P Networks. In *IEEE ICWS*, 2007 – *Selectivity rate 18%*

[14] V. Gabrel, M. Manouvrier, I. Megdiche, and C. Murat. A new 0-1 linear program for QoS and transactional-aware web service composition. In *IEEE ISCC*, 2012

[15] V. Gabrel, M. Manouvrier, and C. Murat. Optimal and Automatic Transactional Web Service Composition with Dependency Graph and 0-1 Linear Programming. In *ICSOC* - LNCS 8831, 2014 – *CORE 2014 A-ranked conf., Selectivity rate 15%*

[16] M. Mouhoub, D. Grigori, and M. Manouvrier. A Framework for Searching Semantic Data and Services with SPARQL. In *ICSOC* - LNCS 8831, 2014 – *CORE 2014 A-ranked conf., Selectivity rate 15%*

[17] M. Mouhoub, D. Grigori, and M. Manouvrier. LIDSEARCH: A SPARQL-Driven Framework for Searching Linked Data and Semantic Web Services. In *ESWC* - LNCS 9341, 2015 – *CORE 2014 A-ranked conf., Demo Session* n Non-Functional Properties for Service-oriented Systems: Future Directions ,

[18] Y. Cardinale, J. El Haddad, M. Manouvrier, and M. Rukoz. chapter Transactional-aware web service composition: A survey. *IGI Global-Advances in Knowledge Management (AKM) Book Series* - S. Reiff-Marganiec and M. Tilly Ed(s)., 2011 – *Chapter accepted after a peer-review process*