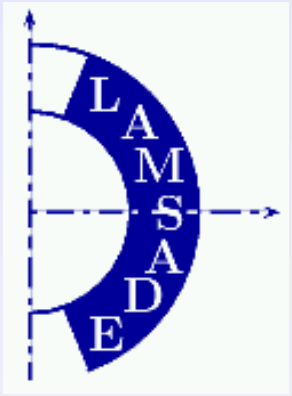


Laboratoire d'Analyse et Modélisation
de Systèmes pour l'Aide à la DEcision

QoS-driven Selection of Web Services for Transactional Composition

Joyce El Haddad, Maude Manouvrier, Guillermo Ramirez, Marta Rukoz

Université Paris Dauphine, Universidad Central de Venezuela, Université Paris X-Nanterre

IEEE International Conference on Web Services

September 23-26, 2008

Outline

Goals

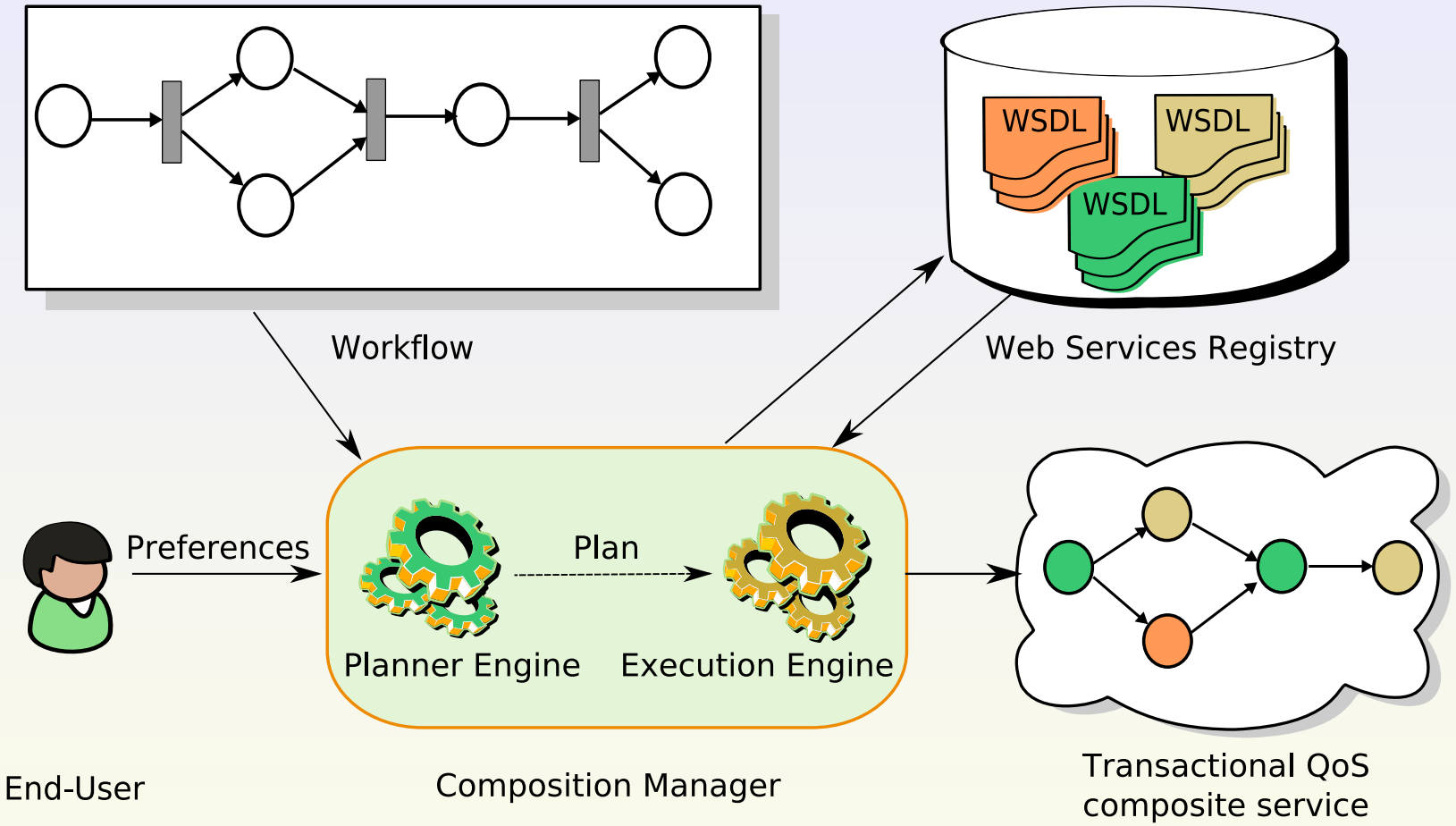
1. To support user-tailored composition of web services
2. To exploit the transactional properties of the component Web services to derive the transactional properties for the composite Web service
3. To select the best QoS component Web services

Our contribution

A selecting algorithm for composing Web services not only according to their functional requirements but also to their transactional properties and QoS characteristics

Outline

System Architecture



Web Service Description

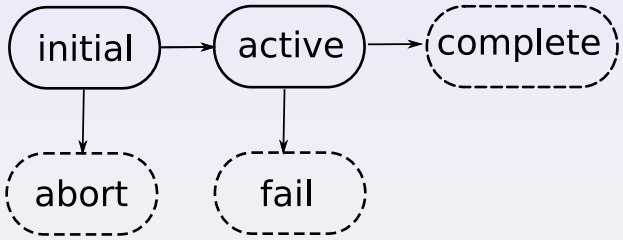
1. Web service behavioral properties

Web Service Description

1. Web service behavioral properties
   - transactional behavior

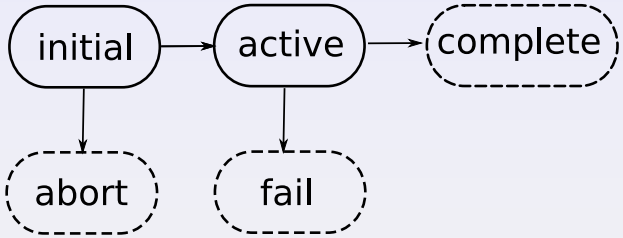
Web Service Description

1. Web service behavioral properties
   - transactional behavior
     - pivot (p)

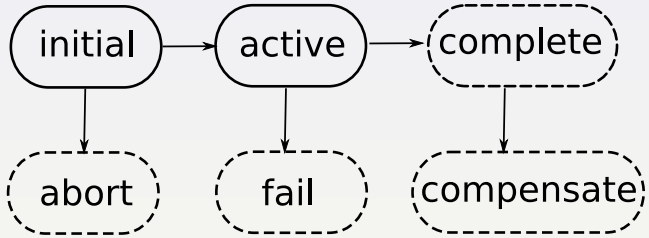


Web Service Description

1. Web service behavioral properties
   - transactional behavior
     - pivot (p)

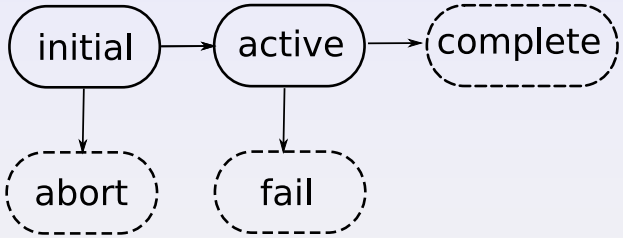


     - compensatable (c)

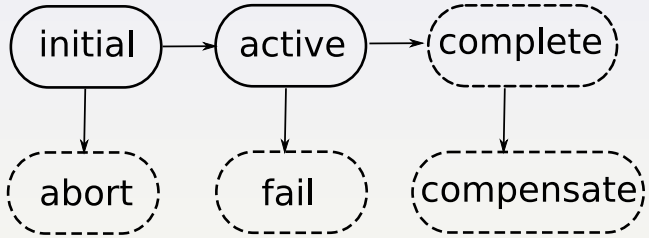


Web Service Description

1. Web service behavioral properties
   - transactional behavior
     - pivot (p)



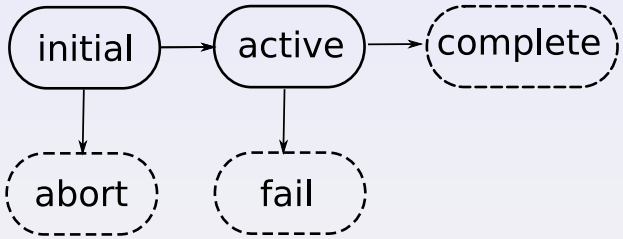
     - compensatable (c)



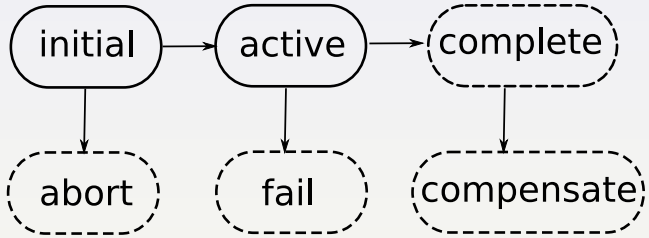
   - non-transactional behavior
     - retriable (r)

Web Service Description

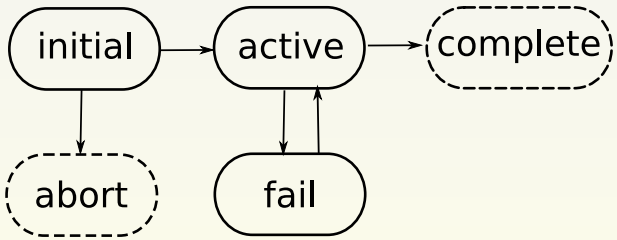
1. Web service behavioral properties
   - transactional behavior
     - pivot (p)



     - compensatable (c)



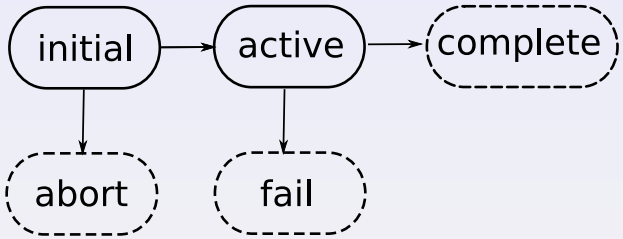
   - non-transactional behavior
     - retriable (r)



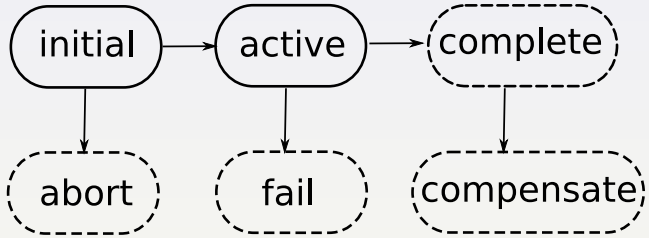
Pivot and retriable service

Web Service Description

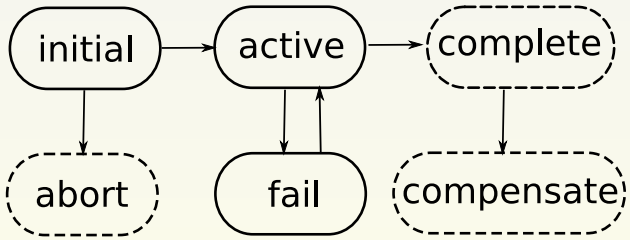
1. Web service behavioral properties
   - transactional behavior
     - pivot (p)



     - compensatable (c)



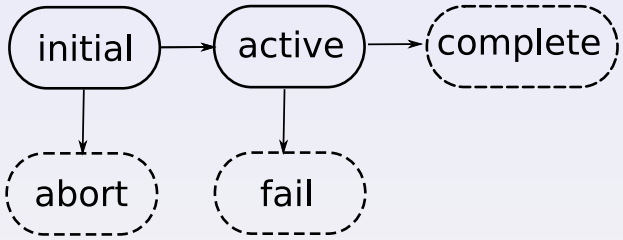
   - non-transactional behavior
     - retriable (r)



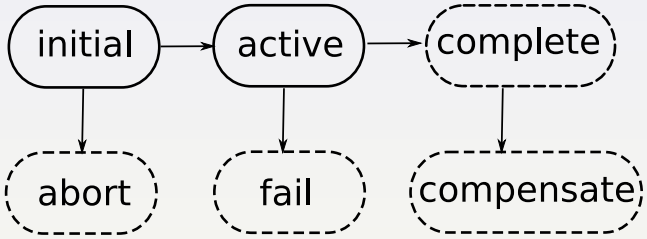
Compensatable and retriable service

Web Service Description

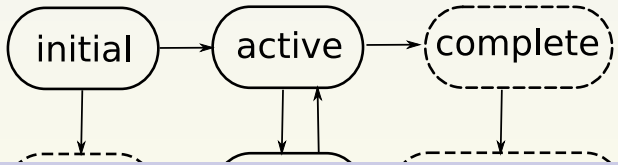
1. Web service behavioral properties
   - transactional behavior
     - pivot (p)



     - compensatable (c)



   - non-transactional behavior
     - retriable (r)



Behavioral properties

The set of all possible combinations for the behavioral property of a Web service is $\{p; c; pr; cr\}$

Web Service Description

2. Web service non-functional properties
   - Price ($q_{ep}(s)$) : the fee that a requester has to pay for invoking of the Web service s
   - Duration ($q_{ed}(s)$) : the measure of the expected delay time between the moment when a requester of Web service s is sent and when the results are received
   - Reputation ($q_r(s)$): the measure of trustworthiness of service s, generally this measure is defined as the average ranking given to the service by end users
   - Successful execution rate ($q_{sr}(s)$) : the probability that service s responds correctly to the user request
   - Availability ($q_a(s)$) : the probability that a service s is accessible

Composite transactional model

L. Li, C. Liu and J. Wang
Deriving Transactional Properties of Composite Web Services
IEEE International Conference on Web Services *(ICWS)*, July, 2007.

Atomic workflow ($\vec{a}$)

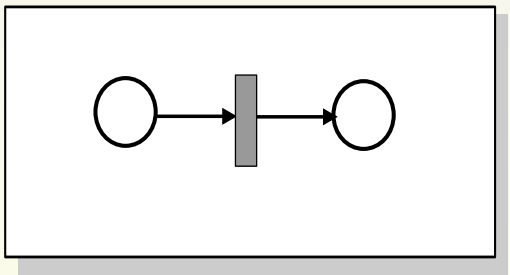
- All the activities complete successfully $\Longrightarrow$ their effect remain forever and cannot be semantically undone
- One activity fails $\Longrightarrow$ all previous activities compensate

Composite transactional model

L. Li, C. Liu and J. Wang
Deriving Transactional Properties of Composite Web Services
IEEE International Conference on Web Services *(ICWS)*, July, 2007.

Atomic workflow ($\vec{a}$)

- All the activities complete successfully $\Longrightarrow$ their effect remain forever and cannot be semantically undone
- One activity fails $\Longrightarrow$ all previous activities compensate

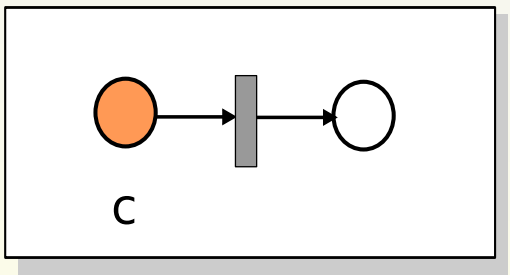


Composite transactional model

L. Li, C. Liu and J. Wang
Deriving Transactional Properties of Composite Web Services
IEEE International Conference on Web Services *(ICWS)*, July, 2007.

Atomic workflow ($\vec{a}$)

- All the activities complete successfully $\Longrightarrow$ their effect remain forever and cannot be semantically undone
- One activity fails $\Longrightarrow$ all previous activities compensate

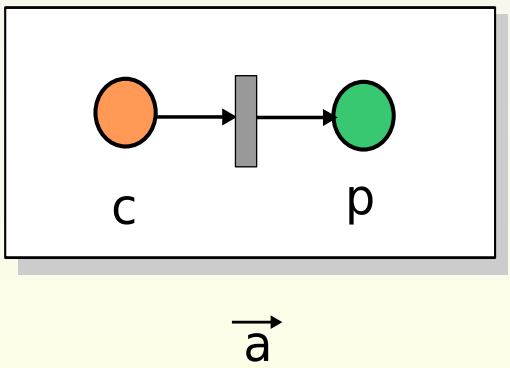


Composite transactional model

L. Li, C. Liu and J. Wang
Deriving Transactional Properties of Composite Web Services
IEEE International Conference on Web Services *(ICWS)*, July, 2007.

Atomic workflow ($\vec{a}$)

- All the activities complete successfully $\Longrightarrow$ their effect remain forever and cannot be semantically undone
- One activity fails $\Longrightarrow$ all previous activities compensate

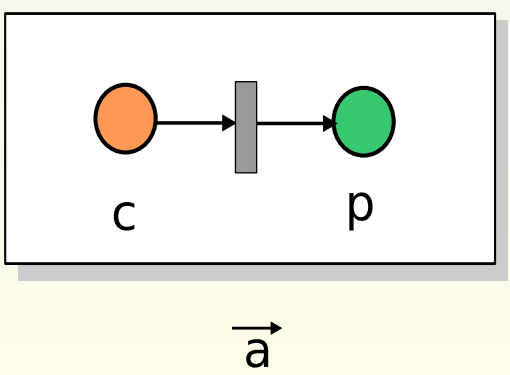


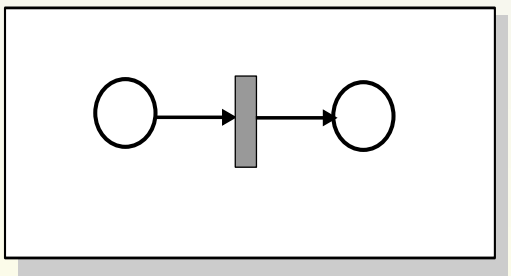
Composite transactional model

L. Li, C. Liu and J. Wang
Deriving Transactional Properties of Composite Web Services
IEEE International Conference on Web Services *(ICWS)*, July, 2007.

Atomic workflow ($\vec{a}$)

- All the activities complete successfully $\Longrightarrow$ their effect remain forever and cannot be semantically undone
- One activity fails $\Longrightarrow$ all previous activities compensate



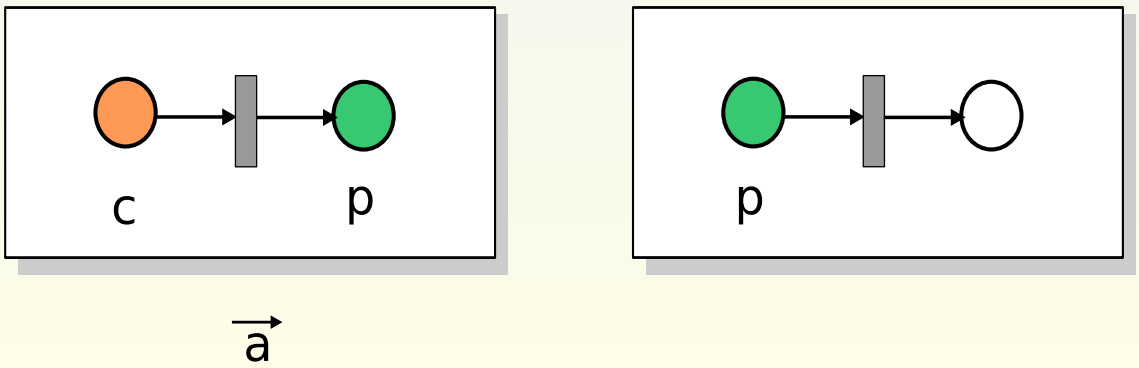


Composite transactional model

L. Li, C. Liu and J. Wang
Deriving Transactional Properties of Composite Web Services
IEEE International Conference on Web Services *(ICWS)*, July, 2007.

Atomic workflow ($\vec{a}$)

- All the activities complete successfully $\Longrightarrow$ their effect remain forever and cannot be semantically undone
- One activity fails $\Longrightarrow$ all previous activities compensate

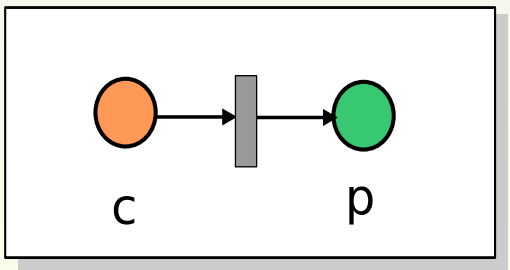


Composite transactional model

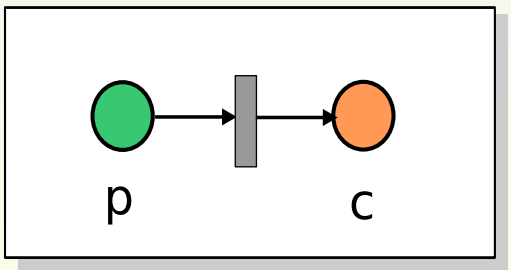
L. Li, C. Liu and J. Wang
Deriving Transactional Properties of Composite Web Services
IEEE International Conference on Web Services *(ICWS)*, July, 2007.

Atomic workflow ($\vec{a}$)

- All the activities complete successfully $\Longrightarrow$ their effect remain forever and cannot be semantically undone
- One activity fails $\Longrightarrow$ all previous activities compensate



$\overrightarrow{a}$



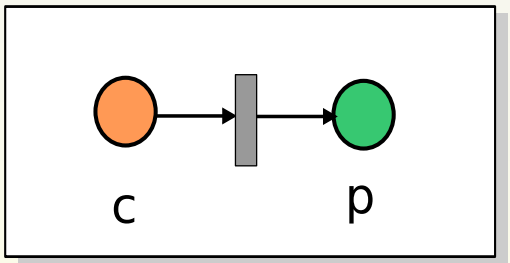
$\widetilde{a}$

Composite transactional model

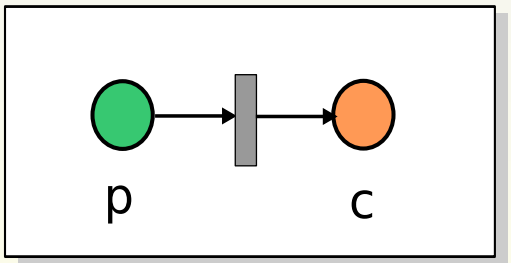
L. Li, C. Liu and J. Wang
Deriving Transactional Properties of Composite Web Services
IEEE International Conference on Web Services *(ICWS)*, July, 2007.

Atomic workflow ($\vec{a}$)

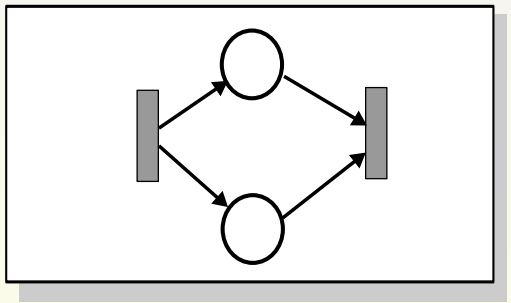
- All the activities complete successfully $\Longrightarrow$ their effect remain forever and cannot be semantically undone
- One activity fails $\Longrightarrow$ all previous activities compensate



$\overrightarrow{a}$



$\widetilde{a}$

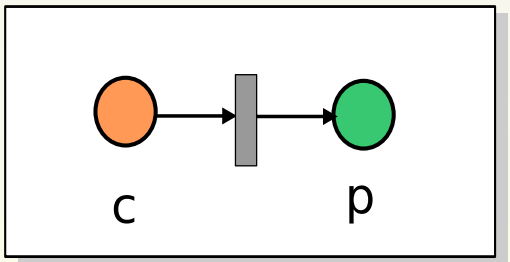


Composite transactional model

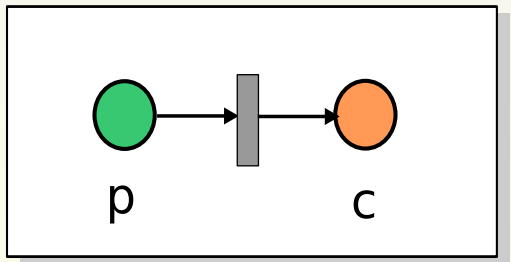
L. Li, C. Liu and J. Wang
Deriving Transactional Properties of Composite Web Services
IEEE International Conference on Web Services *(ICWS)*, July, 2007.

Atomic workflow ($\vec{a}$)

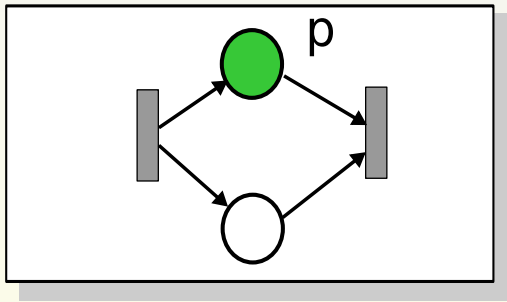
- All the activities complete successfully $\Longrightarrow$ their effect remain forever and cannot be semantically undone
- One activity fails $\Longrightarrow$ all previous activities compensate



$\overrightarrow{a}$



$\widetilde{a}$

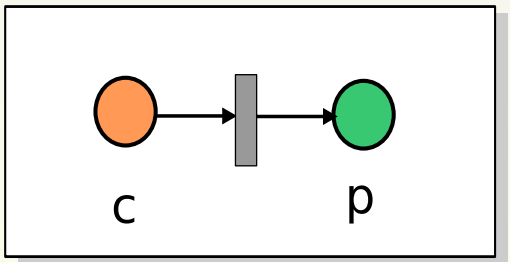


Composite transactional model

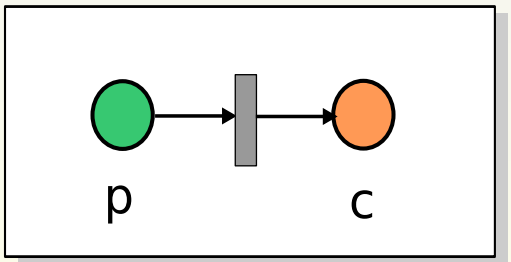
L. Li, C. Liu and J. Wang
Deriving Transactional Properties of Composite Web Services
IEEE International Conference on Web Services *(ICWS)*, July, 2007.

Atomic workflow ($\vec{a}$)

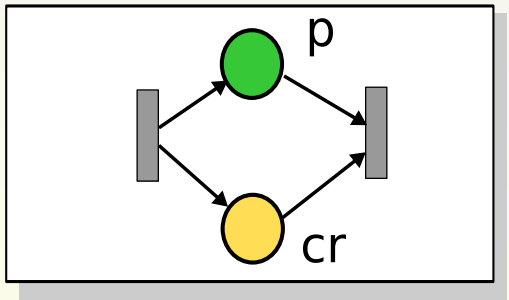
- All the activities complete successfully $\Longrightarrow$ their effect remain forever and cannot be semantically undone
- One activity fails $\Longrightarrow$ all previous activities compensate



$\overrightarrow{a}$



$\widetilde{a}$



$\overrightarrow{a}$

Composite transactional model

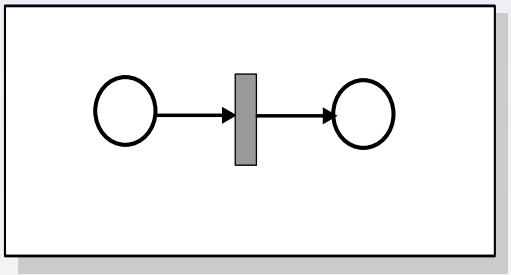
Compensatatable workflow (c)

A workflow is compensatable if all its activities can be compensated

Composite transactional model

Compensatatable workflow (c)

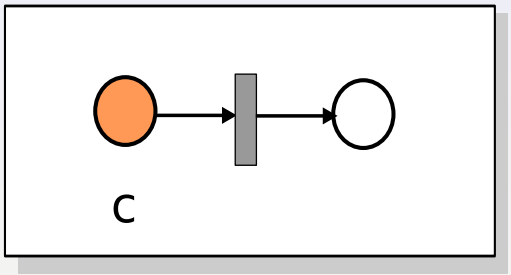
A workflow is compensatable if all its activities can be compensated



Composite transactional model

Compensatatable workflow (c)

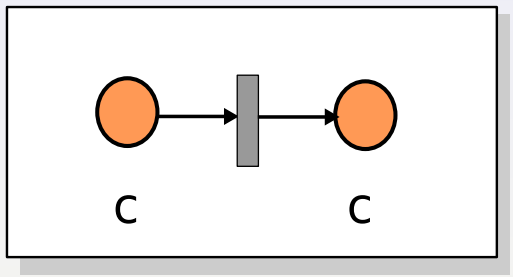
A workflow is compensatable if all its activities can be compensated



Composite transactional model

Compensatatable workflow (c)

A workflow is compensatable if all its activities can be compensated

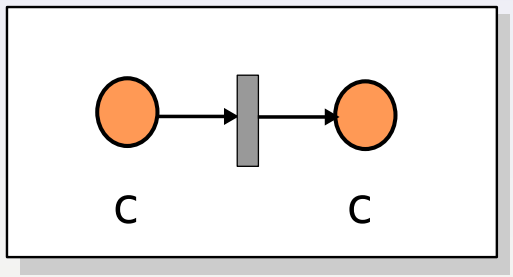


C

Composite transactional model

Compensatatable workflow (c)

A workflow is compensatable if all its activities can be compensated



c

Transactional Composite Web Service (TCWS)

A TCWS is a worklfow that can be atomic or compensatable $\Longrightarrow$ The set of behavioral property of a TCWS is $\{\vec{a}; pr; c; cr\}$

Composite quality model

L. Zeng, B.Benatallah, M.Dumas, J.Kalagnanam and H.Chang
QoS-Aware Middleware for Web Services Composition
IEEE Transactions on Software Engineering, 30(5):311–327, May 2004.

Criteria	**Aggregation function**
Price	$q_{ep}(CWS) = \sum_{i=1}^{n} q_{ep}(s_i)$
Duration	$q_{ed}(CWS) = \sum_{i=1}^{n} q_{ed}(s_i)$
Reputation	$q_r(CWS) = \frac{1}{n} \sum_{i=1}^{n} q_r(s_i)$
Success rate	$q_{sr}(CWS) = \prod_{i=1}^{n} q_{sr}(s_i)$
Availability	$q_a(CWS) = \prod_{i=1}^{n} q_a(s_i)$

$Score(s_i) = \sum w_j q_{ij}$, where $w_j \in [0,1]$ is the weight assigned to the quality criterion, $\sum w_j = 1$ and q_{ij} is the value of criterion j

Outline

Definition of risk

The users can express their transactional criteria

- *Risk* 0 : the system guarantees that if the execution is successful, the obtained results can be compensated by the user

Definition of risk

The users can express their transactional criteria

- *Risk* 0 : the system guarantees that if the execution is successful, the obtained results can be compensated by the user $\Longrightarrow$ the selecting process generates a compensatable workflow

Definition of risk

The users can express their transactional criteria

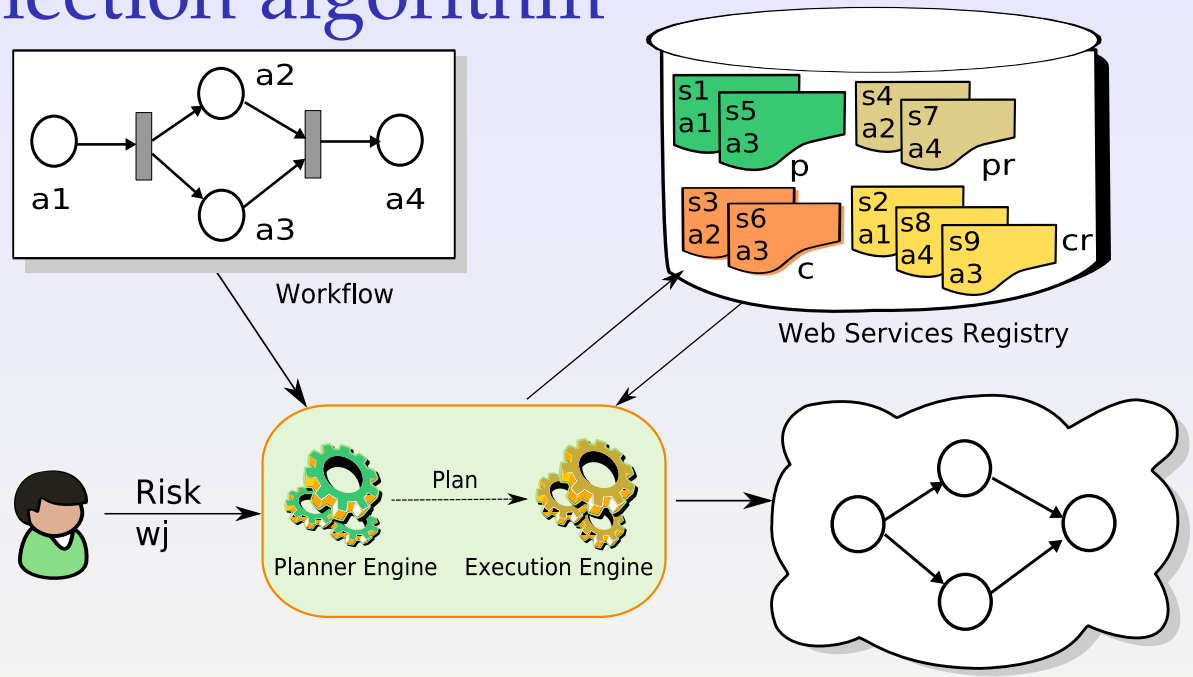
- *Risk* 0 : the system guarantees that if the execution is successful, the obtained results can be compensated by the user $\Longrightarrow$ the selecting process generates a compensatable workflow
- *Risk* 1 : the system does not guarantee the successful execution but if it achieves the results cannot be compensated by the user

Definition of risk

The users can express their transactional criteria

- *Risk* 0 : the system guarantees that if the execution is successful, the obtained results can be compensated by the user $\Longrightarrow$ the selecting process generates a compensatable workflow
- *Risk* 1 : the system does not guarantee the successful execution but if it achieves the results cannot be compensated by the user $\Longrightarrow$ the selecting process generates an atomic workflow

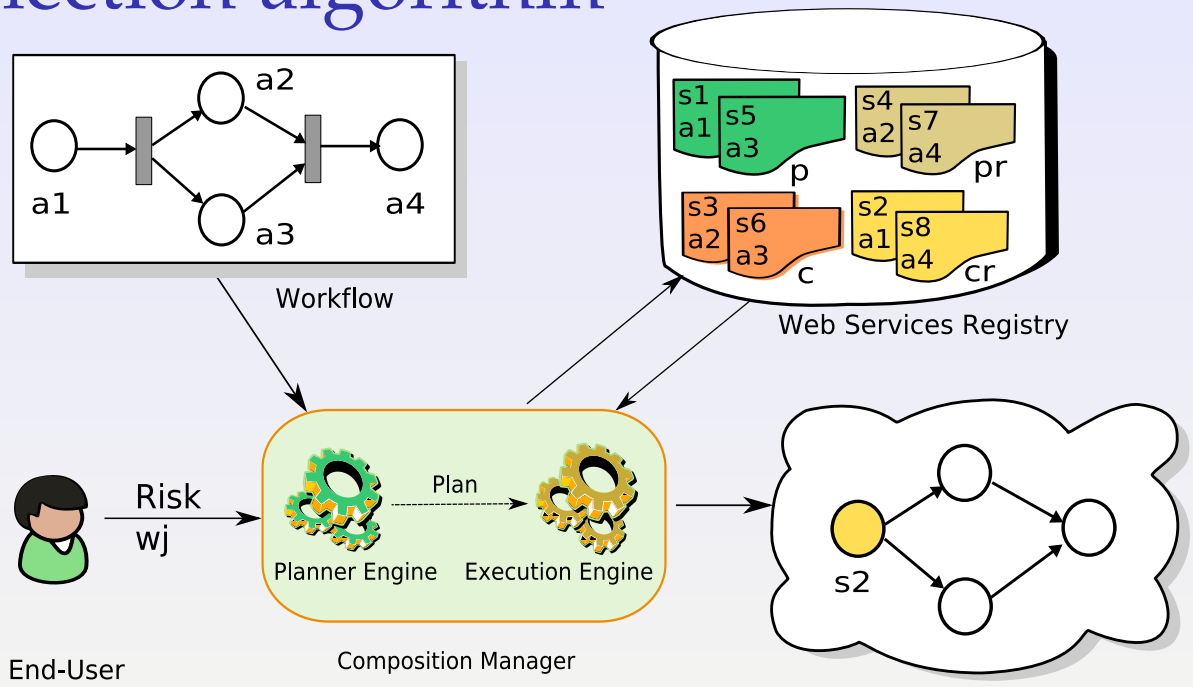
Service selection algorithm



Risk 0

$\forall\ a_i, getBestQoS(c \bigcup cr)$

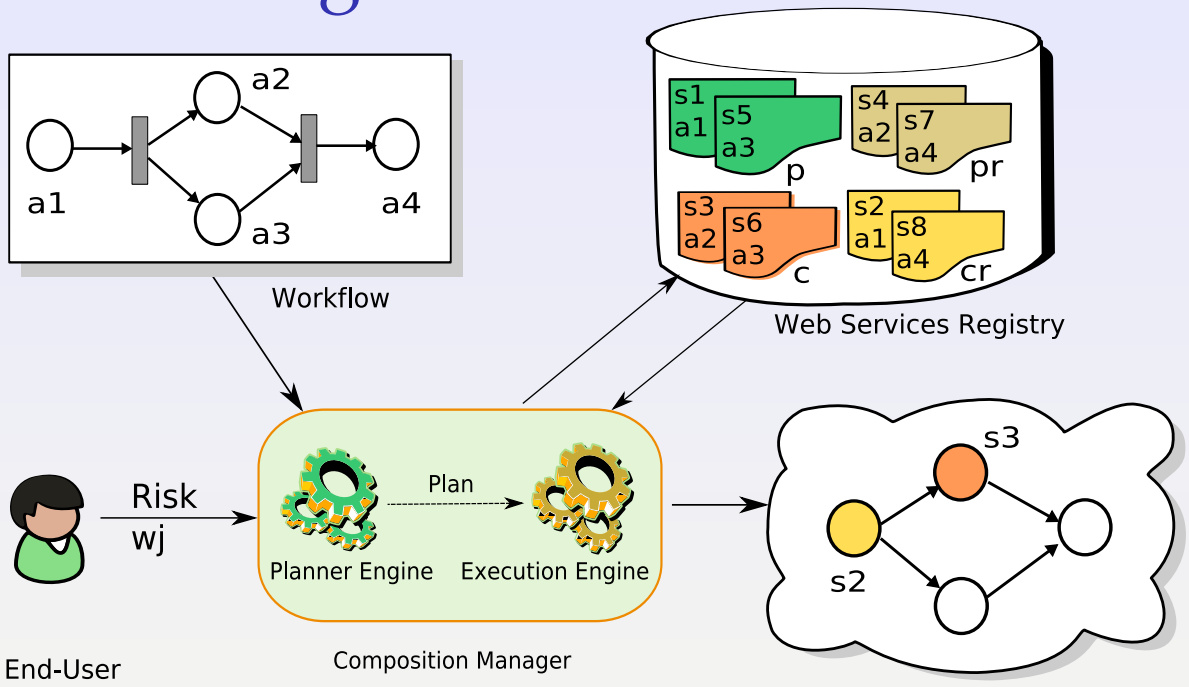
Service selection algorithm



Risk 0

$\forall\ a_i, getBestQoS(c \bigcup cr)$

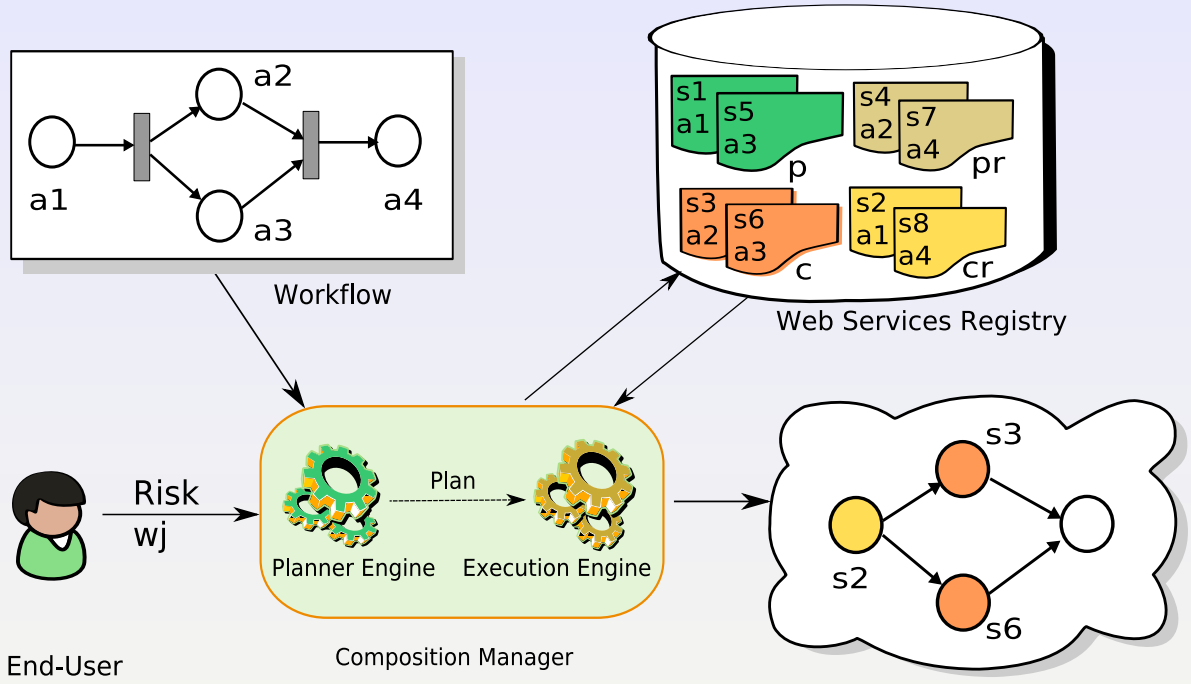
Service selection algorithm



Risk 0

$\forall\ a_i, getBestQoS(c \bigcup cr)$

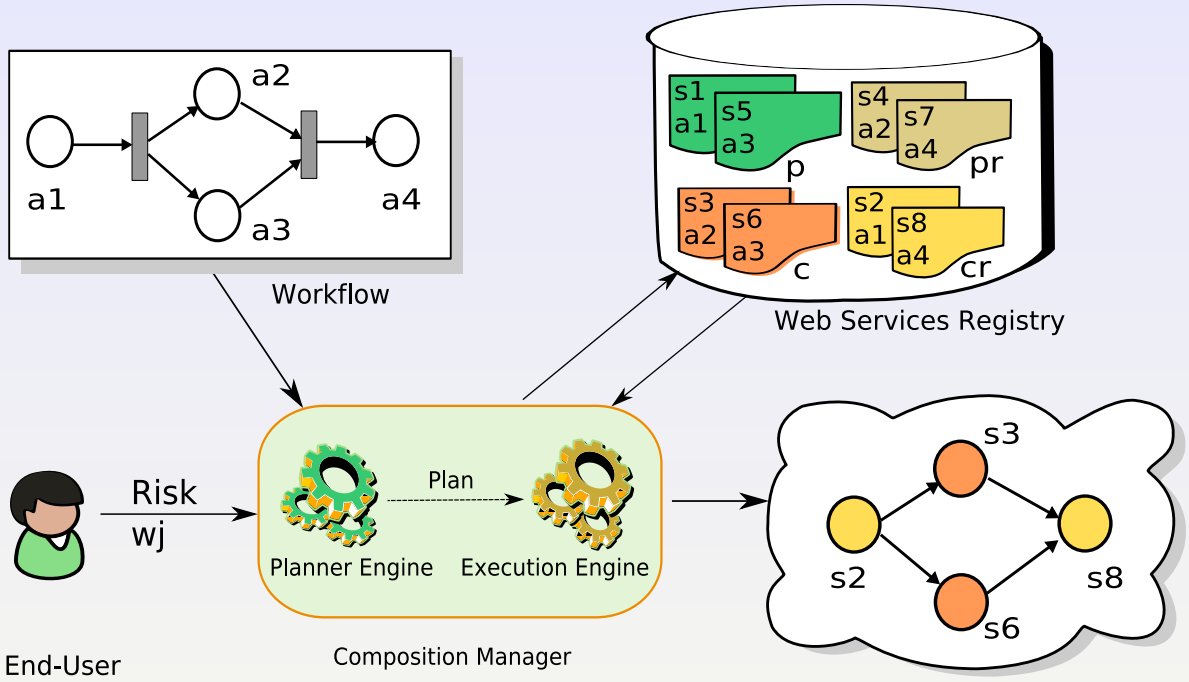
Service selection algorithm



Risk 0

$\forall\, a_i, getBestQoS(c \bigcup cr)$

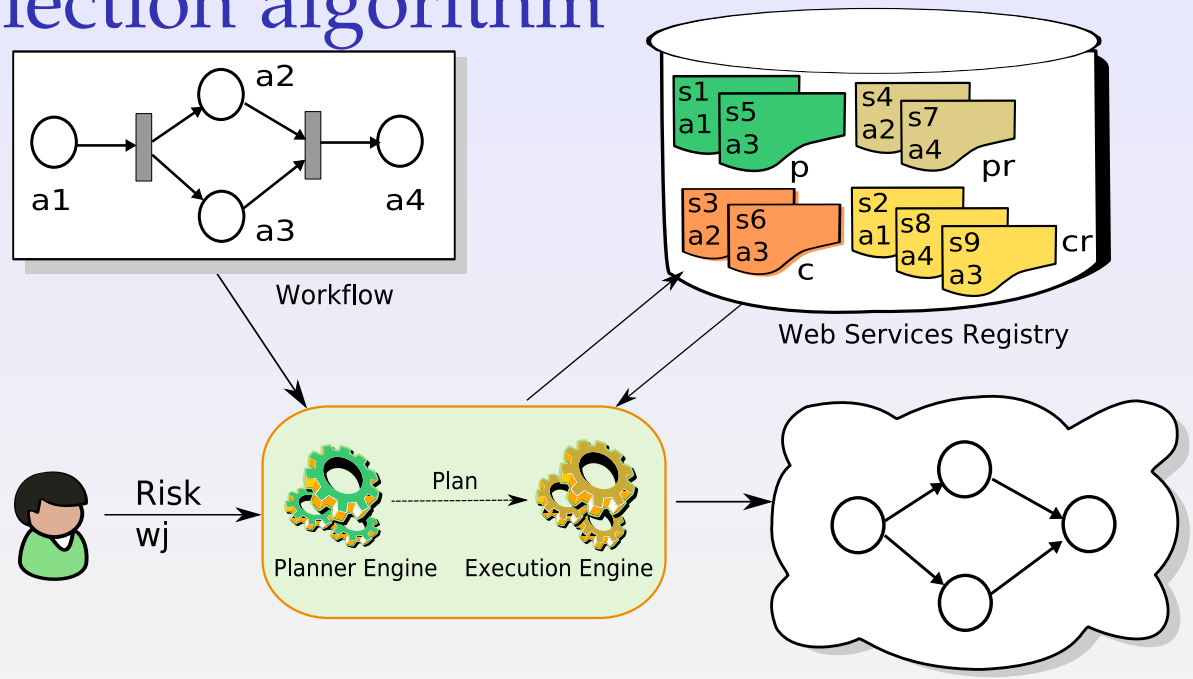
Service selection algorithm



Risk 0

$\forall\ a_i, getBestQoS(c \bigcup cr)$

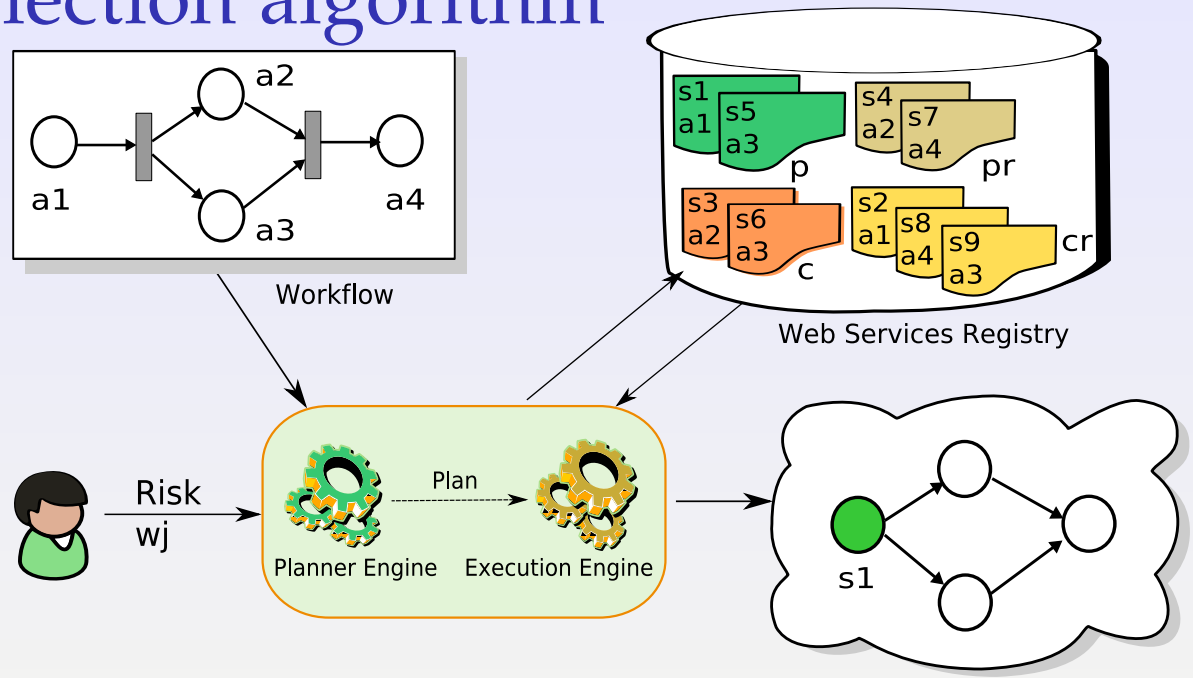
Service selection algorithm



Risk 1

- sequential path $(a_{i-1}; a_i)$: previously selected service has either
  - a transactional property in $\{p, pr\}$ or $nbp = 1 \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c, cr\} \Rightarrow$ for a_i, $getBestQoS(p \bigcup pr \bigcup c \bigcup cr)$
- concurrent path (split-pattern $\ldots a_i \ldots a_j$ join-pattern) : one previously selected service has either
  - a transactional property in $\{p\} \Rightarrow$ for a_i to a_j, $getBestQoS(cr)$
  - a transactional property in $\{pr\} \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c\} \Rightarrow$ for a_i to a_j, $getBestQoS(c \bigcup cr)$

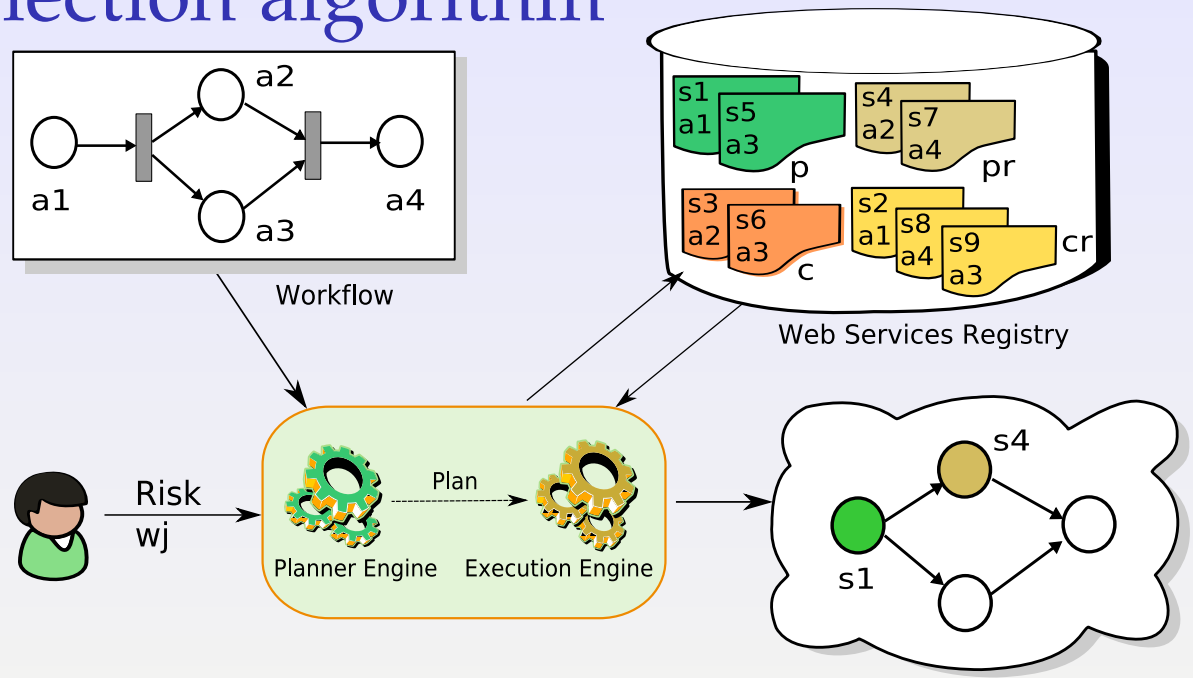
Service selection algorithm



Risk 1

- sequential path $(a_{i-1}; a_i)$: previously selected service has either
  - a transactional property in $\{p, pr\}$ or $nbp = 1 \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c, cr\} \Rightarrow$ for a_i, $getBestQoS(p \bigcup pr \bigcup c \bigcup cr)$
- concurrent path (split-pattern $\ldots a_i \ldots a_j$ join-pattern) : one previously selected service has either
  - a transactional property in $\{p\} \Rightarrow$ for a_i to a_j, $getBestQoS(cr)$
  - a transactional property in $\{pr\} \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c\} \Rightarrow$ for a_i to a_j, $getBestQoS(c \bigcup cr)$

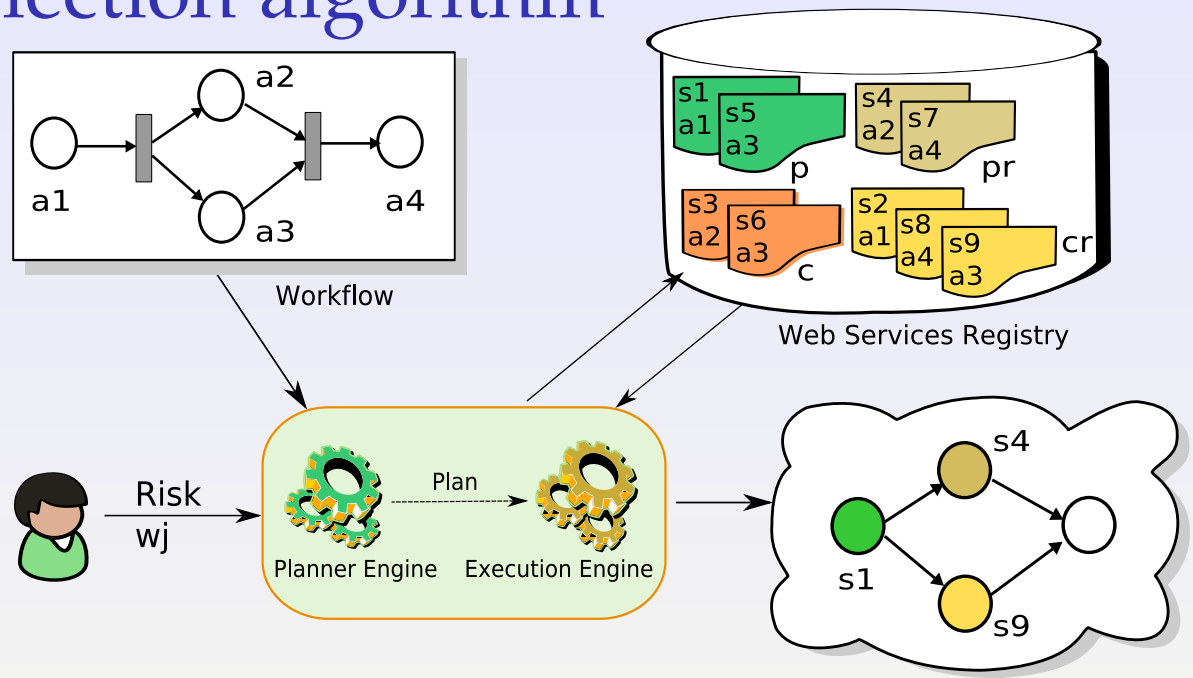
Service selection algorithm



Risk 1

- sequential path $(a_{i-1}; a_i)$: previously selected service has either
  - a transactional property in $\{p, pr\}$ or $nbp = 1 \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c, cr\} \Rightarrow$ for a_i, $getBestQoS(p \bigcup pr \bigcup c \bigcup cr)$
- concurrent path (split-pattern $\ldots a_i \ldots a_j$ join-pattern) : one previously selected service has either
  - a transactional property in $\{p\} \Rightarrow$ for a_i to a_j, $getBestQoS(cr)$
  - a transactional property in $\{pr\} \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c\} \Rightarrow$ for a_i to a_j, $getBestQoS(c \bigcup cr)$

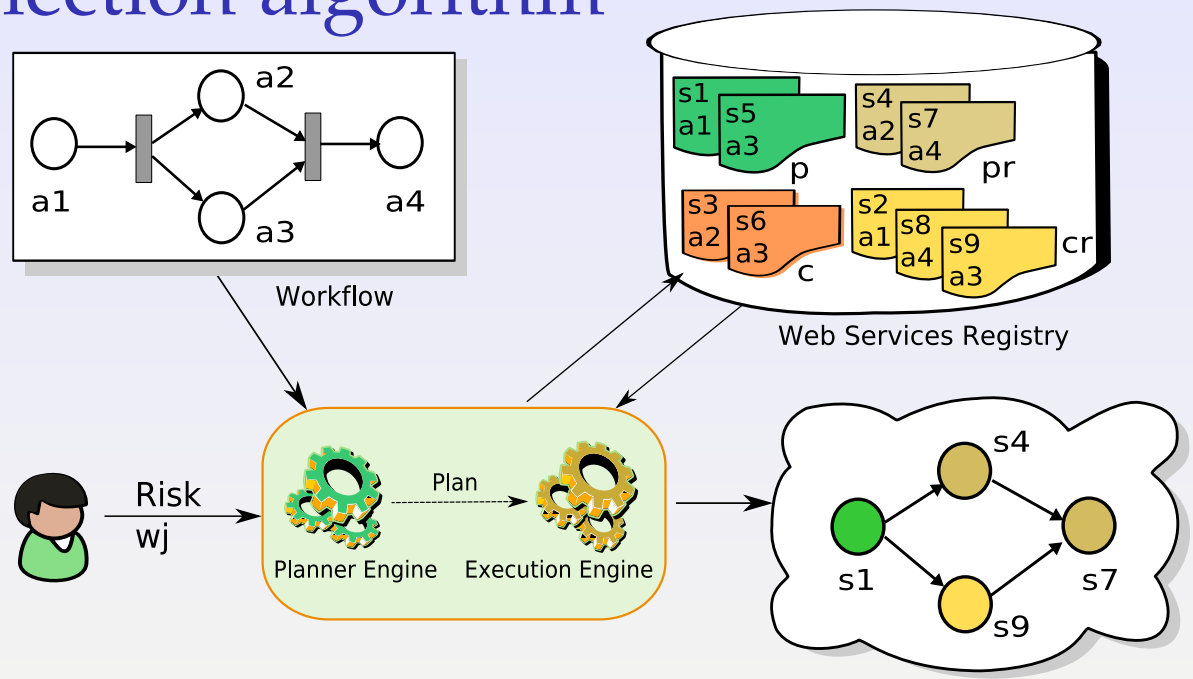
Service selection algorithm



Risk 1

- sequential path $(a_{i-1}; a_i)$: previously selected service has either
  - a transactional property in $\{p, pr\}$ or $nbp = 1 \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c, cr\} \Rightarrow$ for a_i, $getBestQoS(p \bigcup pr \bigcup c \bigcup cr)$
- concurrent path (split-pattern $\ldots a_i \ldots a_j$ join-pattern) : one previously selected service has either
  - a transactional property in $\{p\} \Rightarrow$ for a_i to a_j, $getBestQoS(cr)$
  - a transactional property in $\{pr\} \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c\} \Rightarrow$ for a_i to a_j, $getBestQoS(c \bigcup cr)$

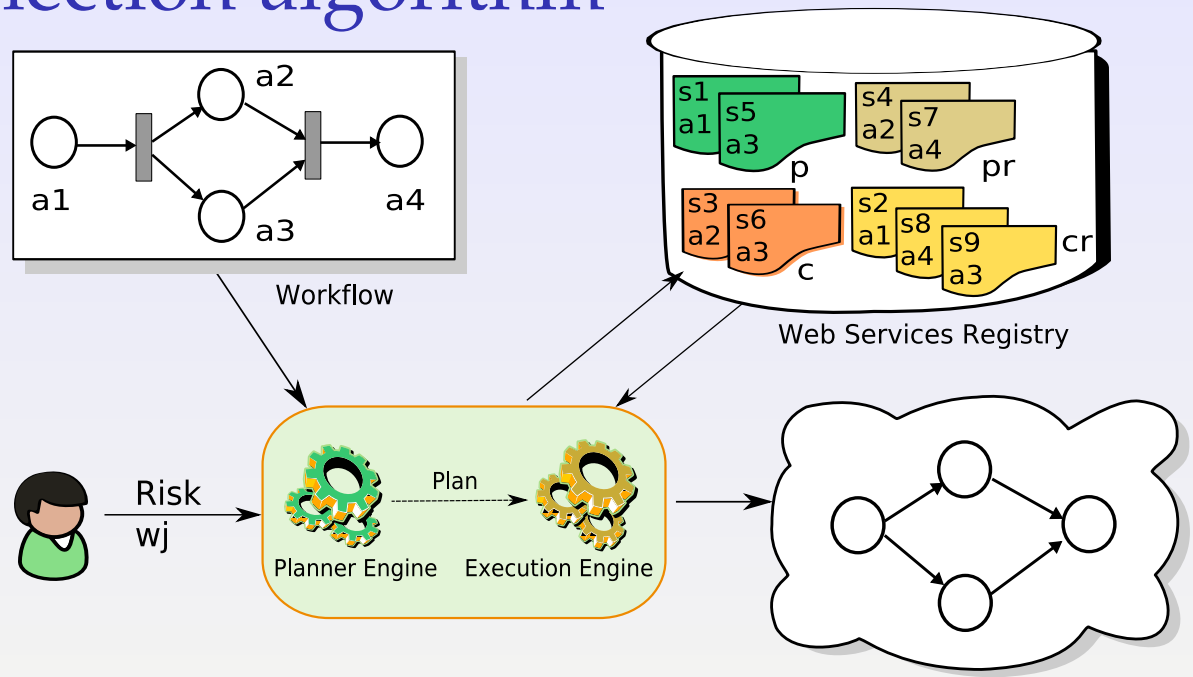
Service selection algorithm



Risk 1

- sequential path $(a_{i-1}; a_i)$: previously selected service has either
  - a transactional property in $\{p, pr\}$ or $nbp = 1 \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c, cr\} \Rightarrow$ for a_i, $getBestQoS(p \bigcup pr \bigcup c \bigcup cr)$
- concurrent path (split-pattern $\ldots a_i \ldots a_j$ join-pattern) : one previously selected service has either
  - a transactional property in $\{p\} \Rightarrow$ for a_i to a_j, $getBestQoS(cr)$
  - a transactional property in $\{pr\} \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c\} \Rightarrow$ for a_i to a_j, $getBestQoS(c \bigcup cr)$

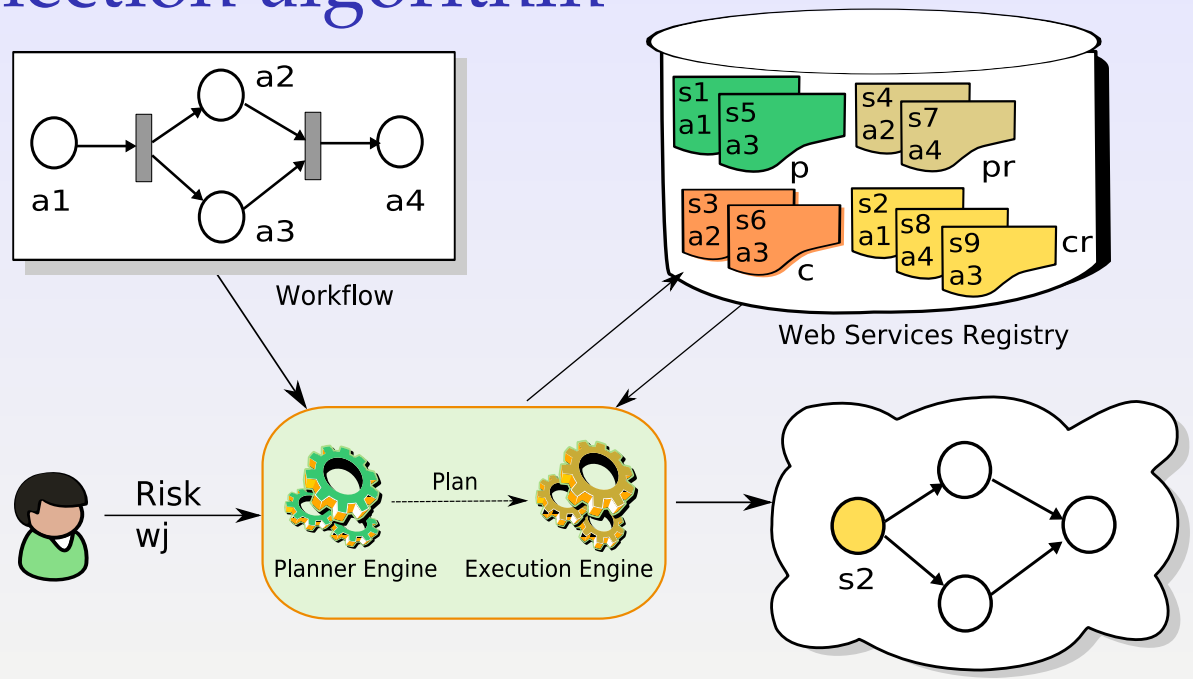
Service selection algorithm



Risk 1

- sequential path $(a_{i-1}; a_i)$: previously selected service has either
  - a transactional property in $\{p, pr\}$ or $nbp = 1 \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c, cr\} \Rightarrow$ for a_i, $getBestQoS(p \bigcup pr \bigcup c \bigcup cr)$
- concurrent path (split-pattern $\ldots a_i \ldots a_j$ join-pattern) : one previously selected service has either
  - a transactional property in $\{p\} \Rightarrow$ for a_i to a_j, $getBestQoS(cr)$
  - a transactional property in $\{pr\} \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c\} \Rightarrow$ for a_i to a_j, $getBestQoS(c \bigcup cr)$

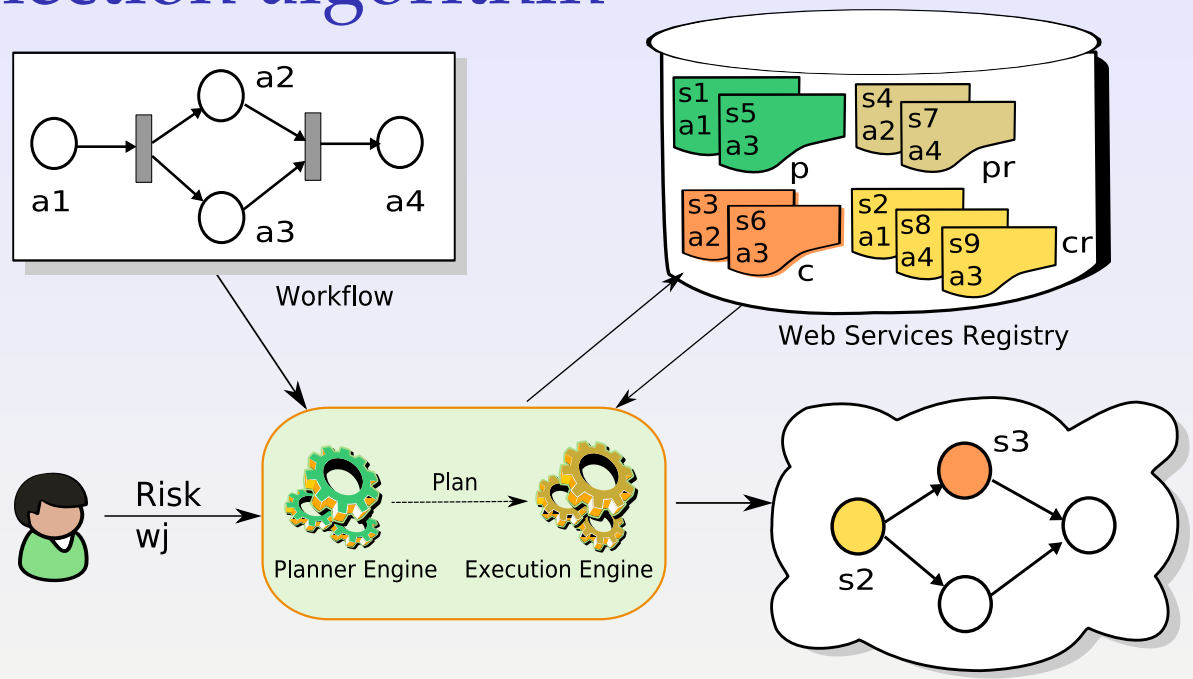
Service selection algorithm



Risk 1

- sequential path $(a_{i-1}; a_i)$: previously selected service has either
  - a transactional property in $\{p, pr\}$ or $nbp = 1 \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c, cr\} \Rightarrow$ for a_i, $getBestQoS(p \bigcup pr \bigcup c \bigcup cr)$
- concurrent path (split-pattern $\ldots a_i \ldots a_j$ join-pattern) : one previously selected service has either
  - a transactional property in $\{p\} \Rightarrow$ for a_i to a_j, $getBestQoS(cr)$
  - a transactional property in $\{pr\} \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c\} \Rightarrow$ for a_i to a_j, $getBestQoS(c \bigcup cr)$

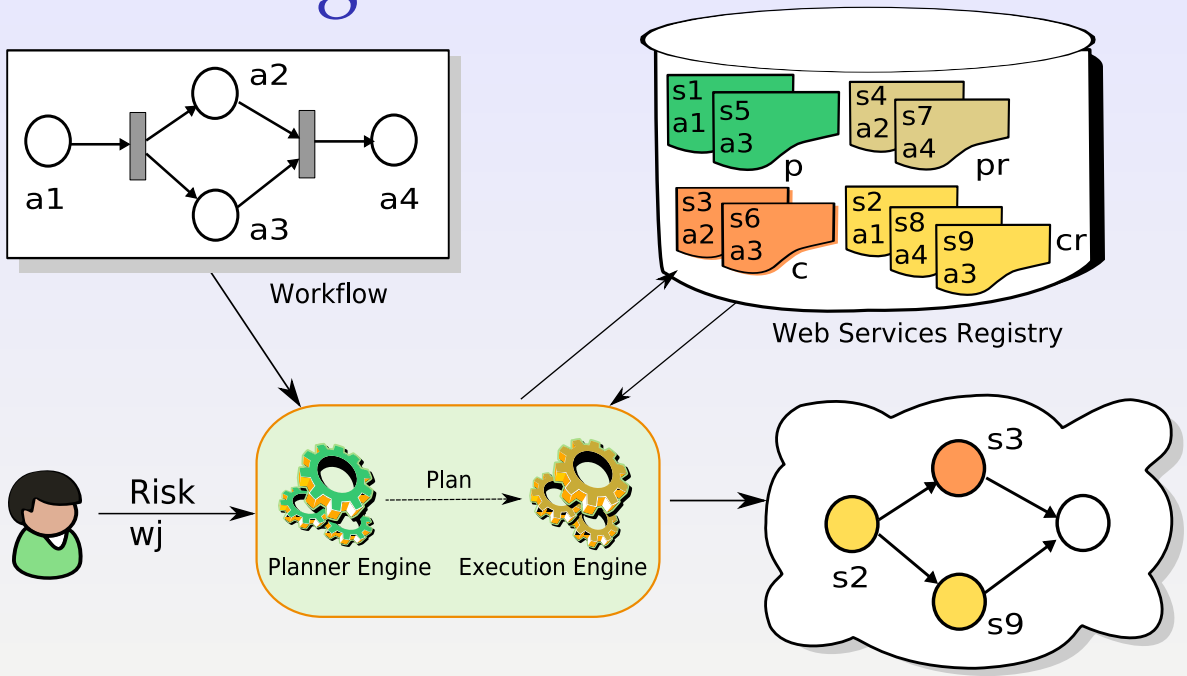
Service selection algorithm



Risk 1

- sequential path $(a_{i-1}; a_i)$: previously selected service has either
  - a transactional property in $\{p, pr\}$ or $nbp = 1 \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c, cr\} \Rightarrow$ for a_i, $getBestQoS(p \bigcup pr \bigcup c \bigcup cr)$
- concurrent path (split-pattern $\ldots a_i \ldots a_j$ join-pattern) : one previously selected service has either
  - a transactional property in $\{p\} \Rightarrow$ for a_i to a_j, $getBestQoS(cr)$
  - a transactional property in $\{pr\} \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c\} \Rightarrow$ for a_i to a_j, $getBestQoS(c \bigcup cr)$

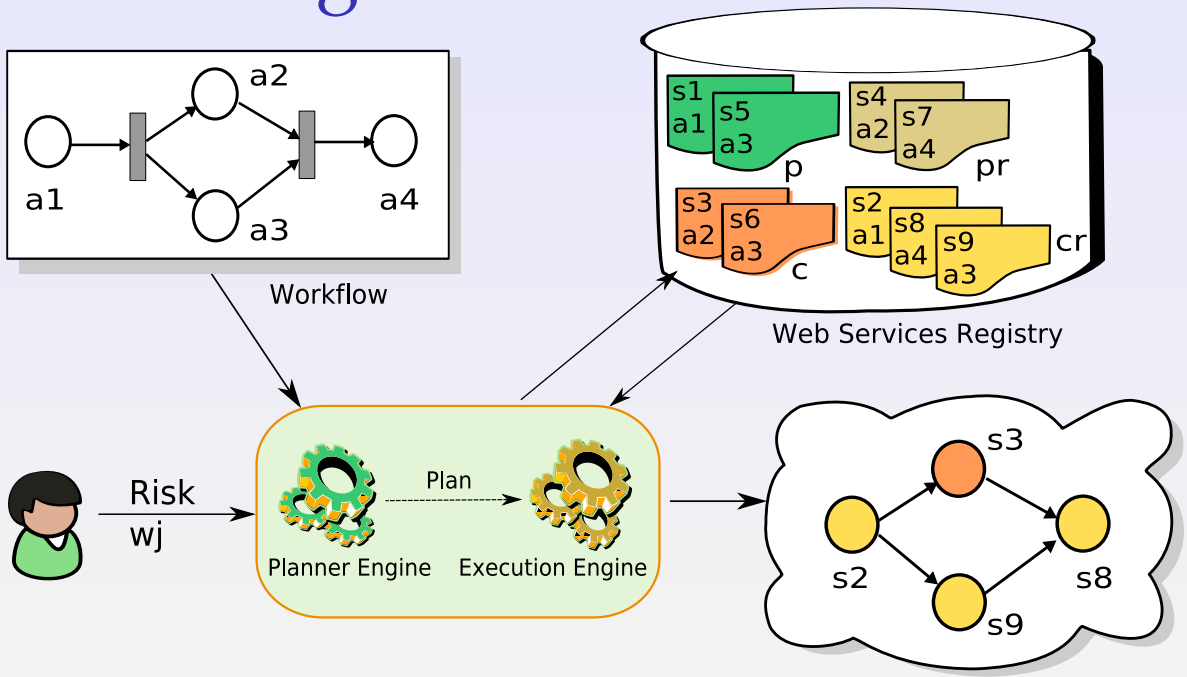
Service selection algorithm



Risk 1

- sequential path $(a_{i-1}; a_i)$: previously selected service has either
  - a transactional property in $\{p, pr\}$ or $nbp = 1 \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c, cr\} \Rightarrow$ for a_i, $getBestQoS(p \bigcup pr \bigcup c \bigcup cr)$
- concurrent path (split-pattern $\ldots a_i \ldots a_j$ join-pattern) : one previously selected service has either
  - a transactional property in $\{p\} \Rightarrow$ for a_i to a_j, $getBestQoS(cr)$
  - a transactional property in $\{pr\} \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c\} \Rightarrow$ for a_i to a_j, $getBestQoS(c \bigcup cr)$

Service selection algorithm

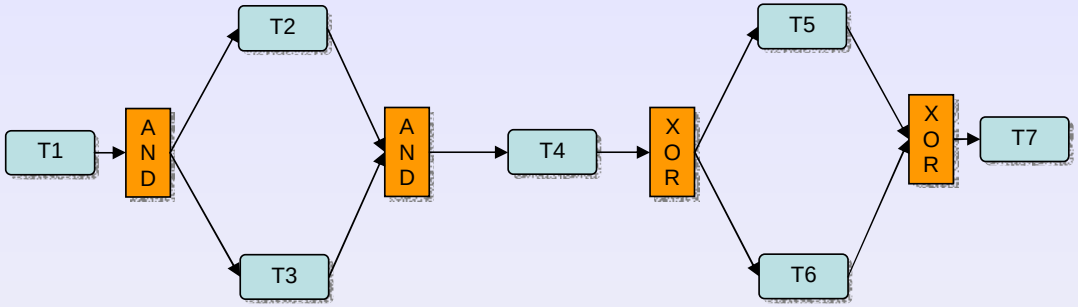


Risk 1

- sequential path $(a_{i-1}; a_i)$: previously selected service has either
  - a transactional property in $\{p, pr\}$ or $nbp = 1 \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c, cr\} \Rightarrow$ for a_i, $getBestQoS(p \bigcup pr \bigcup c \bigcup cr)$
- concurrent path (split-pattern $\ldots a_i \ldots a_j$ join-pattern) : one previously selected service has either
  - a transactional property in $\{p\} \Rightarrow$ for a_i to a_j, $getBestQoS(cr)$
  - a transactional property in $\{pr\} \Rightarrow$ for a_i to a_n, $getBestQoS(pr \bigcup cr)$
  - a transactional property in $\{c\} \Rightarrow$ for a_i to a_j, $getBestQoS(c \bigcup cr)$

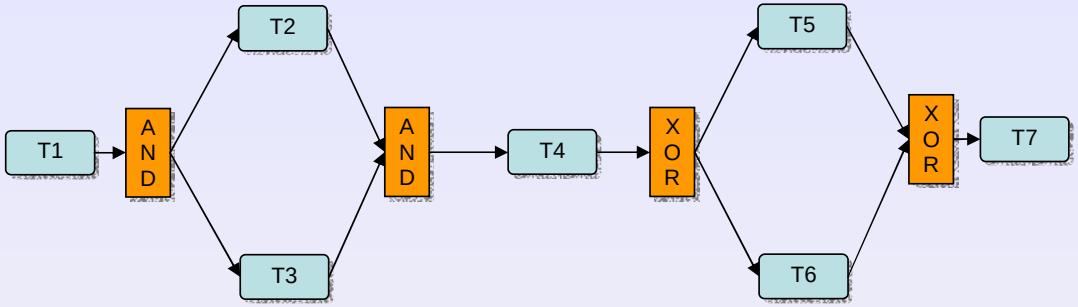
Implementation

- A workflow



Implementation

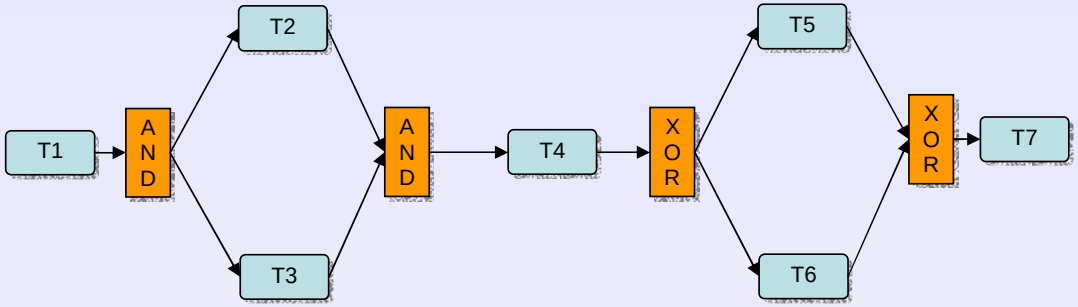
- A workflow



- A random generation of different services that can implement the activities
  - For each activity, uniformly generate 15 Web services
  - For each service, randomly generates transactional property and a QoS vector

Criteria	cr	c	pr	p
$q_{ep}(s)$	$0.20 - 0.30$	$0.20 - 0.30$	$0.00 - 0.10$	$0.00 - 0.10$
$q_{ed}(s)$	$0.20 - 0.30$	$0.01 - 0.10$	$0.20 - 0.30$	$0.01 - 0.10$
$q_r(s)$	$0 - 5$	$0 - 5$	$0 - 5$	$0 - 5$
$q_{sr}(s)$	$0.00 - 1.00$	$0.00 - 1.00$	$0.00 - 1.00$	$0.00 - 1.00$
$q_a(s)$	$0.00 - 1.00$	$0.00 - 1.00$	$0.00 - 1.00$	$0.00 - 1.00$

Implementation



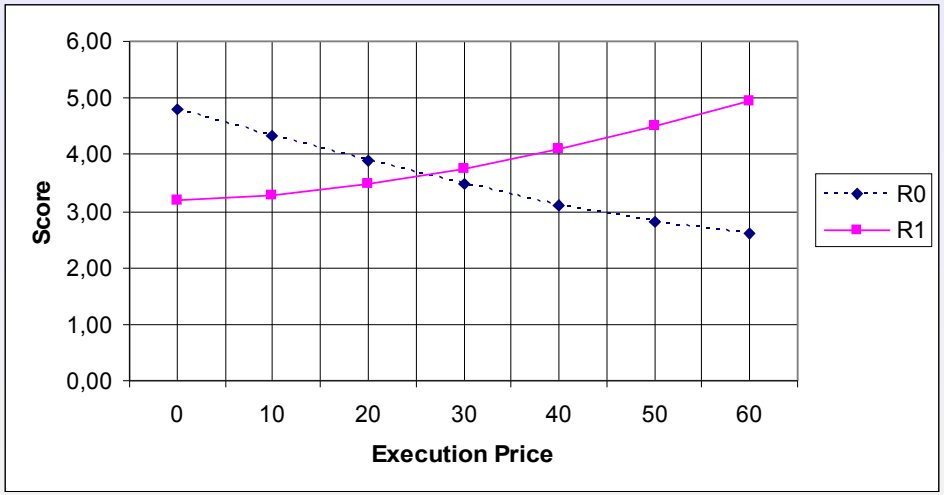
- A workflow
- A random generation of different services that can implement the activities
  - For each activity, uniformly generate 15 Web services
  - For each service, randomly generates transactional property and a QoS vector

Criteria	cr	c	pr	p
$q_{ep}(s)$	$0.20 - 0.30$	$0.20 - 0.30$	$0.00 - 0.10$	$0.00 - 0.10$
$q_{ed}(s)$	$0.20 - 0.30$	$0.01 - 0.10$	$0.20 - 0.30$	$0.01 - 0.10$
$q_r(s)$	$0 - 5$	$0 - 5$	$0 - 5$	$0 - 5$
$q_{sr}(s)$	$0.00 - 1.00$	$0.00 - 1.00$	$0.00 - 1.00$	$0.00 - 1.00$
$q_a(s)$	$0.00 - 1.00$	$0.00 - 1.00$	$0.00 - 1.00$	$0.00 - 1.00$

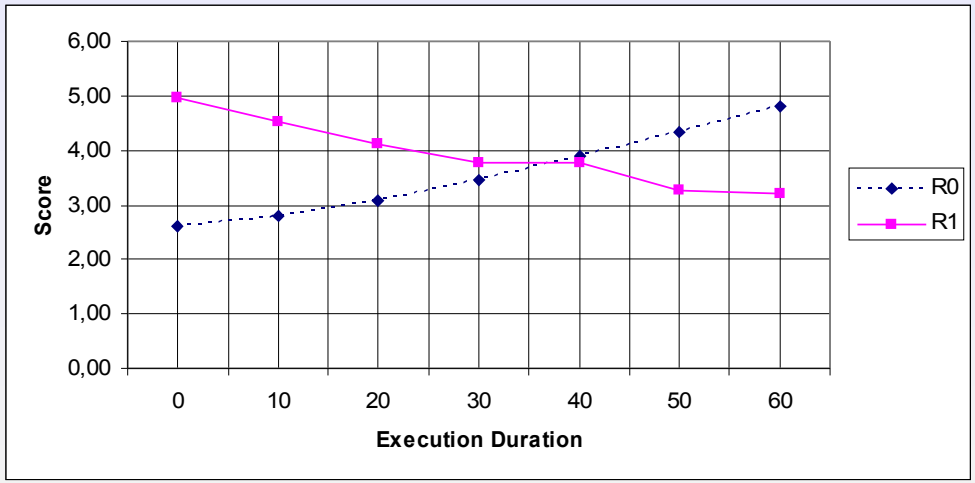
- User assigned weight with price and duration constraints have always 60% of the total weight

Criteria	(1)	(2)	(3)	(4)	(5)	(6)	(7)
$q_{ep}(s)$	0	10	20	30	40	50	60
$q_{ed}(s)$	60	50	40	30	20	10	0
$q_r(s)$	10	10	10	10	10	10	10
$q_{sr}(s)$	15	15	15	15	15	15	15
$q_a(s)$	15	15	15	15	15	15	15

Implementation



(a)



(b)

- More the price criteria is important to the user, the best is a composition with risk 1
- More the duration criteria is important to the user, the best is a composition with risk 0

Related work

L. Zeng, A. N. B. Benatallah, M. Dumas, J. Kalagnanam and H. Chang
QoS-Aware Middleware for Web services Composition
IEEE Transactions on Software Engineering, 30(5), May 2004.

M. C. Jaeger, G. Muehl and S. Golze
Qos-aware composition of web services: An evaluation of selection algorithms
On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, LNCS 3760, October 2005.

W. Zhang, Y. Yang, S. Tang and L. Fang
Qos-driven service selection optimization model and algorithms for composite web services
31st Annual Int. Computer Software and Applications Conf. (COMPSAC'2007), 2007.

\+ QoS-aware Web services selection

\− no transactional behavior for the composite Web service

Related work

S. Bhiri, O. Perrin and C. Godart
Ensuring required failure atomicity of composite web services
14th Int. Conf. on World Wide Web (WWW'2005), 2005.

F. Montagut and R. Molva
Augmenting web services composition with transactional requirements
IEEE Int. Conf. on Web Services (ICWS'2006), September 2006.

L. Li, C. Liu and J. Wang
Deriving Transactional Properties of Composite Web Services
IEEE Int. Conf. on Web Services (ICWS'2007), July 2007.

\+ Transactional composition mechanism

– no QoS-aware selection

Related work

A. Liu, L. Huang and Q. Li.
QoS-Aware Web Services Composition Using Transactional Composition Operator
7th Int. Conf. Advances in Web-Age Information Management (*WAIM*), LNCS 4016, June 2006.

+ Composition of Web services with various transactional requirements
− Transactional property of the composite service is determined by its component services (if all component are p then the composite service is p)
+ Evaluation of the QoS of the composite service
− not a QoS-aware approach to service composition

Conclusion

- Web service selection approach supporting transactional and quality driven Web service composition
- Transactional properties of composite Web service are established based on the transactional properties of its component Web services
- The selection is realized depending on transactional and QoS user requirements
  - User transactional requirements are established by means of a risk notion that indicates if the user has or not the obligation to take the execution results
  - User QoS requirements are expressed as weight over each QoS criterion

Perspectives

- Replaning the selection of Web services in order to take into account dynamic changes
- Web service selection at run-time
- Take into account other considerations about the component Web services such *non-retriable* services with higher execution time than *retriable* ones
- Experimentations

References

F. Montagut and R. Molva

Augmenting web services composition with transactional requirements

IEEE Int. Conf. on Web Services (ICWS'2006), September 2006.

L. Li, C. Liu and J. Wang

Deriving Transactional Properties of Composite Web Services

IEEE Int. Conf. on Web Services (ICWS'2007), July 2007.

L. Zeng, A. N. B. Benatallah, M. Dumas, J. Kalagnanam and H. Chang

QoS-Aware Middleware for Web services Composition

IEEE Transactions on Software Engineering, 30(5), May 2004.

A. Liu, L. Huang and Q. Li.

Qos-aware web services composition using transactional composition operator

7th Int. Conf. Advances in Web-Age Information Management (*WAIM*), LNCS 4016, June 2006.