

Planning and monitoring the execution of web service requests

Alexander Lazovik^{1,2}, Marco Aiello¹, Mike Papazoglou^{1,3}

¹DIT, University of Trento

{lazovik, aiellom}@dit.unitn.it

² ITC-IRST, Trento

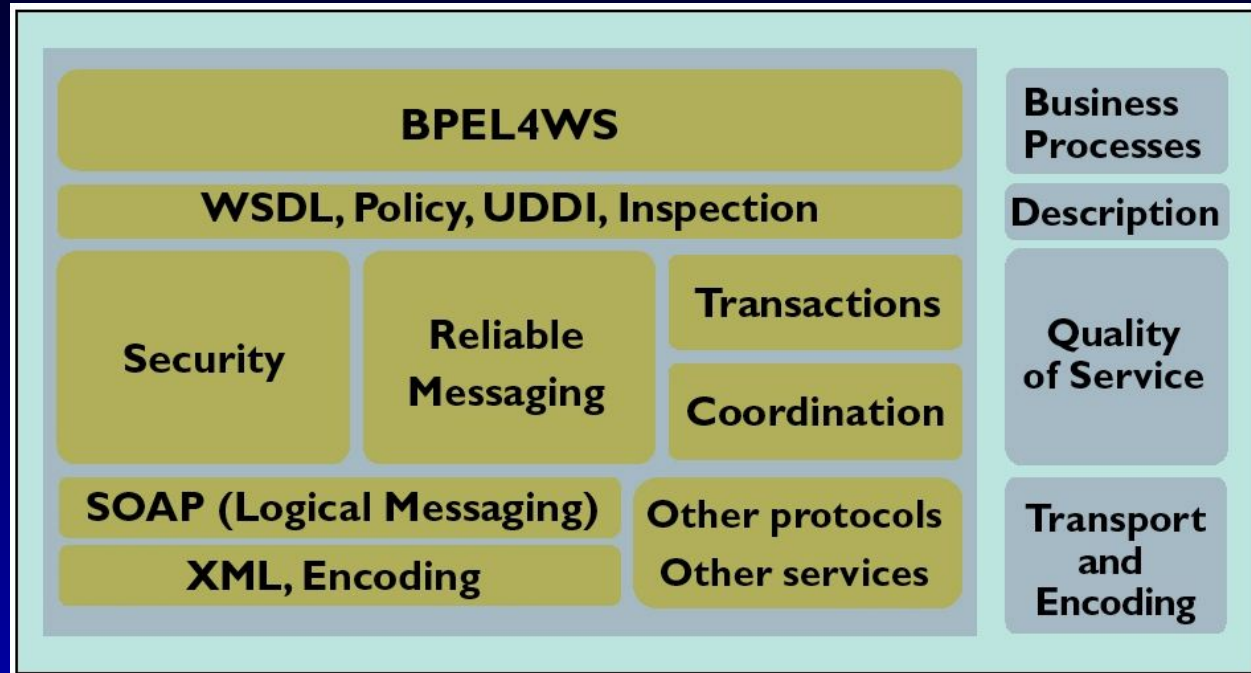
³ Infolab, Tilburg University

mikep@uvt.nl

Outline

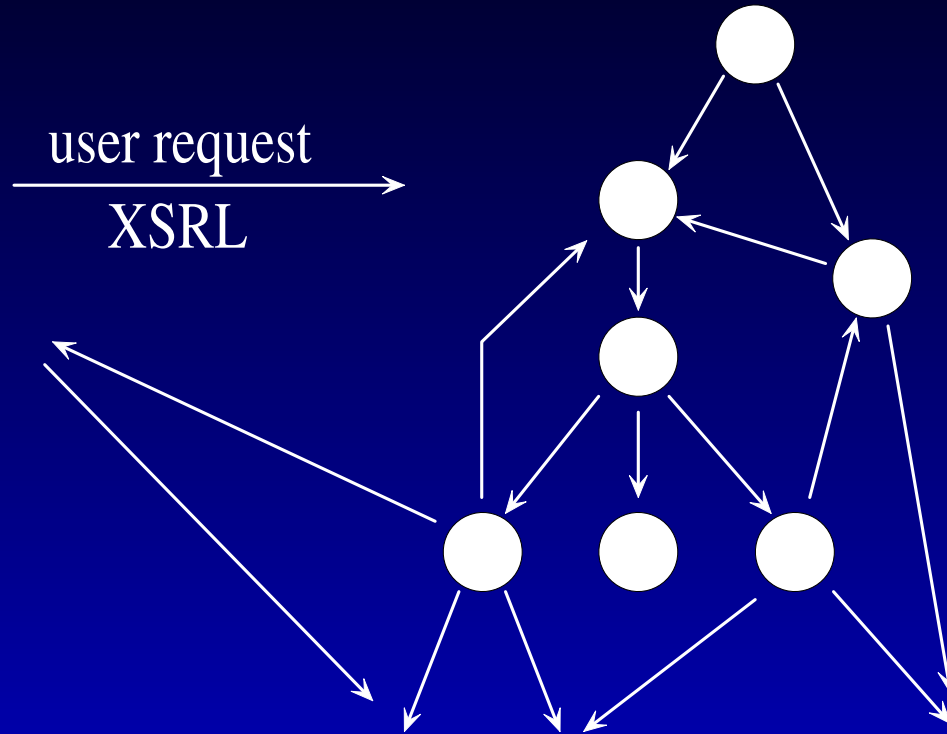
- Web service composition
- Planning a trip
- XSRL
- Framework
- A run of an XSRL request

Web Service Composition



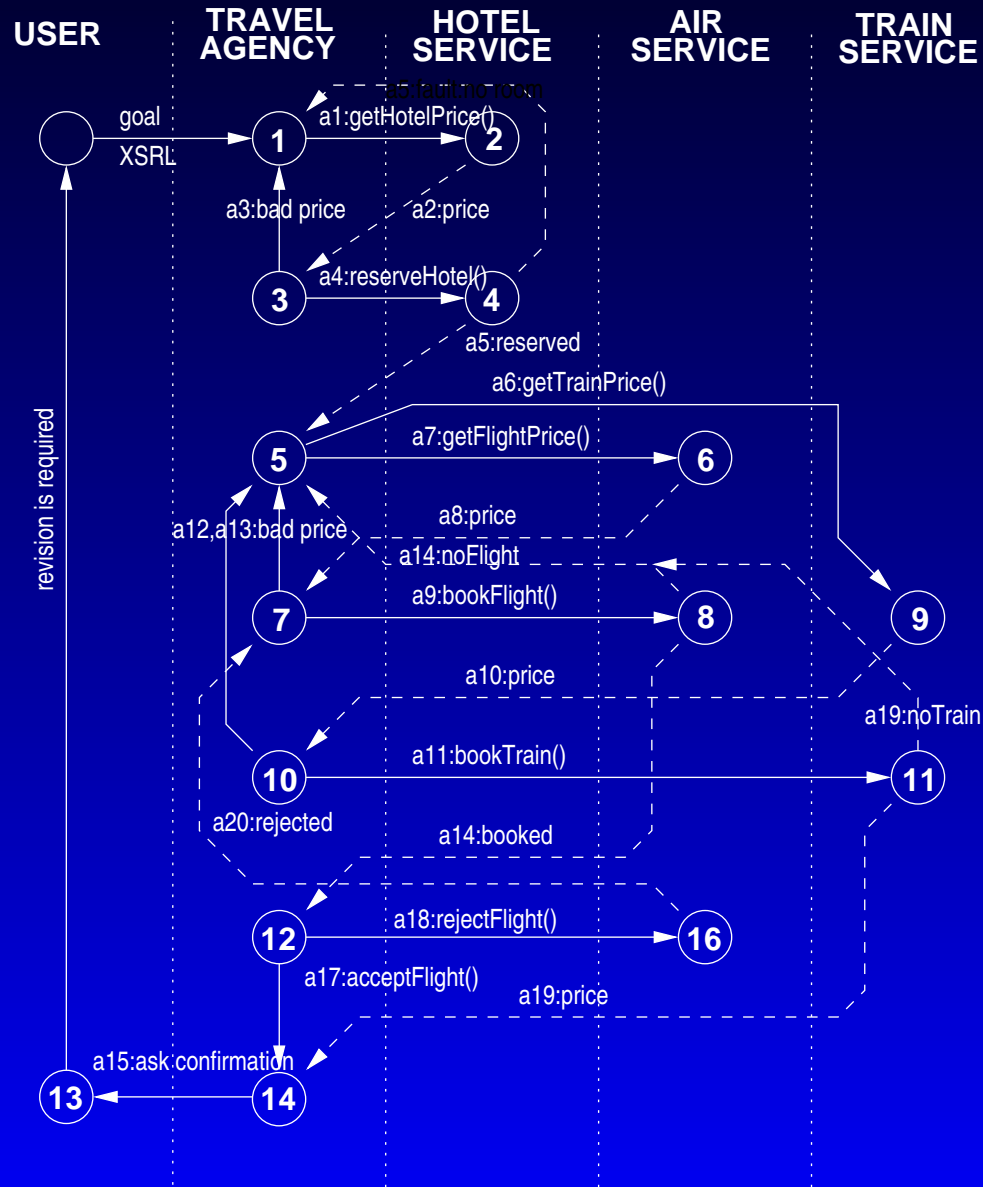
- Automatic Composition vs. Composition by design
- How to interact with a generic composition of services?

Request-driven process



- dynamically adapted business process
- reduced development cost
- better tool for user request specification

Business Process for Traveling



Sample user request

Given a business process of a travelling domain user requests:

- reserving one night trip to Paris with hotel
- avoiding to travel by train, if possible
- spending overall amount less than 300 euros
- spending less than 100 euros per hotel, if possible, and 200 at most
- receiving, finally, a confirmation of payment

XSRL

Xml Service Request Language, based on an extension of temporal logic (EaGLE and CTL).

- `achive-all, vital, atomic`
reachability
- `before...then`
sequencing
- `prefer...to, optional`
preference
- `vital-maint, optional-maint`
maintainability goals
- `=, <, >`
linear constraints

Organizing a trip

achieve-all

before

achieve-all

prefer vital-maint $hotelPrice < 100$ to

vital-maint $hotelPrice < 200$

optional-maint $\neg trainBooked$

vital $confirmed \wedge$

$location = \text{"Paris"} \wedge$

$hotelReserved$

then

atomic $final$

vital-maint $price < 300$

XSRL approach

- Business process is defined by a *state-transition system*
- Problem: find and execute a *plan*, such that its *execution structure* satisfies the XSRL goal

From the planning perspective:

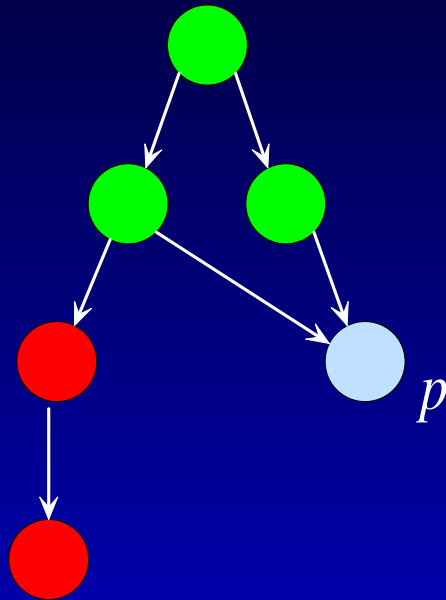
Our framework	Planning
XSRL	goal language
Web Services operations	actions
Business processes	domain
Web Service implementations and UDDI	environment

Actions: two types of uncertainty - nondeterministic failures and unknown outcomes from invocations

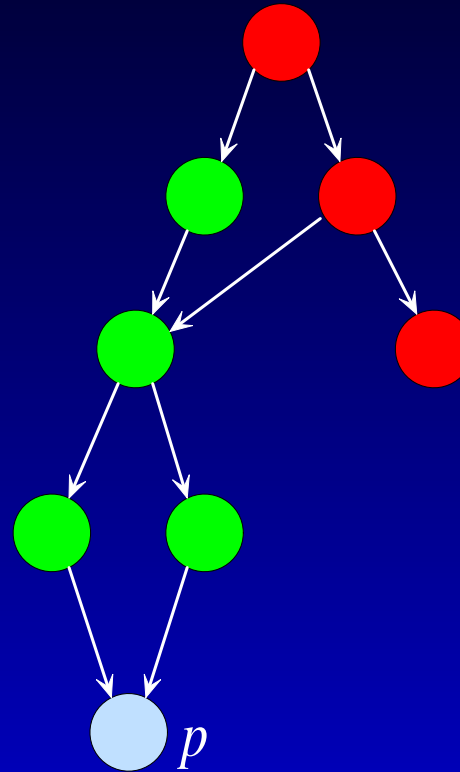
Solution: Interleaved planning and execution!

Reachability goals

vital p :

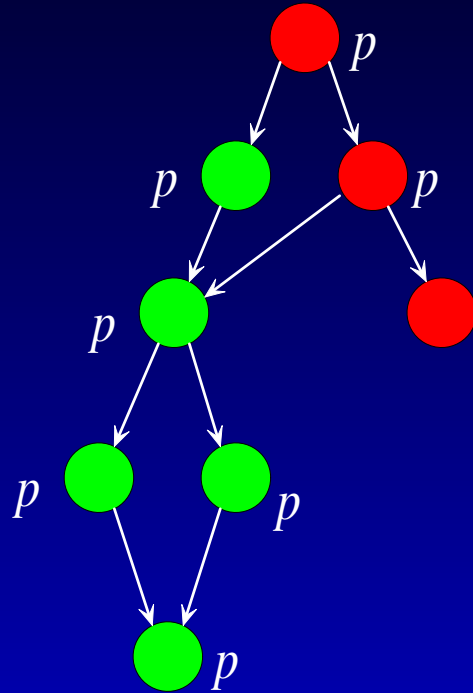


atomic p :

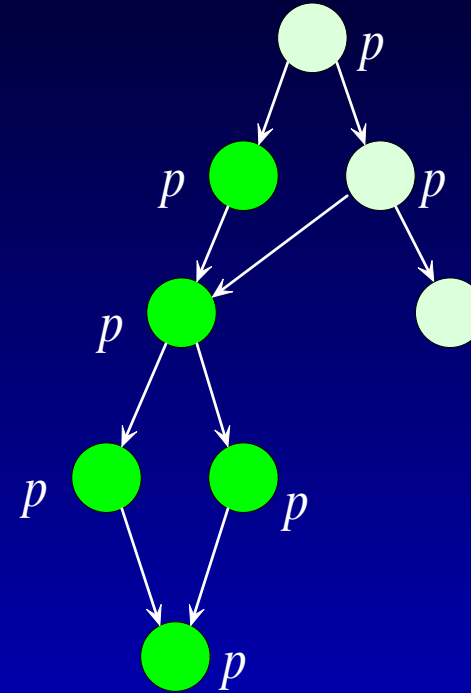


Maintainability goals

vital-maint p :

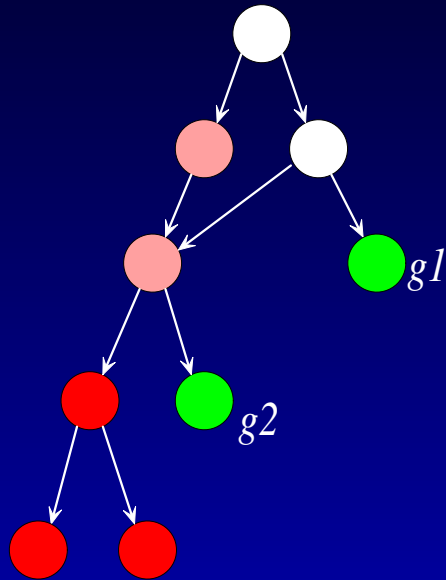


optional-maint p :

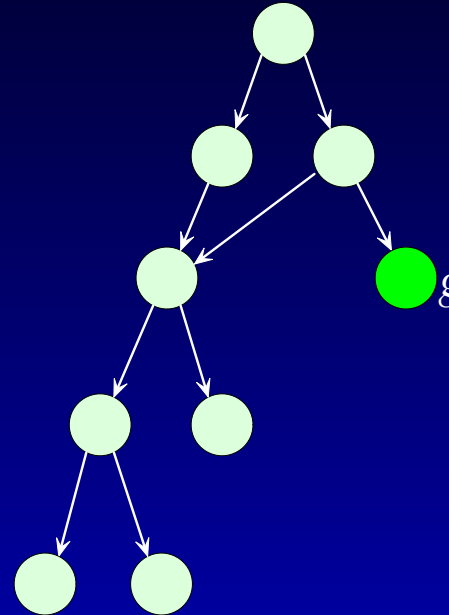


Preference goals

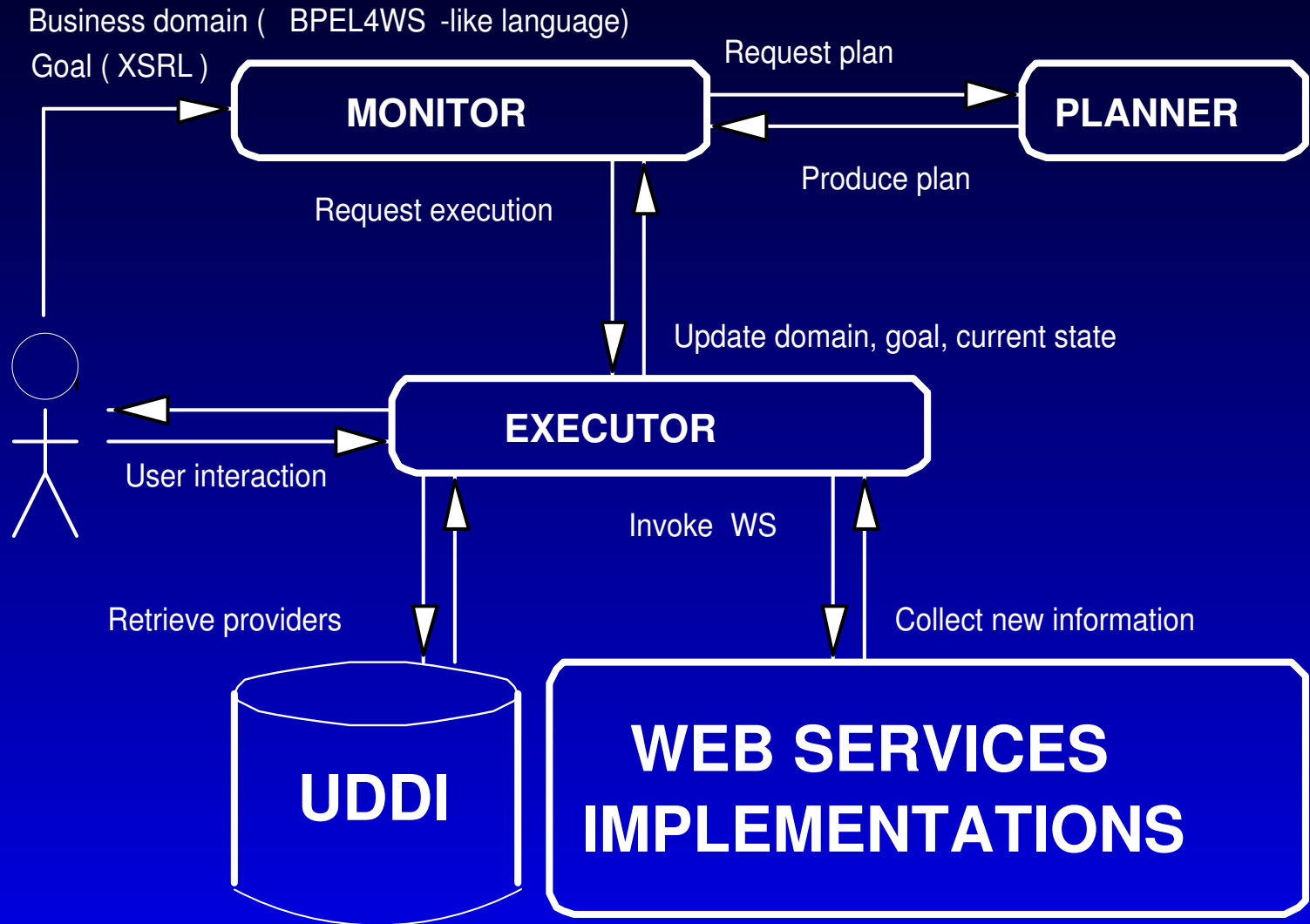
prefer g_1 to g_2 :



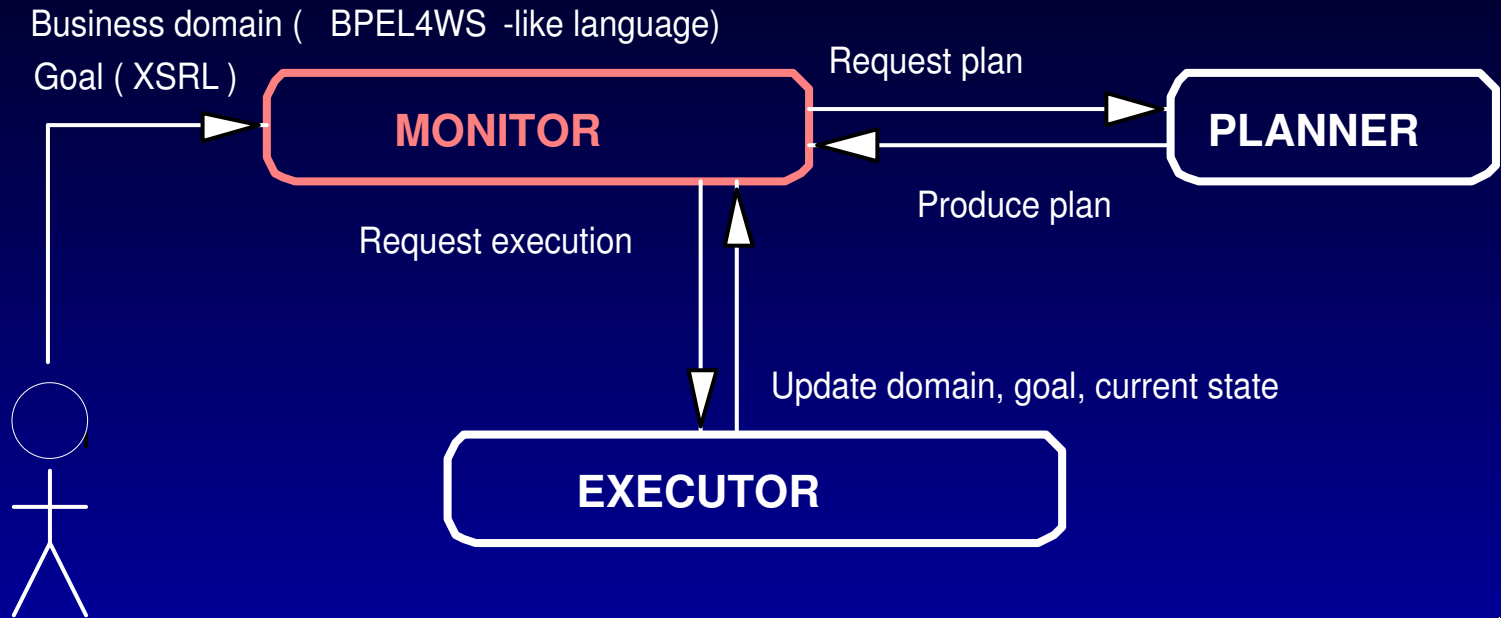
optional g :



XSRL Framework

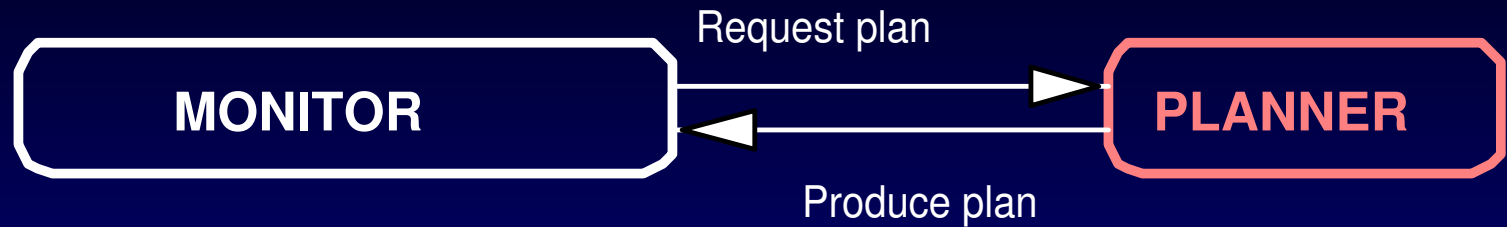


Monitor



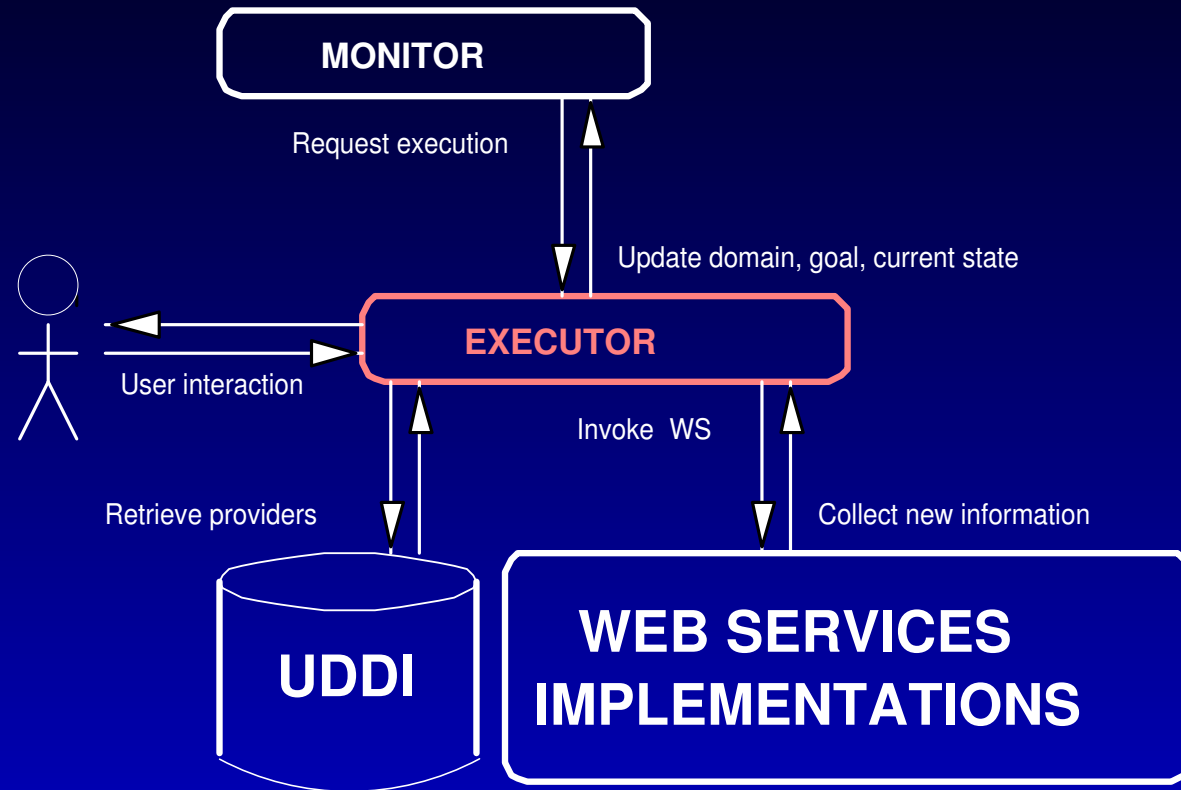
- ...invokes planner
- ...recovers from failure
- ...requests execution of plan actions

Planner



- ... synthesizes a plan
- ... currently relies on *MBP*, planning as model checking under uncertainty
- ... *MBP* works with booleanized domain

Executor



- ...executes actions from a plan
- ...contacts UDDI and assigns providers
- ...requests replanning in case of knowledge-gathering actions

Organizing a trip - revisited

achieve-all

before

achieve-all

prefer vital-maint $hotelPrice < 100$ to

vital-maint $hotelPrice < 200$

optional-maint $\neg trainBooked$

vital $confirmed \wedge$

$location = \text{"Paris"} \wedge$

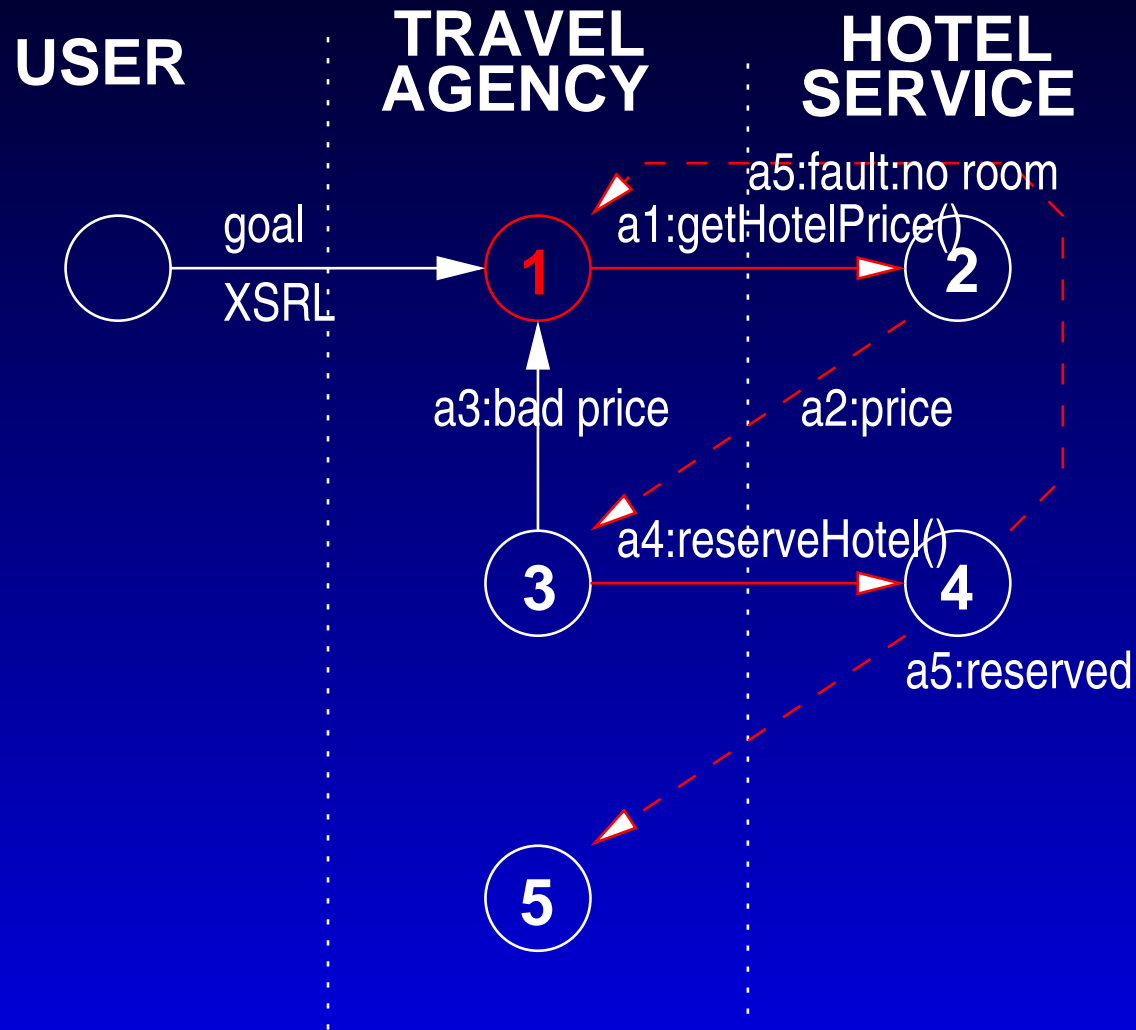
$hotelReserved$

then

atomic $final$

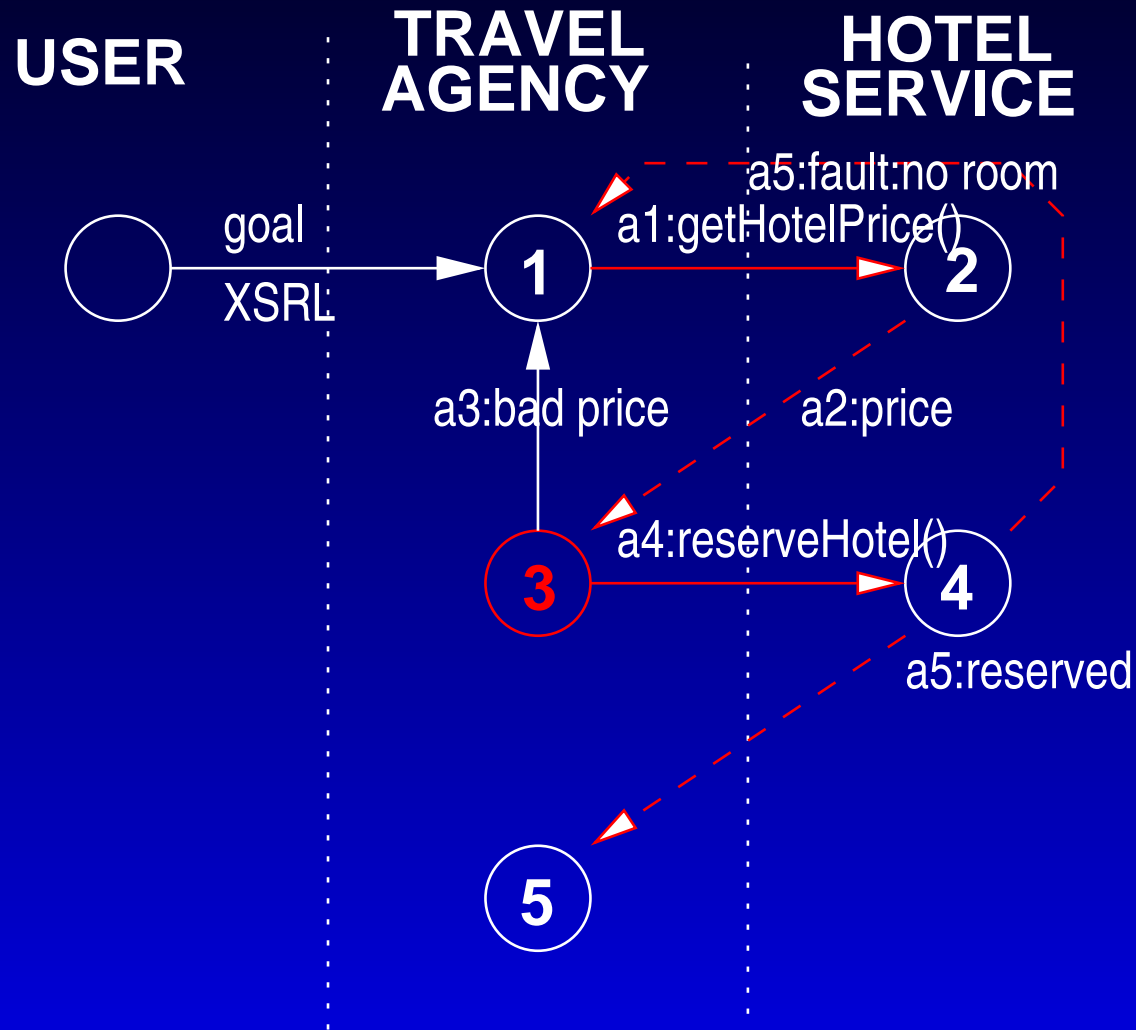
vital-maint $price < 300$

Running the example - 1



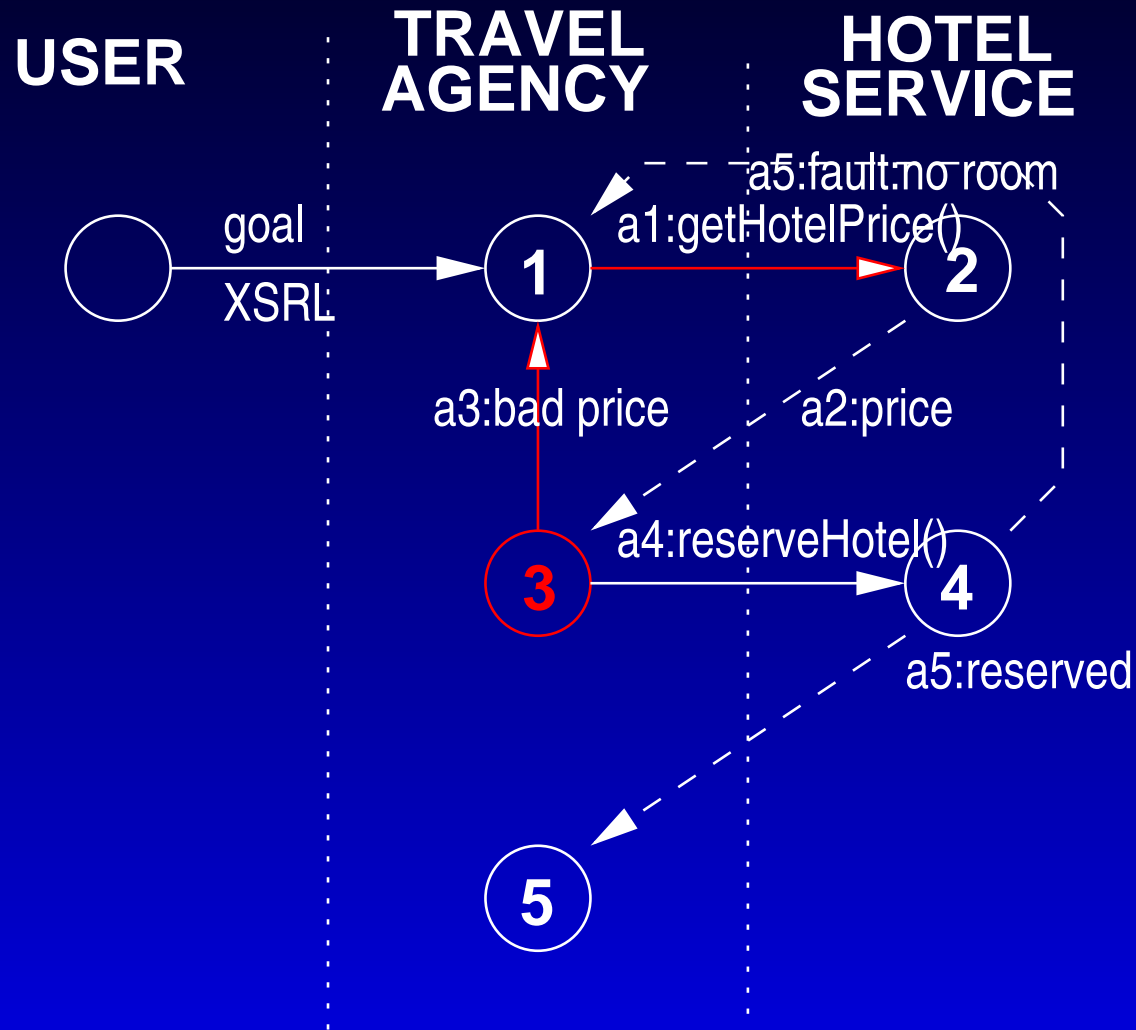
Planner synthesizes a plan...

Running the example - 2



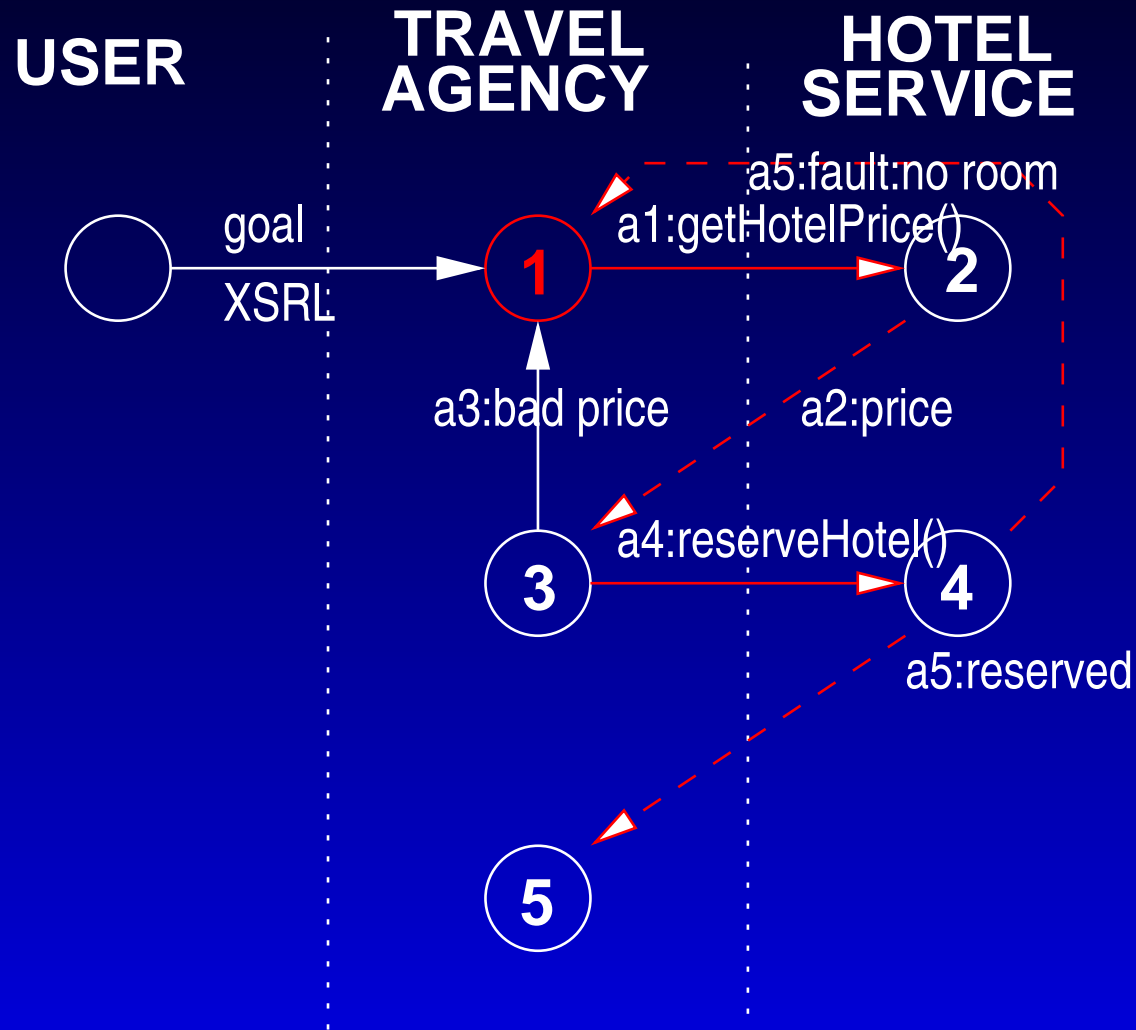
...hotel price is 300!...

Running the example - 3



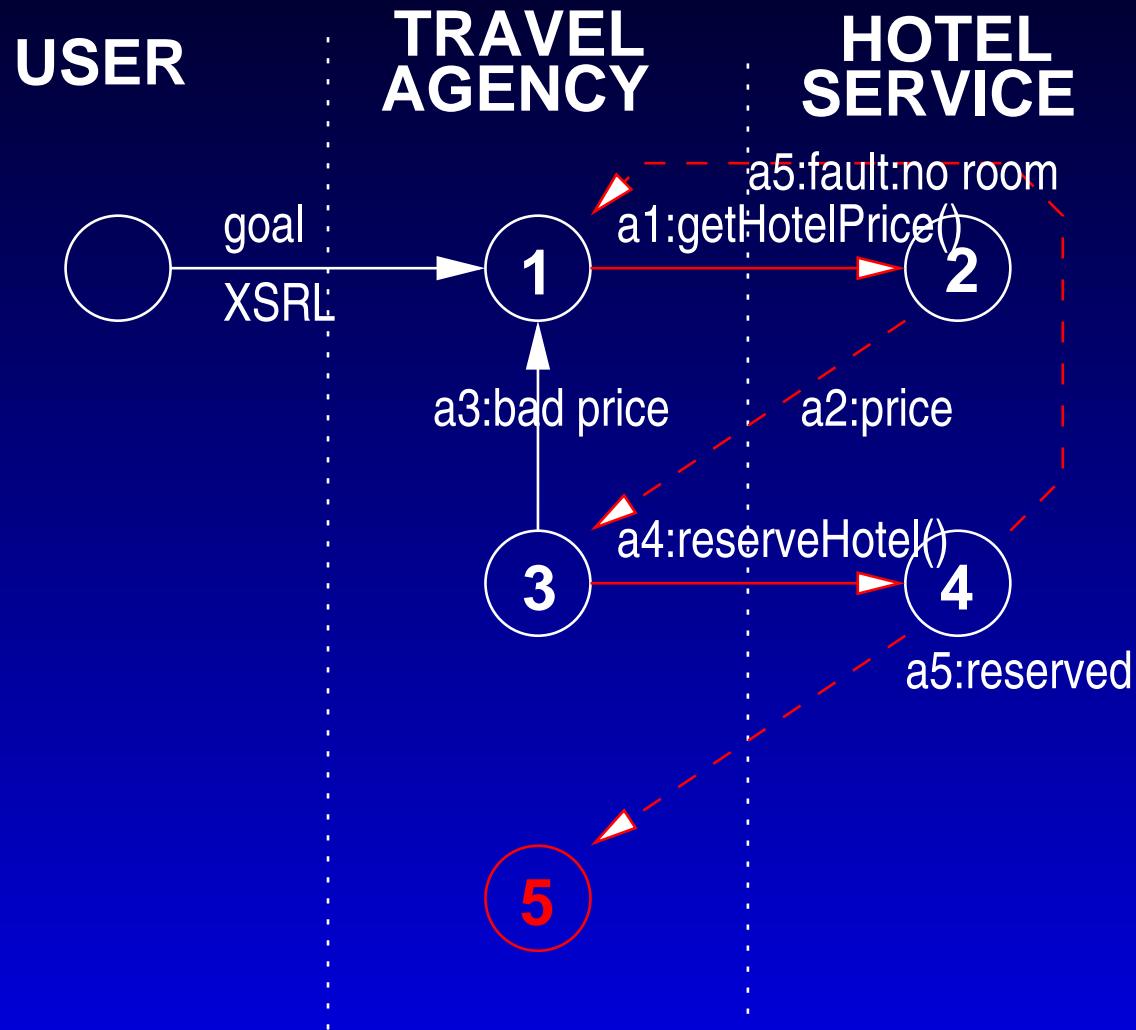
...replanning phase...

Running the example - 4



...choose a different provider and generate a new plan...

Running the example - 6



...finally hotel is reserved!

A note about using XSRL



- One does not expect the user to “talk” XSRL
- Template of requests for a given business process can be implemented rapidly



Summary

- XSRL: a request language for interacting with compositions of web services
- Formal semantics of XSRL fully specified
- Framework for interpreting requests and monitoring their execution
- XSRL not for direct specification by end user

Open issues

- Algorithmic properties:
 - completeness
 - complexity
- Implementation:
 - efficiency in replanning
 - mediating the interaction with UDDI
 - balance between off-line vs. interleaved planning and execution
 - planning under partial observability
- Identifying all appropriate extensions to BPEL4WS and WSDL for domain representation