

Composition of services: algorithms and complexity

Fahima Cheikh

IRIT-Université de Toulouse

June 19, 2009

Introduction

Building complex software systems

- Hierarchic systems
- Subsystems are able to realize a functionality
- Subsystems communicate to achieve parent functionality

Introduction

Building complex software systems

- Hierarchic systems
- Subsystems are able to realize a functionality
- Subsystems communicate to achieve parent functionality

Coalition formation for composition of agents (MAS)

- Set of autonomous agents
- Each agent can realize a partial goal
- Coalitions of agent allow the realization of a common complex goal

Introduction

Robotic supervision

- A robot is composed by several modules
- Each module realizes a specific task
- Supervisor interacts with modules in order to realize a complex task

Introduction

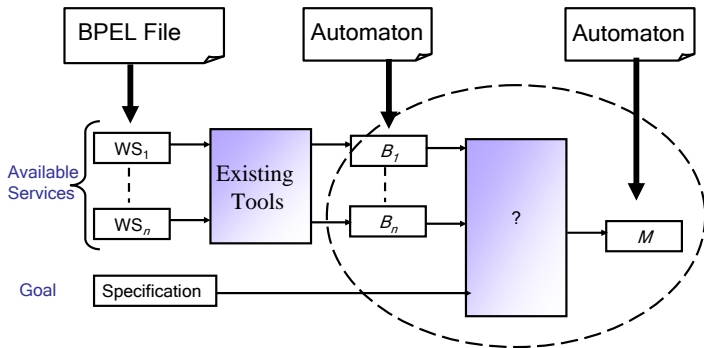
Robotic supervision

- A robot is composed by several modules
- Each module realizes a specific task
- Supervisor interacts with modules in order to realize a complex task

Composition of services

- A set of available services
- Each service realizes a functionality
- Composing services in order to realize a complex functionality

Orchestration of services



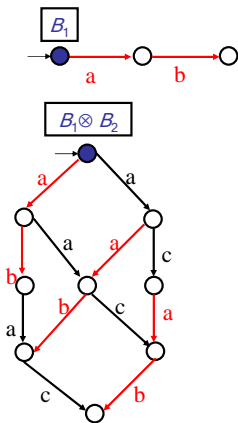
M orchestrates the available services in order to satisfy the goal

- 1 Introduction
- 2 Existing approaches
- 3 Our approach
- 4 Conclusion and future works

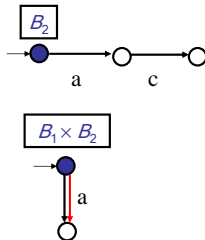
Existing approaches: choice of the model

- Kind of automata considered? **With precondition and effects/ with communication/...**
- Formalization of the specification? **Formula/ automata/...**
- Satisfaction of the goal? **Formula satisfiability/ automata equivalence /...**
- Orchestration method? **Synchronous product/ asynchronous product/...**

Some definitions: asynchronous and synchronous product



Asynchronous product

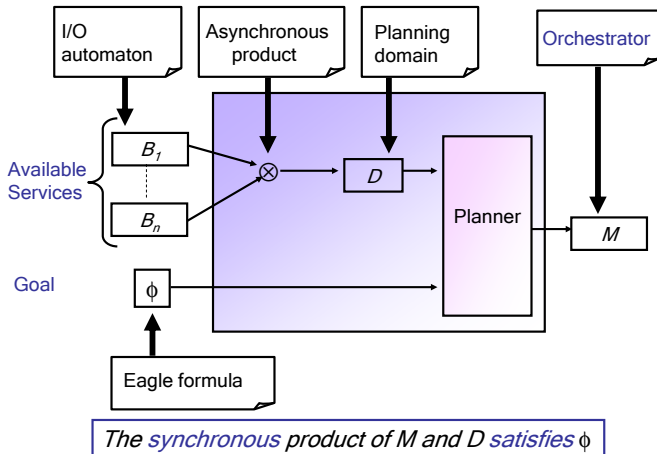


Synchronous product

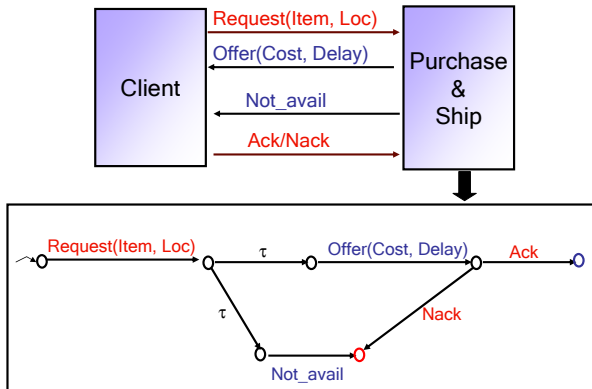
Planning based approach: M. Pistore, P. Traverso and P. Bertoli

- Kind of automata considered? [Input/output automata](#)
- Formalization of the specification? [Eagle formula](#)
- Satisfaction of the goal? [Satisfiability of the Eagle formula](#)
- Orchestration method? [Synchronous product](#)

Planning based approach: overview



Planning based approach: service example



Note: the state Success (resp. Failure) is represented by \circ (resp. \circ)

Planning based approach: Eagle formula

TryReach

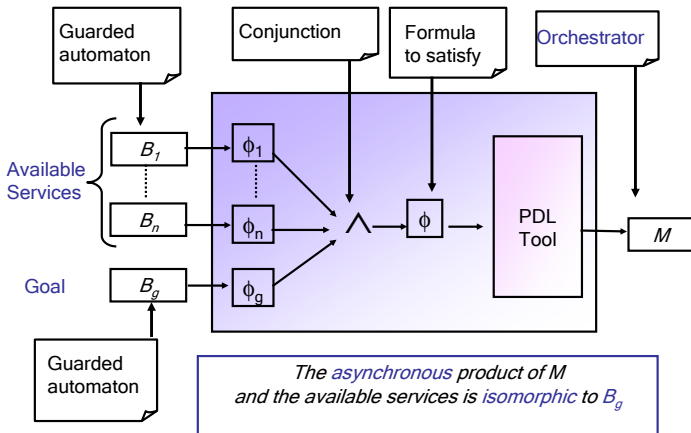
$(Client.pc = Success \wedge Shipper.pc = Success \wedge Production.pc = Success)$ **Fail**

$(Client.pc = Failure \wedge Shipper.pc = Failure \wedge Production.pc = Failure)$

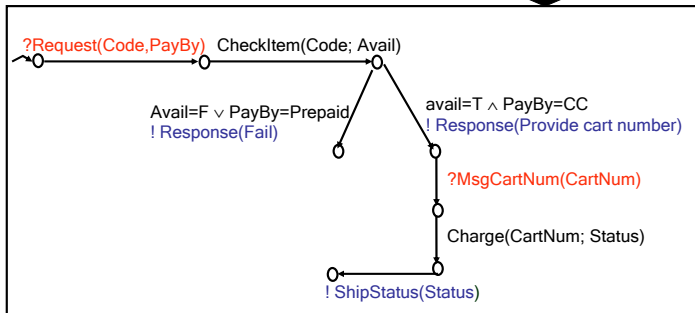
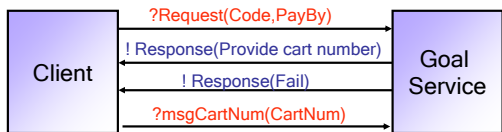
PDL based approach: D. Berardi, D. Calvanese, G. De Giacomo, R. Hull and M. Mecella

- Kind of automata considered? [Guarded automata](#)
- Formalization of the specification? [Guarded automata](#)
- Satisfaction of the goal? [Isomorphism](#)
- Orchestration method? [Asynchronous product](#)

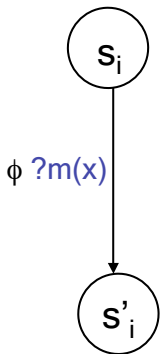
PDL based approach: overview



PDL based approach: service example



PDL based approach: PDL formula



Service B_i receives a message m

$$exec_i \wedge s_i \wedge \alpha \rightarrow (\langle ?m \rangle \top \wedge [?m] s'_i \wedge [?m] \alpha')$$

where

ϕ evaluates true with respect to α

α' is obtained from α by updating the value of x

Comparison of the existing approaches

Approaches	Automata	Orchestration	Specification	Computational results
Planning based	I/O automata	Synchronous product	Eagle formula (satisfiability)	Experimental
PDL based	Guarded automata	Asynchronous product	Guarded automaton (isomorphism)	Complexity upper bound (EXPTIME)

- 1 Introduction
- 2 Existing approaches
- 3 Our approach**
- 4 Conclusion and future works

Our approach: general model

- Kind of automata considered? **Conditional communicating automata (CCA)**
- Formalization of the specification? **CCA**
- Satisfaction of the goal? **Trace inclusion, trace equivalence, simulation and bisimulation**
- Orchestration method? **Asynchronous product**

Composition problem

Instance: Finite set of actions Σ , a finite set of unbounded channels Π , a finite set of atomic formulas At and finite CCA B_1, \dots, B_n and B_g over Σ, Π and At

Question: Does there exist a CCA M over Π and At such that

$$B_g \approx B_1 \otimes \dots \otimes B_n \otimes M?$$

Relation	Decidable
Trace inclusion	<i>Yes</i>
Trace equivalence	<i>No</i>
Simulation	<i>Yes</i>
Bisimulation	<i>No</i>

Considered variants

First variant: pure comparison

Instance: Finite set of actions Σ , a finite set of atomic formulas At and finite Conditional Automata (CA) B_1, \dots, B_n and B_g over Σ and At

Question: $B_g \approx B_1 \otimes \dots \otimes B_n$?

Results for the pure comparison variant

Relation	Complexity
Trace inclusion	<i>EXPSPACE-complete</i>
Trace equivalence	<i>EXPSPACE-complete</i>
Simulation	<i>EXPTIME-complete</i>
Bisimulation	<i>EXPTIME-complete</i>

Considered variants

Second variant: formula synthesis

Instance: Finite set of actions Σ , a finite set of channels Π , a finite set of atomic formulas At and finite CCA B_1, \dots, B_n and B_g over Σ, Π and At

Question: Synthesize ϕ such that for all valuations V of atomic formulas: $B_g \approx B_1 \boxtimes \dots \boxtimes B_n$ iff $V(\phi) = 1$

Results for the formula synthesis variant

Relation	Complexity
Trace inclusion	<i>EXPSPACE-complete</i>
Trace equivalence	<i>EXPSPACE-complete</i>
Simulation	<i>EXPTIME-complete</i>
Bisimulation	<i>PSPACE-hard</i> is in <i>EXPTIME</i>

Considered variants

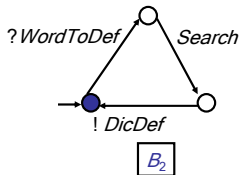
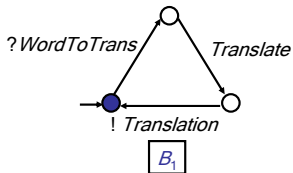
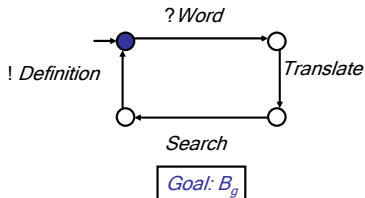
Third variant: orchestrator synthesis

Instance: Finite set of actions Σ , a finite set of bounded channels Π , a finite set of atomic formulas At and finite CCA B_1, \dots, B_n and B_g over Σ , Π and At

Question: Does there exist a CCA M over Π and At such that

$$B_g \approx B_1 \otimes \dots \otimes B_n \otimes M?$$

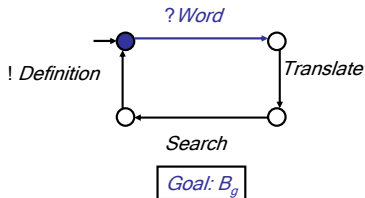
Orchestrator synthesis variant: example of services



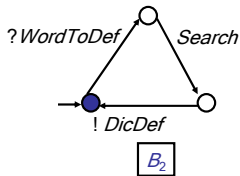
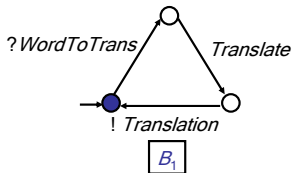
?

M

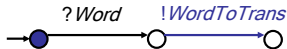
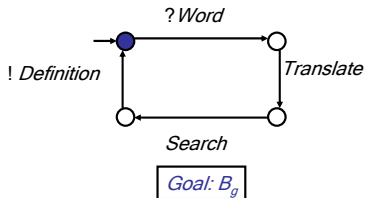
Orchestrator synthesis variant: example of services



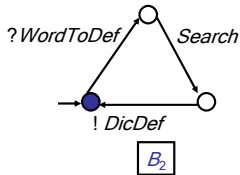
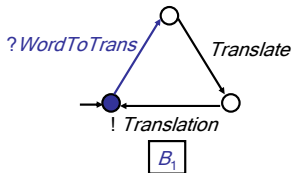
M



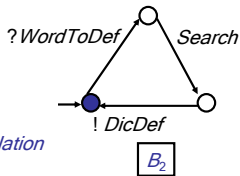
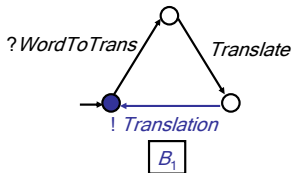
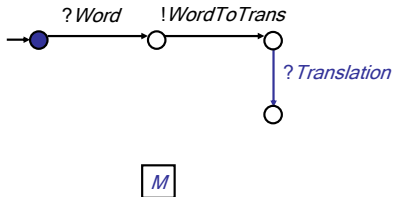
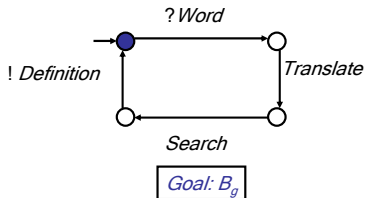
Orchestrator synthesis variant: example of services



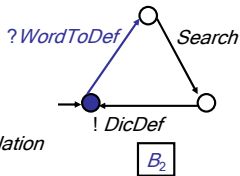
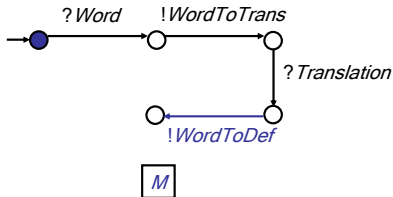
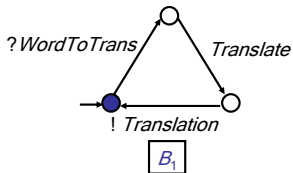
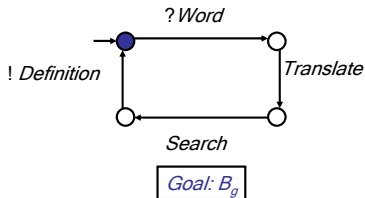
M



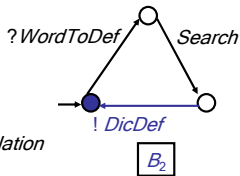
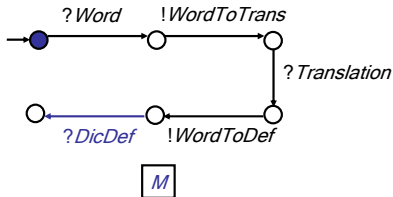
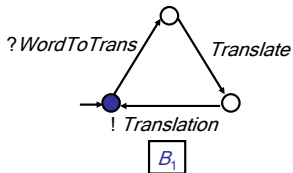
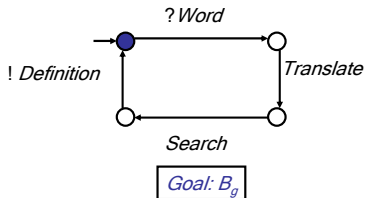
Orchestrator synthesis variant: example of services



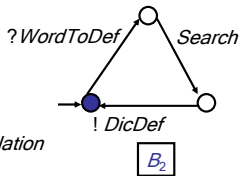
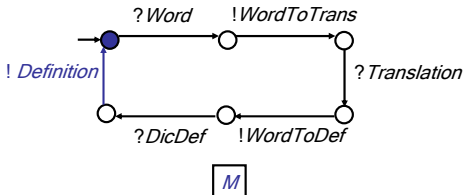
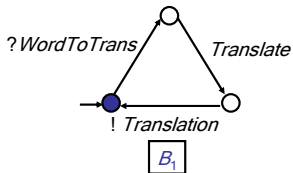
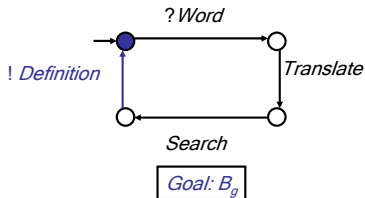
Orchestrator synthesis variant: example of services



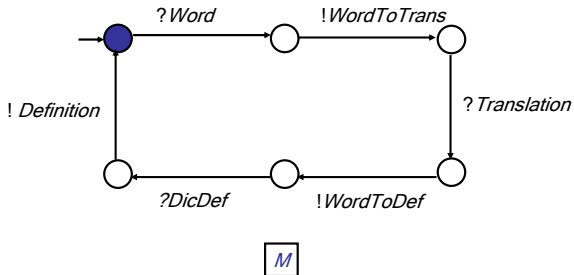
Orchestrator synthesis variant: example of services



Orchestrator synthesis variant: example of services



Orchestrator synthesis variant: example of services



M is such that B_g is Bisimilar to $B_1 \otimes B_2 \otimes M$

Orchestrator synthesis variant: composition problem

Instance: Finite set of actions Σ , a finite set of bounded channels Π , a finite set of atomic formulas At and finite CCA B_1, \dots, B_n and B_g over Σ , Π and At

Question: Does there exist a CCA M over Π and At such that B_g is bisimilar to $B_1 \otimes \dots \otimes B_n \otimes M$?

Orchestrator synthesis variant: upper bound

Theorem

The composition problem can be solved in **2-EXPTIME** for bisimulation

Proof

Reduction of the composition problem to the control problem

Control problem: observability and controllability constraints

Σ is partitioned into the set Σ_{ob} of observable events and the set Σ_{uob} of unobservable events.

Σ is also partitioned into the set Σ_{ct} of controllable events and the set Σ_{uct} uncontrollable events.

Control problem: observability and controllability constraints

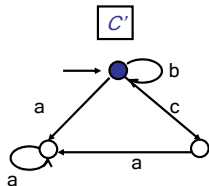
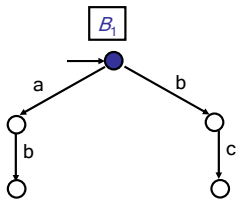
Controllability constraints

For any state q of C and for any uncontrollable event $a \in \Sigma_{uct}$, there is a transition from q labelled by a .

Observability constraints

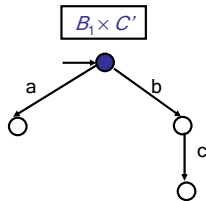
For any state q of C and for any unobservable event $a \in \Sigma_{uob}$, if there is a transition from q labelled by a then this transition is a loop over q .

Control problem: example



$$\Sigma_{uob} = \{b\}$$

$$\Sigma_{uct} = \{a\}$$



Control problem

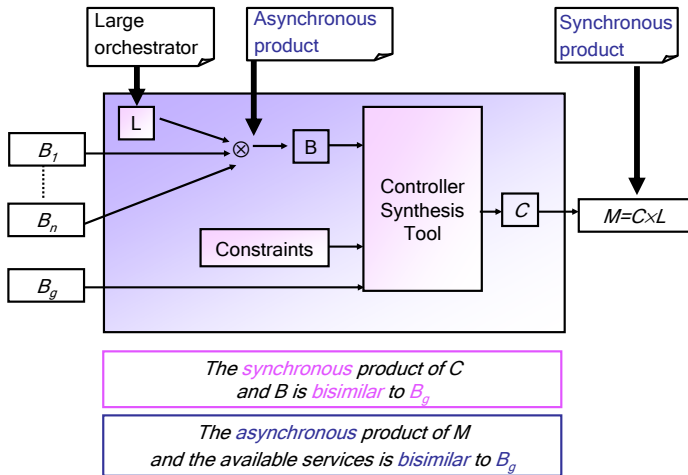
Instance: Finite set of actions Σ' , finite automata A and B over Σ' and observability and controllability constraints ?

Question: Does there exist a finite automaton C over Σ' that satisfies the constraints and such that

A is bisimilar to $B \times C$?

- This problem can be solved in **EXPTIME**
- Arnold, Vincent and Walukiewicz 2003
- Pinchinat and Raclet 2005.
- Basu and Kumar 2006.

Orchestrator synthesis variant: overview



From the orchestrator synthesis problem to the control problem

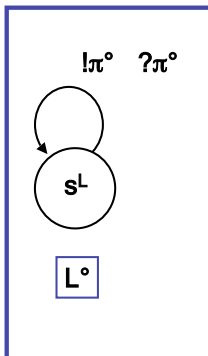
Orchestrator synthesis problem

Does there exist a finite automaton M over Π such that A is bisimilar to $B \otimes M$?

Control problem

If $\Sigma_{ob} = \Sigma_{ct} = \{!, ?\} \times \Pi^\circ$, does there exist a finite automaton C over Σ and $\Pi \cup \Pi^\circ$ that satisfies observability and controllability constraints and such that A is bisimilar to $(B \otimes L^\circ) \times C$?

From the orchestrator synthesis problem to the control problem



Orchestrator synthesis problem

Does there exist a finite automaton M over Π such that A is bisimilar to $B \otimes M$?

Control problem

If $\Sigma_{ob} = \Sigma_{ct} = \{!, ?\} \times \Pi^\circ$, does there exist a finite automaton C over Σ and $\Pi \cup \Pi^\circ$ that satisfies observability and controllability constraints and such that A is bisimilar to $(B \otimes L^\circ) \times C$?

Orchestrator synthesis variant: algorithm

Solve($\Sigma, \Pi, B_g, B_1, \dots, B_n$)

$\Pi^\circ := \text{Copy}(\Pi);$

$L^\circ := \text{LargMed}(\Pi^\circ);$

$\Sigma' := \Sigma \cup \{!, ?\} \times (\Pi \cup \Pi^\circ);$

$B := B_1 \otimes \dots \otimes B_n \otimes L^\circ;$

$\Sigma_{ob} := \{!, ?\} \times \Pi^\circ;$

$\Sigma_{ct} := \{!, ?\} \times \Pi^\circ;$

return $\text{Control}(\Sigma', \Sigma_{ob}, \Sigma_{ct}, B_g, B)$

Orchestrator synthesis variant: lower bound

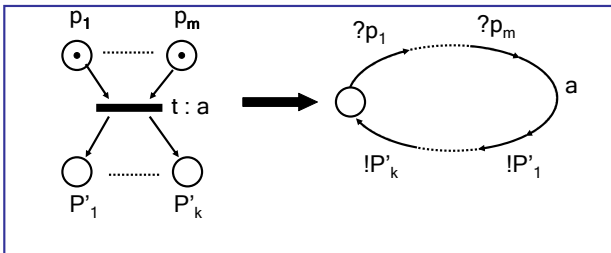
Theorem

The composition problem for bisimulation is **EXPTIME-hard**

Proof

A reduction of the equivalence problem of 1-safe Petri nets to our problem

From 1-safe Petri nets to CCA



1-safe Petri nets:

N
 O



CCA:

B_g
 B_1

N and O are bisimilar
iff

$\exists M$ such that B_g is bisimilar to $B_1 \otimes M$

Results for the orchestrator synthesis variant

Relation	Complexity
Trace inclusion	<i>EXPSPACE-complete</i>
Trace equivalence	<i>EXPSPACE-hard</i>
Simulation	<i>EXPTIME-complete</i>
Bisimulation	<i>EXPTIME-hard</i> <i>2-EXPTIME</i>

Conclusion

- Formal model of services
- Study of the composition problem decidability when the channels are unbounded
- Study of the composition problem complexity for three variants with four relations

Future works

- Open problems
- Consider messages as terms of first order logic
- Consider temporal conditions
- Consider failures during the composition
- Consider trust, argumentation and negotiation between agents

Thank you