Composition of interactive Web services based on controller synthesis

Philippe Balbiani
Philippe.Balbiani@irit.fr

Fahima Cheikh
Fahima.Cheikh@irit.fr

Guillaume Feuillade
Guillaume.Feuillade@irit.fr

Institut de Recherche en Informatique de Toulouse, Université Paul Sabatier, France

Abstract

We study an abstract form of service composition where Web services are represented as nondeterministic communicating automata. Considering the case in which communication is done via channels able to hold at most one message at a time, the service composition problem consists, given a client service, a goal service and a community of available services, to determine whether there exists a mediator service able to communicate with the client and the services of the given community in such a way that their global behavior satisfies the client service request expressed as the given goal service. We demonstrate the decidability of this problem via a reduction to a decidable control problem.

1. Introduction

Actual development of web services and service-oriented programming motivates the development and the use of formal methods for the interaction between electronic agents. Web services are distributed applications available to the users of a network by means of communication protocols. These protocols involve three classes of electronic agents: available services, potential clients, and mediators which relate the need of clients to the available services. Access to a unique service through an interface allowing information exchange with one unique client may prove useful. However the composition of electronic services offers the possibility to generate new services and to satisfy queries that no service can individually answer. The Web service composition problem is thus the one of abstract representation and algorithmic treatment of communication between agents within the setting of service oriented programming. We focus on the following problem: how do the potential clients express their needs and how do the mediators manage to relate available services and client needs. Tools for solving this problem are not yet completely defined.

In this paper, services are abstracted as non-deterministic finite communicating automata. A set of available services constitutes the *community* of services. A *client* is a query/answer based service. A *mediator* is a service which role is to be the interface between the client and the available services: it exchanges messages with the client and with services in the community. The composition synthesis problem we consider is, given a community of services and a goal service, to synthesize a *mediator* such that the triplet client/mediator/community is equivalent to the goal service; the equivalence is given by a bisimulation relation modulo some hidden internal actions and communications. Our main result is that automatic synthesis is decidable. We prove that it reduces into a *control* problem. The control problem is, given an automaton and a logical formula, to find an automaton called the *controller* satisfying some controllability and observability constraints and such that the synchronous product between the automaton and the controller satisfies the formula.

The composition problem we consider, namely the *synthesis* one, differs from the *orchestration* one addressed in [2] in the following way: in the orchestration problem, the objective is to find a proper scheduling for the services of the community to match the goal services, while in the synthesis problem the objective is to build a mediator which communicates with the services. The second difference is that in our approach we consider communicating services.

However, studies where services are able to send and to receive messages already exists [7, 5, 4]. In these studies, client specification is given by a logical formula which represents the client's goals. By communicating together, services modify their knowledge and those of their client. It is the approach considered by [7]. In all cases, to compose services together is to interleave their actions sequences in accordance with a client specification. The composition problem is difficult to solve, as shown by theoretical complexity results obtained in the papers mentioned above [2, 7, 5, 4].

In this paper, communication is done via a set of *ports* which are communication channels able to hold at most one message at a time. Thus the communication process is asynchronous. Note that this setting allows to encode any kind of bounded communication channels. In our case, the internal actions are not necessarily hidden actions as in [5, 4].

The paper is organized as follows. In section 2, we formally present the synthesis composition problem and we show a reduction into a simpler composition problem. In Section 3, we present the controller synthesis problem and recalls the decidability results. In Section 4, we show that the simplified composition problem is decidable by a reduction into control problem. Finally, in Section 5, we compare this approach to related work and we conclude.

2. Services composition

The composition problem for services is to decide whether for a given client service, a goal service and a set of available services, there exists a mediator service that when combined to the available services and the client service has the same behavior as the client service combined with the goal service. Here, the client service can only communicate and the goal service represents the request of the client. The purpose of this section is to give the definition of all notions needed to define formally the notion of service and service composition problem.

2.1. Finite automata

First we present the classical model of finite automaton.

Definition 1. A *finite automaton* is a structure $A = (Q, q^0, \Sigma, \delta)$ where

- Q is a finite set of states,
- q^0 is the initial state,
- Σ is a finite set of actions,
- $\delta : Q \times \Sigma \to 2^Q$ is the transition function.

We write $q \xrightarrow{a}_A q'$ instead of $q' \in \delta(q, a)$. Let q and q' be two states in Q and Θ be a finite set of actions. We say that q' is Θ-*accessible* in A from q, in symbols $q \xrightarrow{\Theta}_A q'$ iff there exists $\tau \in \Theta$ such that $q \xrightarrow{\tau}_A q'$. Let $\xrightarrow{\Theta}_A{}^*$ be the smallest reflexive and transitive relation containing $\xrightarrow{\Theta}_A$. We write $q \overset{a,\Theta}{\rightsquigarrow}_A q'$ iff there exist $q^1, q^2 \in Q$ such that $q \xrightarrow{\Theta}_A{}^* q^1 \xrightarrow{a}_A q^2 \xrightarrow{\Theta}_A{}^* q'$. We define two equivalence relations between automata: isomorphism and Θ-bisimulation. Let $A_1 = (Q_1, q_1^0, \Sigma_1, \delta_1)$ and $A_2 = (Q_2, q_2^0, \Sigma_2, \delta_2)$ be two finite automata.

Definition 2. We say that A_1 is *isomorphic* to A_2 if $\Sigma_1 = \Sigma_2$ and there exists a bijection $g : Q_1 \to Q_2$, such that $g(q_1^0) = q_2^0$ and for all $q_1, q_1' \in Q_1$ and for all $a \in \Sigma_1$, $q_1 \xrightarrow{a}_{A_1} q_1'$ iff $g(q_1) \xrightarrow{a}_{A_2} g(q_1')$.

Definition 3. Let $\Theta \subseteq \Sigma_1 \cup \Sigma_2$ be a set of actions. We shall say that a binary relation $Z \subseteq Q_1 \times Q_2$ is a Θ-*bisimulation* from A_1 to A_2 iff for all $q_1 \in Q_1$ and for all $q_2 \in Q_2$, if $(q_1, q_2) \in Z$ then for all $a \in \Sigma_1 \cup \Sigma_2$,

- if, for some $q_1' \in Q_1$, $q_1 \overset{a,\Theta}{\rightsquigarrow}_{A_1} q_1'$ then, for some $q_2' \in Q_2$, $q_2 \overset{a,\Theta}{\rightsquigarrow}_{A_2} q_2'$ and $(q_1', q_2') \in Z$,
- if, for some $q_2' \in Q_2$, $q_2 \overset{a,\Theta}{\rightsquigarrow}_{A_2} q_2'$ then, for some $q_1' \in Q_1$, $q_1 \overset{a,\Theta}{\rightsquigarrow}_{A_1} q_1'$ and $(q_1', q_2') \in Z$.

A_1 and A_2 are Θ-*bisimilar* if $(q_1^0, q_2^0) \in Z$ for some Θ-bisimulation Z. We introduce the notion of synchronous product of automata, needed in next sections. Let $A_1 = (Q_1, q_1^0, \Sigma_1, \delta_1)$ and $A_2 = (Q_2, q^0{}_2, \Sigma_2, \delta_2)$ be two finite automata.

Definition 4. The *synchronous product* of A_1 and A_2 is the finite automaton $A_1 \times A_2 = (Q, q^0, \Sigma, \delta)$ where

- $Q = Q_1 \times Q_2$,
- $q^0 = (q_1^0, q_2^0)$,
- $\Sigma = \Sigma_1 \cap \Sigma_2$,
- the transition function $\delta : Q \times \Sigma \to 2^Q$ is such that $(q_1, q_2) \xrightarrow{a}_{A_1 \times A_2} (q_1', q_2')$ iff $q_1 \xrightarrow{a}_{A_1} q_1'$ and $q_2 \xrightarrow{a}_{A_2} q_2'$.

2.2. Communicating automata

Our model of services is based on communicating automata.

Definition 5. A *communicating automaton* is a structure $A = (Q, q^0, Port, \Sigma, \delta)$ where

- Q is a finite set of states,
- q^0 is the initial state,
- $Port$ is a finite set of ports,
- Σ is a finite set of actions,
- $\delta : Q \times (\Sigma \cup (\{?, !\} \times Port)) \to 2^Q$ is the transition function.

For the sake of simplicity, the elements $(?, p)$ and $(!, p)$ in $\{?, !\} \times Port$ will be denoted $?p$ and $!p$ respectively. The action $?p$ consists in receiving a message on the port p, whereas the action $!p$ consists in sending a message to the port p. In this paper we consider that ports are communication channels able to hold at most one message at a time. This restriction was also considered by [3].

Every communicating automaton is associated to a finite automaton in the following way.

Definition 6. Let $A = (Q, q^0, Port, \Sigma, \delta)$ be a communicating automaton. The *associated finite automaton* $\mathrm{FA}(A) = (Q', q^{0\prime}, \Sigma', \delta')$ is defined as follows

- $Q' = \{(q, V) \mid q \in Q \text{ and } V \subseteq Port\}$,
- $q^{0\prime} = (q^0, \emptyset)$,
- $\Sigma' = \Sigma \cup (\{?, !\} \times Port)$,
- the transition function $\delta' : Q' \times \Sigma' \to 2^{Q'}$ is such that $(q, V) \xrightarrow{a}_{\mathrm{FA}(A)} (q', V')$ iff one of the three following conditions is satisfied
  - $a \in \Sigma$, $q \xrightarrow{a}_A q'$ and $V = V'$,
  - $a = ?p$, $q \xrightarrow{a}_A q'$, $p \in V$ and $V' = V \setminus \{p\}$,
  - $a = !p$, $q \xrightarrow{a}_A q'$, $p \notin V$ and $V' = V \cup \{p\}$.

In the definition above, V represents the current set of nonempty ports. Observe that initially all ports are empty. Intuitively, the second condition, in the definition of δ', means that receiving a message on port p is possible only if port p is nonempty. In a dual way, the third condition means that sending a message to port p is possible only if port p is not full. The behavior of a set of communicating automata is formally defined by their asynchronous product or their synchronous product.

Definition 7. The *asynchronous product* of two communicating automata $A_1 = (Q_1, q_1^0, Port_1, \Sigma_1, \delta_1)$ and $A_2 = (Q_2, q_2^0, Port_2, \Sigma_2, \delta_2)$ is the communicating automaton $A_1 \otimes A_2 = (Q, q^0, Port, \Sigma, \delta)$ where

- $Q = Q_1 \times Q_2$,
- $q^0 = (q_1^0, q_2^0)$,
- $Port = Port_1 \cup Port_2$,
- $\Sigma = \Sigma_1 \cup \Sigma_2$,
- the transition function $\delta : Q \times (\Sigma \cup (\{?, !\} \times Port)) \to 2^Q$ is such that $(q_1, q_2) \xrightarrow{a}_{A_1 \otimes A_2} (q_1', q_2')$ iff one of two following conditions is satisfied
  - $q_1 \xrightarrow{a}_{A_1} q_1'$ and $q_2' = q_2$,
  - $q_2 \xrightarrow{a}_{A_2} q_2'$ and $q_1' = q_1$.

Definition 8. The *synchronous product* of two communicating automata $A_1 = (Q_1, q_1^0, Port_1, \Sigma_1, \delta_1)$ and $A_2 = (Q_2, q_2^0, Port_2, \Sigma_2, \delta_2)$ is the communicating automaton $A_1 \times A_2 = (Q, q^0, Port, \Sigma, \delta)$ where

- $Q = Q_1 \times Q_2$,
- $q^0 = (q_1^0, q_2^0)$,
- $Port = Port_1 \cap Port_2$,
- $\Sigma = \Sigma_1 \cap \Sigma_2$,
- the transition function $\delta : Q \times (\Sigma \cup (\{?, !\} \times Port)) \to 2^Q$ is such that $(q_1, q_2) \xrightarrow{a}_{A_1 \times A_2} (q_1', q_2')$ iff $q_1 \xrightarrow{a}_{A_1} q_1'$ and $q_2 \xrightarrow{a}_{A_2} q_2'$.

Definition 9. Let Θ be a finite set of actions. we say that two communicating automata A_1 and A_2 *have equivalent behaviors with respect to* Θ, in symbols $A_1 \approx_\Theta A_2$, iff $\mathrm{FA}(A_1)$ and $\mathrm{FA}(A_2)$ are Θ-bisimilar.

Definition 10. Let $A = (Q, q^0, Port, \Sigma, \delta)$ be a communicating automaton. $Auto(A) = (Q', q^{0\prime}, \Sigma', \delta')$ is the finite automaton defined as follows

- $Q' = Q$,
- $q^{0\prime} = q^0$,
- $\Sigma' = \Sigma \cup (\{?, !\} \times Port)$,
- $\delta' = \delta$.

Definition 11. We say that two communicating automata A_1 and A_2 are *isomorphic* iff $Auto(A_1)$ and $Auto(A_2)$ are isomorphic.

2.3. Service composition problem

Following the line of reasoning suggested by [3], we need to introduce the following notions:

- client services,
- goal services,
- available services and
- mediator services.

We define a *client service* as a communicating automaton $A_c = (Q_c, q_c^0, Port_c, \Sigma_c, \delta_c)$ such that $Q_c = \{q_c^0, q_c^1\}$ is a pair of states, $\Sigma_c = \emptyset$ and for all $p \in Port_c$, $\delta_c(q_c^0, ?p) = \emptyset$ and $\delta_c(q_c^0, !p) = \emptyset$. As an example, we consider the client service represented in figure 1. In this figure, $Word$ and $Definition$ are ports. A *goal service* for A_c is a communicating automaton $A_g = (Q_g, q_g^0, Port_g, \Sigma_g, \delta_g)$ such that $Port_g = Port_c$. As an example, we consider the goal service represented in figure 1. In this figure $Translate$ and $Search$ are actions. We define a *mediator service* as a communicating automaton $M = (Q_M, q_M^0, Port_M, \Sigma_M, \delta_M)$ such that $\Sigma_M = \emptyset$. We simply define *available services* as communicating automata. As an example, we consider two availables services: Dictionary service and Translator service. This two services are represented in figure 2. In this figure, $WrdToTrans$, $Translation$, $WordToDef$ and $DicDef$ are ports. We suppose that the Dictionary service can give only definitions of english words,

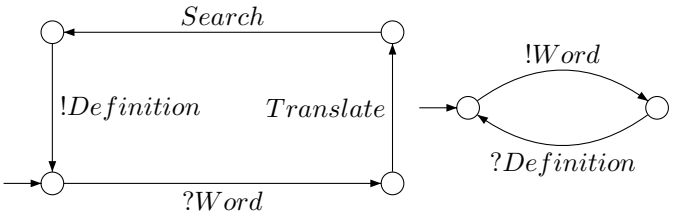

Figure 1. From left to right a goal service and a client service

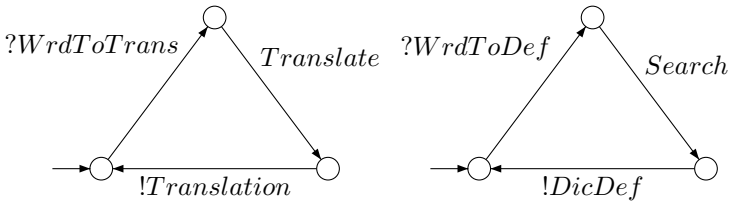

Figure 2. From left to right, Dictionary available service and Translator available service

while the translator service can translate an italian word to an english one. Let $A_c = (Q_c, q_c^0, Port_c, \Sigma_c, \delta_c)$ be a client service and $A_1 = (Q_1, q_1^0, Port_1, \Sigma_1, \delta_1), \ldots, A_n = (Q_n, q_n^0, Port_n, \Sigma_n, \delta_n)$ be available services. Without loss of generality, we will always assume that $Port_c$ and $Port_1 \cup \ldots \cup Port_n$ are disjoint sets of ports. Let $M = (Q_M, q_M^0, Port_M, \Sigma_M, \delta_M)$ be a mediator service. We say that M is a *mediator for* A_c and $A_1, \ldots, A_n$ iff $Port_M = Port_c \cup Port_1 \cup \ldots \cup Port_n$. As an example, we consider the mediator service represented in figure 3.

We consider now the decision problem P_{comp} defined as follows:

Problem 1 (Composition problem P_{comp})**.**

Instance a client service A_c, a goal service A_g for A_c, n available services $A_1, \ldots, A_n$ such that $Port_c$ and $Port_1 \cup \ldots \cup Port_n$ are disjoint,

Question does there exist a mediator service M for A_c and $A_1, \ldots, A_n$ such that $A_c \otimes A_g \approx_\Theta A_c \otimes A_1 \otimes \ldots \otimes A_n \otimes M$, where $\Theta = \{?, !\} \times (Port_1 \cup \ldots \cup Port_n)$?

The above definition captures formally the intuitive notion of the service composition problem. This problem can

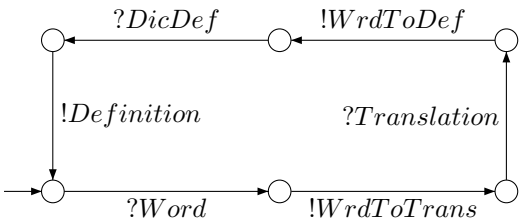

Figure 3. Mediator service

be informally stated as follows: given a client service, a goal service and available services, synthesize a mediator service establishing a connection between the client service and the available services in such a way that the behavior of the available services communicating with the client service through the intermediary of the mediator service is equivalent, leaving aside the communications executed by the available services, to the behavior of the goal service communicating with the client service. Such problems has been considered by a lot of people including [3]. Its complete solution, nevertheless, has never been given. The decision problem P_{comp} can be reduced to the following more abstract problem:

Problem 2 (Composition problem P^1_{comp})**.**

Instance communicating automata A and B such that $Port_A \subseteq Port_B$,

Question does there exist a mediator service M such that $Port_M = Port_B$ and $A \approx_\Theta B \otimes M$, where $\Theta = \{?, !\} \times (Port_B \setminus Port_A)$?

Proposition 1. The decision problem P_{comp} can be reduced to the decision problem P^1_{comp}.

Proof. Let $A_c = (Q_c, q_c^0, Port_c, \Sigma_c, \delta_c)$ be a client service, $A_g = (Q_g, q_g^0, Port_g, \Sigma_g, \delta_g)$ be a goal service for A_c and $A_1 = (Q_1, q_1^0, Port_1, \Sigma_1, \delta_1), \ldots, A_n = (Q_n, q_n^0, Port_n, \Sigma_n, \delta_n)$ be n available services such that $Port_c$ and $Port_1 \cup \ldots \cup Port_n$ are disjoint. Let $A = A_c \otimes A_g$ and $B = A_c \otimes A_1 \otimes \ldots \otimes A_n$. Clearly, $Port_A \subseteq Port_B$. Moreover, $Port_B \setminus Port_A = Port_1 \cup \ldots \cup Port_n$. The reader can easily verify that there exists a mediator service $M = (Q_M, q_M^0, Port_M, \Sigma_M, \delta_M)$ for A_c and $A_1, \ldots, A_n$ such that $A_c \otimes A_g \approx_\Theta A_c \otimes A_1 \otimes \ldots \otimes A_n \otimes M$, where $\Theta = \{?, !\} \times (Port_1 \cup \ldots \cup Port_n)$ iff there exists a mediator service $M' = (Q_{M'}, q_{M'}^0, Port_{M'}, \Sigma_{M'}, \delta_{M'})$ such that $Port_{M'} = Port_B$ and $A \approx_{\Theta'} B \otimes M'$, where $\Theta' = \{?, !\} \times (Port_B \setminus Port_A)$.

In order to apply, in section 4, the techniques of controller synthesis for solving the decision problem P^1_{comp}, we need to introduce the notion of complete mediator. Let $B = (Q_B, q_B^0, Port_B, \Sigma_B, \delta_B)$ be a communicating automaton where $Port_B = \{p_1, \ldots, p_m\}$. Let us consider the finite set of port copies $Port'_B = \{p'_1, \ldots, p'_m\}$ such that $Port'_B \cap Port_B = \emptyset$. *The complete mediator with respect to* B is the mediator service $L = (Q_L, q_L^0, Port_L, \Sigma_L, \delta_L)$ where $Q_L = \{q_L^0\}$, $Port_L = Port'_B$, $\Sigma_L = \emptyset$ and for all port $p' \in Port'_B$, $\delta(q_L^0, ?p') = \{q_L^0\}$ and $\delta(q_L^0, !p') = \{q_L^0\}$.

Let $A = (Q_A, q_A^0, Port_A, \Sigma_A, \delta_A)$ be a communicating automaton such that $Port_A = Port_B \cup Port_L$ or $Port_A = Port_L$. The *renaming of* A *with respect to* $Port'_B$ is the communicating automaton $Ren(A) = (Q, q^0, Port, \Sigma, \delta)$ where:

- $Q = Q_A$,
- $q^0 = q^0_A$,
- $Port = Port_B$,
- $\Sigma = \Sigma_A$ and
- the transition function $\delta : Q \times (\Sigma \cup (\{?,!\} \times Port)) \rightarrow 2^Q$ is such that for all $a \in \Sigma$, $q \xrightarrow{a}_{Ren(A)} q'$ iff $q \xrightarrow{a}_A q'$ where as for all $p \in Port$, $q \xrightarrow{\sigma p}_{Ren(A)} q'$ iff $q \xrightarrow{\sigma p}_A q'$ or $q \xrightarrow{\sigma p'}_A q'$, where $\sigma \in \{?,!\}$.

Intuitively $Ren(A)$ consists to replace the transitions of A labeled by $?p'$ (resp. $!p'$), where p' is a port in $Port'_B$ by transitions labeled by $?p$ (resp. $!p$), such that p' is the copy of p.

Let us now consider the decision problem P^2_{comp} defined as follows:

Problem 3 (Composition problem P^2_{comp})**.**

Instance communicating automata A and B such that $Port_A \subseteq Port_B$,

Question does there exist a communicating automaton C such that $Port_C = Port_B \cup Port_L$, $\Sigma_C = \Sigma_B$ and $A \approx_\Theta Ren(B \otimes (C \times L))$, where $\Theta = \{?,!\} \times (Port_B \setminus Port_A)$?

Intuitively, the decision problem P^2_{comp} captures the idea that, instead of synthesizing a mediator service M such that $Port_M = Port_B$, we synthesize a communicating automaton C that play the role of a supervisor for the complete mediator service L of B. When the communicating automaton C is combined to the complete mediator L, it restricts the communications of L.

Proposition 2. The decision problem P^1_{comp} is equivalent to the decision problem P^2_{comp}.

Proof. Let $A = (Q_A, q^0_A, Port_A, \Sigma_A, \delta_A)$ and $B = (Q_B, q^0_B, Port_B, \Sigma_B, \delta_B)$ be communicating automata such that $Port_A \subseteq Port_B$ and let $\Theta = \{?,!\} \times (Port_B \setminus Port_A)$ be a set of actions. We have to prove that there exists a mediator service $M = (Q_M, q^0_M, Port_M, \Sigma_M, \delta_M)$ such that $Port_M = Port_B$ and $A \approx_\Theta B \otimes M$ iff there exists a communicating automaton $C = (Q_C, q^0_C, Port_C, \Sigma_C, \delta_C)$ such that $Port_C = Port_B \cup Port_L$, $\Sigma_C = \Sigma_B$ and $A \approx_\Theta Ren(B \otimes (C \times L))$. Remind that $Port_L = Port'_B$ consists of copies of ports in $Port_B$.

Concerning the left to right implication, suppose that there exists a mediator service $M = (Q_M, q^0_M, Port_M, \Sigma_M, \delta_M)$ such that $Port_M = Port_B$ and $A \approx_\Theta B \otimes M$. Let us consider the communicating automaton $C = (Q_C, q^0_C, Port_C, \Sigma_C, \delta_C)$ where

- $Q_C = Q_M$,
- $q^0_C = q^0_M$,
- $Port_C = Port_B \cup Port_L$,
- $\Sigma_C = \Sigma_B$ and
- the transition function $\delta_C : Q_C \times (\Sigma_C \cup (\{?,!\} \times Port_C)) \rightarrow 2^Q$ is such that for all $q \in Q_C$, for all $a \in \Sigma_C$, $\delta_C(q,a) = \emptyset$ and for all $p \in Port_B$, $\delta_C(q, \sigma p) = \emptyset$ and $\delta_C(q, \sigma p') = \delta_M(q, \sigma p)$, where $\sigma \in \{?,!\}$.

Since $A \approx_\Theta B \otimes M$, it suffices to demonstrate that FA$(B \otimes M)$ and FA$(Ren(B \otimes (C \times L)))$ are isomorphic. Let $g : Q_B \times Q_M \times 2^{Port_B} \rightarrow Q_B \times Q_C \times Q_L \times 2^{Port_B}$ be the bijection defined by $g((q_B, q_M), V) = ((q_B, (q_M, q^0_L)), V)$. The reader may easily verify that $g((q^0_B, q^0_M), \emptyset) = ((q^0_B, (q^0_M, q^0_L)), \emptyset)$ and for all $q_B, q'_B \in Q_B$, for all $q_M, q'_M \in Q_M$ and for all $V, V' \in 2^{Port_B}$, there is a transition in FA$(B \otimes M)$ between $((q_B, q_M), V)$ and $((q'_B, q'_M), V')$ iff there is a similar transition in FA$(Ren(B \otimes (C \times L)))$ between $g((q_B, q_M), V)$ and $g((q'_B, q'_M), V')$.

Concerning the right to left implication, suppose that there exists a communicating automaton $C = (Q_C, q^0_C, Port_C, \Sigma_C, \delta_C)$ such that $Port_C = Port_B \cup Port_L$, $\Sigma_C = \Sigma_B$ and $A \approx_\Theta Ren(B \otimes (C \times L))$. Let us consider that $M = Ren(C \times L)$. Consequently, $Port_M = (Port'_B \setminus Port'_B) \cup Port_B = Port_B$. Clearly, $Ren(B \otimes (C \times L)) = B \otimes Ren(C \times L)$. Thus $Ren(B \otimes (C \times L))$=$B \otimes M$. Since $A \approx_\Theta Ren(B \otimes (C \times L))$ then $A \approx_\Theta B \otimes M$. □

3. Controller synthesis

Ramadge and Wonham initiate in [9] the control theory of discrete event systems. We briefly present this theory and an extension, the one of [1] and a subproblem which is a step in the direction of service composition. We fix a finite alphabet Σ of actions.

3.1. Control with observability and controllability

Given a finite automaton G over Σ, also called the *plant*, the control problem is to find a particular finite automaton C over Σ, called the *controller*, which role is to prevent the plant to perform some unwanted sequences of actions. The controller must also take into account some *observability* and *controllability* constraints.

These constraints are given by the partition of Σ into the set Σ_{ob} of observable events and the set Σ_{uob} of unobservable events. The alphabet Σ is also partitioned into the set

Σ_{ct} of controllable events and the set Σ_{uct} uncontrollable events.

The observability constraint $(\mathcal{O})$ and the controllability constraint $(\mathcal{C})$ for a controller C are defined as follows:

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

$(\mathcal{O})$ for any state q of C and for any unobservable event $a \in \Sigma_{uct}$, if there is a transition from q labeled by a then this transition is a loop over q.

The controller satisfies $(\mathcal{C})$ iff it reacts to every uncontrollable event whereas it satisfies $(\mathcal{O})$ iff it cannot detect the occurrence of any unobservable event. The *controlled system* is the synchronized product $G \times C$. Thus, the controller operates by restricting the set of *finite traces* of the plant, where a *trace* of an automaton is a legal sequence of actions for this automaton.

Informally, the basic control problem can be stated as follows:

Problem 4 (Basic control problem)**.**

Instance a plant G and a language $K \subseteq \Sigma^*$ of finite traces,

Question does there exist a controller C satisfying $(\mathcal{C})$ and $(\mathcal{O})$ and such that the finite traces of $G \times C$ are all in K ?

Many variants of this control problem have been studied. In particular, the language K of legal behaviors can be replaced by the general notion of *control objective*. We focus on control objectives given as logical formulas. In [1], the authors propose a solution for the following control problem:

Problem 5 (Formula-based control problem)**.**

Instance a plant G and a formula ϕ of the modal μ-calculus,

Question does there exist a controller C satisfying $(\mathcal{C})$ and $(\mathcal{O})$ and such that $G \times C \models \phi$?

The solution provides an algorithm for constructing the controller. In fact, the authors solve a more general problem where $(\mathcal{C})$ and $(\mathcal{O})$ are encoded in a formula satisfied by the controller. The problem we present here is sufficient for the present paper. Note that non-deterministic plants are allowed since Problem 5 with non-deterministic plants can be reduced to the same problem with deterministic plant [8]. The problem can also be solved without this reduction as shown in [6].

3.2. Tau bisimulation as control objective

In Section 2, we introduced service composition and Θ-bisimulation as behavioral equivalence for validating the composition. Let Θ be a subset of Σ. We propose a variant of the Control Problem 5 where the control objective is based upon Θ-bisimulation.

Formally, consider the following problem:

Problem 6 (P_{bbcp}: bisimulation control problem)**.**

Instance Given a plant G and a finite automaton A,

Question does there exist a controller C satisfying $(\mathcal{C})$ and $(\mathcal{O})$ and such that $Ren(G \times C) \approx_\Theta A$?

We show that this problem is a particular case of Problem 5. Indeed, the Θ-bisimulation between two finite automata can be reduced to the satisfiability of a μ-calculus formula as stated by the following proposition.

Proposition 3. Given a finite automaton A, there exists a μ-calculus formula ϕ_A such that for every finite automaton P, the following assertion holds:

$$P \models \phi_A \Leftrightarrow Ren(P) \approx_\Theta A$$

Proof. This is an obvious variant of a standard result. □

Thus we get:

Corollary 1. The problem P_{bbcp} is decidable, and the controller synthesis is effective.

Proof. From Proposition 3, each instance of P_{bbcp} is equivalent to an instance of Problem 5, which have an effective decision procedure. □

4. From services to controllers

In this section, we show that the service composition problem, as formally defined in section 2, can be seen as a controller synthesis problem, as formally defined in the previous section. In what follows we prove that for solving the service composition problem, the techniques of controller synthesis can be applied. For this aim, we first give the definition of the functions $Comp$ and FA_{Copies}. Then, we propose lemma 1. The propositions 4 and 6 below are necessary to establish the proof of the lemma.

Definition 12. Let $A = (Q, q^0, \Sigma, \delta)$ be a finite automaton and $Port$ be a finite set of ports. The communicating automaton $Com(A, Port) = (Q', q^{0\prime}, Port', \Sigma', \delta')$ associated to A with respect to $Port$ is defined as follows

- $Q' = Q$,
- $q^{0\prime} = q^0$,

- $Port' = Port$,
- $\Sigma' = \Sigma \setminus (\{?,!\} \times Port)$,
- $\delta' = \delta$.

Definition 13. Let $A = (Q_A, q_A^0, Port_A, \Sigma_A, \delta_A)$ be a communicating automaton. Let $Port'_A$ be a finite set of ports copies. The finite automaton $\mathrm{FA}_{Copies}(A) = (Q, q^0, \Sigma, \delta)$ associated to A is defined as follows

- $Q = \{(q_A, V) \mid q_A \in Q_A \text{ and } V \subseteq Port_A\}$,
- $q^0 = (q_A^0, \emptyset)$,
- $\Sigma = \Sigma_A \cup (\{?,!\} \times (Port_A \cup Port_{A'}))$,
- the transition function $\delta : Q \times \Sigma \to 2^Q$ is such that $(q_A, V) \xrightarrow{a}_{\mathrm{FA}_{\mathrm{Copies}}(A)} (q'_A, V')$ iff one of the three following conditions is satisfied
  - $a \in \Sigma_A$, $q_A \xrightarrow{a}_A q'_A$ and $V = V'$,
  - $a = ?p$, $q_A \xrightarrow{a}_A q'_A$, $p \in V$ and $V' = V \setminus \{p\}$,
  - $a = !p$, $q_A \xrightarrow{a}_A q'_A$, $p \notin V$ and $V' = V \cup \{p\}$,
  - $a = ?p'$, $q_A \xrightarrow{a}_A q'_A$, $p \in V$ and $V' = V \setminus \{p\}$,
  - $a = !p'$, $q_A \xrightarrow{a}_A q'_A$, $p \notin V$ and $V' = V \cup \{p\}$.

Proposition 4. Let B, C be communicating automata such that $Port_C = Port_B \cup Port'_B$, $\Sigma_C = \Sigma_B$. If $Auto(C)$ satisfies $(\mathcal{O})$ and $(\mathcal{C})$, when $\Sigma_{ob} = \Sigma_{ct} = \{?,!\} \times Port'_B$ then $\mathrm{FA}_{Copies}(B \otimes (C \times L))$ and $\mathrm{FA}_{Copies}(B \otimes L) \times Auto(C)$ are isomorphic.

Proof. Let $B = (Q_B, q_B^0, Port_B, \Sigma_B, \delta_B)$ and $C = (Q_C, q_C^0, Port_C, \Sigma_C, \delta_C)$ be communicating automata such that $Port_C = Port_B \cup Port'_B$, $\Sigma_C = \Sigma_B$. Suppose that $Auto(C)$ satisfies $(\mathcal{O})$ and $(\mathcal{C})$, when $\Sigma_{ob} = \{?,!\} \times Port'_B$. Consequently, for all state $q_{Auto(C)} \in Q_{Auto(C)}$ and for all action $a \in \Sigma_B \cup (\{?,!\} \times Port'_B)$, $q_{Auto(C)} \xrightarrow{a}_{Auto(C)} q_{Auto(C)}$. Let $g : (Q_B \times (Q_C \times Q_L)) \times 2^{Port_B} \to ((Q_B, Q_L) \times 2^{Port_B}) \times Q_C$ be the bijection defined by $g((q_B, (q_C, q_L^0)), V) = (((q_B, q_L^0), V), q_C)$. The reader may easily verify that $g((q_B^0, (q_C^0, q_L^0)), \emptyset) = (((q_B^0, q_L^0), \emptyset), q_C^0)$ and for all $q_B, q'_B \in Q_B$, for all q_C, q'_C and for all $V, V' \in 2^{Port_B}$, there is a transition in $\mathrm{FA}_{Copies}(B \otimes (C \times L))$ between $((q_B, (q_C, q_L^0)), V)$ and $((q'_B, (q'_C, q_L^0)), V')$ iff there is a similar transition in $\mathrm{FA}_{Copies}(B \otimes L) \times Auto(C)$ between $g((q_B, (q_C, q_L^0)), V)$ and $g((q'_B, (q'_C, q_L^0)), V')$. □

Proposition 5. Let B, C be communicating automata such that $Port_C = Port_B \cup Port'_B$, $\Sigma_C = \Sigma_B$. If $Auto(C)$ satisfies $(\mathcal{O})$ and $(\mathcal{C})$, when $\Sigma_{ob} = \Sigma_{ct} = \{?,!\} \times Port'_B$ then $\mathrm{FA}(B \otimes (C \times L))$ and $\mathrm{FA}(B \otimes L) \times Auto(C)$ are isomorphic.

Proof. The same argument used to prove 4, can be used to prove this proposition. □

Proposition 6. Let B, C be communicating automata such that $Port_C = Port_B \cup Port'_B$. Then there exists a communicating automaton C' such that $Port_{C'} = Port_C$, $\Sigma_{C'} = \Sigma_C$, $Auto(C')$ satisfies $(\mathcal{O})$ and $(\mathcal{C})$, when $\Sigma_{ob} = \Sigma_{ct} = \{?,!\} \times Port'_B$ and $C \times L$ is isomorphic to $C' \times L$.

Proof. Let $B = (Q_B, q_B^0, Port_B, \Sigma_B, \delta_B)$ and $C = (Q_C, q_C^0, Port_C, \Sigma_C, \delta_C)$ be communicating automata such that $Port_C = Port_B \cup Port'_B$. Let us consider the communicating automaton $C' = (Q_{C'}, q_{C'}^0, Port_{C'}, \Sigma_{C'}, \delta_{C'})$ such that $Q_{C'} = Q_C$, $q_{C'}^0 = q_C^0$, $Port_{C'} = Port_C$, $\Sigma_{C'} = \Sigma_C$ and the transition function $\delta_{C'} : Q_{C'} \times (\Sigma_{C'} \cup (\{?,!\} \times Port_{C'})) \to 2^{Q_{C'}}$ is such that: for all $a \in \Sigma_C$, $q \xrightarrow{a}_{C'} q$, for all $p \in Port_B$, $q \xrightarrow{\sigma p}_{C'} q$ and for all $p' \in Port'_B$, $q \xrightarrow{\sigma p'}_{C'} q'$ iff $q \xrightarrow{\sigma p'}_C q'$, $\sigma \in \{?,!\}$.

The reader may easily verify that $Auto(C \times L) = Auto(C' \times L)$. Hence, $C \times L$ and $C' \times L$ are isomorphic. □

Lemma 1. The decision problem P^2_{comp} can be reduced to the decision problem P_{bbcp}.

Proof. Let $A = (Q_A, q_A^0, Port_A, \Sigma_A, \delta_A)$ and $B = (Q_B, q_B^0, Port_B, \Sigma_B, \delta_B)$ be communicating automata such that $Port_A \subseteq Port_B$ and let $\Theta = \{?,!\} \times (Port_B \setminus Port_A)$. Let $A' = \mathrm{FA}(A)$, $G = \mathrm{FA}_{Copie}(B \otimes L))$, $\Sigma_{ob} = \Sigma_{ct} = \{?,!\} \times Port'_B$. We have to prove that there exists a communicating automaton C such that $Port_C = Port_B \cup Port_L$, $\Sigma_C = \Sigma_B$ and $A \approx_\Theta Ren(B \otimes (C \times L))$ iff there exists a controller C' satisfying $(\mathcal{C})$ and $(\mathcal{O})$ and such that $G \times C'$ and A' are Θ-bisimilar.

Concerning the left to right implication, suppose that $C = (Q_C, q_C^0, Port_C, \Sigma_C, \delta_C)$ is such that $Port_C = Port_B \cup Port_L$, $\Sigma_C = \Sigma_B$ and $A \approx_\Theta Ren(B \otimes (C \times L))$. Since $Port_C = Port_B \cup Port_L$, then according to proposition 6 there exists a communicating automaton C'' such that $Port_{C''} = Port_C$, $\Sigma_{C''} = \Sigma_C$, $Auto(C'')$ satisfies $(\mathcal{O})$ and $(\mathcal{C})$ and $C \times L$ is isomorphic to $C'' \times L$. Clearly, $A \approx_\Theta Ren(B \otimes (C \times L))$ and $C \times L$ is isomorphic to $C'' \times L$ implies that $A \approx_\Theta Ren(B \otimes (C'' \times L))$. Thus, $\mathrm{FA}(A)$ and $\mathrm{FA}(Ren(B \otimes (C'' \times L))$ are Θ-bisimilar. It is easy to prove that $\mathrm{FA}(Ren(B \otimes (C'' \times L))) = Ren(\mathrm{FA}_{Copies}(B \otimes (C'' \times L)))$. According to proposition 4, $\mathrm{FA}_{Copies}(B \otimes (C'' \times L))$ and $\mathrm{FA}_{Copies}(B \otimes L) \times Auto(C'')$ are isomorphic. Consequently, $\mathrm{FA}(A)$ and $Ren(\mathrm{FA}_{Copies}(B \otimes L) \times Auto(C''))$ are Θ-bisimilar. If we consider that $C' = Auto(C'')$ then A' and $Ren(G \times C')$ are Θ-bisimilar.

Concerning the right to left implication, suppose that C' is a finite automaton that satisfies $(\mathcal{O})$ and $(\mathcal{C})$ and such that $\mathrm{FA}(A)$ and $Ren(\mathrm{FA}_{Copies}(B \otimes L) \times C')$ are Θ-bisimilar. Let us consider $C = Com(C', Port_B \cup Port'_B)$.

Thus, $Port_C = Port_B \cup Port'_B$ and $\Sigma_C = \Sigma_B$. One can observe that $Auto(C) = C'$. According to proposition 4, $\mathrm{FA}_{Copies}(B \otimes (C \times L))$ and $\mathrm{FA}_{Copies}(B \otimes L) \times C'$ are isomorphic, which implies that $Ren(\mathrm{FA}_{Copies}(B \otimes (C \times L)))$ and $Ren(\mathrm{FA}_{Copies}(B \otimes L) \times C')$ are isomorphic. It is easy to prove that $Ren(\mathrm{FA}_{Copies}(B \otimes (C \times L))) = \mathrm{FA}(Ren(B \otimes (C \times L)))$. Consequently, $\mathrm{FA}(A)$ and $\mathrm{FA}(Ren(B \otimes (C \times L)))$ are Θ-bisimilar. Thus $A \approx_\Theta Ren(B \otimes (C \times L))$. □

With this established, we now come the main result of this paper.

Proposition 7. The composition problem P_{Comp} is decidable.

Proof. By propositions 1, 2, corollary 1 and lemma 1. □

5. Variants and open problems

Our main result, stated in proposition 7, is that service composition is decidable. An interesting (and still open) question is to evaluate the exact complexity of (P_{comp}). In other respect, our controller-based approach constitutes the basis of an algorithm for solving (P_{comp}). This algorithm can be informally described as follows:

- first, given, as input, a client service A_c, a goal service A_g and available services A_1, ..., A_n, compute $A = A_c \otimes A_g$ and $B = A_c \otimes A_1 \otimes \ldots \otimes A_n$,
- second, compute $A' = \mathrm{FA}(A)$ and $G = Ren(\mathrm{FA}(B \otimes L))$ where L is the complete mediator with respect to B,
- third, using a decision procedure for solving P_{bbcp}, determine whether there exists a controller C' such that A' and $G \times C'$ have equivalent behaviors.

Remark that A' and G can be deterministically computed in exponential time with respect to the size of the input A_c, A_g and A_1, ..., A_n. Moreover, the size of A' and G is exponential with respect to the size of the input. Seeing that P_{bbcp} can be solved in deterministic exponential time, our approach provides an algorithm that solves (P_{comp}) in deterministic double-exponential time. Nevertheless, the exact complexity of (P_{comp}) is still unknown.
Variants of (P_{comp}) can be considered as well. For example, one may ask whether, given A_c, A_g and A_1, ..., A_n, there exists a mediator M such that $A_c \otimes A_g$ and $A_c \otimes A_1 \otimes \ldots \otimes A_n \otimes M$ are trace-equivalent. In other respect, one may consider that ports are communication channels that can contain more than one message at a time. In this case, the problem complexity remains the same as in the case studied in this paper. The reason is that the size of the finite automaton associated to a communicating automaton will be $(k+1)^{|Ports|}$ instead of $2^{|Ports|}$ (k is the maximal number of messages that a port can contain). It would be interesting to find an efficient algorithm that works in exponential time. For this aim, we will possibly use heuristics or on-the-fly algorithms. Finally, one may define the notion of a communicating automaton in a less abstract way. More precisely, one may consider that real messages, i.e. closed terms in a first-order setting, are exchanged between services.

Acknowledgements

The preparation of this paper has been financially supported by the french project COPS and the european project AVANTSSAR.

References

[1] A. Arnold, A. Vincent, and I. Walukiewicz. Games for synthesis of controllers with partial observation. *Theoretical Computer Science*, 303:7–34, 2003.

[2] D. Berardi, D. Calvanese, G. De Giacomo, and M. Mecella. Composing web services with nondeterministic behavior. In *Proceedings of the International Conference of Web Services*, 2006.

[3] D. Berardi, D. Calvanese, G. D. Giacomo, R. Hull, and M. Mecella. Automatic composition of transition-based semantic web services with messaging. In *Proceedings of the International Conference on Very Large Data Bases*, 2005.

[4] X. Fu, T. Bultan, and J. Su. Conversation protocols: A formalism for specification and verification of reactive electronic services. In *Proceedings of the International Conference on Implementation and Application of Automata*, 2003.

[5] J. Hoffmann, P. Bertoli, and M. Pistore. Web service composition as planning, revisited: In between background theories and initial state uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2007.

[6] R. Kumar and M. A. Shayman. Centralized and decentralized supervisory control of nondeterministic systems under partial observation. *SIAM Journal of Control and Optimization*, 35:363–383, 1997.

[7] M. Pistore, A. Marconi, P. Bertoli, and P. Traverso. Automated composition of web services by planning at the knowledge level. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2005.

[8] J. Raclet and S. Pinchinat. The supervisory control problems for non-deterministic descrite-event systems: a logical approach. In *Proceedings of the IFAC Word Congress*, 2005.

[9] P. Ramadge and W. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77:81–98, 1989.