

# A formal model for semantic Web service composition

Freddy Lécué<sup>1,2</sup> and Alain Léger<sup>1</sup>

<sup>1</sup> France Telecom R&D, France,  
4, rue du clos courtel F-35512 Cesson Sévigné  
{freddy.lecue, alain.leger}@orange-ft.com  
<sup>2</sup> École Nationale Supérieure des Mines de Saint-Étienne, France  
158, cours Fauriel F-42023 Saint-Étienne cedex 2

**Abstract.** Automated composition of Web services or the process of forming new value added Web services is one of the most promising challenges in the semantic Web service research area. Semantics is one of the key elements for the automated composition of Web services because such a process requires rich machine-understandable descriptions of services that can be shared. Semantics enables Web service to describe their capabilities and processes, nevertheless there is still some work to be done. Indeed Web services described at functional level need a formal context to perform the automated composition of Web services. The suggested model (i.e., Causal link matrix) is a necessary starting point to apply problem-solving techniques such as regression-based search for Web service composition. The model supports a semantic context in order to find a correct, complete, consistent and optimal plan as a solution. In this paper an innovative and formal model for an AI planning-oriented composition is presented.

**Key words:** Semantic Web, Web service, AI planning, Automated composition, Automated reasoning.

## 1 Introduction

Web service [1] provides the feature richness, flexibility and scalability needed by enterprises to manage the SOA challenges. By Web services we mean loosely coupled, reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard internet protocols.

Web services proliferation over the web implies difficulties to find specific services that can perform specialized tasks. Nevertheless a combination of existing services is an alternative and promising approach although manual Web service combination from scratch can be difficult and time consuming. That is why new abilities are necessary to support dynamic and automated tasks such as discovery, selection and composition. The main ability is to describe capability (inputs, outputs, preconditions, and effects: IOPEs) and process model (Web services activities, interaction protocol) of Web services. The latter needs are covered by means of semantic Web services. Indeed a semantic Web service [2] is described as a Web service whose internal and external description is in a language that has well-defined semantics.

Composition of Web services is probably the most interesting challenge spawned by this paradigm. Most of the work in semantic Web services composition has focused on two main levels of composition: functional [3–5] and process [6–8] levels. The former level considers Web services as “atomic” components described in terms of their IOPEs, and executed in a simple request-response step. The latter level supposes Web services as stateful processes with an interaction protocol involving in different sequential, conditional, and iterative steps. The functional and process level composition are complementary methods to propose solutions for composition. In this paper, we study an AI planning-oriented functional composition of Web services through a new formal model i.e., the Causal link matrix (CLM). The CLM aims at not only storing all relevant Web services in a semantic way but also pre-chaining Web services according to a semantic link i.e., the causal link. According to a CLM, the *Ra4C* algorithm proposes a **Regression-based Approach for Composition**. Thus the issue of the paper is an automated process of chaining Web services according to their functional description.

The rest of the paper is organized as follows. Section 2 introduces a motivating example through an e-healthcare scenario. Section 3 presents the causal link matrix as a formal model to describe Web services at functional level. In section 4, an AI planning-oriented method is presented to solve a Web service composition with a specific CLM. We briefly comment on related work in section 5. Finally in section 6, we draw some conclusions and we talk about possible future directions.

## 2 A motivating example: an e-healthcare scenario

One of the most challenging problems in healthcare domain is providing a way to order and compose medical devices. Such a composition does not only improve the patient follow-up but also reduce the number of consultations, examinations, medical check-ups and consequently their price. Indeed a long-standing clinical observation in hospital is no longer a realistic issue for cost reasons since the elderly. In order to tackle this problem and propose an automated process of composition, we propose an AI planning oriented composition approach through the Causal link matrix. For this purpose, the existing applications and medical devices (e.g., sphygmomanometer) are wrapped as Web services. Thus telemedical collaborations are possible through the Web service paradigm. A solution of such a problem consists in implementing a composite and value-added Web service that can automate the patient follow-up by a reliable Web service interoperation, hence a long distance follow-up.

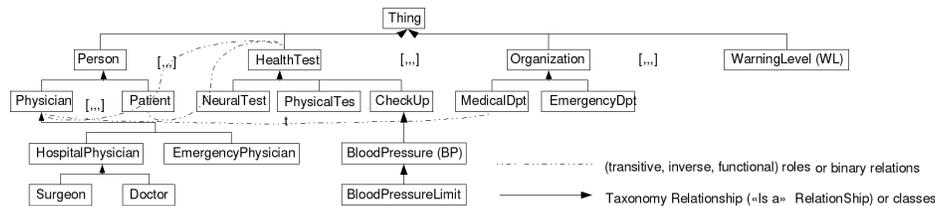


Fig. 1. A sample of an e-healthcare ontology  $\mathcal{T}$ .

Consider the above scenario with six different Web services:  $S_a$  returns the blood pressure (BP) of a patient given his PatientID (PID) and DeviceAddress (Add);  $S_b$  and  $S_{b'}$  return respectively the supervisor (Person) and a physician of an organisation (Org);  $S_c$  returns a Warning level (WL) given a blood pressure;  $S_d$  returns the Emergency department given a level of Warning;  $S_e$  returns the Organization given a Warning level.

### 3 Formal model

#### 3.1 Motivation

Algorithms for Web service composition have to not only find feasible plans with relevant Web services, but also find the optimal plan according to an optimization criteria. The latter criteria will be viewed as a quality of semantic connection between Web services (Input and output parameters relation). Indeed the semantic connection between Web services is considered as essential to form new value-added Web services. The formal model (i.e., the Causal link matrices) aims at storing all those connections (i.e., causal links) in order to find the best Web service composition. The CLM pre-computes all semantic links between Web services as an Output-Input matching because a Web service composition is mainly made up of semantic connections. Indeed a solution of a Web service composition have to design and define a plan of Web services wherein all Web services are semantically well ordered and well linked. The latter links are computed and stored in CLMs.

The idea behind the CLM is a formal model to store Web services in an adequate and semantic context for functional level composition of Web services hence a clear formalization of the Web service composition. The CLM aims at proposing a composition model for a finite set of Web services. The latter Web services are supposed to be relevant according to a discovery criteria [9, 2]. In such a case, the CLM pre-computes and defines all the possible semantic matching functions between Web services to improve the performance of Web service composition, and also to make Web service composition easier. Moreover CLMs allow us to consider a simpler composition problem i.e., the causal link composition. Thus the Web service composition is mapped to a causal link composition wherein causal links inform about semantic connections between Web service. A composition solution is mainly oriented by the CLM of the domain.

#### 3.2 Semantic web context

Parameters (i.e., input and output) of Web services are concepts referred to in an ontology  $\mathcal{T}$  (e.g., OWL-S profile [10], WSMO capability [11]). Finding a semantic similarity between two parameters  $Out_{s_y}$  and  $In_{s_x}$  is similar to find a mapping [12] between two knowledge representations encoded using the same ontology  $\mathcal{T}$ . Causal links store this semantic similarity between parameters of Web services. Indeed a causal link describes a semantic relation between an output parameter  $Out_{s_y} \in \mathcal{T}$  of a Web service  $s_y$  and an input parameter  $In_{s_x} \in \mathcal{T}$  of a Web service  $s_x$ . Thereby  $s_x$  and  $s_y$  are semantically and partially linked according to a matchmaking function  $Sim_{\mathcal{T}}(Out_{s_y}, In_{s_x})$  with  $\mathcal{T}$  a terminology (e.g., Figure 1).

Match Type	<i>Exact</i>	<i>Plug-in</i>	<i>Subsume</i>	<i>Fail</i>
$Sim_{\mathcal{T}}(Out_{s_y}, In_{s_x})$	1	$\frac{2}{3}$	$\frac{1}{3}$	0
Logic meaning	$Out_{s_y} \equiv In_{s_x}$	$Out_{s_y} \subset In_{s_x}$	$Out_{s_y} \supset In_{s_x}$	Otherwise

**Table 1.** Semantic matching functions described by  $Sim_{\mathcal{T}}$ .

Despite some methods [13–15], solving a mapping problem is hard because the syntactic form of two knowledge representations rarely matches exactly. Four kinds of semantic matching functions [13] are considered in our model to check semantic similarity between a concept  $Out_{s_y}$  and a concept  $In_{s_x}$ . The semantic similarity is valued by the  $Sim_{\mathcal{T}}$  function (Table 1) in order to estimate the semantic degree of link between parameters of Web services. In other words the semantic similarity valuation is necessary to chain Web services parameters with the most appropriate links. For example, the *Plug-in match* means that an output parameter of a service  $s_y$  is subsumed by an input parameter of the succeeding service  $s_x$  whereas the *Subsume match* means that an output parameter of a service  $s_y$  subsumes an input parameter of the succeeding service  $s_x$ . Besides these four semantic matching functions (Table 1), non-empty intersection [14], concept abduction or contraction [15] might be proposed in order to add expressivity of the  $Sim_{\mathcal{T}}$  function.

Suppose two Web services  $s_y$  and  $s_z$  with a respective output parameter  $Out_{s_y}$  and  $Out_{s_z}$ . In case one finds a Web service  $s_x$  such that  $Out_{s_y}$  and  $Out_{s_z}$  semantically match with  $In_{s_x}$ , a semantic similarity function is necessary to value Web services connections. Thus the latter function aims at ordering the different kinds of matching ( $Out_{s_y}, In_{s_x}$ ). The similarity function described as  $Sim_{\mathcal{T}}(Out_{s_y}, In_{s_x})$  is clearly analogous to  $degreeOfMatch(Out_{s_y}, In_{s_x})$  function [13].

### 3.3 Web service composition formalism

Web service composition is close to function composition in the mathematical area. A trivial Web service composition of two Web services  $s_y$  and  $s_x$  is considered as a mathematical composition  $s_x \circ s_y$ . The latter composition means that  $s_y$  precedes  $s_x$  and there exists a positive value of  $Sim_{\mathcal{T}}$  between all input parameters of  $s_x$  and some output parameters of  $s_y$ . CLMs (i.e., matrices of semantic connections) are introduced with the aim of finding not only trivial but also more complex composition.

### 3.4 Causal link matrices

The CLMs contribute to the automated process of Web service composition by classifying Web services according to a formal link called “causal link”. A causal link is related to a logical dependency among input and output parameters of different Web services.

A causal link<sup>3</sup> [16, 17] is refined as a triple  $\langle s_y, Sim_{\mathcal{T}}(Out_{s_y}, In_{s_x}), s_x \rangle$  such that  $s_x$  and  $s_y$  refer to two Web services in a set of available Web services  $S_{Ws}$ . The concept  $Out_{s_y}$  is an output parameter of the service  $s_y$  whereas the concept

<sup>3</sup> In AI planning area, some authors call causal link **protection intervals** [16].

$In_{s_x}$  is an input parameter of the service  $s_x$ . The function  $Sim_{\mathcal{T}}$  is the function of semantic similarity described in Table 1.  $Sim_{\mathcal{T}}$  returns a value in  $[0, 1]$  depending on the matching degree between the concepts  $Out_{s_y}, In_{s_x} \in \mathcal{T}$ . A causal link  $\langle s_y, Sim_{\mathcal{T}}(Out_{s_y}, In_{s_x}), s_x \rangle$  requires that i)  $s_y$  precedes  $s_x$ , ii) no Web service is interleaved between  $s_x$  and  $s_y$ .

**Definition 1** (*Valid Causal link*)

A causal link  $\langle s_y, Sim_{\mathcal{T}}(Out_{s_y}, In_{s_x}), s_x \rangle$  is valid iff  $Sim_{\mathcal{T}}(Out_{s_y}, In_{s_x}) > 0$ .

**Example 1** (*Valid Causal link illustration*)

According to the motivating example,  $\langle S_d, Sim_{\mathcal{T}}(EmergencyDpt, Organization), S_b \rangle$  is a valid causal links whereas  $\langle S_b, Sim_{\mathcal{T}}(Person, Organization), S_b \rangle$  is not.

A causal link matrix contains all enabled, legal and valid transitions for a composition goal because causal links help to detect inconsistencies (Fail case in Table 1) of semantic link between Web services. Indeed all valid causal links between Web services are explicitly represented with a value pre-computed by the  $Sim_{\mathcal{T}}$  function. The latter value is based on the semantic quality of valid causal links. The Causal link matrix aims at storing all those valid causal links in an appropriate way. The more valid causal links there are, the better it is for a functional composition problem.

**Definition 2** (*Causal link matrix CLM*)

The set of  $p \times q$  CLMs<sup>4</sup> is defined as  $M_{p,q}(\mathcal{P}((S_{W_s} \cup \mathcal{T}) \times (0, 1]))$ . Columns  $c_j, j \in \{1, \dots, q\}$  are labelled by  $(Input(S_{W_s}) \cup \beta) \subseteq \mathcal{T}$ , the inputs parameters of services in  $S_{W_s}$  and/or the concepts described by the goal set  $\beta \subseteq \mathcal{T}$ . Rows  $r_i, i \in \{1, \dots, p\}$  are labelled by  $Input(S_{W_s})$ , the inputs parameters of services in  $S_{W_s}$ . Each entry  $m_{i,j}$  of a CLM  $\mathcal{M}$  is defined as a set of pairs  $(s_y, score) \in (S_{W_s} \cup \mathcal{T}) \times (0, 1]$  such that

$$(s_y, score) = \begin{cases} (s_y, Sim_{\mathcal{T}}(Out_{s_y}, c_j)) & \text{if } s_y \in S_{W_s}, Out_{s_y} \in Out(s_y) \\ (s_y, 1) & \text{if } s_y \in \mathcal{T} \end{cases} \quad (1)$$

with  $r_i \in \mathcal{T} \cap In(s_y) \subseteq Input(S_{W_s})$  is the label of the  $i^{th}$  row.

with  $c_j \in \mathcal{T} \cap (Input(S_{W_s}) \cup \beta)$  is the label of the  $j^{th}$  column.

$Out(s_y)$  is the set of output parameters of the Web services  $s_y$  whereas  $In(s_y)$  is its set of input parameters.  $\beta$  contains the set of goals, described as concepts in a terminology  $\mathcal{T}$ . Those concepts have to be reached. The variable *score* refers to the degree of match  $Sim_{\mathcal{T}}(Out_{s_y}, c_j)$  between an output parameter  $Out_{s_y} \in \mathcal{T}$  of  $s_y$  and  $c_j \in Input(S_{W_s}) \cup \beta$  in case  $s_y \in S_{W_s}$ . In the alternative case  $s_y \in \mathcal{T}$ , the value *score* is 1. A CLM pre-computes the semantic similarities between all output and input parameters of a closed set of Web services. All entries defined in  $\mathcal{P}((S_{W_s} \cup \mathcal{T}) \times (0, 1])$  are valid causal links. Indeed  $Sim_{\mathcal{T}}$  is restricted on  $(0, 1]$  according to definition 2.

A CLM is seen as a matrix with entries in  $\mathcal{P}((S_{W_s} \cup \mathcal{T}) \times (0, 1])$ . Thus each entry of a CLM refers to a set of pairs  $(s_y, score)$  such that the score refers to a semantic similarity between an output parameter of a Web service  $s_y$  and an input parameter

<sup>4</sup>  $\mathcal{P}(S)$  refers to power set of  $S$  whereas  $\#S$  refers to the Cardinality of  $S$ .

i/j index	1	2	3	4	5	6
$r_i.label$	Address (Add)	BloodPressure (BP)	Org	Patient (PID)	Warning Level (WL)	
$c_j.label$	Address (Add)	BloodPressure (BP)	Org	Patient (PID)	Warning Level (WL)	Person

**Table 2.** Labels of the rows  $r_i$  and columns  $c_j$  of the  $5 \times 6$  matrix  $\mathcal{M}$ .

Web Services	$S_a$	$S_b$	$S_c$	$S_d$	$S_e$
Input	PID ( $r_4, c_4$ ), Add ( $r_1, c_1$ )	Org ( $r_3, c_3$ )	BP ( $r_2, c_2$ )	WL ( $r_5, c_5$ )	WL ( $r_5, c_5$ )
Output	BP ( $r_2, c_2$ )	Person ( $c_6$ )	WL ( $r_5, c_5$ )	Emerg. Dpt	Org ( $r_3, c_3$ )

**Table 3.** Semantic Web services of  $S_{W_s}$  and their capabilities.

of another Web service in  $S_{W_s}$ . All semantic connections (i.e., Causal links) are pre-computed in such a matrix to make Web service composition easier.

According to definition 2, Causal link matrices are defined with  $p$  rows and  $q$  columns, with  $\#(\beta)$  is the cardinality of goals:

$$p = \#(Input(S_{W_s})) \quad (2)$$

$$q = p + \#(\beta) - \#(\beta \cap Input(S_{W_s})) \quad (3)$$

The variables  $p$  and  $q$  refer, respectively, to the cardinality of input parameters of all Web services in  $S_{W_s}$  and the cardinality of input parameters of all Web services in  $S_{W_s}$  and  $\beta$ . In compliance with [18], dimension of a causal link matrix in  $M_{p,q}(\mathcal{P}(S_{W_s} \times (0, 1]))$  is defined by  $dim_{\mathcal{P}(S_{W_s} \times (0, 1])} M_{p,q}(\mathcal{P}(S_{W_s} \times (0, 1])) = p \times q$ . In the general case, CLMs are not square matrices since  $q > p$ .

**Example 2** (Illustration of Causal link matrix indexes and labels.)

Let  $\{S_a, S_b, S_c, S_d, S_e\}$  be the set of Web services  $S_{W_s}$  (section 2) and  $\{Person\}$  be the goal  $\beta$ .  $p$  and  $q$  are respectively equal to 5 and 6 (Tables 2, 3) according to equalities (2), (3) and Definition 2. Thus rows, columns of the CLM  $\mathcal{M}$  are respectively indexed by  $\{1, \dots, 5\}$ ,  $\{1, \dots, 6\}$  and labelled by concepts  $r_{i,i \in \{1, \dots, 5\}}$ ,  $c_{j,j \in \{1, \dots, 6\}}$  of  $\mathcal{T}$ .  $\mathcal{M}$  refers to a causal link matrix with entries in  $\mathcal{P}((S_{W_s} \cup \mathcal{T}) \times \{\frac{1}{3}, \frac{2}{3}, 1\})$ .

The causal link matrices construction is function of the cardinality of output and input parameters of Web services in  $S_{W_s}$ . Suppose  $\#(Output(S_{W_s}))$  and  $\#(Input(S_{W_s}))$  be respectively the cardinality of output parameters of Web services in  $S_{W_s}$  and the cardinality of input parameters of Web services in  $S_{W_s}$ . The algorithmic complexity for the causal link matrix construction is  $\theta(\#(Input(S_{W_s})) \times \#(Output(S_{W_s}))$  or  $\theta((Max\{\#(Input(S_{W_s})), \#(Output(S_{W_s}))\})^2)$  so square in the worst case [19]. In other words, the CLMs construction consists of finding a semantic similarity *score* between the output parameters of all Web services  $s_y \in S_{W_s}$  and the input parameters of another Web service in  $S_{W_s}$ . In case *score* is not null, the pair  $(s_y, score)$  is added in the CLM according to the Definition 2. For further details, [19] studies the whole process of the CLM construction.

**Example 3** (Causal link matrix illustration with Tables 2, 3)

The entry  $m_{5,3}$  (i.e.,  $m_{WarningLevel, Organization}$ ) is equal to  $\{(S_d, \frac{2}{3}), (S_e, 1)\}$ . Indeed a Web service  $S_d$  with one input parameter *WarningLevel* and an output *Emer-*

gencyDpt semantically similar to Organization exists in  $S_{Ws}$ .  $\langle S_d, Sim_{\mathcal{T}}(EmergencyDpt, Organization), S_b \rangle$  is a valid causal link. The EmergencyDpt and Organization concepts match with the Plug-in match according to the definition of  $Sim_{\mathcal{T}}$ . According to examples 1 and 2, the causal link matrix  $\mathcal{M}$  follows:

$$\mathcal{M} = \begin{pmatrix} \emptyset & \{(S_a,1)\} & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \{(S_c,1)\} & \emptyset \\ \emptyset & \emptyset & \emptyset & \{(S_b, \frac{1}{3})\} & \emptyset & \{(S_b,1)\} \\ \emptyset & \{(S_a,1)\} & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \{(S_d, \frac{2}{3}), (S_e,1)\} & \emptyset & \emptyset & \emptyset \end{pmatrix}$$

Given a set of instantiated concepts in  $\mathcal{KB}$ , definition 3 initialises a CLM  $\mathcal{M}$  and the property 1 follows.

**Definition 3** (Causal link matrix initialisation)

Let  $\mathcal{M}$  be a CLM in  $M_{p,q}(\mathcal{P}((S_{Ws} \cup \mathcal{T}) \times (0, 1]))$  and  $\mathcal{KB}$  be the set of instantiated concepts  $\{C_1, \dots, C_t\}$  such that  $\mathcal{KB} \subseteq Input(S_{Ws}) \cap \mathcal{T}$ .  $\mathcal{M}$  is initialised with  $\mathcal{KB}$  iff

$$m_{i,k} \supseteq (C_k, 1), \forall i \in \{1, \dots, p\}, \forall k \in \{1, \dots, t\}$$

**Example 4** (Illustration of a causal link matrix initialisation)

Let  $\{Address, PatientID\}$  be the knowledge base  $\mathcal{KB}$  and  $\mathcal{M}$  be the CLM (example 3). According to the definition 2, Tables 2 and 3, the initialised CLM is:

$$\mathcal{M}_0 = \begin{pmatrix} \{(Add,1)\} & \{(S_a,1)\} & \emptyset & \{(PID,1)\} & \emptyset & \emptyset \\ \{(Add,1)\} & \emptyset & \emptyset & \{(PID,1)\} & \{(S_c,1)\} & \emptyset \\ \{(Add,1)\} & \emptyset & \emptyset & \{(PID,1), \{(S_b, \frac{1}{3})\}\} & \emptyset & \{(S_b,1)\} \\ \{(Add,1)\} & \{(S_a,1)\} & \emptyset & \{(PID,1)\} & \emptyset & \emptyset \\ \{(Add,1)\} & \emptyset & \{(S_d, \frac{2}{3}), (S_e,1)\} & \{(PID,1)\} & \emptyset & \emptyset \end{pmatrix}$$

**Property 1** An entry  $m_{i,j}$  from a causal link matrix  $\mathcal{M} \in M_{p,q}(\mathcal{P}((S_{Ws} \cup \mathcal{T}) \times (0, 1]))$  is different from the empty set if and only if one of the following conditions is satisfied:

- i)  $\exists s_y \in S_{Ws}$  with at least one input  $r_{i.label} \in \mathcal{T}$  and one output  $Out_{s_y} \in Out(s_y) \cap \mathcal{T}$  such that  $Sim_{\mathcal{T}}(Out_{s_y}, c_{j.label}) \neq 0$  (definition 2);
- ii)  $c_{j.label}$  is a concept in  $\mathcal{KB}$  (definition 3).

Once all Web services in  $S_{Ws}$  are semantically chained according to the causal link criteria, the Web service composition problem is mapped to an AI planning problem.

### 3.5 Causal link matrix issues

The key contribution of the Causal link matrix is a formal and semantic model to control a set of Web services which are relevant for a Web service composition. Web services of  $S_{Ws}$  are supposed to be relevantly discovered in a discovery process [2, 9]. Thus the set of Web services  $S_{Ws}$  is closed in order to limit the dimension of the Causal link matrix. This model allows performance analysis of proposed plans with a concrete view of the composition background: causal links and their semantic dependency. The Causal link matrix aims at pre-chaining Web services according to a semantic similarity based on

their Output/Input specification. Thus the CLM describes all possible interactions between all the known Web services in  $S_{Ws}$  as semantic connections. Moreover the CLM model in an interesting trade-off to support processes such as Web service verification (valid causal link) or repairing by insertion and deletion of Web services. The Causal link matrix is able to prepare a suitable context for an AI planning problem [7, 20] with the purpose of obtaining complete, correct, consistent and optimal plan.

A set of ontologies  $\mathcal{T}$ , a set of Web services  $S_{Ws}$ , a goal  $\beta$ , a knowledge base  $\mathcal{KB}$  and a semantic similarity function  $Sim_{\mathcal{T}}$  are required in order to satisfy such a challenging solution. With a terminology  $\mathcal{T}$ , we deal with conceptual analysis (inference problems) and knowledge representation. A set of Web services refers to a set of actions for a planning problem.  $\beta$  informs about plan directions (as searching concepts). A knowledge base  $\mathcal{KB}$  informs about initial conditions (instantiated concepts). Finally the similarity function  $Sim_{\mathcal{T}}$  semantically compares two parameters as concepts in  $\mathcal{T}$ .

#### 4 AI planning and Causal link matrices

The planning problem is formalized as a triple  $\Pi = \langle S_{Ws}, \mathcal{KB}, \beta \rangle$ .  $S_{Ws}$  refers to a set of possible state transitions,  $\mathcal{KB}$  is an *Initial state* and  $\beta \subseteq \mathcal{T}$  is an explicit goal representation. The Web service composition method consists of finding a plan that produces the desired outputs  $\beta$  according to a knowledge base  $\mathcal{KB}$ . The causal link score allows the early detection of impossible, feasible and best links between Web services (Definitions 1 and 2). That is why our method is based on the causal link validity between Web service. The CLM of a specified domain allows to detect all Web service composition with semantic connections. Composition as sequences of Web service is a necessary requirement to propose a solution plan. Such a composition is defined by the sequence-composability. The latter composability defines a composition  $s_x \circ s_y$  if an output of  $s_y$  is consumed by an input of another Web service  $s_x$ . The sequence-composability knowledge is expressed in CLMs according to the Theorem 1.

**Theorem 1** *Let  $\mathcal{M}$  be a CLM, and  $s_x, s_y$  be two Web services in  $S_{Ws}$ .  $s_x$  and  $s_y$  are sequence-composable iff*

- $\exists i \in \{1, \dots, p\}, \exists j \in \{1, \dots, q\}, \exists v \in (0, 1]$  such that  $(s_y, v) \subseteq m_{i,j} \cdot c_{j.label}$  and  $r_{i.label}$  are respectively inputs of  $s_x$  ( $In(s_x)$ ) and  $s_y$  ( $In(s_y)$ ).

*Proof.* Consider the proof of theorem 1 as the following two implications.

( $\Rightarrow$ ) Let  $s_x, s_y$  be two Web services in  $S_{Ws}$  and  $\mathcal{M}$  be a CLM with entries in  $\mathcal{P}((S_{Ws} \cup \mathcal{T}) \times (0, 1])$ . Moreover, we consider the *Sequence-composability* of  $s_x$  and  $s_y$  such that an output of the Web service  $s_y$  is consumed by the input of another Web service  $s_x$  i.e.,  $s_x \circ s_y$ . According to the CLM definition, input parameters of  $s_x$  are labelled in  $\mathcal{M}$  as concepts in  $\mathcal{T}$ . Thus we may suppose  $\{1, \dots, p_{s_x}\}$  as the index of the  $s_x$  input parameters in  $\mathcal{M}$  without loss of generalities. According to the *Sequence-composability* definition,  $\exists j \in \{1, \dots, q_{s_y}\}$  such that  $Sim_{\mathcal{T}}(Out_{s_y}, c_{j.label}) > 0$  since an output  $Out_{s_y} \in Out(s_y)$  of one Web service  $s_y$  is consumed by an input  $c_{j.label}$  of another web service  $s_x$ . Consequently  $\langle s_y, Sim_{\mathcal{T}}(Out_{s_y}, c_{j.label}), s_x \rangle$  is a valid causal link. According to the

property 1.i), an entry  $m_{i,j}$  from  $\mathcal{M}$  is different from the empty set. Finally  $\exists i \in \{1, \dots, p_{s_x}\} \subseteq \{1, \dots, p\}, \exists j \in \{1, \dots, q_{s_x}\} \subseteq \{1, \dots, q\}$  such that  $(s_y, Sim_{\mathcal{T}}(Out_{-s_y}, c_{j.label})) \subseteq m_{i,j}$  with  $c_{j.label} \in In(s_x)$  and  $r_{i.label} \in In(s_y)$ .

( $\Leftarrow$ ) Suppose  $\exists i \in \{1, \dots, p\}, \exists j \in \{1, \dots, q\}, \exists score \in (0, 1]$  such that  $(s_y, score) \subseteq m_{i,j}$  with  $c_{j.label} \in In(s_x) \subseteq \mathcal{T}$  and  $r_{i.label} \in In(s_y) \subseteq \mathcal{T}$ . According to definition 2 and property 1.i), an entry  $m_{i,j}$  from  $\mathcal{M}$  is different from the empty set. Thus  $\exists s_y \in S_{W_s}$  with at least one input  $r_{i.label} \in \mathcal{T}$  and one output  $Out_{-s_y} \in \mathcal{T}$  such that  $Sim_{\mathcal{T}}(Out_{-s_y}, c_{j.label}) \neq 0$ . Since  $c_{j.label} \in In(s_x)$ , two Web services  $s_x$  and  $s_y$  in  $S_{W_s}$  exist such that an output of the Web service  $s_y$  is consumed by an input of another Web service  $s_x$ . Thus  $s_x$  and  $s_y$  are sequence-composable.

**Remark 1** *In case of more complex composition, more than one Web service needs to be chained with  $s_x$  in order to produce input parameters of  $s_x$  (in case of a regression-based search). So parallel constructs may be applied. The latter constructs is conceivable in case the entry cardinality (in the CLM) is greater than 1.*

**Example 5** *Suppose the CLM  $\mathcal{M}$  in section 3.  $S_c$  and  $S_d$  are sequence-composable in  $S_{W_s}$  if and only if  $S_d \circ S_c$  (Theorem 1). Indeed there exists  $(i, j) = (2, 5)$  in  $\mathcal{M}$  such that  $(r_{i.label}, c_{j.label}) = (BloodPressure, WarningLevel)$ .  $(S_c, 1) \subseteq m_{i,j}$  with  $c_{j.label} \in In(S_d) \subseteq \mathcal{T}$  and  $r_{i.label} \in In(S_c) \subseteq \mathcal{T}$ . Therefore the output  $S_c$  is consumed by the input of  $S_d$  because  $Sim_{\mathcal{T}}(Out_{-S_c}, In_{-S_d}) \neq 0$  (Table 1).*

#### 4.1 AI planning context and regression-based approach

A simpler form of AI planning is introduced to avoid problems [21] from planning-based Web services composition, e.g., non determinism and implicit goal. The set of Web services  $S_{W_s}$  (i.e., Actions) is closed by assumption and the *goal* set  $\beta$  refers to a set of concepts in a terminology  $\mathcal{T}$ . Thus we propose a solution plan in a well-defined domain: goals are explicitly given, initial state is well defined and Web services are strictly defined at functional level. So non determinism, implicit goal, fuzzy Web service description and behaviour are out of the question. Therefore it does seem possible to directly apply current AI planning methods to our specific problem.

The composition process consists of a recursive and regression-based approach. The main idea is to propose a controlled and adequate matrix parsing. Thus each causal link takes place in the solution plan as a semantic link between Web services. According to the *Ra4C* algorithm, a goal  $\beta$  needs to be solved. In case the previously goal is fulfilled by the initial condition i.e., the knowledge base  $\mathcal{KB}$ , the process of consistent plans discovery is stopped. Otherwise a Web service  $s_x$  with a goal  $\beta$  as an output parameter should be discovered in  $S_{W_s}$ . This discovery process is eased by the CLM of the domain. In case of a discovery success, the process is iterated with the  $s_x$  input parameters as new goals. Alternatively, the process is stopped and the (or a part of the) plan is reduced to  $\emptyset$ . All the process is recursive until all goals and new goals are concepts in  $\mathcal{KB}$  (stop condition). The algorithm 1 presents the complete process of composition and returns a disjunction of consistent plans consisted of valid and “*sequence-composable*” causal links. CLMs ease the regression-based search because

all Web services are semantically well ordered in a robust and formal model. The solutions are plans wherein Web services are semantically chained by causal links. The complexity of the algorithm 1 depends on the filling rate of the CLM. The more the CLM is sparse the faster the  $Ra_4C$  algorithm is. Instead a regression-based approach, other problem-solving techniques - called heuristic reasoning - may be applied [22].

Plan constructs are necessary to describe a partial ordering [23] of Web services in  $\Pi$ , hence  $\wedge$  the conjunction operator (parallel construct),  $\vee$  the disjunction operator (non determinism construct),  $\circ$  the sequence construct, and  $\wedge > \vee > \circ$  their priority order. The operator  $\circ$  defines the sequence-composability between two Web services.

---

**Algorithm 1:** Composition by regression-based approach  $Ra_4C$ .

---

**Input:** A CLM  $\mathcal{M}$  ( $[m_{i,j}]$ ), a (or disjunction of) plan(s)  $\pi$ , a planning problem  $\langle S_{Ws}, \mathcal{KB}, \beta \rangle$ , a set of solved goals  $G$ , a set of non valid goals  $\beta_{nv}$ .

**Result:** A disjunction of consistent plans  $\pi$ .

```

begin
   $S_c \leftarrow \emptyset$ ; // Temporary set of pairs in  $(S_{Ws} \cup \mathcal{T}) \times (0, 1]$ .
  // Stop condition of the  $Ra_4C$  algorithm.
  if  $((\exists C_k \in \mathcal{KB}) \ \& \ (Sim_{\mathcal{T}}(C_k, \beta) \neq 0))$  then  $\pi \leftarrow \beta$ ;
  // Web services discovery with  $\beta$  output.
  foreach  $I_i \in Input(S_{Ws})$  do
     $\lfloor$  if  $\exists (s_y, v) \in m_{I_i, \beta}$  then  $Add((s_y, v), S_c)$ ;
  // Plan for Web service composition.
  if  $S_c \neq \emptyset$  then
    foreach pair  $(s_y, v) \in S_c$  such that  $s_y \in S_{Ws}$  do
       $\pi \leftarrow \pi \vee s_y$ ;
      foreach  $In_{-s_y} \in In(s_y)$  do
        if  $\beta \in G$  then  $\pi \leftarrow \pi \wedge \emptyset$ ;  $Add(G, \beta_{nv})$ ; // inconsistent plan
        else
           $Add(\beta, G)$ ;  $\Pi \leftarrow \langle S_{Ws}, \mathcal{KB}, In_{-s_y} \rangle$ ;
           $\pi \leftarrow \pi \vee (\bigwedge_{In(s_y)} Ra_4C(\mathcal{M}, \pi, \Pi, G))$ ;
        else  $\pi \leftarrow \pi \wedge \emptyset$ ; // inconsistent plan
      return  $\pi$ ;
  end

```

---

$s_x \circ s_y$  if  $\exists Out_{-s_y}, In_{-s_x} \in \mathcal{T} \mid \langle s_y, Sim_{\mathcal{T}}(Out_{-s_y}, In_{-s_x}), s_x \rangle$  is a valid causal link. The conjunction operator is used to express parallel plans. Such a situation is possible if a Web service contains more than one input parameter (e.g.,  $m_{1,2}, m_{4,2}$ ). The latter parameters consider new parallel goals in the  $Ra_4C$  algorithm. The disjunction operator is used if more than one output parameter is consumed by the goal (e.g.,  $m_{5,3}$ ).

## 4.2 Consistency, completeness and correctness properties of solutions

Consistency is a necessary condition for a solution plan. Such a condition is satisfied by plans which contain no cycle in the ordering constraints and no causal link conflicts [17]. The  $Ra_4C$  algorithm builds such a plan and avoid cycles and conflicts to dispose

of inconsistent causal links. The latter inconsistency is tackled by the Algorithm 1 with an update of solved goals. Thus the  $Ra_4C$  algorithm do not solve goals already solved. The correctness proof of algorithm 1 is detailed in [24].

**Example 6** (*Set of consistent plans*)

Let  $\mathcal{M}_0$  be the CLM (section 3) and  $\Pi = \langle \{S_a, S_b, S_c, S_d, S_e\}, \{Add, PatientID\}, \{Person\} \rangle$  be the planning-oriented Web service composition problem. We are looking for a “Person” with skills to understand hypertension troubles. The result is a disjunction of four consistent plans:  $\pi_{a1} = S_b \circ ((S_d \circ S_c \circ S_a(Add \wedge PID)))$ ,  $\pi_{b1} = S_b \circ ((S_e \circ S_c \circ S_a(Add \wedge PID)))$ ,  $\pi_{a2} = S_b \circ ((S_d \circ S_c \circ S_a(Add \wedge (S_b \circ \emptyset))))$ ,  $\pi_{b2} = S_b \circ ((S_e \circ S_c \circ S_a(Add \wedge (S_b \circ \emptyset))))$ .

Plans suggested by Algorithm 1 do not necessarily satisfy the correctness and completeness properties of plan. Regarding a complete plan [17] as a plan where every input of every Web service is achieved by some other previous Web service, a complete plan is a partial order of well-ordered causal links. By definition, a CLM contains all necessary information about complete plans because a CLM explicitly stores all valid causal links between Web services. Non-complete plans contain empty plan  $\emptyset$  (Algorithm 1) hence open goals. Plans with open goals (e.g.,  $\pi_{a3}, \pi_{b3}$ ) are removed from the solutions set since those goals can not be satisfied by  $\mathcal{KB}$  or the  $S_{Ws}$  Web services.

The plans refinement follows a backward chaining strategy from a goal to initial states. In other words the goal  $\beta \in \mathcal{T}$  is recursively produced from a (or some) valid causal link(s)  $\langle s_y, Sim_{\mathcal{T}}(Out_{s_y}, \beta), s_x \rangle$ . So correctness of the solution plans is guaranteed by the causal link between the input and output parameters of Web services.

Therefore the algorithm 1 returns a set of correct, complete and consistent plans. However such a set may contain a large number of plans. So pruning strategies for plan-space is necessary to propose a solution. A “causal link”-based optimization criteria is proposed to detect the optimal plan, hence the computation of best causal links in a regression process. The process is recursively executed until the plan is a solution or until the inputs  $In(s_y) \subseteq \mathcal{T}$  of the service  $s_y$  are concepts in  $\mathcal{KB}$ . The weight of the optimal plan is computed by means of the CLM and algorithm 1 previously introduced:

$$W_{Max}(\beta) = Max_{S_c} \left\{ \frac{1}{\#In(s_y)^2} \sum_{In(s_y)} m_{I_i, \beta}.score \times \left( \prod_{In(s_y)} (W_{Max}(I_i)) \right) \right\} \quad (4)$$

The recursive function  $W_{Max}$  returns the weight of the best plan depending on the goal  $\beta$ . (4) is based on the weight of valid causal links of suggested plans.  $S_c$  is a set of couple  $(s_y, v)$  such that  $s_y$  is a Web service with an output  $\beta$  and input  $I_i$ . In other words,  $\langle s_y, Sim_{\mathcal{T}}(Out_{s_y}, \beta), s_x \rangle$  is a valid causal link. The  $In(s_y)$  set is the inputs set of  $s_y \in S_{Ws}$  whereas  $I_i$  is an input of  $s_y$ .  $\mathcal{M}$  is a CLM with coefficients in  $\mathcal{P}((S_{Ws} \cup \mathcal{T}) \times (0, 1])$ .  $m_{I_i, \beta}.score$  is the second component of a couple  $(s_y, v) \subseteq m_{I_i, \beta}$ .  $Max_S$  is a  $n$ -arity function which returns the maximum value between  $n$  float value(s). Given a CLM, the combination of algorithm 1 and (4) is an interesting trade-off to find an optimal, consistent, correct and complete plan when one exists.

**Example 7** According to example 6,  $\pi$  is divided into a disjunction of four consistent plans  $\{\pi_{ai}, \pi_{bi}\}_{1 \leq i \leq 2}$ . The plans  $\pi_{a2}$  and  $\pi_{b2}$  are not complete. The weights of different

plans have been computed with formula (4).  $Weight(\pi_{a1}) = 1 \times \frac{2}{3} \times 1 \times \frac{1}{2^2} \times (1+1) = \frac{1}{3}$  whereas  $Weight(\pi_{b1}) = 1 \times 1 \times 1 \times \frac{1}{2^2} \times (1+1) = \frac{1}{2}$ . Thus  $\pi_{b1}$  is the optimal plan.

### 4.3 The flexibility and scalability of the model

The flexibility of Web service composition models is a fundamental criteria for a relevant evaluation. In particular such models should be as robust as possible in order to evolve in a volatile environment such as the Web services area. The formal model for semantic Web service composition introduced in this paper takes into account this flexibility criteria. Indeed the alteration and modification (e.g., addition, deletion, and the update of Web services in the set  $S_{W_s}$ ) is the scope of CLMs. Each new update of  $S_{W_s}$  is supported by a Causal link matrix revision since the CLM is responsible for storing Web service in a semantic way through the causal relationship. Thus incremental systems wherein new Web services are progressively added, are supported by the previous model of composition and especially the CLMs. For instance the integration of a new Web service is related to the insertion of new labelled rows and columns in the worst case. In the alternative case the integration of a Web service  $s_y$  means a simple insertion of  $s_y$  in the relevant entry(ies) of the specific CLM. The flexibility of the model allows us to apply a dynamic process of Web service discovery. The only constraint is a simple update of the CLM before applying the  $Ra_4C$  algorithm.

Scalability [25] of Web service composition and discovery models is still an open issue. However the formal model scales well in France Telecom scenarios (about twenty Web services) such as the e-healthcare (six Web services and a  $5 \times 6$  CLM) scenario.

## 5 Related work

Two different approaches [26, 27] propose matrices to represent the Web services domain. [26] solve an AI planning problem where actions are viewed as tasks. Actions are formally described with Preconditions and Effects. These tasks are executed by concrete Web services, according to a service/task (row/column) matrix. [27] proposes a simple method to store Web service according to an input/output (row/column) matrix. The Matrix model used in [26, 27] does not propose reasoning about those matrices. In fact, such the matrices are simply considered as representation models. Moreover no semantic feature is introduced in their models.

From HTNs [28] to regression planning based on extensions of PDDL [29], different planning approaches have been proposed for the composition of Web services. However there is still the issue of how to deal with non determinism in these frameworks. Some composition planners [20] propose output/input mapping with type characteristics of these parameters and initial, final state predicates to generate compositions (i.e., plans). However, services need to be composed using a specification technique that characterizes ongoing behaviour of the service in order to ensure a sound composition. Situation calculus is proposed in [7] to represent Web service and Petri nets for describing the execution behaviours of Web services. A planner is declared as a state chart in [30], and the resulting composite services are executed by replacing the roles in the chart by selected individual services. With the aim of generating a composite service plan out of

existing services, [31] propose a composition path, which is a sequence of operators that compute data, and connectors that provide data transport between operators. The search for possible operators to construct a sequence is based on the shortest path algorithm on the graph of operator space. However, they only considered two kinds of services operator and connector with one input and one output parameter (i.e., the simplest case for a service composition). [32] propose a forward chaining approach to solve a planning problem. Their composition process terminates when a set of Web services that matches all expected output parameters given the inputs provided by a user is found.

## 6 Conclusion and future work

Despite the fact that Web service composition is in its infancy some proposals are being studied, but no theoretical model has been proposed to help automation of composition at the best stage of our knowledge. Nevertheless many work directions may need such clearer formalizations, for instance for verification purposes. In this paper we outlined the main challenge faced in semantic Web services. Indeed we showed how the CLM tackles this challenge by providing a necessary formal model which draws a concrete context for automatic Web service composition. This concrete context captures semantic connections between Web services. The composition model has its roots in AI planning domain and takes advantage of causal link expressivity by extending its definition in a semantic context. Semantically weighted by the  $Sim_{\mathcal{T}}$  function, the latter link refers to a local optimization criteria in order to find solution plans. Moreover solution plans have properties of completeness, correctness, consistency and optimality. The model of functional level composition is easily applied to Web services which are described according to OWL-S (service profile) or WSMO (capability model) specification. Finally, contrary to [26, 27], our matrix model pre-computes the semantic similarities between Web services (individual inputs and outputs) according to causal links. Web service composition is viewed as causal link composition.

For further studies new optimization algorithms and scalability of the model need to be studied. Finally a process level composition needs to be associated to our functional level composition to guarantee a full correctness of the composition process.

## References

1. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures and Applications. Springer-Verlag (2004)
2. Sycara, K.P., Paolucci, M., Ankolekar, A., Srinivasan, N.: Automated discovery, interaction and composition of semantic web services. *J. Web Sem* **1**(1) (2003) 27–46
3. Paolucci, M., Sycara, K.P., Kawamura, T.: Delivering semantic web services. In: Proceedings of the international conference on WWW (Alternate Paper Tracks). (2003) 829–837
4. Sirin, E., Parsia, B., Hendler, J.A.: Filtering and selecting semantic web services with interactive composition techniques. *IEEE Intelligent Systems* **19**(4) (2004) 42–49
5. Klusch, M., Fries, B., Khalid, M., Sycara, K.: Owls-mx: Hybrid owl-s service matchmaking. In: AAI Fall Symposium Series. (2005)
6. Berardi, D., Calvanese, D., Giacomo, G.D., Lenzerini, M., Mecella, M.: Automatic composition of e-services that export their behavior. In: 1st ICSOC. (2003) 43–58 volume 2910.

7. Narayanan, S., McIlraith, S.: Simulation, verification and automated composition of web services. *Eleventh International World Wide Web Conference (2002)* 7–10
8. Pistore, M., Roberti, P., Traverso, P.: Process-level composition of executable web services: "on-the-fly" versus "once-for-all" composition. In: *ESWC. (2005)* 62–77
9. Benatallah, B., Hacid, M.S., Leger, A., Rey, C., Toumani, F.: On automating web services discovery. *VLDB J* **14**(1) (2005) 84–96
10. Ankolenkar, A., Paolucci, M., Srinivasan, N., Sycara, K.: The owl services coalition, owl-s 1.1 beta release. Technical report (2004)
11. Fensel, D., Kifer, M., de Bruijn, J., Domingue, J.: Web service modeling ontology (wsmo) submission, w3c member submission. (2005)
12. Küsters, R.: *Non-Standard Inferences in Description Logics*. Volume 2100 of *Lecture Notes in Computer Science*. Springer (2001)
13. Paolucci, M., Kawamura, T., Payne, T., Sycara, K.: Semantic matching of web services capabilities. In: *Proceedings of the First International Semantic Web Conference, LNCS 2342, Springer-Verlag (2002)* 333–347
14. Li, L., Horrocks, I.: A software framework for matchmaking based on semantic web technology. In: *Proceedings of the Twelfth International Conference on WWW. (2003)* 331–339
15. Colucci, S., Noia, T.D., Sciascio, E.D., Mongiello, M., Donini, F.M.: Concept abduction and contraction for semantic-based discovery of matches and negotiation spaces in an e-marketplace. In: *Proceedings of the 6th ICEC, ACM Press (2004)* 41–50
16. McAllester, D., Rosenblitt, D.: *Systematic nonlinear planning*, Menlo Park, CA, AAAI (1991) 634–639
17. Russell, S., Norvig, P.: *Artificial Intelligence: a modern approach*. Prentice-Hall (1995)
18. Baker, A.: *Matrix Groups: An Introduction to Lie Group Theory*. Springer undergraduate mathematics series. Springer-Verlag, London (2002)
19. Lécué, F., Léger, A.: Semantic web service composition through a matchmaking of domain. In: *4th IEEE European Conference on Web Services (ECOWS) (to appear)*. (2006)
20. Desjardins, M., Sheshagiri, M., Finin, T.: A planner for composing services described in DAML-S. In: *AAMAS Workshop on Web Services and Agent-based Engineering. (2003)*
21. Srivastava, B., Koehler, J.: Web service composition - current solutions and open problems. In: *ICAPS 2003 Workshop on Planning for Web Services. (2003)*
22. Ghallab, M., Nau, D., Traverso, P.: *Automated Planning: Theory and Practice*. Morgan Kaufmann Publishers (2004)
23. Sacerdoti, E.: The nonlinear nature of plan. In: *IJCAI-4. (1975)* 206–214
24. Lécué, F., Léger, A.: Semantic web service composition based on a closed world assumption. In: *4th IEEE European Conference on Web Services (ECOWS) (to appear)*. (2006)
25. Constantinescu, I., Faltings, B., Binder, W.: Type-based composition of information services in large scale environments. In: *The International Conference on Web Intelligence. (2004)*
26. Claro, D.B., Albers, P., Hao, J.K.: Selecting web services for optimal composition. In: *ICWS International Workshop on Semantic and Dynamic Web Processes, Orlando - USA (2005)*
27. Constantinescu, I., Faltings, B., Binder, W.: Type based service composition. In: *WWW (Alternate Track Papers & Posters). (2004)* 268–269
28. Wu, D., Parsia, B., Sirin, E., Hendler, J.A., Nau, D.S.: Automating DAML-S web services composition using SHOP2. In: *ISWC. (2003)* 195–210
29. Dermott, D.M.: *PDDL - the planning domain definition language (1997)*
30. Benatallah, B., Sheng, Q.Z., Ngu, A.H.H., Dumas, M.: Declarative composition and peer-to-peer provisioning of dynamic web services. In: *ICDE. (2002)* 297–308
31. Mao, Z.M., Katz, R.H., Brewer, E.A.: Fault-tolerant, scalable, wide-area internet service composition. Technical report (2001)
32. Zhang, R., Arpinar, I.B., Aleman-Meza, B.: Automatic composition of semantic web services. In: *ICWS. (2003)* 38–41