

Semantic Web Service Composition Based on a Closed World Assumption

Freddy Lécué^{1,2}, Alain Léger¹

¹France Telecom R&D, France
4 rue du clos courtel, F-35512 Cesson Sévigné
{(freddy.lecue, alain.leger)}@orange-ft.com}

² École Nationale Supérieure
des Mines de St-Etienne, France
158, cours Fauriel, F-42023 Saint-Étienne

Abstract

Automation of Web service composition i.e., the process of forming new value added Web services is one of the most promising challenges in Web service research area. Such an automation does not involve only a richer semantic for Web service description but also reasoning processes about those descriptions. Semantics is supposed to be one of the key elements for the automation of Web service composition. Indeed semantic Web services enable a rich machine-understandable descriptions of their capabilities and processes in order to ease automation of processes such as discovery, selection and composition. In this paper one studies the functional level description of Web services and its impact on Web service composition hence the presentation of a formal model CLM (Causal link matrix). The model supports a semantic context in order to find a correct, complete and consistent plan as a AI planning-based composition. The innovative and formal model follows a forward chaining-oriented composition. Moreover two methods of optimization (i.e., local-oriented and a global-oriented) of Web service composition are introduced to discover the best plan according to a semantic criterion i.e., the causal link between Web services.

1 Introduction

The Web service paradigm [1] provides the feature richness, flexibility and scalability needed by enterprises to manage the SOA challenges. Web services are defined as loosely coupled, reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard internet protocols. Thus Web services enable improved coordination amongst multiple computing platforms, applications, and business partners. However the proliferation of Web services over the web implies difficulties to discover specific Web services that can perform specialized tasks. Despite such difficulties the combination of existing Web services

may be an alternative and also a promising approach.

Manual Web service combination from scratch can be difficult and time consuming. That is why new functionalities are necessary to support dynamic and automated tasks such as discovery, selection and composition. In one hand a necessary ability is the functional description of Web services. Those Web services are described in terms of their IOPEs i.e., inputs, outputs, precondition, and effects at functional level. Inputs and outputs describe members of concepts in the underlying ontology \mathcal{T} . Preconditions and effects are logical formulas, that must hold prior to/will hold after execution of the service, respectively. In the other hand the process description of Web services is of prime importance to describe Web services activities and interaction protocol. These two levels of description are covered by means of semantic Web services. A semantic Web service [33] is described as a Web service whose internal and external description is in a language that has well-defined semantics.

Most of the work in semantic Web services composition has focused on these two main levels of composition: functional [24, 29, 9, 15] and process [5, 6, 22, 25] levels. Functional composition comes down to finding a sequence of atomic components that matches a given query – meaning that the sequence can be executed from the start conditions provided by the query, so that the query goal is satisfied at the end of the sequence. Process level composition creates a workflow for composite service with respect to choreography and orchestration constraints. Functional and process level composition as complementary methods is an interesting trade-off to propose solutions for Web service composition.

In this paper one introduces *CLMs* (Causal link matrix) i.e., formal model for functional level composition wherein Web services are chain according to their functional description. In such a model, a progression-based approach is exposed for Web service composition. Moreover two models of optimization for Web service composition are suggested i.e., the local optimization and local optimization oriented composition. The criterion of optimization is based

on the quality of semantic connection between Web service parameters i.e., the causal links.

The rest of the paper is organized as follows. Section 2 presents a motivating example through an e-healthcare scenario. Section 3 introduces the *CLM* 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. 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 An e-Healthcare Scenario

Let an e-healthcare scenario be the motivating example wherein telemedical collaborations of medical devices (e.g., sphygmomanometer) are involved. Medical devices have become an increasingly important health care area in relation to their impact on health and health care expenditure. In case one supposes medical devices interfaced by Web services, a main issue is to expose early interaction and composition between these devices. Indeed automation of device discovery, selection and composition are supposed to be necessary in the next few years. More specially medical device composition aims at reducing useless and time-consuming consultation, medical check up by proposing an innovative method to compose medical devices. Thus the patient follow-up may occurred at home although surgeons and physicians stay at hospital. The patient follow-up is not only reduced to a discovery, selection and composition of devices but also linked to a relevant analysis of results provided by medical devices (e.g., blood pressure).

Web Services	Input	Output
S_a	Pid (p_1),Add (p_4)	BP (q_2)
S_b	Organiz. (p_3)	Person (q_5)
S_c	BP (p_2)	WL (q_6)
S_d	WL (p_5)	Emerg. Dpt
S_e	WL (p_5)	Organiz. (q_3)

Table 1. Semantic Web services of S_{WS} .

Consider the above scenario with six different Web services: S_a returns the BP (Blood pressure) of a patient given a Pid (PatientID) and an Add (DeviceAddress); S_b and S_b' return respectively the supervisor (Person) and a physician of an Org (Organisation); S_c returns a WL (level of warning) given a Blood pressure; S_d returns the EmerDpt (Emergency department) given a level of Warning; S_e returns the Organization given a level of Warning. In this scenario Web services are described at a restricted form of functional level i.e., Inputs and Outputs according to the table 1.

A solution of the above problem consists of implementing a composite and value-added Web service that can au-

tomate the patient follow-up.

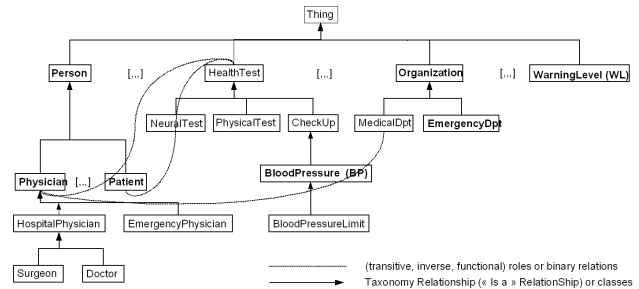


Figure 1. An e-healthcare ontology \mathcal{T} .

3 Formal Model

3.1 Motivation

Main issues in Web service composition are related to discovery of complex but feasible plans of relevant Web services. However discovery of the optimal plan according to an optimization criterion is more challenging. This criterion may depend on non-functional parameters such as QoS (Quality of Service), but also on functional parameters such that the quality of semantic connection between Web services i.e., the relation between Input and output parameters. This semantic connection between Web service parameters is considered as essential to form new value-added Web services in our model. The suggested model *CLM* aims at computing and storing all those connections as causal links with an aim of discovering the best Web service composition depending on semantic links. The *CLM* pre-computes all semantic links between Web services as an Output-Input matching. Indeed a Web service composition must be designed and defined by a plan of Web services wherein all Web services are semantically well ordered and well linked.

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 composition model evolves in a finite set of Web services wherein those Web services are supposed to be relevant according to a discovery criterion [4, 33]. 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, but also to make Web service composition easier. Moreover *CLMs* allow us to consider a simplest composition problem i.e., causal link composition. Thus the Web service composition is mapped to a causal link composition

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

Table 2. Semantic matching functions.

wherein causal links inform about semantic connections between Web service. Thus Web service composition is *CLM*-oriented.

3.2 Semantic Web Context

The functional parameters (i.e., inputs and outputs) of Web services are concepts referred to in an ontology \mathcal{T} e.g., defined by the OWL-S profile [2], or the WSMO capability [13]. A causal link describes a semantic similarity 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). The discovery of a semantic similarity between two parameters Out_{s_y} and In_{s_x} is similar to find a mapping [16] between two knowledge representations encoded using the same ontology \mathcal{T} .

Despite some methods [23, 20, 8], solving a mapping problem is hard because the syntactic form of two knowledge representations rarely matches exactly. Four kinds of semantic matching functions [23] 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 2) in order to estimate the semantic degree of link between parameters of Web services. The valuation of semantic similarity between Web service parameters is necessary to chain parameters with the most appropriate links. For example, the *Plug-in match* means that an output parameter of a Web service s_y is subsumed by an input parameter of a succeeding service s_x . Besides these four semantic matching functions (table 2), concept abduction or contraction [8] 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 between s_y and s_x , and also between s_z and s_x . The $Sim_{\mathcal{T}}$ function aims at ordering these different kinds of matching e.g., $Sim_{\mathcal{T}}(Out_{s_y}, In_{s_x})$ and $Sim_{\mathcal{T}}(Out_{s_z}, In_{s_x})$.

3.3 Web Service Composition Formalism

In the proposed model Web service composition is compared 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 of Web services $s_x \circ s_y$. The latter composition means that s_y precedes s_x i.e., there exists a positive value of $Sim_{\mathcal{T}}$ between all input parameters of s_x and output parameters of s_y . This kind of composition is called a trivial composition because such a composition implies a similarity between two domains (Input, Output) of only two Web services. However a composite Web service does not still satisfy this trivial definition of Web service composition. Indeed more than one Web service s_y may be necessary to achieve the input parameters of s_x hence $s_x \circ (s_{y1} \wedge \dots \wedge s_{yn})$.

The *CLMs* are introduced with the aim of discovering not only trivial $s_x \circ s_y$ but also these more complex cases of composition $s_x \circ (s_{y1} \wedge \dots \wedge s_{yn})$.

3.4 Causal Links and *CLM*

3.4.1 Causal Links

A causal link¹ [21, 26] 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 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 2. $Sim_{\mathcal{T}}$ 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.

3.4.2 Causal Link Matrix

The *CLMs* contribute to the automated process of Web service composition by classifying Web services according to a formal link called *causal link*.

A *CLM* contains all enabled, legal and valid transitions in a composition process because causal links help to detect

¹In AI planning area causal link referred to **protection intervals** [21].

inconsistencies (fail case in table 2) 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 CLM aims at storing all those valid causal links in an appropriate way. The more valid causal links, the better is for a functional composition problem.

Definition 2. (Causal link matrix CLM)

The set of $p \times q$ CLMs² 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

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

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

$Out(s_y)$ is the set of output parameters of the Web service s_y whereas $In(s_y)$ is its set of input parameters. β contains the set of goals, described as concepts in a terminology \mathcal{T} . Those concepts have to be reached and instantiated. 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 other words such an 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 of another Web service 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 $(0, 1]$ (definition 2).

According to definition 2, CLMs 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 β . In compliance with [3], dimension of a CLM 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, CLM are not square matrices since $q > p$.

² $\mathcal{P}(S)$ is the power set of S whereas $\#S$ refers to its Cardinality.

Example 2. (Illustration of CLM 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 β . p and q are respectively equal to 5 and 6 (tables 1, 2) 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} (table 3).

i/j index	$r_{i.label}$	$c_{j.label}$
1	Address (Add)	Address (Add)
2	BloodPressure (BP)	BloodPressure (BP)
3	Organization	Organization
4	Patient	Patient
5	Warning Level	Warning Level
6		Person

Table 3. Labels of rows r_i , columns c_j of \mathcal{M} .

Example 3. (CLM 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 parameter *EmergencyDpt* semantically similar to *Organization* exists in S_{W_s} . Indeed $\langle S_d, Sim_{\mathcal{T}}(EmergencyDpt, Organization), S_b \rangle$ is a valid causal link wherein the semantic similarity between the concepts *EmergencyDpt* and *Organization* is the Plug-in match (definition of $Sim_{\mathcal{T}}$). \mathcal{M} refers to a CLM with entries in $\mathcal{P}((S_{W_s} \cup \mathcal{T}) \times \{\frac{1}{3}, \frac{2}{3}, 1\})$. According to examples 1 and 2, the CLM \mathcal{M} follows:

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

3.4.3 CLM Construction

The CLMs construction depends on 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 CLM 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]. The CLM construction consists of discovering 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 the value *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.

3.4.4 CLM Initialisation

Given a set of instantiated concepts in \mathcal{KB} (Knowledge Base), the definition 3 initialises a CLM \mathcal{M} .

Definition 3. (CLM initialisation)

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

Let $\{\text{Address}, \text{PatientID}\}$ be the knowledge base \mathcal{KB} and \mathcal{M} be the CLM (example 3). According to the definition 2, tables 1 and 3, the initialised CLM \mathcal{M}_{init} is:

$$\left(\begin{array}{ccc|ccc} \{(Add,1)\} \{(S_a,1)\} & \emptyset & \{(Pid,1)\} & \emptyset & \emptyset & \emptyset \\ \{(Add,1)\} \emptyset & \emptyset & \{(Pid,1)\} & \{(S_c,1)\} & \emptyset & \emptyset \\ \{(Add,1)\} \emptyset & \emptyset & \{(Pid,1), (S_b, \frac{1}{3})\} & \emptyset & \{(S_b,1)\} & \emptyset \\ \{(Add,1)\} \{(S_a,1)\} & \emptyset & \{(Pid,1)\} & \emptyset & \emptyset & \emptyset \\ \{(Add,1)\} \emptyset & \{(S_d, \frac{2}{3}), (S_e,1)\} & \{(Pid,1)\} & \emptyset & \emptyset & \emptyset \end{array} \right)$$

Property 1. (Domain of definition of a CLM.)

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

- i) $\exists s_y \in S_{W_s}$ 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).

3.5 Causal Link Matrix Issues

The key contribution of the CLM is a formal and semantic model to control a set of Web services which are relevant for a Web service composition. The set of Web services S_{W_s} is closed in order to limit the dimension of the CLM. This model allows performance analysis of proposed plans with a concrete view of the composition background i.e., causal links and their semantic dependency. The CLM 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_{W_s} as semantic connections. Moreover the CLM model is an interesting trade-off to support tasks such as Web service verification (valid causal link) or repairing by insertion and deletion of Web services in CLMs. The CLM is able to prepare a suitable context for an AI planning problem [22, 12] with the purpose of obtaining a complete, correct, consistent and optimal plan.

A set of ontologies \mathcal{T} , a set of Web services S_{W_s} , a goal β , a knowledge base \mathcal{KB} and a semantic similarity function

$Sim_{\mathcal{T}}$ are required in order to satisfy such a challenging solution. The conceptual analysis and knowledge representation are supported by a terminology \mathcal{T} . The set of Web services S_{W_s} refers to a set of actions for a planning problem. The goal β 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

In this section one formalizes an AI planning problem as triple $\Pi = \langle S_{W_s}, \mathcal{KB}, \beta \rangle$. S_{W_s} 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 proposed method for Web service composition consists of finding a plan that produces the desired outputs β according to a knowledge base \mathcal{KB} .

Such a plan is composed of causal links such that the scores of causal links allows the early detection of feasible and best semantic links between Web services (definitions 1 and 2). The approach is based on the causal link validity between Web service i.e., a CLM of the specified domain. The CLM aims at detecting all trivial and more complex Web service compositions with semantic connections hence the identification of sequences of Web service through the sequence-composability property. This composability exposes a composition $s_x \circ s_y$ in case an output of s_y is consumed by an input of another Web service s_x . The knowledge of sequence-composability between Web services is expressed in CLMs according to the theorem 1 [18].

Theorem 1. (Sequence-composability)

Let \mathcal{M} be a CLM, and s_x, s_y be two Web services in S_{W_s} . 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 $(s_y, v) \subseteq m_{i,j}$. $c_{j,label}$ and $r_{i,label}$ are respectively inputs of s_x ($In(s_x)$) and s_y ($In(s_y)$).

Remark 1. In case of more complex cases of composition, more than one Web service needs to be chained with s_x in order to produce input parameters of s_x i.e., $s_x \circ (s_{y_1} \wedge \dots \wedge s_{y_n})$. Thus parallel and non-deterministic constructs may be applied. The non-deterministic constructs is conceivable in case the number of pairs in a CLM entry is greater than 1 (e.g., $m_{5,3}$ in \mathcal{M}) whereas the parallel construct is applied in case more than one input parameter is necessary to call a Web service (e.g., $m_{1,2}$ and $m_{4,2}$).

Example 5. (Sequence-composability illustration)

Suppose the CLM \mathcal{M} in section 3. S_c and S_d (table 1) are sequence-composable in S_{W_s} if and only if $S_d \circ S_c$ (theorem 1). Indeed there exists $(i, j) = (2, 6)$ 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 2).

4.1 An AI Progression-based Approach

The studied AI planning based Web services composition is viewed as a simple form of AI planning wherein non determinism and implicit goal [31] are out of question. The set of Web services S_{Ws} i.e., *Actions* is closed by assumption and Web services are supposed to be atomic. The set of *Goals* β refers to a set of instantiated concepts in a terminology \mathcal{T} . Thus the suggested solution is a plan in a well-defined domain i.e., *Goals* are explicitly given, the *Initial state* is well defined and *Actions* are strictly defined in a restricted form of the functional level. So non determinism, implicit goal, fuzzy description of Web service are not considered in our simple form of AI planning. Therefore it does seem possible to directly apply current AI planning methods to our specific problem.

4.1.1 The Core Operators to Construct Plans

A minimum set of operators are necessary to describe the partial ordering [27] of Web services in an AI planning-based Web service composition. This set of operators is denoted by $\{\wedge, \vee, \circ\}$. The \wedge operator represents the conjunction construct in order to express the parallel execution of Web services. Such a situation is possible in case a Web service contains more than one input parameter (e.g. $m_{1,2}$, $m_{4,2}$ in the example 3). For instance $s_x \wedge s_y$ means that Web services s_x and s_y are executed in parallel. The \vee operator is the disjunction construct for non determinism operations. For example $s_x \vee s_y$ means that the Web service s_x or s_y may be executed e.g., $m_{5,3}$ in the example 3. The \circ operator stands for the sequence construct. Let s_x and s_y be two Web services, $s_y \circ s_x$ illustrates the sequence composability between s_x and s_y i.e., $\exists Out_{s_x} \in Out(s_x)$ and $In_{s_y} \in In(s_y)$ such that $\langle s_y, Sim_{\mathcal{T}}(Out_{s_y}, In_{s_x}), s_x \rangle$ is a valid causal link.

Definition 4. (Priority order on the core operators)

Let $\langle S_{Ws}, \mathcal{KB}, \beta \rangle$ be an AI planning based Web service composition problem, the priority of each operator of $\{\wedge, \vee, \circ\}$ is defined by $\wedge > \vee > \circ$ such that $>$ is the priority order.

Example 6. (Illustration of the core operators)

Suppose a set of Web service $\{S_a, S_{a1}, S_{a2}, S_b, S_c, S_d, S_e\}$. According to the definition 4 the following expression:

$$S_b \circ S_d \vee S_e \circ S_c \circ S_a \circ S_{a1} \wedge S_{a2}$$

becomes more legible by removing possible ambiguities using parentheses signs (and) i.e.,

$$S_b \circ ((S_d \vee S_e) \circ (S_c \circ (S_a \circ (S_{a1} \wedge S_{a2}))))$$

4.1.2 Pa_4C as a Four Steps Approach for Automation of Web Service Composition

The composition process consists of a recursive and progression-based approach Pa_4C wherein the AI planning based Web service composition is $\Pi = \langle S_{Ws}, \mathcal{KB}, \beta \rangle$. The ultimate goal is the discovery of composite Web services as plans. The suggested plans must satisfy the goal β according to initial conditions i.e., \mathcal{KB} . The Pa_4C process is explained through the algorithm 1 and its four main steps.

The first step of the Pa_4C algorithm (line 6) consists on valuating the semantic similarity between concepts in \mathcal{KB} and the goal concept β . In case a positive value is returned the Pa_4C process is stopped because the goal β is satisfy by elements of the knowledge base. Alternatively in case $Sim_{\mathcal{T}}(C_k, \beta) = 0$ the proposed solution is not yet able to solve the goal β . This step is responsible for the verification of the stop condition.

The second step (lines 8 to 11) consists of a process of Web service discovery. In other words this step is responsible for discovering all Web services with elements of \mathcal{KB} as input parameters. All those Web services are stored in a set S_c wherein elements are in $S_{Ws} \times (0, 1]$. Thus a Web service $s_y \in S_c$ is a Web service with at least one input parameter in \mathcal{KB} and one output parameter *semantically close* ($Sim_{\mathcal{T}}$) to an input parameter of another Web service.

The third step (lines 13 to 24) is the core process of the progression-based approach to compose Web services according to their causal links. All Web services of S_c which have been discovered in the second step are then parsed in order to obtain a subset of Web services. This parsing stage allows to prune the set S_c in order to obtain a set of Web services such that all their input parameters are in \mathcal{KB} (line 15). Once the pruning phase is completed the output parameters of all the latter Web services are added to the knowledge base \mathcal{KB}' (line 17). Then the process is iterated with the new AI planning problem $\langle S_{Ws}, \mathcal{KB}', \beta \rangle$ (line 23).

According to the algorithm 1 each new output parameter Out_{s_y} of a Web service s_y is added to the knowledge base \mathcal{KB}' (line 17) in order to consider a new knowledge base \mathcal{KB}' hence a new AI planning problem $\langle S_{Ws}, \mathcal{KB}', \beta \rangle$ (line 23). However several output parameters Out_{s_y} are responsible for accumulating loops in the final plan. The detection of those kind of output parameters is necessary to propose solutions without loop in their plans. The operation in line 19 of the Pa_4C algorithm is in charge of detecting loops in proposed plans and the operation in line 21 is responsible for reducing the new knowledge base \mathcal{KB}' hence the reduction of the set of solution of $\langle S_{Ws}, \mathcal{KB}', \beta \rangle$.

Algorithm 1: Composition by progression-based approach Pa_4C .

```

1 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 temporary solved goals  $G \equiv G \cup \mathcal{KB}$ .
2 Result: A disjunction of consistent plans  $\pi$ .
3 begin
4    $S_c \leftarrow \emptyset$ ; // Set of elements in  $S_{Ws} \times (0, 1]$ .
5   // Stop condition of the  $Pa_4C$  algorithm.
6   if  $\exists C_k \in \mathcal{KB} \ \& \ Sim_{\mathcal{T}}(C_k, \beta) \neq 0$  then  $\pi \leftarrow \beta$ ;
7   // Web services discovery with  $kb_i$  as input.
8   foreach  $kb_i \in \mathcal{KB}$  do
9     foreach  $I_j \in Input(S_{Ws})$  do
10      if  $\exists (s_y, v) \in m_{kb_i, I_j}$  then
11         $Add((s_y, v), S_c)$ ;
12  // Plan for Web service composition.
13  if  $S_c \neq \emptyset$  then
14    foreach pair  $(s_y, v) \in S_c$  do
15      if  $In(s_y) \subset \mathcal{KB}$  then
16         $\pi \leftarrow s_y \vee \pi$ ;
17         $\mathcal{KB}' \leftarrow \mathcal{KB} \cup Out(s_y)$ ;
18        foreach  $Out_{s_y} \in Out(s_y)$  do
19          if  $Out_{s_y} \in G$  then
20             $\pi \leftarrow \pi \wedge \emptyset$ ; // Inconsistent plan.
21             $\mathcal{KB}' \leftarrow \mathcal{KB}' \setminus Out_{s_y}$ ;
22           $Add(Out_{s_y}, G)$ ;
23           $\Pi \leftarrow \langle S_{Ws}, \mathcal{KB}', \beta \rangle$ ;
24           $\pi \leftarrow$ 
             $\bigwedge_{Out(s_y)} Pa_4C(\mathcal{M}, \pi, \Pi, G) \circ \pi$ ;
25  else  $\pi \leftarrow \pi \wedge \emptyset$ ; // Incorrect plan
26  return  $\pi$ ;
27 end

```

Alternatively, in case the set of discovered Web services is empty in the third step (lines 13 to 24), the recursive process is stopped and the plan is reduced to \emptyset in the fourth step (line 25) because no Web service with input parameters in \mathcal{KB} are available in S_{Ws} .

The algorithm 1 presents the complete progression-based Web service composition and returns a plan composed of valid causal links (definition 2) and *sequence-composable* Web services (theorem 1). The two latter properties are ensured by the definition of the *CLM* of the domain. *CLMs* ease the progression-based search because all Web services are semantically well ordered in a robust and formal model. Web service composition are designed as plans wherein Web services are semantically chained by valid causal links. Instead a progression-based approach, other problem-solving techniques such as backward-chaining [18] - called heuristic reasoning [14] -

may be applied. The composition by regression-based approach Ra_4C is applied until concepts are in the knowledge base whereas the Pa_4C algorithm [18] is a recursive process which acts from elements of the knowledge base \mathcal{KB} until the goal β is satisfied.

4.2 Properties of Plans-based Web Service Composition

In the exposed approach Web service composition is compared to a simpler AI planning problem. The proposed solutions i.e., plans have to satisfy properties of consistency, completeness, correctness and optimality (depending on a criterion) [26, 14] of plans.

4.2.1 Consistency of Plans

Consistency of plans is a necessary property in order to discover a solution for Web service composition. This property is satisfied by plans which do not contain cycles (line 19 to 21) in their ordering constraints and no causal link conflicts [26]. By definition of a *CLM* (definition 2) the causal links of plans are valid hence no causal link conflicts in the proposed plans. Thus a consistent plan is considered without infamous *Sussman* anomaly [32]. The Pa_4C algorithm builds such a plan and avoid cycles and conflicts to dispose of inconsistent causal links.

Example 7. (Set of consistent plans)

Let \mathcal{M} 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. According to the Pa_4C algorithm, the result is a disjunction of two consistent plans π_{a1} , π_{b1} and an inconsistent plan π_{a2} .

$$\begin{aligned}
\pi_{a1} &= S_b \circ ((S_d \circ S_c \circ S_a(Add \wedge Pid))) \\
\pi_{b1} &= S_b \circ ((S_e \circ S_c \circ S_a(Add \wedge Pid))) \\
\pi_{a2} &= S_a(Add \wedge Pid) \wedge \emptyset
\end{aligned}$$

All plans suggested by the Algorithm 1 do not necessarily satisfy the correctness and completeness properties.

4.2.2 Completeness of Plans

A complete plan [26] is a plan wherein every input parameter of every Web service is achieved by output parameters of other Web services. In such a context a complete plan is a partial order of well-ordered causal links.

By definition, a *CLM* contains all necessary information about the complete plans because a *CLM* explicitly stores all valid causal links between Web services. Thus a valid causal link between two Web services informs that those

two Web services are *sequence composable* (theorem 1). Moreover the completeness of plans is justified by the operation $In(s_y) \subset \mathcal{KB}$ (line 15) of the algorithm 1. Indeed only Web services with their input parameters in \mathcal{KB} are proposed in the final disjunction of plans.

Example 8. (*Set of complete plans*)

The set of consistent plans of the example 7 is also a set of complete plans. Indeed all input parameters of Web services implied in the plans are achieved by output parameters of other Web services. For instance the input parameter of S_d are achieved by the output parameter of S_c hence the sequence-composability $S_d \circ S_c$ between S_d and S_c .

4.2.3 Correctness of Plans

In classical AI planning, a plan is correct [14] if it is executable, and produces a state that satisfies the goal formula.

The process of plan refinement follows a forward chaining method from the initial states \mathcal{KB} to the goal β such that the final plan is composed of valid causal links. In case a solution of $\langle S_{W_s}, \mathcal{KB}, \beta \rangle$ exists the final plan is a disjunction of conjunction of sequence-composable Web services. The correct plans are plans without \emptyset (lines 20, 25) in their plans. Indeed the first assignment (line 20) aims at locating loops whereas the second assignment (line 25) eliminates the incorrect plans.

Example 9. (*Set of correct plans*)

The set of consistent, complete and correct plans of the example 7 is reduced to

$$\begin{aligned} \pi_{a1} &= S_b \circ ((S_d \circ S_c \circ S_a(Add \wedge Pid))) \\ \pi_{b1} &= S_b \circ ((S_e \circ S_c \circ S_a(Add \wedge Pid))) \end{aligned}$$

Indeed π_{a1} and π_{b1} are correct plans because these two plans produce the goal $\{Person\}$ according to the initial state $\{Add, PatientID\}$.

In the context of semantic Web service, the correctness of a Web service composition is guaranteed by the semantic link (i.e., causal link) between the input and output parameters of Web services.

4.2.4 Local Optimality of Plans

The Pa_4C algorithm is able to return a set of correct, complete and consistent plans. However such a set may contain a large number of plans. Thus one chooses to prune the set of correct, complete and consistent plans to expose an optimal solution for Web service composition. A *causal link*-based optimization criterion is introduced to detect the optimal plan. The heuristic is presented as follows i.e., *The more valid causal links between Web services and the less input parameters of Web services the better is the plan.* The

optimal plan is a plan with a local optimization of its valid causal links. Indeed each Web service is locally chosen according to the causal link criterion wherein a Web service with a minimum of input parameters is preferred. The weight of the optimal plan is computed by means of the *CLM* and algorithm 1. Contrary to the algorithm 1, the optimal plan is computed according to a regression-based strategy:

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

The recursive function W_{Max}^{local} returns the weight of the best plan depending on the goal β and the initial conditions \mathcal{KB} . The formula (4) is based on the score 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 parameter β and inputs parameters $I_i, 1 \leq i \leq \#In(s_y)$. In other words, $\langle s_y, Sim_{\mathcal{T}}(Out_{s_y}, \beta), s_x \rangle$ is a valid causal link hence a *sequence-composability* between s_y and s_x i.e., $s_x \circ s_y$. The $In(s_y)$ set is the set of input parameters of $s_y \in S_{W_s}$ whereas I_i is an input parameter of s_y . \mathcal{M} is a *CLM* with coefficients in $\mathcal{P}((S_{W_s} \cup \mathcal{T}) \times (0, 1])$. $m_{I_i, \beta}.score$ is the second component of a couple $(s_y, v) \subseteq m_{I_i, \beta}$ i.e., the score of a valid causal link. Max_S is a n -arity function which returns the maximum value between n float value(s). In case one supposes a Web service with more input parameters than another Web service is not preferred in the composition schema the inverse proportionality $\frac{1}{2^{(\#In(s_y)-1)}}$ may be exchanged by the computation of the average $\frac{1}{\#In(s_y)}$ without loss of generalities. The only change is about the heuristic. Given a *CLM* the combination of the algorithm 1 and formula (4) is an interesting trade-off to find an optimal (4), consistent (algorithm 1), correct (definition 2) and complete (theorem 1 and algorithm 1) plan when one exists.

Example 10. (*Local optimality of plans.*)

According to example 9, π is divided into a disjunction of two consistent, correct and complete plans $\{\pi_{a1}, \pi_{b1}\}$. The Weights of the different plans have been computed with (4).

$Weight(\pi_{a1}) = 1 \times \frac{2}{3} \times 1 \times \frac{1}{2^{(2-1)}} \times (1 + 1) = \frac{2}{3}$ whereas $Weight(\pi_{b1}) = 1 \times 1 \times 1 \times \frac{1}{2^{(2-1)}} \times (1 + 1) = 1$. Thus π_{b1} is the optimal plan in our context.

4.2.5 Global Optimality of Plans

The global optimality of plans consists on computing all weights of each plan and choose the best one depending on the score value. However the computation of the optimal global plan is more time consuming in case the Pa_4C

algorithm returns a large number of plans for Web service composition. The discovery of all plans may be very costly according to the algorithm 1. Indeed the complexity of the global plan computation θ_{global} is close to $(\theta_{local} \times \#\pi)$ wherein θ_{local} is the complexity of the optimal plan with a local optimization and $\#\pi$ is the cardinality of disjunctive plans returned by the Pa_4C algorithm.

5 Related Work

The Matrix as a formal model to represent Web services [7, 10] at functional level seems to be an interesting and promising approach. The authors of [7] solve an AI planning problem wherein actions are viewed as tasks. In this model the actions are formally described with Preconditions and Effects. These tasks are executed by concrete Web services, according to a service/task (row/column) matrix. A simpler method to store Web service according to an input/output (row/column) matrix is exposed in [10]. Contrary to our model which exposes an input/input matrix with a pre-computation of causal links, [10] presents a simple representation model for Web service. The Matrix model used in [7, 10] do not propose reasoning about those matrices. In fact, such matrices are simply considered as a formal model for Web services description without semantic.

Others authors use the Petri Nets model for Web service composition. In [22] the authors use situation calculus for representing web service description and Petri nets for describing the execution behaviours of web services. This approach is very promising for detecting deadlocks and concurrency process, augmenting the probability of reaching the goal of involved services.

Recently the authors of [17] have addressed in more detail the problem of interleaving Web service discovery and composition but have considered only simple workflows where Web services have one input and one output parameter. The plan of a Web service composition is restricted to a sequence of limited Web services hence a linear workflow of Web services. In such a case the solution proposed in this paper returns a sequence of causal links between Web services hence a linear and total order plan. Contrary to [17], the model proposed in this paper may also considers Web services with more than one input and output parameter.

Composition of Web services has been discussed in [28]. It provides a composer that will aid the user to select Web services for each activity in the composition and to create flow specifications to link them. Upon selecting a Web service, the Web services that can produce output that could be fed as the input of the selected service are listed after filtering based on profile descriptions. The user can manually select the service that he wants to fit in at a particular activity. After selecting all the services, the system generates a composite process in DAML-S. The execution is done by calling

each service separately and passing the results between services according the flow specifications. However the composition is still semi-automatic because the user must select a Web service in a restricted list. Our formal model presented in this paper aims at automating the process of Web service selection according to the causal link criterion and the two models of optimization.

From HTNs [30] to regression planning based on extensions of PDDL [11], different planning approaches have been proposed for Web service composition. Hierarchical Task Network (HTN) planning for Web Service composition [30] is an example of AI planning based composition approach. An HTN planning problem is formulated in terms of tasks to be accomplished where a task represents an abstract activity. A task is defined with its name and the number of parameters it has. The key concept of this model is the possible decomposition of task into subtasks by operators which are single-step atomic actions with no internal structure. Tasks are matched against method descriptions, which is a prescription for how to decompose a task into subtasks. The tasks that are accomplished by operators are named primitive tasks. An HTN planning problem starts with one or more tasks, methods are matched with the tasks and recursively decomposed until each step is reduced to an operator. The resulting plan is a sequence of operators. In the category of AI planning based Web service composition [34] propose a forward chaining approach. 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. Nevertheless no formal model has been proposed to help representation and automation of Web service composition at the best stage of our knowledge. In this paper we outlined the main challenge faced in semantic Web services. Indeed we showed how the *CLM* tackles the problem of Web service composition by providing a formal and semantic model by suggesting a concrete context for automation of Web service composition. The context captures semantic connections between Web services as causal links between an output parameter of a Web service and an input parameter of another Web service. Semantically weighted by the Sim_T function, the latter link refers to a local optimization criteria in order to discover plans as solutions. Moreover one introduced a way to compute the best Web service composition according to a local criterion of optimization. This criterion i.e., the causal link ensures the *sequence-composability* between Web services. The proposed algorithm (Pa_4C) returns a set of consistent plans. Once this set is pruned, a set of consis-

tent, complete and correct plans is proposed as solutions. The model of functional level composition is easily applied to Web services which are described according to the OWL-S service profile [2] or the WSMO capability model [13]. Finally, contrary to [7, 10], our matrix model pre-computes the semantic similarities between Web services (individual inputs and outputs) in order to make Web service composition easier in a closed set of Web service. *CLMs* do not only allow to bound the Web service domain but also provide a semantic context for the service composition. Thus Web service composition is viewed as causal link composition wherein the composition plan is built from a simple causal link matrix analysis (Pa_4C).

For further studies we plan to improve and extend the set of semantic Web service matching functions for optimization reasons. Causal link matrices as sparse matrices need to be studied in more detail. Moreover a process level composition of Web services needs to be associated to our functional level composition in order to guarantee a full correctness of the composition process.

References

- [1] G. Alonso, F. Casati *et al.* *Web Services: Concepts, Architectures and Applications*. Springer-Verlag, 2004.
- [2] A. Ankolenkar, M. Paolucci *et al.* The owl services coalition, owl-s 1.1 beta release. Technical report, July 2004.
- [3] A. Baker. *Matrix Groups: An Introduction to Lie Group Theory*. Springer undergraduate mathematics series. Springer-Verlag, London, 2002.
- [4] B. Benatallah, M.-S. Hacid *et al.* On automating web services discovery. *VLDB J*, 14(1):84–96, 2005.
- [5] D. Berardi, D. Calvanese *et al.* Automatic composition of e-services that export their behavior. In *1st Int. Conf. on Service Oriented Computing (ICSOC)*, pages 43–58, 2003. volume 2910.
- [6] T. Bultan, X. Fu *et al.* Conversation specification: a new approach to design and analysis of e-service composition. In *International conference on WWW*, pages 403–410, 2003.
- [7] D. B. Claro, P. Albers *et al.* Selecting web services for optimal composition. In *ICWS International Workshop on Semantic and Dynamic Web Processes*, Orlando - USA, 2005.
- [8] S. Colucci, T. D. Noia *et al.* Concept abduction and contraction for semantic-based discovery of matches and negotiation spaces in an e-marketplace. In *Proceedings of the 6th ICEC*, pages 41–50. ACM Press, 2004.
- [9] I. Constantinescu and B. Faltings. Efficient matchmaking and directory services. In *IEEE/WIC International Conference on Web Intelligence (WI'03)*, page 75, 2003.
- [10] I. Constantinescu, B. Faltings *et al.* Type based service composition. In *WWW (Alternate Track Papers & Posters)*, pages 268–269, 2004.
- [11] D. M. Dermott. PDDL - the planning domain definition language, Aug. 07 1997.
- [12] M. Desjardins, M. Sheshagiri *et al.* A planner for composing services described in DAML-S, May 05 2003.
- [13] D. Fensel, M. Kifer *et al.* Web service modeling ontology (wsmo) submission, w3c member submission. June 2005.
- [14] M. Ghallab, D. Nau *et al.* *Automated Planning: Theory and Practice*. Morgan Kaufmann Publishers, May 2004.
- [15] M. Klusch, B. Fries *et al.* Owls-mx: Hybrid owl-s service matchmaking. In *AAAI Fall Symposium Series*, November 2005.
- [16] R. Küsters. *Non-Standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Computer Science*. Springer, 2001.
- [17] O. Lassila and S. Dixit. Interleaving discovery and composition for simple workflows. In *Semantic Web Services, AAAI Spring Symposium Series*, 2004.
- [18] F. Lécué and A. Léger. A formal model for semantic web service composition. In *ISWC the 5th International Semantic Web*, November 2006.
- [19] F. Lécué and A. Léger. Semantic web service composition through a matchmaking of domain. In *4th IEEE European Conference on Web Services (ECOWS)*, December 2006.
- [20] L. Li and I. Horrocks. A software framework for matchmaking based on semantic web technology. In *Proceedings of the Twelfth International Conference on WWW*, pages 331–339, 2003.
- [21] D. McAllester and D. Rosenblitt. Systematic nonlinear planning. pages 634–639, Menlo Park, CA, July 1991. AAAI.
- [22] S. Narayanan and S. McIlraith. Simulation, verification and automated composition of web services,. *Eleventh International World Wide Web Conference*, pages 7–10, May 2002.
- [23] M. Paolucci, T. Kawamura *et al.* Semantic matching of web services capabilities. In *Proceedings of the First International Semantic Web Conference, LNCS 2342*, pages 333–347. Springer-Verlag, june 2002.
- [24] M. Paolucci, K. P. Sycara *et al.* Delivering semantic web services. In *Proceedings of the international conference on WWW (Alternate Paper Tracks)*, pages 829–837, 2003.
- [25] M. Pistore, P. Roberti *et al.* Process-level composition of executable web services: "on-the-fly" versus "once-for-all" composition. In *ESWC*, pages 62–77, 2005.
- [26] S. Russell and P. Norvig. *Artificial Intelligence: a modern approach*. Prentice-Hall, 1995.
- [27] E. Sacerdoti. The nonlinear nature of plan. In *IJCAI-4*, pages 206–214, 1975.
- [28] E. Sirin, J. A. Hendler *et al.* Semi-automatic composition of web services using semantic descriptions. In *WSMAI*, pages 17–24, 2003.
- [29] E. Sirin, B. Parsia *et al.* Filtering and selecting semantic web services with interactive composition techniques. *IEEE Intelligent Systems*, 19(4):42–49, 2004.
- [30] E. Sirin, B. Parsia *et al.* HTN planning for web service composition using SHOP2. *J. Web Sem*, 1(4):377–396, 2004.
- [31] B. Srivastava and J. Koehler. Web service composition - current solutions and open problems. In *ICAPS 2003 Workshop on Planning for Web Services*, July 22 2003.
- [32] G. J. Sussman. *A Computer Model of Skill Acquisition*. New York: New American Elsevier, 1975.
- [33] K. P. Sycara, M. Paolucci *et al.* Automated discovery, interaction and composition of semantic web services. *J. Web Sem*, 1(1):27–46, 2003.
- [34] R. Zhang, I. B. Arpinar *et al.* Automatic composition of semantic web services. In *ICWS*, pages 38–41, 2003.