ECAI 2008
M. Ghallab et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-891-5-45

Optimizing Causal Link Based Web Service Composition

Freddy Lécué[1,2] and **Alexandre Delteil**[2] and **Alain Léger**[2]

Abstract. Automation of Web service composition is one of the most interesting challenges facing the Semantic Web today. Since Web services have been enhanced with formal semantic descriptions, it becomes conceivable to exploit causal links i.e., semantic matching between their functional parameters (i.e., outputs and inputs). The semantic quality of causal links involved in a composition can be then used as a innovative and distinguishing criterion to estimate its overall semantic quality. Therefore non functional criteria such as quality of service (QoS) are no longer considered as the only criteria to rank compositions satisfying the same goal. In this paper we focus on semantic quality of causal link based semantic Web service composition. First of all, we present a general and extensible model to evaluate quality of both elementary and composition of causal links. From this, we introduce a global causal link selection based approach to retrieve the optimal composition. This problem is formulated as an optimization problem which is solved using efficient integer linear programming methods. The preliminary evaluation results showed that our global selection based approach is not only more suitable than the local approach but also outperforms the naive approach.

1 Introduction

The semantic web [6] is considered to be the future of the current web. Web services in the semantic web are enhanced using rich description languages such as the Web Ontology Language (OWL) [19]. Formally the latter semantic descriptions are expressed by means of Description Logics concepts [4] in ontologies. An ontology is defined as a formal conceptualization of a domain we require to describe the semantics of services e.g., their functional input, output parameters. Intelligent software agents can, then, use these descriptions to reason about web services and automate their use to accomplish intelligent tasks e.g., selection, discovery, composition.

In this work we focus on web service composition and more specifically on its functional level (aka causal link composition). Starting from an initial set of web services, such a level of composition aims at selecting and inter-connecting web services by means of their (semantic) causal links according to a goal to achieve. The functional criterion of causal link, first introduced in [14], is defined as a semantic connection between an output of a service and an input parameter of another service. Since the quality of the latter links are valued by a semantic matching between their parameters, causal link compositions could be estimated and ranked as well. From their estimation results, some compositions can be considered as unsuitable in case of under specified causal links. Indeed a composite service that does not provide acceptable quality of causal links might be as useless as a service not providing the desired functionality.

Unlike most of approaches [5, 22, 23] which focus on the quality of composition by means of non functional parameters i.e., quality of service (QoS), the quality of causal links can be considered as a distinguishing functional criterion for semantic web service compositions. Here we address the problem of optimization in service composition with respect to this functional criterion. Retrieving such a composition is defined as the global selection of causal links maximizing the quality of the composition, taking into account preferences and constraints defined by the end-user. To this end, an objective function maximizing the overall quality subject to causal links constraints is introduced. This leads to an NP-hard optimization problem [8] which is solved using integer linear programming methods.

The remainder of this paper is organised as follows. In the next section we briefly review i) causal links, ii) a distinguishing criterion i.e., their robustness and iii) the causal link composition model. Section 3 defines the causal link quality criteria we require during the global selection phase. Section 4 formulates the problem of global causal link selection and describes an integer linear programming method to efficiently solve it. Section 5 presents its computational complexity and some experimentations. Section 6 briefly comments on related work. Finally section 7 draws some conclusions and talk about possible future directions.

2 Background

First of all, we present causal links. Then we remind the definition of their robustness, and finally describe causal link composition.

2.1 Web Service Composition & its Causal Links

In the semantic web, parameters (i.e., input and output) of services referred to concepts in a common ontology[3] or Terminology $\mathcal{T}$, where the OWL-S profile [1] or SA-WSDL [18] can be used to describe them (through semantic annotations). At functional level web service composition consists in retrieving some semantic links between output parameters $Out\_s_i \in \mathcal{T}$ of services s_i and input parameters $In\_s_j \in \mathcal{T}$ of other services s_j. Such a link i.e., causal link [14] $cl_{i,j}$ (Figure 1) between two functional parameters of s_i and s_j is formalized as $\langle s_i, Sim_{\mathcal{T}}(Out\_s_i, In\_s_j), s_j\rangle$. Thereby s_i and s_j are partially linked according to a matching function $Sim_{\mathcal{T}}$. This function expresses which matching type is employed to chain services. The range of $Sim_{\mathcal{T}}$ is reduced to the four well known matching type introduced by [16] and the extra type Intersection [15]:

- **Exact** If the output parameter Out_s_i of s_i and the input parameter In_s_j of s_j are equivalent; formally, $\mathcal{T} \models Out\_s_i \equiv In\_s_j$.
- **PlugIn** If Out_s_i is sub-concept of In_s_j; formally, $\mathcal{T} \models Out\_s_i \sqsubseteq In\_s_j$.
- **Subsume** If Out_s_i is super-concept of In_s_j; formally, $\mathcal{T} \models In\_s_j \sqsubseteq Out\_s_i$.
- **Intersection** If the intersection of Out_s_i and In_s_j is satisfiable; formally, $\mathcal{T} \not\models Out\_s_i \sqcap In\_s_j \sqsubseteq \perp$.

---

[1] Ecole de Mines de Saint-Etienne, France, email: freddy.lecue@gmail.com
[2] Orange Labs, France, email: {firstname.lastname}@orange-ftgroup.com

[3] Distributed ontologies are not considered here but are largely independent of the problem addressed in this work.

- **Disjoint** Otherwise Out_s_i and In_s_j are incompatible i.e., $\mathcal{T} \models Out\_s_i \sqcap In\_s_j \sqsubseteq \bot$.

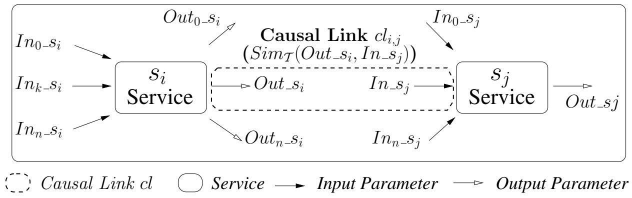

Figure 1. Illustration of a Semantic Causal Link $cl_{i,j}$.

2.2 Robust Causal Link

The latter matching function $Sim_{\mathcal{T}}$ enables, at design time, finding some levels of semantic compatibilities (i.e., Exact, PlugIn, Subsume, Intersection) and incompatibilities (i.e., Disjoint) among independently defined web service descriptions. However, as emphasized by [13], the matching types Intersection and Subsume need some refinements to be fully efficient for causal links composition.

Example* 1.** ***(Causal Link & Subsume Matching Type)
Suppose s_1 and s_2 be two services such that the output parameter `NetworkConnection` *of s_1 is (causal) linked to the input parameter* `SlowNetworkConnection` *of s_2 ($cl^1_{1,2}$ in Figure 3). This causal link is valued by a Subsume matching type since $NetworkConnection \sqsupset SlowNetworkConnection$ (Figure 2). It is obvious that such a causal link should not be directly applied in a service composition since the* `NetworkConnection` *is not specific enough to be used by the input* `SlowNetworkConnection`*. Indeed the output parameter* `NetworkConnection` *requires some* Extra Descriptions *to ensure a composition of s_1 and s_2.*

A causal link valued by the Intersection matching type requires a comparable refinement. From this, [13] defined a robust causal link.

$$NetworkConnection \equiv \forall netPro.Provider \sqcap \forall netSpeed.Speed$$
$$SlowNetworkConnection \equiv NetworkConnection \sqcap \forall netSpeed.Adsl1M$$
$$Adsl1M \equiv Speed \sqcap \forall mBytes.1M$$

Figure 2. Sample of an $\mathcal{ALE}$ domain ontology $\mathcal{T}$.

Definition 1. ***(Robust Causal link)***
A causal link $\langle s_i, Sim_{\mathcal{T}}(Out\_s_i, In\_s_j), s_j\rangle$ is robust iff the matching type between Out_s_i and In_s_j is either Exact or PlugIn.

Property 1. ***(Robust Web Service Composition)***
A composition is robust iff all its causal links are robust.

A possible way to replace a link $\langle s_i, Sim_{\mathcal{T}}(Out\_s_i, In\_s_j), s_j\rangle$ valued by Intersection or Subsume in its robust form consists in computing the information contained in the input In_s_j and not in the output Out_s_i. To do this, the difference or subtraction operation [7] for comparing $\mathcal{ALE}$ DL descriptions is adapted in [13]. Even if [20] previously presented an approach to capture the real semantic difference, the [7]'s difference is preferred since its result is unique. From this, in case a causal link $\langle s_i, Sim_{\mathcal{T}}(Out\_s_i, In\_s_j), s_j\rangle$ is neither valued by a Disjoint matchmaking nor robust, Out_s_i and In_s_j are compared to obtain two kinds of information, a) the *Extra Description* $In\_s_j \backslash Out\_s_i$ that refers to the information required but not provided by Out_s_i to semantically link it with the input In_s_j of s_j, and b) the *Common Description* $Out\_s_i \sqcap In\_s_j$ that refers to the information required by In_s_j and effectively provided by Out_s_i.

Example* 2.** ***(Robustness, Extra & Common Description)
Suppose the causal link presented in Example 1. Such a link is not robust enough (Definition 1) to be applied in a composition. The description missing in `NetworkConnection` *to be used by the input parameter* `SlowNetworkConnection` *is defined by the* Extra Description $SlowNetworkConnection \backslash NetworkConnection$ *i.e., $\forall netSpeed.Adsl1M$. However the* Common Description *is not empty since this is defined by $SlowNetworkConnection \sqcap NetworkConnection$ i.e., $\forall netPro.Provider$.*

Robust causal links can be obtained by retrieving *Extra Description* that changes an Intersection in a PlugIn matching type, and a Subsume by an Exact matching type.

2.3 Causal Link Composition Model

In this work, the process model of web service composition and its causal links is specified by a statechart [10]. Its states refer to services whereas its transitions are labelled with causal links. In addition some basic composition constructs such as sequence, conditional branching (i.e., OR-Branching), structured loops, concurrent threads (i.e., AND-Branching), and inter-thread synchronization can be found. To simplify the presentation, we assume that all considered statecharts are acyclic and consists of only sequences, OR-Branching and AND-Branching. In case of cycle, a technique for *unfolding* statechart into its acyclic form needs to be applied beforehand. Details about this unfolding process are omitted for space reasons.

Example* 3.** ***(Process Model of a Causal Link Composition)
Suppose $s_{i,3\leq i\leq 8}$ be six services extending Example 1 in a more complex composition. The process model of this composite service is illustrated in Figure 3. The composition consists in an OR-Branching and AND-Branching wherein nine causal links are involved.

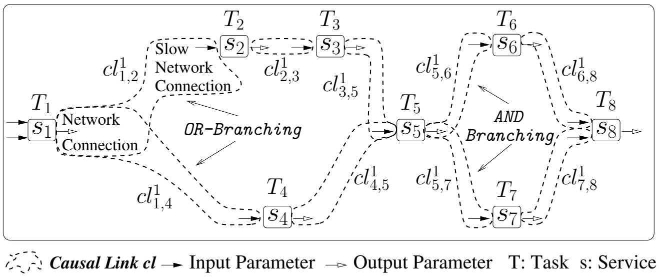

Figure 3. Illustration of an (Executable) Causal Link Composition.

The example 3 illustrates an executable composition wherein tasks T_i have been concretized by one of their candidate services e.g., here s_i. Indeed some services with common functionality, preconditions and effects although different input and output parameters are given and can be used to perform a target task in the composition. In this way we address the issue of composing a large and changing collection of semantic web services. In our approach the choice of services is done at composition time, only based on their causal links with other services. Thus each abstract causal link $cl^A_{i,j}$ between two tasks T_i, T_j of an abstract composition needs to be concretized. Ideally, a relevant link is selected among its n candidate causal links $cl^{k,1\leq k\leq n}_{i,j}$ between two of their services to obtain an executable composition.

Example* 4.** ***(Tasks, Candidate Services & Causal Links)
Let s'_2 be a candidate service for T_2 with `NetworkConnection` *as input parameter. The causal link $cl^2_{1,2}$ between s_1 and s'_2 is then more robust than $cl^1_{1,2}$. Indeed $cl^2_{1,2}$ is valued by an Exact matching type whereas $cl^1_{1,2}$ is valued by a Subsume matching type.*

3 Causal Link Quality Model

As previously presented, several candidate services are grouped together in every task of an abstract composition. A way to differentiate their causal links (e.g., $cl^1_{1,2}$ and $cl^2_{1,2}$ in example 4) consists in considering their different functional quality criteria. To this end, we adopt a causal link quality model, effective to any causal link.

In this section, we first present the quality criteria used for elementary causal links, before turning our attention to composite causal links. For each criterion, we provide a definition and indicates rules to compute its value for a given causal link.

3.1 Quality Criteria for Elementary Causal Links

We consider three generic quality criteria for elementary causal links $cl_{i,j}$ defined by $\langle s_i, Sim_{\mathcal{T}}(Out\_s_i, In\_s_j), s_j\rangle$: its i) *Robustness*, ii) *Common Description* rate, and iii) *Matching Quality*.

- **Robustness.** The Robustness q_r of a causal link $cl_{i,j}$ is defined by 1 in case the link $cl_{i,j}$ is robust (see Definition 1), and 0 otherwise.
- **Common Description rate.** This rate[4] $q_{cd} \in (0, 1]$ is defined by:

$$q_{cd}(cl_{i,j}) = \frac{|Out\_s_i \sqcap In\_s_j|}{|In\_s_j \backslash Out\_s_i| \ + \ |Out\_s_i \sqcap In\_s_j|} \quad (1)$$

  This criterion estimates the rate of descriptions which is well specified for upgrading a non robust causal link into its robust form. In (1), $Out\_s_i \sqcap In\_s_j$ is supposed to be satisfiable since only relevant links between two services are considered in our model.
- **Matching Quality.** The Matching Quality q_m of a link $cl_{i,j}$ is a value in $(0, 1]$ defined by $Sim_{\mathcal{T}}(Out\_s_i, In\_s_j)$ i.e., either 1 (Exact), $\frac{3}{4}$ (PlugIn), $\frac{1}{2}$ (Subsume) and $\frac{1}{4}$ (Intersection). The Disjoint match type is not considered since $Out\_s_i \sqcap In\_s_j$ is satisfiable.

In case we consider $Out\_s_i \sqcap In\_s_j$ to be not satisfiable, it is straightforward to extend and adapt our quality model by computing contraction [9] between Out_s_i and In_s_j. Given the above quality criteria, the quality vector of a causal link $cl_{i,j}$ is defined as follows:

$$q(cl_{i,j}) = \big(q_r(cl_{i,j}), q_{cd}(cl_{i,j}), q_m(cl_{i,j})\big) \quad (2)$$

In case of services s_i and s_j related by more than one causal link, the value of each criterion is retrieved by computing their average.

3.2 Quality Criteria for Causal Link Composition

The above quality criteria are also applied to evaluate the quality of any causal link composition c. To this end, Table 1 provides aggregation functions for such an evaluation. A brief explanation of each criterion's aggregation function follows (here cl stands for $cl_{i,j}$):

- **Robustness.** On the one hand the robustness Q_r of both a sequential and an AND-Branching composition c is defined as the average of its causal link cl's robustness $q_r(cl)$. On the other hand the robustness of an OR-Branching causal link composition is a sum of $q_r(cl)$ weighted by p_r i.e., the probability that causal link *cl* be chosen at run time.
- **Common Description rate.** This Description rate Q_{cd} of c is defined as its robustness, by simply changing $q_r(cl)$ by $q_{cd}(cl)$.
- **Matching Quality.** The matching quality Q_m of a sequential and AND-Branching causal link composition c is defined as a product of $q_m(cl)$. The matching quality of an OR-Branching causal link composition c is defined as $Q_r(c)$, by changing $q_r(cl)$ by $q_m(cl)$.

Using the above aggregation functions, the quality vector of an executable causal link composition is defined by (3). For each criterion $l \in \{r, cd, m\}$ the higher the value Q_l for c the higher its l^{th} quality.

$$Q(c) = (Q_r(c), Q_{cd}(c), Q_m(c)) \quad (3)$$

Even if criteria q_r, q_m used to value a single causal link are correlated, their aggregated values of compositions Q_r, Q_m for Sequential, AND-Branching are independent since they are computed from different functions i.e., linear for Q_r, not for Q_m. Thus a composition c with a high robustness may have either a high or low overall matching quality. We have the same conclusion on the other criteria.

Composition Construct	Quality Criterion		
	Robustness Q_r	Com. Desc. rate Q_{cd}	Match. Qual. Q_m
Sequential/ AND- Branching	$\frac{1}{\lvert cl\rvert}\sum_{cl} q_r(cl)$	$\frac{1}{\lvert cl\rvert}\sum_{cl} q_{cd}(cl)$	$\prod_{cl} \ q_m(cl)$
OR-Branching	$\sum_{cl} q_r(cl).p_{cl}$	$\sum_{cl} q_{cd}(cl).p_{cl}$	$\sum_{cl} q_m(cl).p_{cl}$

Table 1. Quality Aggregation Rules for Causal Link Composition.

4 Global Causal Link Selection

In the following we study the optimal composition[5] as the selection of causal links that optimize the overall quality of the composition.

On the one hand the selection can be locally optimized at each abstract causal link $cl^A_{i,j}$ of the composition, but two main issues arise. First, the local selection of a candidate link $cl^k_{i,j}$ enforces a specific service for both tasks T_i and T_j. Thus, these constraints can no longer ensure to select neither the best links for its closest abstract links $cl^A_{\alpha,i}$ and $cl^A_{j,\beta}$ nor the optimal composition (e.g., the best local selection in $cl^A_{1,2}$ i.e., $cl^1_{1,2}$ does not lead to the optimal composition in Figure 4). Secondly, quality constraints may be not satisfied, leading to a suboptimal composition e.g., a constraint with a robustness more than 70% cannot be enforced. On the other hand, the naive global approach considers an exhaustive search of the optimal composition among all the executable compositions. Let $|cl^A_{i,j}|$ be the number abstract links in an composition and n be the number of candidate services by task, the total number of executable causal link compositions is $n^{2.|cl^A_{i,j}|}$, making this approach impractical for large scale composition.

Here, we address these issues by presenting an integer linear programming (IP) [21] based global causal link selection, which i) further constrains causal links, and ii) meets a given objective.

4.1 IP Based Global Selection & Objective Function

There are 3 inputs in an IP problem: an *objective function*, a set of integer decision *variables* (restricted to value 0 or 1), and a set of *constraints* (equalities or inequalities), where both the objective function and the constraints must be linear. IP attempts to maximize or minimize the value of the objective function by adjusting the values of the variables while enforcing the constraints. The problem of retrieving an optimal executable composition is mapped into an IP problem.

Here we suggest to formalize its objective function. To this end, the robustness, common description rate and matching values of the p potential executable compositions i.e., $Q^{\lambda,1\leq\lambda\leq p}_{l,l\in\{r,cd,m\}}$ have been first determined by means of aggregation functions in Table 1. Then, the latter quality values $Q^{\lambda}_r, Q^{\lambda}_{cd}, Q^{\lambda}_m$ has been scaled according to (4).

$$\tilde{Q}{}^{\lambda}_{l} = \begin{cases} \frac{Q^{\lambda}_l - Q^{\min}_l}{Q^{\max}_l - Q^{min}_l} & \text{if } Q^{\max}_l - Q^{min}_l \neq 0 \\ 1 & \text{if } Q^{\max}_l - Q^{min}_l = 0 \end{cases} \quad l \in \{r, cd, m\} \quad (4)$$

In (4), $Q^{\max}_l$ is the maximal value of the l^{th} quality criteria whereas $Q^{\min}_l$ is the minimal value of the l^{th} quality criteria. This scaling phase complexity is linear in the number of abstract links in the composition. Finally, the objective function (5) of the IP problem follows.

---

[4] $|.|$ refers to the size of $\mathcal{ALE}$ concept descriptions ([12] p.17) i.e., $|\top|$, $|\bot|$, $|A|$, $|\neg A|$ and $|\exists r|$ is 1; $|C \sqcap D| \doteq |C| + |D|$; $|\forall r.C|$ and $|\exists r.C|$ is $1 + |C|$. For instance $|Adsl1M|$ is 3 in Figure 2.

[5] The relation and combination with quality of services is not addressed here.

$$\max_{1\leq\lambda\leq p}\left(\sum_{l\in\{r,cd,m\}}\left(\tilde{Q}_l^{\lambda}\times\omega_l\right)\right) \tag{5}$$

where $\omega_l \in [0,1]$ is the weight assigned to the l^{th} quality criterion and $\sum_{l\in\{r,cd,m\}}\omega_l = 1$. In this way preferences on quality of the desired executable compositions can be done by simply adjusting ω_l e.g., the Common Description rate could be weighted higher.

4.2 Integer Variables & Constraints of IP Problem

For every candidate link $cl_{i,j}^{k,1\leq k\leq n}$ of an abstract link $cl_{i,j}^{A}$, we include an integer variable $y_{i,j}^{k}$ in the IP problem indicating the selection or exclusion of link $cl_{i,j}^{k}$. By convention $y_{i,j}^{k}$ is 1 if the k^{th} candidate link $cl_{i,j}^{k}$ is selected to concretize $cl_{i,j}^{A}$ between tasks T_i and T_j, 0 otherwise. The selected links will form an optimal executable composition satisfying (5) and meeting the following constraints:

Allocation Constraint. Only one candidate link should be selected for each abstract link $cl_{i,j}^{A}$ between tasks T_i and T_j. This constraint is formalized by exploiting the integer variables $y_{i,j}^{k,1\leq k\leq n}$ in (6).

$$\sum_{k=1}^{n} y_{i,j}^{k} = 1,\ \forall cl_{i,j}^{A} \tag{6}$$

Example 5. (Allocation Constraint)
Suppose the sequential composition of tasks T_1, T_2, T_3 in Figure 4. Two candidate causal links can be applied between tasks T_1 and T_2 i.e., $cl_{1,2}^{1}, cl_{1,2}^{2}$. Since only one candidate between two tasks will be selected, we have $y_{1,2}^{1}+y_{1,2}^{2}=1$. We have $y_{2,3}^{1}+y_{2,3}^{2}=1$ for $cl_{2,3}^{A}$.

Incompatibility Constraint. Since the selection of a candidate causal link $cl_{i,j}^{k}$ for $cl_{i,j}^{A}$ enforces a specific service for both tasks T_i (e.g., s_i) and T_j (e.g., s_j), the number of candidate links concretizing its closest abstract links $cl_{\alpha,i}^{A}$ and $cl_{j,\beta}^{A}$ is highly reduced. Indeed the candidate links for $cl_{j,\beta}^{A}$ ($cl_{\alpha,i}^{A}$) have to use only input (output) parameters of s_j (s_i). Thus, a constraint (7) for each pair of incompatible candidate links $(cl_{i,j}^{k}, cl_{j,\beta}^{l})$ is required in our IP problem.

$$y_{i,j}^{k}+y_{j,\beta}^{l}\leq 1,\ \forall cl_{i,j}^{A}\ \forall cl_{j,\beta}^{A} \tag{7}$$

Example 6. (Incompatibility Constraint)
Suppose the composition in Figure 4. According to (7), the incompatibility constraints are i) $y_{1,2}^{1}+y_{2,3}^{2}\leq 1$, ii) $y_{1,2}^{2}+y_{2,3}^{1}\leq 1$. Indeed $(cl_{1,2}^{1}, cl_{2,3}^{2})$, $(cl_{1,2}^{2}, cl_{2,3}^{1})$ are pairs of incompatible candidate links since task T_2 cannot be performed by two distinct services s_a and s_b.

Besides (6), (7), IP constraints on the quality criteria of the whole abstract composition are required. Here, we focus on the sequential, AND-Branching compositions, but a similar formalization for OR-Branching compositions and a fortiori their combinations is required.

Robustness Constraint. Let $r_{i,j}^{k}$ be a function of (i,j,k) representing the robustness quality of a causal link $cl_{i,j}^{k}$. Constraint (8) is required to capture the robustness quality of a causal link composition.

$$Q_r = \frac{1}{|cl_{i,j}^{A}|}\sum_{cl_{i,j}^{A}}\sum_{k=1}^{n} r_{i,j}^{k}.y_{i,j}^{k} \tag{8}$$

An additional constraint (9) can be used to constrain the robustness quality of the executable composition to not be lower than L.

$$\frac{1}{|cl_{i,j}^{A}|}\sum_{cl_{i,j}^{A}}\sum_{k=1}^{n} r_{i,j}^{k}.y_{i,j}^{k}\geq L,\ L\in[0,1] \tag{9}$$

Common Description Rate Constraint. Let $cd_{i,j}^{k}$ be a function of (i,j,k) representing the Common Description rate of a link $cl_{i,j}^{k}$. Its constraint is defined as (8), (9) by replacing Q_r by Q_{cd}, $r_{i,j}^{k}$ by $cd_{i,j}^{k}$.

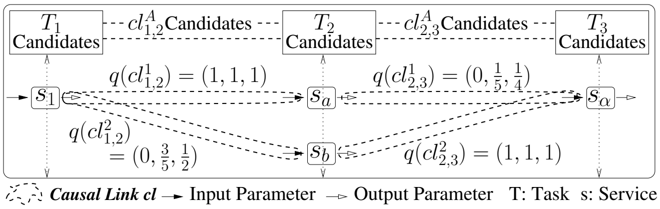

Figure 4. Tasks, Candidate Services & Causal Links.

Matching Quality Constraint. Among the criteria used to select causal links, the Matching quality is associated with a nonlinear aggregation function (see Table 1). A transformation in a linear function is then required to capture it in the IP problem. Assume $m_{i,j}^{k}$ be a function of (i,j,k) representing the Matching quality of causal link $cl_{i,j}^{k}$. The overall Matching quality of the executable composition is:

$$Q_m = \prod_{cl_{i,j}^{A}}\Big(\prod_{k=1}^{n}(m_{i,j}^{k})^{y_{i,j}^{k}}\Big) \tag{10}$$

The Matching quality constraints can be linearised by applying the logarithm function ln. Equation (10) then becomes:

$$\ln(Q_m) = \sum_{cl_{i,j}^{A}}\left(\sum_{k=1}^{n}\ln(m_{i,j}^{k}).y_{i,j}^{k}\right) \tag{11}$$

since $\sum_{k=1}^{n} y_{i,j}^{k} = 1$ and $y_{i,j}^{k} = 1$ or 0 for each causal link $cl_{i,j}^{k}$. $\ln(Q_m)$ is formalized to capture the Matching quality in our work.

Changing a nonlinear constraint in its linear form requires also to linearise the objective function. Thus, (12) is replaced by (13) in (4).

$$\frac{Q_m^{\lambda}-Q_m^{\min}}{Q_m^{max}-Q_m^{\min}} \quad (12) \qquad \frac{\ln(Q_m^{\lambda})-\ln(Q_m^{\min})}{\ln(Q_m^{max})-\ln(Q_m^{\min})} \quad (13)$$

Local Constraint. The IP problem can also include local selection and encompass local constraints. Such constraints can then predicate on properties of a single link and can be formally included in the model. In case a target causal link $cl_{i,j}^{A}$ requires its local robustness to be higher than a given value v, this constraint is defined by (14).

$$\sum_{k=1}^{n} r_{i,j}^{k}.y_{i,j}^{k} > v,\ v\in[0,1] \tag{14}$$

Local constraints are enforced during the causal links selection. Those which violate the local constraints are filtered from the list of candidate links, reducing the number of variables of the model.

The proposed method for translating the problem of selecting an optimal execution composition into an IP problem is generic and, although it has been illustrated with criteria introduced in Section 3, other semantic criteria to value causal links can be accommodated.

5 Computational Complexity & Experimentation

The optimization problem formulated in section 4 , which is equivalent to an IP problem, is NP-hard [17]. In case the number of abstract and candidate causal links is expected to be very high, finding the exact optimal solution to such a problem takes exponential run-time complexity in the worst case, so no practical. However our approach scales well by running a heuristic based IP solver wherein hundreds of abstract and candidate causal links are involved. This is a suitable upper bound for practicable industrial applications.

We conducted experiments on an Intel(R) Core(TM)2 CPU, 1.86GHz with 512 RAM. Compositions with up to 500 abstract causal links and 100 candidates for each abstract link have been considered. In our experiments we assumed that robustness, common description rate and matching quality of each causal link have been inferred in a pre-processing step of semantic reasoning. From these, the IP model formulation is computed, and the optimization problem is solved by running CPLEX, a state of the art integer linear programming solver based on the branch and cut technique [6][21].

The experimentation (Figure 5) aimed at comparing the global selection based approach by IP with the local optimization and naive global selection (i.e., exhaustive search). We measured the computation cost (in ms) of selecting causal links to create an optimal executable composition under the three different selection approaches.

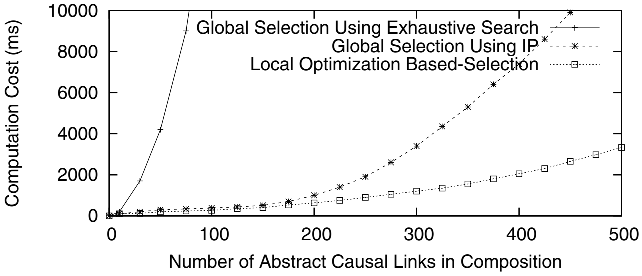

Figure 5. Number of Abstract Causal Links vs. Computation Cost for Optimal Executable Composition. (100 candidates for each causal links).

The computation cost of global selection by exhaustive search is very high even in very small scale in aspect of the number of abstract causal links and their candidates. Although the computation cost of global selection by IP is higher than that of local optimization, it is still acceptable. Finding the optimal solution to the optimization problem takes 10 seconds for a composition of 450 abstract causal links with 100 candidate links (i.e., 10 candidate services by task).

In case of higher number of links, the problem can be, for instance, divided in several *global selection problems*. Alternatively, suboptimal solutions satisfying revisited quality thresholds can be sufficient.

6 Related Work

Despite considerable work in the area of service composition, few efforts have specifically addressed optimization in 'causal link'-based service composition. Even if [13] introduce validity and robustness in causal link composition, no quality model is explicitly supported. In addition, the most valid and robust compositions are only addressed in their future work. In contrast, we present a model with various types of quality criteria used for optimizing the composition.

Unlike our work that considers quality of causal links, [23, 2] focused on QoS-aware service composition. To this end, they suggest a QoS-driven approach to select candidate services valued by non functional criteria such as price, execution time, and reliability. In the same way as our approach, they consider their problem as an optimization problem. Towards this issue different strategies as optimization techniques can be adopted, e.g., Integer Programming [23], Genetic Algorithms (GAs) [8], or Constraint Programming [11]. As discussed in [8], GAs better handle non-linearity of aggregation functions, and better scale up when the number of candidate services for each abstract service is high. In IP based approaches all quality criteria are used for specifying both constraints and objective function. In contrast to our problem the incompatibility constraints are not required since they assume independence between the services of any task. The global selection problem is also modelled as a knapsack problem [22], wherein [3] performed dynamic programming to solve the problem. Unfortunately all the previous QoS-aware service composition approaches consider only causal links valued by an Exact match. The causal link quality is then disregarded by these approach.

7 Conclusion and Future Work

In this work we study causal links based semantic web service composition. Our approach has been directed to meet the main challenge facing this problem i.e., how effectively retrieve optimal compositions of causal links. To this end we have first presented a general and extensible model to evaluate quality of both elementary and composition of causal links. Since the global causal link selection is formalized as an optimization problem, IP techniques are used to compute optimal executable composition of services. Our global selection based approach is not only more suitable than the local approach but also outperforms the naive approach. Moreover the experimental results show an acceptable computation cost of the IP-based global selection for a high number of abstract and candidates causal links.

Since several executable compositions maximizing the overall quality of causal links may be retrieved, the main direction for future work is to consider optimality for quality of service (driven by empirical analysis of compositions usage) to further optimize them.

REFERENCES

[1] Anupriya Ankolenkar, Massimo Paolucci, Naveen Srinivasan, and Katia Sycara, 'The owl-s coalition, owl-s 1.1', Technical report, (2004).
[2] Danilo Ardagna and Barbara Pernici, 'Adaptive service composition in flexible processes', *IEEE Trans. Software Eng.*, **33**(6), 369–384, (2007).
[3] Ismailcem Budak Arpinar, Ruoyan Zhang, Boanerges Aleman-Meza, and Angela Maduko, 'Ontology-driven web services composition platform', *Inf. Syst. E-Business Management*, **3**(2), 175–199, (2005).
[4] F. Baader and W. Nutt, in *The Description Logic Handbook: Theory, Implementation, and Applications*, (2003).
[5] Rainer Berbner, Michael Spahn, Nicolas Repp, Oliver Heckmann, and Ralf Steinmetz, 'Heuristics for qos-aware web service composition', in *ICWS*, pp. 72–82, (2006).
[6] Tim Berners-Lee, James Hendler, and Ora Lassila, 'The semantic web', *Scientific American*, **284**(5), 34–43, (2001).
[7] S. Brandt, R. Kusters, and A. Turhan, 'Approximation and difference in description logics', in *KR*, pp. 203–214, (2002).
[8] Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, and Maria Luisa Villani, 'An approach for qos-aware service composition based on genetic algorithms', in *GECCO*, pp. 1069–1075, (2005).
[9] Simona Colucci, Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, and Marina Mongiello, 'Concept abduction and contraction in description logics', in *DL*, (2003).
[10] David Harel and Amnon Naamad, 'The statemate semantics of statecharts', *ACM Trans. Softw. Eng. Methodol.*, **5**(4), 293–333, (1996).
[11] Ahlem Ben Hassine, Shigeo Matsubara, and Toru Ishida, 'A constraint-based approach to horizontal web service composition', in *ISWC*, pp. 130–143, (2006).
[12] Ralf Küsters, *Non-Standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Computer Science*, Springer, 2001.
[13] Freddy Lécué and Alexandre Delteil, 'Making the difference in semantic web service composition.', in *AAAI*, pp. 1383–1388, (2007).
[14] Freddy Lécué and Alain Léger, 'A formal model for semantic web service composition', in *ISWC*, pp. 385–398, (2006).
[15] L. Li and I. Horrocks, 'A software framework for matchmaking based on semantic web technology', in *WWW*, pp. 331–339, (2003).
[16] M. Paolucci, T. Kawamura, T.R. Payne, and K. Sycara, 'Semantic matching of web services capabilities', in *ISWC*, pp. 333–347, (2002).
[17] Christos H. Papadimitriou, 'On the complexity of integer programming', *J. ACM*, **28**(4), 765–768, (1981).
[18] K. Sivashanmugam, K. Verma, A. Sheth, and J. Miller, 'Adding semantics to web services standards', in *ICWS*, pp. 395–401, (2003).
[19] Michael K. Smith, Chris Welty, and Deborah L. McGuinness, 'Owl web ontology language guide', W3c recommendation, W3C, (2004).
[20] Gunnar Teege, 'Making the difference: A subtraction operation for description logics', in *KR*, pp. 540–550, (1994).
[21] L. Wolsey, *Integer Programming*, John Wiley and Sons, 1998.
[22] Tao Yu, 'Service selection algorithms for composing complex services with multiple qos constraints', in *ICSOC*, pp. 130–143, (2005).
[23] Liangzhao Zeng, Boualem Benatallah, Marlon Dumas, Jayant Kalagnanam, and Quan Z. Sheng, 'Quality driven web services composition', in *WWW*, pp. 411–421, (2003).

---

[6] LINDO API version 5.0, Lindo Systems Inc. http://www.lindo.com/