A survey using constraints to decision-making for fault tolerance in Business processes

M. T. Gómez-López, Rafael Martínez Gasca and Carlos Arévalo Maldonado
E.T.S de Ingeniería Informática. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Spain
{maytegomez,gasca,carlosarevalo}@us.es

Abstract

Sometimes the business processes do not work how it is expected. In these cases, a diagnosis process has to be executed to determine the responsible activity or activities of the fault in order to substitute it or them for a correct activity. The aim of this paper is describe the necessary steps to find out another service that can replace it in an efficient way. In order to automate the search and substitution of activities, we propose to describe the functionality of the tasks using constraints, making easier the determination of the possible activities that could substitute everyone faulty activities in the business process. In this paper, it is also analyzed how to adapt the communication protocol with XML messages to a behavior described using constraints.

Keywords: *Business Process, Fault tolerance, Constraint Satisfaction Problem.*

1. Introduction

A business process consists of a set of activities that are performed in coordination in an organizational and technical environment [1]. While a Business Process Management (BPM) includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business process. The base of BPM is the explicit representation of business processes with their activities and the execution constraints between them. In order to describe the workflow of the process, it is possible to use constraints to model de conditions that define the instance executed for each example, but it is also possible to describe the behavior/semantics of the activities using constraints.

A business process instance represents a certain case in the operational business for a model. Each business process model acts as a blueprint for a set of business process instances.

When an execution of a business process is monitored, some errors could be detected. The diagnosis process is used to determine which activity or activities are responsible of the incorrect behavior in the full business process. In this paper, we are centered in the business processes that fail for certain instances, or its behavior does not correspond with the expected.

Fault diagnosis and diagnosability have been analyzed in previous works [2], since the transformation and adaptation of Artificial Intelligence techniques used in classic diagnosis [3][4] to process business management is not automatic. One of the most important problems in diagnosis of business process is derived from the distribution of the model and that there is no a global knowledge of the system behavior.

Related to BPM diagnosis, there are works related to conflicts detection (CDM - Conflict Detecting Mechanism), where the specification of the services is described using XML to design meta-process [5]. The mentioned work is based on the detection of inconsistent activities. In CDM area, the conflicts are defined as a cause that violate and change the normal behavior or expected in an execution state for a business process. In order to detect errors, the business process area also uses the processes monitoring (BPMod-Business Process Monitoring) that is the query languages proposed in [6].

When a business process is formed by web services (WS), BPEL (Business Process Execution Language [7]) standard is very used. BPEL offers a language based on XML, it describes the process interface, its logic operations and its execution workflow. This language has also been used in fault tolerance area for web services [8].

In this work, we will consider that all the activities of a business process are executed by means of WS. When a fault is produced in the execution of a business process it is necessary to isolate the WS which is cause of this fault. Once the responsible WS of the malfunction is determined, this WS has to be substituted. The new WS has to satisfy the same precondition and postcondition of the original, it means to satisfy the same contract. There are works where weakest preconditions are analyzed, where predicates area transformed into another [29][30][31], to guaranty a safe substitution. It is also important that the substitution has to be done in an efficient way. In order to automate this process, it is necessary to know the formal specification of each WS (expressed as a conjunction or disjunction of constraints) and to have a repository with the possible substitutive web services. The use of constraints makes the decision making techniques more efficient to search a substitute service, since there are algorithms [9][10] to optimize the identification of constraints by means of queries to a Constraint Database (CDB) [21].

The search of a new WS, or a set of them, in order to replace an incorrect service imply three problems:

- Representation of the WS functionality by means of a specification language same to a constraint language.
- Storing the semantic description of the WS in a database to make this information persistent and easier an efficient search.
- Efficient search of the best substitute WS, or the combination of some of them, to replace the incorrect WS.

In this paper, all these aspects are analyzed in order to propose a methodology to help for the decision making in the business process orchestration for fault tolerance. Section 2 revises standard languages for WS semantic specification. Section 3 introduces how it is possible to represent WS contracts by means of constraints. Section 4 revises the definition of CDBs to store web services. Section 5 presents how selection operator can be used for an efficient search of the substitution of web services. Finally, in Section 6 conclusion and future works are presented.

2. Standard languages for web services semantic specification

Here we make a compilation of the main standards for web services semantics specification and we emphasized its main characteristics.

The industry standard Universal Description, Discovery and Integration (UDDI) [32] only supports classification (keyword) based search and does not capture the semantics of web services functionality. To overcome this, some works have already been initiated towards a semantic description specification for web services, especially, the Web Ontology Language for Services (OWL-S) [33] specification. The OWL-S specification allows a WS to be specified in terms of its Inputs, Outputs, Pre-conditions, and Effects or Post-conditions (Pre-conditions and Post-conditions can be generalized as constraints, that is, characteristics of a service which need to be considered for successful execution of the WS). Another way to describe web services semantics is the Web Service Modeling Ontology (WSMO) [39] where precondition, postcondition, assumption and effects can be defined, hence it has a more complete semantics than OWL-S.

The two major web standardization bodies are W3C and OASIS. The Web Service Description Language (WSDL) [34] [35] is a standard technology for describing the interface exposed by a service. In WSDL, contracts are basically limited to one-way (asynchronous) and request/response (synchronous) interactions.

The Web Service Conversation Language (WSCL) [36] extends WSDL contracts by allowing the description of arbitrary, possibly cyclic sequences of exchanged messages between communicating parties. Other languages, such as the Web Service Business Execution Language (WS-BPEL) [37], provide even more detailed descriptions of services by defining the subprocess structure and more specific details regarding the service's internals. Such descriptions can be approximated and compared in terms of contracts.

Standard technologies are also available for building repositories of web service descriptions [38], making it possible to perform queries for services according to their contract.

The WSDL message exchange patterns cover only the simplest forms of interaction between a client and a service. More involved forms of interactions cannot be captured if not as informal annotations within the WSDL interface. The Web service conversation language WSCL [36] provides a more general specification language for describing complex conversations between two communicating parties, by means of an activity diagram.

3. Representing web services by means of Constraints

The term '*Constraint'* is used in this paper as a relation between a set of variables defined over a domain [22] and related between them, where there is a limitation over the values for the instantiation of the variables. This relation between the variables can be described in a compact way using combination of equations and inequations by means of Boolean operators. The standard BPEL permits to describe both the process interface and logic operations and the workflow. The notation described by BPEL is based on the specific behavior of WS, represented in the bibliography as BPEL4WS. The processes in WPEL4WS exports and imports functionality using only the web service interfaces. The business processes can be described in two ways [7]:

- Executable business processes model actual behavior of a participant in a business interaction.
- Business protocols, in contrast, use process descriptions that specify the mutually visible message exchange behavior of each of the parties involved

> in the protocol, without revealing their internal behavior. The process descriptions for business protocols are called abstract processes.

BPEL4WS is meant to be used to model the behavior of both executable and abstract processes. BPEL4WS provides a language for the formal specification of business processes and business interaction protocols. By doing so, it extends the Web Services interaction model and enables it to support business transactions.

The following basic specifications originally defined the Web Services space: SOAP, Web Services Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI). SOAP defines an XML messaging protocol for basic service interoperability. WSDL introduces a common grammar for describing services. UDDI provides the infrastructure required to publish and discover web services in a systematic way. Together, these specifications allow applications to find each other and interact following a loosely coupled, platform independent model.

The implementation M of a WS to develop a process has to satisfy a contract C. The contract can be described by means of a precondition and a postcondition represented with constraints. The use of constraints permits to check the correct behavior of the WS. It is also important that the WS have defined variables to be connected with other web services. There are variables are public variables associated with ports of connection, and private variables unknown outside the WS. If M satisfies contract C, and it is defined over the same public variables, it implies that $M \subseteq C$. In order to substitute M for another WS, it is necessary to find another WS implemented by M', defined over the same public variables and that satisfy the assert $M' \subseteq C$.

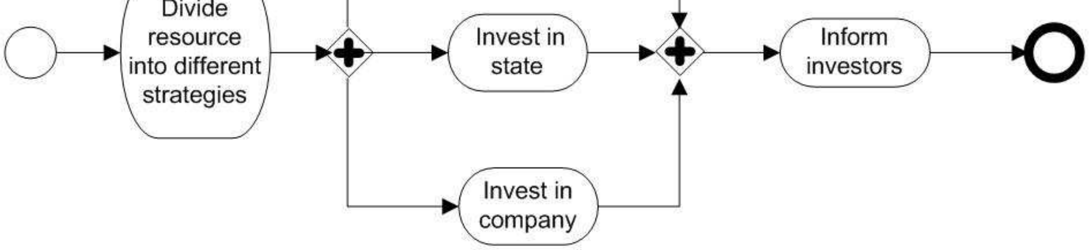
Figure 1 shows an example where the web services are represented using constraints. This example uses a set ~~con~~ of web services that represent an orchestration of activities to carry out some investments in order to maximize the profits for a given quantity. For a determined quantity, the activity named *Divide resource into different strategies* is used to divide the money into different items. The different strategies are represented by means of the following activities: *Invest in stock market*, *Invest in state*, and *Invest in sector company of research and develop.* Each of these web services obtains a profit in function of the invested quantity and the inversion margin establish. It means that each service will receive a pair of values <investment, margin> from the *Divide resource into different strategies* service, thereby the total investment en each item will be investment±margin. As output variable of the web services, the totalProfit will be obtained.

It is possible that after a diagnosis process the service *Invest in stock market* is not working as it is expected, and it has to be substituted for a new one with an equivalent or compatible specification. In order to do that it is necessary to have the description of the web services, being possible the substitution. An example of the description for the task *Invest in stock market* is the following constraint:

$$N_1 * PP_1 + N_2 * PP_2 + \ldots + N_m * PP_m = \text{totalProfit}$$

$$\wedge$$

$$SP_1 * N_1 + SP_2 * N_2 + \ldots + SP_m * N_m \leq \text{investment} + \text{margin}$$



$C_{m,}$ rep
Percen
an spe
are rep

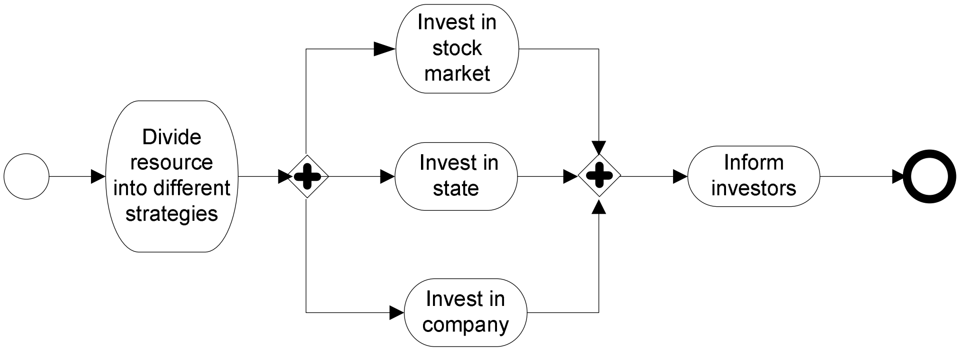


Figure 1: Example of business process

For *Invest in stock market,* both preconditions and postconditions can be described using constraints.

Due to the interchange of information between the web services is carried out using a message protocol, the information sent will define what variables are known between the processes and which variables are private. For the example of "*Invest in stock market*", there are two types of messages, one to receive the information and another to send the results. For the example, the variables are investment, margin and totalProfit, and the messages are:

```
<message name=”StockInvestmentData”>
        <part name=”investment” type =”xsd:integer”/>
        <part name=”magin” type=”xsd:integer”/>
</message>

<message name=”StockInvestmentOutput”>
        <part name=”totalProfit” type=”xsd:integer”/>
</message>
```

4. Storing web services specifications in databases

When a system combines different web services in a business process to obtain a goal, it is useful that each phase of the process works with a set of data of different nature and stored in different databases. As it is analyzed in the previous section, the behavior of the web services can be mapped to constraints, thereby they can be stored in a natural way in a Constraint Database (CDB).

Constraint Databases (CDB) were initially developed in 1990 with a paper by Kanellakis, Kuper and Revesz [11], and were extended through research on a query language [12][13] which is a subset of Prolog from the syntactical point of view (Datalog[23]) and through Constraint Logic Programming (CLP) [24][25]. These paradigms were used to define the first Constraint Databases, and created a new research area [26][27]. The basic idea behind the CDB model is to generalize the notion of a tuple in a relational database to a conjunction of constraints, since a tuple in relational algebra can be represented as an equality constraint between an attribute of the database and a constant value. CDBs permit represent a set of data by means of a constraint (formula) over a set of free variables $x_1, \ldots, x_m$, at the same time that a set of classic attributes in the relational algebra $a_1, \ldots, a_n$ are used. An extension of classic CDBs based in these ideas is presented in [10]. The new definition of CDB proposed is based on:

- A constraint k-tuple with the variables $x_1, \ldots, x_k$, over the vocabulary - is a finite conjunction $\varphi_1 \wedge \ldots \wedge \varphi_N$ where each φ_i, for $1 \leq i \leq N$, is either a constraint such that $\{x_j = \text{Constant}\}$, where $x_j \in \{ x_1, \ldots, x_k \}$, called Classic Attribute, or an Ω-constraint over the variables $x_1, \ldots, x_k$ which do not correspond to a classic attribute, called constraint attribute.
- A constraint relation is defined as a finite set of Classic Attributes and Constraint Attributes. A constraint relation of arity k, is a finite set $r = \{\psi_1, \ldots \psi_M\}$, where each ψ_j for $1 \leq j \leq M$ is a constraint k-tuple over $x_1, \ldots, x_k$. The corresponding formula is the disjunction $\psi_1 \vee \ldots \vee \psi_M$, such that $\psi_j = \varphi_1 \wedge \ldots \wedge \varphi_N$ for each φ_i is a constraint k-tuple, where $1 \leq i \leq N$. If in each $\psi_j \in r$ there is a φ_i such that$\{x = \text{Constant}\}$, where x is the same variable in all φ_i belonging to different ψ_j, and x does not appear in the rest of the φ_i of the same ψ_j, then the x variable is a classic attribute, while the rest of the variables belong to constraint attributes. In Figure 2 the equivalence between a constraint relation and a constraint k-tuple is presented.
- Therefore, a Constraint Database is a finite collection of constraint relations composed of Classic and Constraint Attributes.

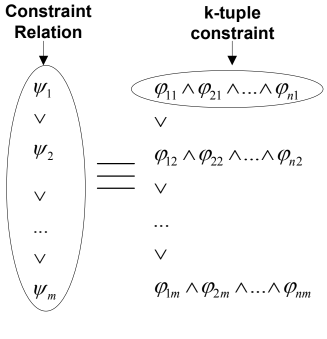

Constraint Relation
k-tuple constraint
ψ_1
$\vee$
ψ_2
$\vee$
...
$\vee$
ψ_m
$\equiv$
$\varphi_{11} \wedge \varphi_{21} \wedge \ldots \wedge \varphi_{n1}$
$\vee$
$\varphi_{12} \wedge \varphi_{22} \wedge \ldots \wedge \varphi_{n2}$
$\vee$
...
$\vee$
$\varphi_{1m} \wedge \varphi_{2m} \wedge \ldots \wedge \varphi_{nm}$

Figure 2: Representing constraint k-tuples and constraint relations

The key idea is to store all the specifications and any another type of information (such as URI, SLA,…) of the web services in a CDB by means of their contract (precondition and postcondition). When a WS is detected as incorrect, a new WS with an equivalent or compatible specification will be selected to substitute it. Having all this information stored, the business process could be fault tolerance systems, since an incorrect WS can be substituted in an automatic way.

The definition of CDB used in this paper permits to describe and store the behavior of web services as constraint attributes, and the public and private port of the web services as classic attributes. An example of representation of web services for Figure 1 is shown in Figure 3. In this example there are two classic attributes, the identification of the web services (IDWS) and the textual description of the service (Description). In the other hand, there are two constraint attributes to represent the pre and postconditions.

In order to connect the different web services, the public variables are used. Since the behavior of the process is represented using constraints, it is possible to execute project operation over the constraints. The projection operation [28] permits to transform the precondition and postcondition depending on what of them are the public or private variables. It is also necessary to take into account that to connect two web services, they have to share some public variables.

IDSW	Precondition	Post-condition	Description
2354	margin≥0∧investment≥0	*N*1**PP*1 + *N*2**PP*2 +...*Nn***PPn* = totalProfit ∧..	Invest in stock...
3487	1.000.000≥investment∧investment≥120.000	investment*1,8-2.500=totalProfit	Invest in state
5561	18.000≥investment∧investment ≥0 ∧700≥fixExpense≥50	investment*1,4-fixExpense=totalProfit	Invest in companies
...	...	...	...

Figure 3: Example of tuples for Web Services

In order to optimize the CDB queries, when a CDB is created, three auxiliary tables are also created (Constraints, Variables and Constraints/Variables) which relate each constraint with its variables. These tables, shown in Figure 3 and presented in [28], improve the computation time for obtaining the constraints related to the variables of a projection. These tables allow the identification of each constraint (Constraints table), each variable (Variables table) and the establishment of the relations between constraints and variables (Constraints/Variables table), thereby making it unnecessary to study all the constraints for a query. The table Constraints/Variables can also store the minimum and maximum value of each variable for every constraint. These tables are not accessible to the users, however these tables implicitly change according to the constraints added to or removed from the CDB. The table Constraints stores the idConstraint, which is the object identification (OID), generated by the system, and the Label according to the type of constraint, in order to decide which technique will be used to handle the constraint. The table Variables stores the names of the variables, their identification and their type.

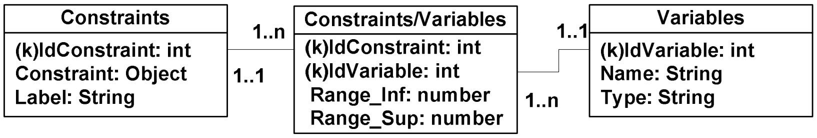


Figure 4. Database tables to index constraints and variables

5. Efficient search for substitute Web Services

There are works [14][15] that are centered in web services queries. In this work we are centered in search new services in order to substitute one. In order to do that, it is important to bear in mind that the implementation M of a WS satisfies a contract C, if it satisfies the postcondition for a known precondition.

Dijkstra's weakest precondition calculus [29] (based on Hoare's Logic) is based on:

A formula A is weaker than a formula B if B ⇒ A

Being WS_A(Pre) and WS_A(Post) the precondition and postcondition of the WS A belong to a business process. In order to obtain a WS B that can be the substitute to WS A, it is necessary that the specification of the WS B satisfies the following asserts:

- WS_b(Pre) □ WS_a(Pre). That means that all the possible input values for WS A are possible for WS B.
- WS_b(Post) $\subseteq$ WS_a(Post). That means that all the output values of WS B are equivalent or are included in WS A.

Our proposal defines the implementation of this decision making by means of the construction of a Constraint Satisfaction Problem (CSP), in our concrete proposal we will use the selection operator of CDB. CSPs represent a reasoning framework consisting of variables, domains, and constraints. Formally, it is defined as a triple <X, D, C> where X = {x_1, x_2, …, x_n} is a finite set of variables, D = {d(x_1), d(x_2), …, d(x_n)} is a set of domains of the values of the variables, and C = {C_1, C_2, …, C_m} is a set of constraints. Each constraint C_i is defined as a relation R on a subset of variables V = {x_i, x_{j+1}, …, x_k}, called the constraint scope. The relation R may be represented as a subset of the Cartesian product d(x_i) × d(x_j) × … × d(xk). A constraint Ci = (X_i, R_i) specifies the possible values of the variables in X_i simultaneously in order to satisfy R. Let X_i = {x_{i1}, x_{i2}, …, x_{il}} be a subset of X. An l-tuple (x_{i1}, x_{i2}, …, x_{il}) from d(x_{i1}), d(x_{i2}), …, d(x_{il}) is called an instantiation of the variables in X_i. An instantiation of all the variables in X is a solution.

In order to make the search more efficient, the implementation of CDB presented in [16] is used. This proposal achieves its efficiency in to avoid the construction of certain CSPs when it is possible. In order to do that, an analysis of the ranges the variables is carried on. The ranges of the variables are stored as it is explained in previous section. There are cases where the analysis of the range of variables avoids the construction of a CSP. Analyzing the maximum of minimum value of a variable, it is possible to infer if the constraints related in a comparison can satisfy a condition. For example, being Cx and Cy two constraints that define the precondition of two web services, both constraints defined over the variables v1 and v2, where the ranges for the variables for each constraint are Cx(v1:[1..15], v2[20..30]) and Cy(v1:[20..25], v2:[40..55]). In the example, the predicate Cx $\subseteq$ Cy is false, and the web services represented by Cy cannot be a substitute of Cx. Using again the previous example, if the ranges of the variables would be Cx(v1:[5..15], v2[20..30]) and Cy(v1:[2..25], v2:[10..55]), an analysis of the range is not enough to know the evaluation of the predicate Cx $\subseteq$ Cy. Although the domain of the variables in Cx are included in the domain for the same variables in Cy, it is no possible ensure that Cx $\subseteq$ Cy. An example is shown in Figure 5.

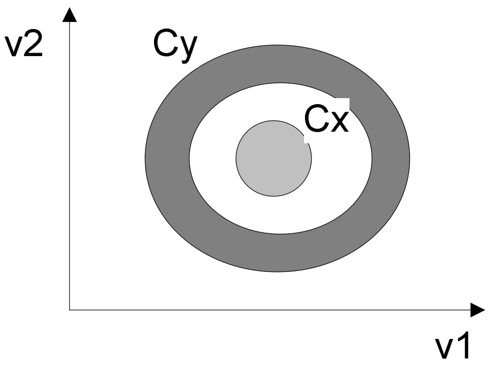


Figure 5: Example where Cx $\not\subset$ Cy

For the range of all the shared public variables of the precondition and postcondition, there are different relations between them (shown in Figure 6). In function of each case, a CSP have to be created or not. The cases are:

a. Cx $\subseteq$ Cy is false
b. Cx $\subseteq$ Cy is false
c. Cx $\subseteq$ Cy is false
d. Cx $\subseteq$ Cy. It is necessary to create and solve a CSP to know the evaluation of the predicate
e. Cx $\subseteq$ Cy. It is necessary to create and solve a CSP to know the evaluation of the predicate

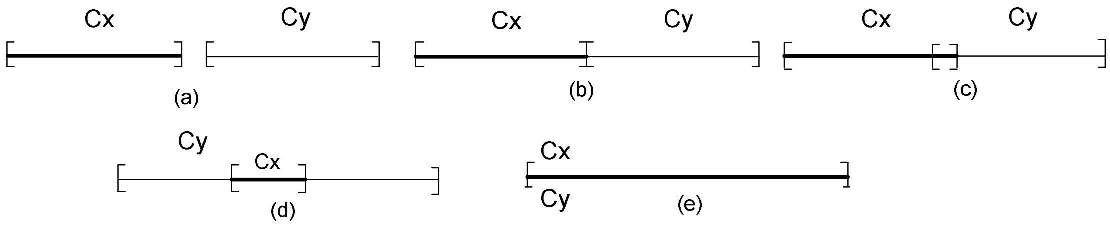


Figure 6. Types of relations between ranges of variables

The creation of the CSPs is based on to determine if all the solutions of Cx are also solutions of Cy. In order to analyze the inclusion operator in constraints, both constraints have to be defined over the same variables, then being Cx and Cy two constraints where $X = \{x_1,x_2, \ldots, x_n\}$ are the variables of Cx and Cy, Cx $\subseteq$ Cy is equal than the implication (Cx $\rightarrow$ Cy) [17]. This comparison determines if all the solutions of Cx are also solutions of Cy, although it is possible that Cy has solutions that do not belong to Cx. The use of the selection operator permits to find out a new WS where the precondition is weaker and the postcondition is stronger.

In order to avoid analyzing all the solutions of Cx, checking if they are solutions of Cy, the CSP is created to look for solutions where the constraint is not satisfiable. It means that the evaluation of the conditional predicate Cx $\subseteq$ Cy corresponds to the formula:

$$\neg (\exists_{Xi \in X}(Cx \wedge \neg Cy))$$

And the CSP is:

$$X = \{x_1, x_2, \ldots, x_n\}$$
$$Cx \wedge \neg Cy$$

If any solution is found for the CSP, the evaluation returns *false*, and true if no solution is found.

Having all the contracts of the web services stored in a database, the decision making process is easier. It is not only efficient, but also we could obtain a better WS.

When it is not possible to find an adequate WS, it will be necessary combine a set of services in order to define a new one. For the composition of services, there are previous related works [18][19][20], but it is not the objective of this work.

6. Conclusions and Future work

This paper presents the necessary steps for an efficient search of substitute services. The found services can substitute to an incorrect service if its precondition is weaker and the postcondition is more restrictive that the original. The necessary steps are:

- Representing web services by means constraints
- Storing web services specification in a CDB
- Defining an efficient search for finding out substitute web services

Finally, it is necessary to highlight that the previous steps when we use the CDBs permits, by means of a set of tables of indexation, to evaluate the selection operation in a efficient way in order to obtain the more adequate substitute WS for a faulty one.

As future work we propose to enlarge the analysis of combination of services that can be studied with the CDBs projection operator. It will be necessary when there are no web services in the CDB whose behavior can replace a web service and then a combination of a set of web services is necessary.

7. Acknowledge

This work has been partially funded by Junta de Andalucía by means la Consejería de Innovación, Ciencia y Empresa (P08-TIC-04095) and by the Ministry of Science and Technology of Spain (TIN2009-13714) and the European Regional Development Fund (ERDF/FEDER).

8. References

[1] Mathias Weske, “Business Process Management. Concepts, Languages, Architectures”, Springer, Berlin, 2007.

[2] Stefano Bocconi, Claudia Picardi, Xavier Pucel, Daniele Theseider Dupré, and Louise Travé-Massuyès. Model-based diagnosability analysis for web services. In AI*IA ’07: Proceedings of the 10th Congress of the Italian Association for Artificial Intelligence on AI*IA 2007, pages 24–35, Berlin, Heidelberg, 2007. Springer-Verlag.

[3] R. Reiter. A theory of diagnosis from first principles. volume 1, pages 57–96, 1987.

[4] R. Greiner, B. A. Smith, and R. W. Wilkerson. A correction to the algorithm in reiter’s theory of diagnosis. volume 41, pages 79–88. Elsevier Science Publishers Ltd, 1989.

[5] Shi-Ming Huang, Yuang-Te Chu, Shing-Han Li, and David C. Yen. Enhancing conflict detecting mechanism for web services composition: A business process flow model transformation approach. Inf. Softw. Technol., 50(11):1069–1087, 2008.

[6] Catriel Beeri, Anat Eyal, Tova Milo, and Alon Pilberg. Monitoring business processes with queries. In VLDB ’07: Proceedings of the 33rd international conference on Very large data bases, pages 603–614. VLDB Endowment, 2007.

[7] Business Process Execution Language for Web Services. http://www.ibm.com/developerworks/library/ws-bpel/, 2003.

[8] Glen Dobson. Using ws-bpel to implement software fault tolerance for web services. In EUROMICRO ’06: Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications, pages 126–133, Washington, DC, USA, 2006. IEEE Computer Society.

[9] María Teresa Gómez-López, Rafael M. Gasca, Carmelo Del Valle, and F. Fernando de la Rosa T. Querying a polynomial constraint object-relational database in model-based diagnosis. In International Conference on Database and Expert Systems Applications-DEXA, volume 3588, pages 848–857. Lecture Notes in Computer Science, Springer, 2005.

[10] María Teresa Gómez-López, Rafael Ceballos, Rafael M. Gasca, and Carmelo Del Valle. Developing a labelled object-relational constraint database architecture for the projection operator. Data Knowl. Eng., 68(1):146–172, 2009.

[11] G. M. Kuper P. C. Kanellakis and P. Z. Revesz. Constraint query languages. Symposium on Principles of Database Systems, pages 299–313, 1990.

[12] G. Kuper, L. Libkin, and J. Paredaes. ”Introduction”, Constraint Databases, G. Kuper and L. Libkin and J. Paredaes. Springer, 2000.

[13] P. Revesz. Datalog and Constraints. Constraint Databases, pages 155–170, 2000.

[14] D. Martin S. Narayanan R.Waldinger G. Denker, J. Hobbs. Querying and accessing information on the semantic web. In Proc. Semantic Web Workshop, in conjunction with 10th international Worldwide Web Conference, 2001.

[15] Catriel Beeri, Anat Eyal, Simon Kamenkovich, and Tova Milo. Querying business processes. In VLDB ’06: Proceedings of the 32nd international conference on Very large data bases, pages 343–354. VLDB Endowment, 2006.

[16] María Teresa Gómez-López. Lorcdb: Gestor de bases de datos objeto-relacionales de restricciones. Ph D thesis, 2007.

[17] Kim Marriott and Peter J. Stuckey. ”Programming with Constraints. An introduction”, Simplification, Optimization and Implication. The Mitt Press, 1998.

[18] Singh M.P. Cheng, Z. and M.A. Vouk. Composition constraints for semantic web services. In Proceedings of WWW2002 Workshop on Real World RDF and Semantic Web Application, 2003.

[19] Srini Narayanan and Sheila A. McIlraith. Simulation, verification and automated composition of web services. In WWW ’02: Proceedings of the 11th international conference on World Wide Web, pages 77–88, New York, NY, USA, 2002. ACM.

[20] Rajesh Thiagarajan and Markus Stumptner. Service composition with consistencybased matchmaking: A csp-based approach. In ECOWS ’07: Proceedings of the Fifth European Conference on Web Services, pages 23–32, Washington, DC, USA, 2007. IEEE Computer Society.

[21] Peter Z. Revesz, Safe Query Languages for Constraint Databases, ACM Trans. Database Syst., volume 23, number 1, 1998, pages 58-99.

[22] R. Dechter. Constraint Processing (The Morgan Kaufmann Series in Artificial Intelligence). Morgan Kaufmann, May 2003.

[23] S. Ceri, G. Gottlob, and L. Tanca. What you always wanted to know about Datalog (and never dared to ask). volume 1, pages 146–166, Piscataway, NJ, USA, 1989. IEEE Educational Activities Department.

[24] J. Jaffar and J.-L. Lassez. Constraint Logic Programming. In POPL, pages 111–119, 1987.

[25] J. Jaffar and M. J. Maher. Constraint Logic Programming: A Survey. J. Log. Program., 19/20:503–581, 1994.

[26] P. Z. Revesz. Datalog Queries of Set Constraint Databases. In ICDT, pages 425–438, 1995.
[27] P. Z. Revesz. Safe query languages for constraint databases. ACM Trans. Database Syst., 23(1):58–99, 1998.
[28] María Teresa Gómez-López, Rafael Ceballos, Rafael M. Gasca and Carmelo Del Valle, Developing a labelled object-relational constraint database architecture for the projection operator, Data Knowledge Engineering, volume 68, number 1, pages 146-172, 2009.
[29] Dijkstra E. W. A Discipline of Programming. Prentice Hall international, 1976.
[30] Lamport, L. win and sin: predicate transformers for concurrency. ACM Transactions on Programming Languages and Systems 12, 3 (July 1990), 396–428
[31] Decai Jiao Sinclair, D. Weakest Preconditions for Applied pi-Calculus. Computing in the Global Information Technology, pages 27-33, 2007
[32] Universal Description, Discovery and Integration (UDDI) Specification. http://www.uddi.org
[33] Web Ontology Language for Services (OWL-S) Specification http://www.daml.org/services/owl-s/
[34] CHINNICI, R., HAAS, H., LEWIS, A.-A., MOREAU, J.-J., ORCHARD, D., AND WEERAWARANA, S. 2007a. Web services description language (WSDL) version 2.0 Part 2: Adjuncts. W3C Recommendation, http://www.w3.org/TR/wsdl20-adjuncts/
[35] CHINNICI, R., MOREAU, J.-J., RYMAN, A., AND WEERAWARANA, S. 2007b. Web services description language(WSDL) version 2.0 Part 1: Core language. W3C Recommendation, http://www.w3.org/TR/wsdl20/
[36] BANERJI, A., BARTOLINI, C., BERINGER, D., CHOPELLA, V., ET AL. 2002. Web services conversation language (WSCL) 1.0. W3C Note, http://www.w3.org/TR/2002/NOTE-wscl10-20020314.
[37]ALVES, A., ARKIN, A.,ASKARY, S., BARRETO, C., ET AL. 2007. Web Services Business Process Execution Language Version 2.0. OASIS Standard, http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html
[38] BELLWOOD, T., CAPELL, S., CLEMENT, L., COLGRAVE, J., ET AL. 2005. UDDI Version 3.0.2. OASIS Standard, http://uddi.org/pubs/uddi-v3.0.2-20041019.htm
[39] Matthias Klusch and Frank Kaufer. WSMO-MX: A hybrid Semantic Web service matchmaker, Web Intelligence and Agent Systems, IOS Press, number 1, volumn 7, pages: 23-42, 2009

Authors

M.Teresa Gómez-López received her degree in Computer Science from the University of Seville in 2001 and her PhD degree in 2007. She is currently a Ph.D. Lecturer in the Department of Languages and Computer Systems at the University of Seville, Spain. She worked as a researcher in the Spanish National Research Council. Her main research interests are Constraint Databases, Model-based Diagnosis and Diagnosis of Business Process Systems.

Rafael Martínez Gasca received his PhD in Computer Science from the University of Seville in 1998. He is Lecturer in the Department of Languages and Computer Systems at the University of Seville, Spain. He is in charge of the Quivir Research Group. His research with the Group is focused on intelligent and security technologies for information systems. In the previous topics, he has led four industrial projects in different R&D Program and he has published numerous articles in international conference proceedings and scientific journals.

Carlos Arévalo is an Industrial Engineer from the University of Seville. He is a Lecturer in the Languages and Computer Science Department of Seville University. His research area is oriented to business object model and web services semantics.