

Process Mediation for Semantic Web Services

Emilia Cimpian

Semantic Technology Institute Innsbruck
emilia.cimpian@sti2.at

1 Research Problem

It is a common scenario for the business environment that one process needs to *communicate* with another process, in order to fulfill its goal. For example, the simple action of paying a bill to a service provider can be seen as two processes that are communicating: one process is defined by the client, its own steps to be taken for paying the bill; the other one belongs to the service, the sequence of activities it performs in order to obtain the payment. If a bank is also involved (which is currently the case in most of this type of situations), we can even talk of three different processes performed by three different entities in order to obtain the final result: the bill had been paid.

The problem addressed by this thesis is how two (or more) processes can successfully interact in order to accomplish a common goal. The processes considered are semantically defined and any inputs and outputs of a process needs to also be represented using an ontology.

This thesis addresses the problem of solving heterogeneity mismatches between previously defined processes. The assumption made is that the processes should not adjust in order to match the processes they want to interact with, from various reasons. Either they are involved in more than one interaction, and this adjustment will damage those, or the business partner owning the processes simply does not want to change anything. In this case, the communication can be hampered even if all the data is available [2].

This thesis makes the distinction between two different heterogeneity problems: process model heterogeneity and communication heterogeneity. In the first case, the processes are incompatible, that is no automatic solution can be developed for overcoming the heterogeneity problem. In this situation the inputs of a business expert are needed, and the process mediator will have to provide semi-automatic support for the domain expert. In the second case the processes are compatible, the mismatch existing only on the message exchange level. In this case the process mediator can provide a completely automatic mediation solution. An example of mismatch that can not be automatically solved is when one process expects a message that the other one will never send. In this case the domain expert can select a third process that will generate that message, or manually create it.

The first step for process mediation is to determine the nature of the problem, if it can be solved automatically or not. The heterogeneity problems that can be automatically solved are called *solvable* (or *communication*) *mismatches*, while the ones that require domain expert interactions are called *unsolvable* (or *process model*) *mismatches* [1].

2 Related Work

Process mediation is still a poorly explored research field, in the context of Semantic Web Services. The existing work represents only visions of mediator systems able to resolve in a (semi-) automatic manner the processes heterogeneity problems, without presenting sufficient details about their architectural elements. Still, these visions represent the starting points and valuable references for the future concrete implementations.

Two integration tools, Contivo¹ and CrossWorlds² seemed to be the most advanced ones in this field.

Contivo is an integration framework which uses metadata representing messages organized by semantically defined relationships. One of its functionalities is that it is able to generate transform code based on the semantic of the relationships between data elements, and to use this code for transforming the exchange messages. However, Contivo is limited by the use of a purpose-built vocabulary and of pre-configured data models and formats.

CrossWorlds is an IBM integration tool, meant to facilitate the B2B collaboration through business processes integration. It may be used to implement various e-business models, including enhanced intranets (improving operational efficiency within a business enterprise), extranets (facilitating electronic trading between a business and its suppliers) and virtual enterprises (allowing enterprises to link to outsourced parts). The draw-backs of this approach is that different applications need to implement different collaboration and connection modules, in order to interact. As a consequence, the integration of a new application can be done only with additional effort.

3 Contributions

The main contribution of this thesis is *the development of a semantic process mediation solution*. This overall accomplishment consists of a number of smaller contributions:

1. Identification and formalization of a set of atomic problems that can be automatically solved by a mediator (solvable or communication mismatches), as well as identification of a set of problems that can not be automatically overcome (unsolvable or process model mismatches).
2. Development of a run-time process mediator able to address the solvable mismatches.
3. Development of a design-time process mediation for allowing the domain expert to accommodate for the unsolvable mismatches.
4. Development of a comprehensive architecture for process mediation.

Because of space constraints this extended summary of the thesis contains only details of how the formalization and of the algebra developed in the thesis.

3.1 Notations and Definition

The service mediator performs an automatic analyze of the two processes involved in a communication. The internal decisions taken inside any of the processes are not relevant

¹ <http://www.contivo.com/>

² <http://www.sars.ws/hl4/ibm-crossworlds.html>

in this case, the mediator operating on the level of messages sent and received during the actual communication. In this sense it can be considered that the mediator operates on one particular branch of each process involved in the communications. That is, if depending on one condition one of the processes can perform one activity or another, the run-time mediator sees only the result of evaluating that condition, only the activity that is performed.

Furthermore, in a semantic environment the messages are important only from the point of view of the semantic information they carry. This information consists of instances of concepts defined in an ontology used in the description of the process (in the process model). If the process description specifies that *message* M_1 contains instance I_1 of concept C_1 , the mediator understands this as M_1 consists of an instance of C_1 , or in other words *an instance of C_1 is being sent or received*. The previous two formulations are further simplified to *Message C_1* .

If a message M_1 consists of multiple instances of multiple concepts (C_1, C_2, \dots, C_n) it will be referred to as: *message C_1 and C_2 and ... and C_n* . This definition still holds if multiple instances of the same concept are part of the same process, in which case the message will refer to every one of these instances.

The notation used for denoting that message C_1 is to be sent by a process is $S(C)$, while a message that should be received by a process is represented by $R(C)$. For denoting that a process should be either sent or received the notation used is $A(C)$ (an action for handling the message C). If the message carries more than one instance, of types C_1, C_2, \dots, C_n , this is denoted by $A(C_1 + C_2 + \dots + C_n)$.

The order of messages is represented by using the symbol \rightarrow .

The message sequence of a process P is represented as $MS(P)$. If P exchanges n messages during a communication, then: $MS(P) = A(C_1) \rightarrow A(C_2) \rightarrow \dots \rightarrow A(C_n)$

For representing the communication between two processes P_1 and P_2 the notation $\frac{MS(P_1)}{MS(P_2)}$ is used.

The fractions for representing a communication can be decomposed in multiple fractions, respecting the messages sequences of the processes involved in the communication.

$$\begin{aligned} \text{If : } & MS(P_1) = MS_1(P_1) \rightarrow MS_2(P_1) \text{ and } MS(P_2) = MS_1(P_2) \rightarrow MS_2(P_2) \\ \text{then : } & \frac{MS(P_1)}{MS(P_2)} = \frac{MS_1(P_1)}{MS_1(P_2)} \rightarrow \frac{MS_2(P_1)}{MS_2(P_2)} \end{aligned}$$

Furthermore, the following terms are defined:

Definition 1. An *Atomic Send/Receive (Atomic S/R)* is considered to be that particular fragment of a communication consisting of one process sending a message and the other process receiving it.

$$\text{Atomic } S/R(C) = \frac{S(C)}{R(C)} \text{ or } \text{Atomic } S/R(C) = \frac{R(C)}{S(C)}$$

Definition 2. A *Projection* of a process, denoted by $\pi(P)$, is a derived process obtained from P as the result transformations performed by the run-time Process Mediator.

The communication between two processes is equivalent with the communication between one process and the projection of the other process, which is denoted by the symbol \approx .

$$\frac{MS(P_1)}{MS(P_2)} \approx \frac{MS(\pi(P_1))}{MS(P_2)}$$

Definition 3. *There is a **Match** between two given processes if the communication between them can be represented as a sequence of Atomic S/R.*

The notation used for denoting that two processes P_1 and P_2 match is:

$$Match(P_1, P_2)$$

Definition 4. *Two processes are considered to be **Compatible** if there is a Match between them or if every mismatch is at the message sequence level.*

The notation used for denoting that two processes P_1 and P_2 are compatible is:

$$Compatible(P_1, P_2)$$

Both *Match* and *Compatible* relationships are symmetric.

3.2 Process Mediation - Lemmas and Theorems

A set of lemmas can be defined for obtaining the projection of a process, given its message sequence. An example of such lemma is:

Lemma 1. *For a given process P where $MS(P) = MS_1(P) \rightarrow S(C) \rightarrow MS_2(P)$, a process P' such as $MS(P') = MS_1(P) \rightarrow MS_2(P)$ is a projection of P (i.e., $P' = \pi(P)$).*

A total of 8 lemmas are defined for governing the creation of the projections, based on the message exchange sequence of all the processes involved in the communication. All of them define the conditions under which the messages can be interchanged in order to create projections.

Furthermore, the thesis defines and proves several theorems for the process interoperability, given the relationships between their projections. The most general one is:

Theorem 1 *Any two processes P_1 and P_2 are compatible if and only if exist two projections P_1^n and P_2^m that match, where $P_1^i = \pi(P_1^{i-1})$ and $P_2^j = \pi(P_2^{j-1})$ where $1 \leq i \leq n$ and $1 \leq j \leq m$.*

As part of this thesis, a run-time process mediator able to apply the projections described above for each process in respect with the process it communicates with was developed. The appropriate projections are determined based on the exchange patterns of the processes involved in the communication, involving a detailed analyze of the processes and the evaluation of the rules that govern the message ordering. For dealing with the heterogeneity problems that cannot be automatically solved, a design-time process mediator which provides support to the domain expert was also developed.

4 Evaluation

The approach and prototypes developed in this thesis will be evaluated based on a two-fold methodology. Firstly, the thesis will consider a real use-case scenario developed as part of the SUPER project; this type of evaluation will prove that the approach is applicable in a real scenario. Secondly, in order to prove the correctness and completeness of the formal modeled developed in this theses, it will be evaluated against the existing workflow data patterns, based on the data *visibility*, *interaction*, *transfer* and *routing* [3].

5 Work Plan

The most important steps in accomplishing the objectives of this theses were already performed:

1. Identification of the types of mismatches that can be automatically solved.
2. Formalization of the operations that can be automatically performed by a mediator without breaking the communication.
3. Development of proof of concepts design-time and run-time prototypes needed for the process mediation.
4. Identification of a real-use case scenario, detailed analyze of the problems raised by the scenario.

However, important phases needed for the completion of the theses are still ongoing, such as:

1. Development of a comprehensive architecture for process mediation which will allow the integration of the two prototypes previously developed, providing complete solutions for process mediation;
2. Evaluation of the prototypes based on the available scenario;
3. Evaluation of the completeness and correctness of the approach based on the existing workflow data patterns

References

1. C. Bussler. *B2B Integration: Concepts and Architecture*. Springer, 2003.
2. E. Cimpian, A. Mocan, and M. Stollberg. Mediation enabled semantic web services usage. *Proceedings of the First Asian Semantic Web Conference*, 09 2006.
3. N. Russell, A. H. ter Hofstede, D. Edmond, and W. M. van der Aalst. Workflow data patterns. Technical report, Workflow Patterns Initiative, 2005.