



**HAL**  
open science

## Interconnecting Workflows Using Services: An Approach for "Case Transfer" with Centralized Control

Saida Boukhedouma, Zaia Alimazighi, Mourad Chabane Oussalah, Dalila Tamzalit

### ► To cite this version:

Saida Boukhedouma, Zaia Alimazighi, Mourad Chabane Oussalah, Dalila Tamzalit. Interconnecting Workflows Using Services: An Approach for "Case Transfer" with Centralized Control. 6th International Conference on Information Systems and Technology Management - ICISTM'2012, Mar 2012, Grenoble, France. pp.396 - 401. hal-01069112

**HAL Id: hal-01069112**

**<https://hal.science/hal-01069112v1>**

Submitted on 27 Sep 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Interconnecting Workflows Using Services: An Approach for “Case Transfer” with Centralized Control

Saida Boukhedouma<sup>(1)</sup>, Zaia Alimazighi<sup>(1)</sup>,  
Mourad Oussalah<sup>(2)</sup>, Dalila Tamzalit<sup>(2)</sup>

<sup>(1)</sup> USTHB- FEI- Departement of Computer Science- LSI Laboratory – ISI Team  
✉: El Alia BP n°32, Bab Ezzouar, Alger, Algérie.  
sboukhedouma@usthb.dz; zalimazighi@usthb.dz;

<sup>(2)</sup> Nantes University - LINA Laboratory – MODAL Team  
✉: 2, Rue de la Houssinière, BP 92208, 44322 – Nantes, cedex 3- France  
Mourad.oussalah@univ-nantes.fr; Dalila.tamzalit@univ-nantes.fr;

**Abstract.** In this paper, we are interested in structured cooperation based on workflow (WF). The current work proposes an approach based on **services** for WF interconnection particularly obeying to the “case transfer” architecture. This late defines a form of cooperation in Inter-Organizational WF involving a range of partners with common business goals, exercising the same business. All partners share the same WF model implemented at each location and a transfer policy to manage transfer for process instances from one partner to another. By the use of services, our goal is to obtain IOWF models flexible enough so they remain easily adaptable to support process changes. The proposed approach is based on centralized control for transfers.

**Key words:** IOWF, Process model, Service, SOA, Transfer policy, Coordinator.

## 1 Introduction

The B2B cooperation was initially supported by concepts and tools of *Inter-Organizational workflow* (IOWF). In our research, we focus on structured inter-organizational processes mainly based on architectures of cooperation well defined in the literature of IOWF [1].

Also, in a context of unstable environment, businesses often face stressful situations like a breach of contract with a partner or needs of additional resources. Thus, these companies must revise their systems, their business processes and their cooperation with other business partners in order to make the necessary adjustments. These adjustments can cover three complementary aspects of the system: data, process and organization where the central aspect is the *process* one. Therefore, our final objective is to define mechanisms for adaptation of IOWF models in order to support process changes. Because the adaptation of a process model depends on the entities composing it and links between these entities, we focus first on the question of *interconnection of workflows* so that they remain flexible enough and easily

adaptable. Thus, for WF interconnection, we adopt an approach based on *services* because of their characteristics: loosely coupled, easily invoked and business oriented.

This paper deals with the *case transfer* architecture [1] where several business partners share the *same WF model* implemented at each partner and define conjointly a *transfer policy* implementing rules to govern transfers of process instances from one location (partner) to another, at runtime. To develop our approach, we focus on two main questions: the *structuring* of the WF process in terms of services and the *control* of transfer of instances. In the following, section 2 defines the context of the work and basic concepts. Section 3 talks about some related works and explains the motivation of this paper. Section 4 describes conceptual and technical aspects of our approach. Section 5 concludes the work and talks about future works.

## 2 Context of the Work

An inter-organizational workflow (IOWF) can be defined as a manager of activities involving two or more WFs (affiliated with business partners) autonomous, possibly heterogeneous and interoperable in order to achieve a common business goal [1].

Several architectures of IOWF have been defined; we talk about the *capacity sharing*, the *chained execution*, the *subcontracting*, the (extended) *case transfer*, and the *loosely coupled* architectures [1].

The "*case transfer*" defines a form of cooperation fairly widespread in B2B, especially between partners engaged in the same profession and aiming to satisfy promptly many potential customers. In the "*case transfer*" architecture, business partners share the same WF model implemented at each partner and hosted by a local WFMS (WF management system). Their cooperation consists of transfer of process instances (cases) from one partner to another in order to achieve their execution.

For example, one can envisage an IOWF involving a set of partners in a process of production of medicines to meet many potential customers. A customer's order may arrive at partner *x* but it is not completely performed by the WF of this partner; the order may be transferred to other partners. The transfer can occur for example for load balancing or because of the lack of skills or resources at partner *x*.

At any moment, a process instance is at a single location. Each transfer is done at a *stable point* of the process in order to avoid any incoherence of execution. Also, transfers take into account the *state data* of instances.

## 3 Related Works and Motivation

With the emergence of SOA (Service Oriented Architectures) [2] and web services [3] standards, many research works deal with orchestration and choreography of web services in order to build business processes by service composition [4], [5]. Other research works such as [6], [7], [8] show the interest of combining BPM, WF and SOA for the re-use of services to construct dynamic business processes.

Also, many platforms and approaches based on WF and SOA have been proposed in the context of structured B2B cooperation, we cite as examples: *CoopFlow* [9],

*CrossWork* [10] and *Pyros* [11]. These approaches provide a certain degree of flexibility since they allow internal adaptation of WF processes.

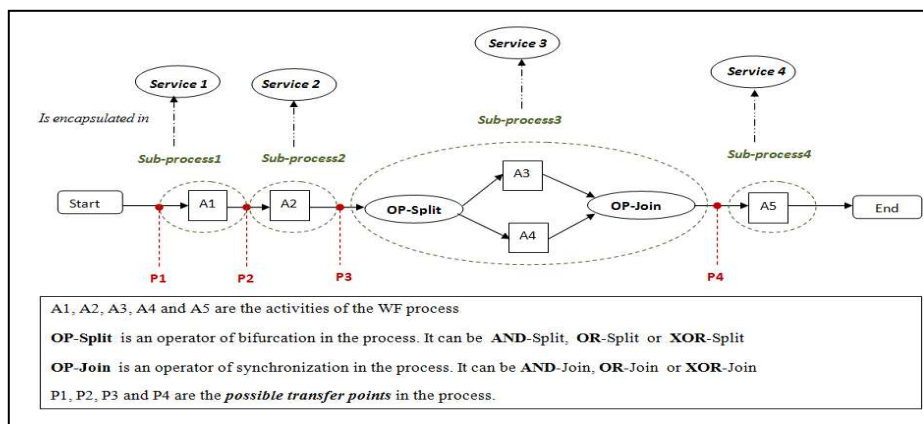
The principal motivation of our works is to achieve the adaptation of IOWF process models, by providing mechanisms to support process changes in context of *structured* cooperation in order to improve them or to satisfy new constraints imposed by the environment. Assuming that the ability of a model to be adaptable depends heavily on its components, we focus first on the question of WF interconnection and we propose *an approach based on services*, since services are software components loosely coupled, easily invoked and business oriented. For structured IOWF models, we rely on patterns defined in [1] because they cover various forms of cooperation that can link business partners together. In [12], we have proposed an approach for interconnecting workflows according to the *subcontracting* architecture. The current paper deals with the *case transfer* architecture, we propose an approach with *centralized* control of transfer which is well adapted for complex transfer policy.

## 4 Our Approach for WF Interconnection

Our approach focuses on two main questions: (i) How to *structure* the WF process into services? (ii) How to *control* the transfer of instances? To answer these two questions, we must define the notions of *transfer point* and *transfer policy*.

### 4.1 Transfer Point and Transfer Policy

- A *Transfer point* can be each state of the process that guarantees the coherence of execution of instances when a transfer is done. In fact, a transfer point should verify the following conditions: (i) It must be before the beginning or after the end of an activity. (ii) It should not interrupt the execution of an activity. (iii) It should not be between a routing operator *Split* and the corresponding operator *Join*; whether a parallel or an alternative branch is involved, the transfer may only occur after *Join*.



**Fig. 1.** Illustration of transfer point and Structuring of a WF process in services

Fig.1. illustrates the notion of *transfer points* on a WF process schema. We can see four transfer points (P1, P2, P3, P4) in the process. Let's notice that transfer points are the states of the process where a *case transfer* can eventually occur, they are fixed by the designers of IOWF at build time.

- A **Transfer policy** is conjointly defined by all partners at build time. It defines the set of *transfer points* and expresses a set of *rules* governing the transfer of process instances from one location to another. A rule is associated to a *transfer point* and can be defined by a pair (condition, action) meaning that if the condition is verified, an action of transfer is done to another location; otherwise the instance continues its execution at its current location.

#### 4.2 Structuring of the Process into Services

Our basic idea is to consider each WF process as a composition of *sub-processes* and then to encapsulate each *sub-process* within a *service* so it becomes easily invoked. A sub-process is a part of a global WF process composed by a single activity, a single bloc of activities delimited by a *Split* operator and the corresponding *Join* operator or a sequence of several activities and/or blocs. The cutting of the WF process into sub-processes is done according to the transfer policy defined. Hence, *transfer points* delimit the *sub-processes* encapsulated into *services* (see Fig. 1).

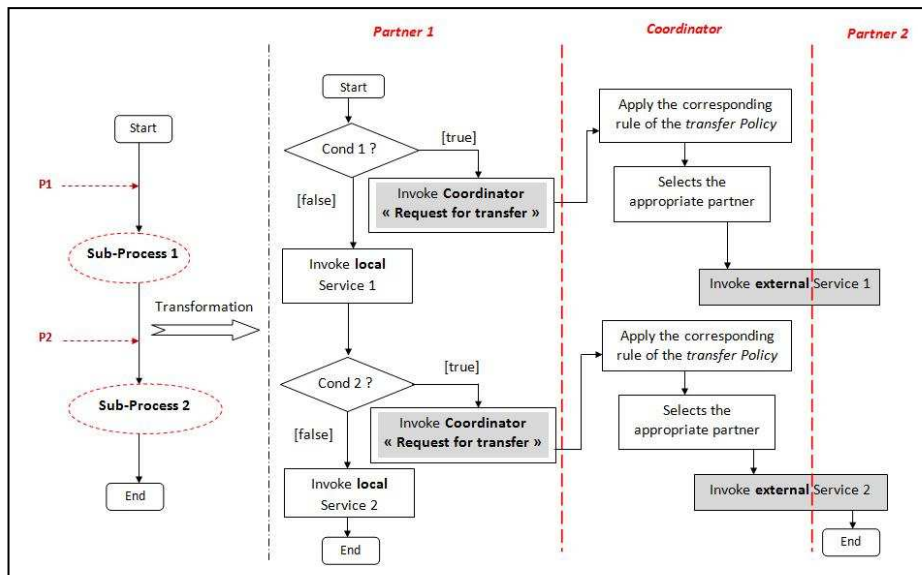


Fig. 2. Schema of WF interconnection with central coordinator

Also, for WF interconnection with *centralized* control of transfers, an additional component (the *coordinator*) is needed in order to manage all transfers to be done through the partners implied in the IOWF process. So, workflows don't interact directly with each other but they must do this through the *coordinator*. This mode is appropriate in case of a complex transfer policy (non deterministic rules), this can

usually occur when transfers are done for load balancing in the system. The process model is transformed to an IOWF model based on *services*. At each partner, the WF process is implemented as a set of *local services* (on Fig.2, Service 1 and Service 2) encapsulating the *sub-processes*. Hence, the execution of an instance is done through *invocations* of services: local invocation if no transfer is necessary and external invocation if a transfer is necessary. Thus, at each transfer point, the appropriate transfer condition (on Fig.2, cond1 or cond2) is evaluated by the system hosting the current instance, if it is true the *coordinator* is invoked (*Request for transfer*). According to the transfer policy and the state of the global system, the coordinator selects the partner to receive the instance and thus invokes it for a transfer with the *state data* of the instance and necessary *artifacts*.

### 4.3 Technical Aspects

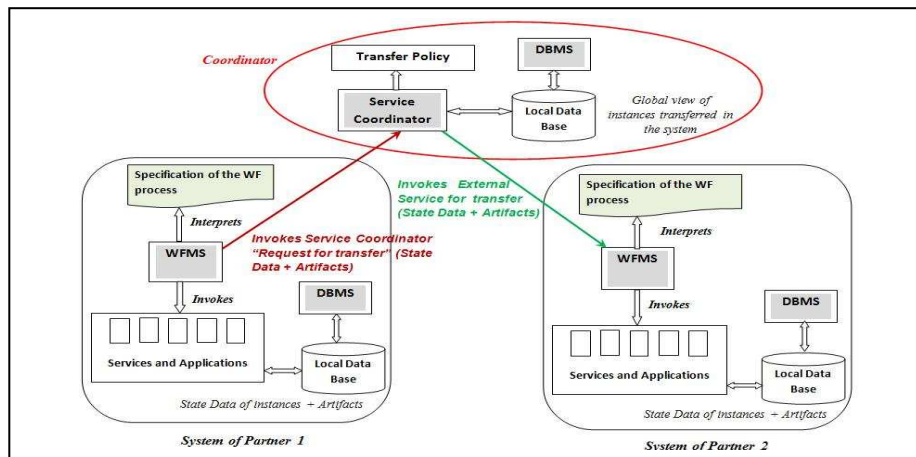


Fig. 3. Architecture of the system

Each partner implements the specification of the WF process locally, a local WFMS interprets the specification, a local DBMS manages local data bases containing state data of all instances locally executed and all artifacts necessary to perform services. The system of each partner implements local applications and services especially those which encapsulate the sub-processes of the WF (see Fig. 3). The *coordinator* is implemented as a service “*Service coordinator*” based on a transfer policy. Also, the coordinator maintains a local data base containing a global view of process instances in the system (this is particularly needed for load balancing). This architecture implies more interactions in the system comparatively to the architecture with decentralized control described in [13], but it is inevitable when the transfer policy is complex.

## 5 Conclusion

In this work, we have presented an approach based on *services* for WF interconnection according to the *case transfer* architecture [1]. The use of *services* for

interconnecting workflows is motivated by the fact that services are software components loosely coupled, easily invoked and business oriented. Thus, the IOWF models obtained remain easily adaptable which is the principal issue of our research.

The proposed approach relies on two main questions related to the *structuring of the WF process into services* and the *control of execution*. In this paper, we have proposed an approach with *centralized control* for transfer which is particularly appropriate to complex transfer policy that implements non deterministic rules. Thus, the systems of the partners interact through an additional component called *coordinator*. We have exhibited only conceptual aspects and general architecture of the system. This work should be completed by improving technical aspects mainly specification of the IOWF based on services (i.e specification of the WF process and the coordinator) using an appropriate specification language like BPEL or YAWL.

Currently, we are working on the question of adaptability of process models, we must inventory all possible changes (or the most important and frequent ones) in the IOWF process and we must formalize them as *adaptation patterns*. After that, we aim to develop a tool that helps the designer of the IOWF process in doing adaptation.

## References

1. Aalst W.V.D., Process-oriented architectures for electronic commerce and inter-organizational workflow, *Journal of IS*, Copyright Elsevier Sciences. (1999).
2. Papazoglou M. P., and Heuvel W., Service Oriented Architectures : approaches, technologies and research issues, *Springer Verlag, The VLDB Journal*, 16: 389-415. (2007).
3. Alonso G., Casati F., and Kuno H., Web services: concepts, architectures and applications, *Springer Verlag, Heidelberg, Germany*. (2004).
4. Decker G., Kopp O., Leymann F., and Weske M., Bpel4chor:Extending BPEL for modeling choreographies, *In Proceedings of the 2007 IEEE International Conference on Web Services (ICWS 2007)*: 296–303. IEEE Computer Society, (2007).
5. Amirreza T., *Web Service Composition Based Interorganizational Workflows*, Sudwestdeutscher Verlag fur Hochschulschriften edition, ISBN 9783838106700.(2009)
6. Leymann F., Roller D., and Schmidt M.T., Web Services and Business Process Management, *IBM Systems Journal*, 41 (2). (2002).
7. Gorton S., Montangero C., Reiff-Marganiec S., and Semini L., StPowla: SOA, Policies and Workflows, *ICSOC workshops, LNCS 4907*: 351-362 (2009).
8. Pedraza Ferrera G.R, An extensible Framework for the Construction of Process-Oriented Applications. PhD Thesis, *University of Grenoble I, France.*. (2009).
9. Chebbi I., *CoopFlow : An approach for workflow ascending cooperation in virtual enterprises. PhD thesis- National Institute of Telecoms, France* (2007).
10. N. Mehandjiev, I. Stalker, K. Fessl, and G.Weichhart. Interoperability contributions of Crosswork. *In invited short paper to Proceedings of INTEROP-ESA'05 Conference, Geneva, Springer-Verlag*. (2005).
11. Belhajjame K., Vargas-Solar G., and Collet. C., Pyros - an environment for building and orchestrating open services. *In SCC '05 : Proceedings of the 2005 IEEE International Conference on Services Computing, pages 155–164, USA, IEEE Computer Society*. (2005).
12. Boukhedouma S., Alimazighi Z., Oussalah M., Tamzalit D., SOA Based Approach for Interconnecting Workflows According to the Subcontracting Architecture. *In proceedings of IADIS- Collaborative Technologies'2011- Rome, Italy, 22-24 july*. (2011).
13. Boukhedouma S., Alimazighi Z., Oussalah M., Tamzalit D., Une approche basée SOA pour l'interconnexion de workflows: Application au transfert de cas. *In proceedings of INFOSID'2011, Lille, France. 24-27 May* (2011).