



HAL
open science

Analyse a priori de l'impact du changement des processus métiers

Oussama Mohamed Kherbouche, Mourad Mohamed Bouneffa, Adeel Ahmad,
Henri Basson

► **To cite this version:**

Oussama Mohamed Kherbouche, Mourad Mohamed Bouneffa, Adeel Ahmad, Henri Basson. Analyse a priori de l'impact du changement des processus métiers. INFORSID, May 2013, Paris, France. pp.257-266. hal-01894118

HAL Id: hal-01894118

<https://hal.science/hal-01894118>

Submitted on 12 Oct 2018

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.

Analyse *a priori* de l'impact du changement des processus métiers

Oussama Mohammed Kherbouche – Mourad Bouneffa – Adeel Ahmad – Henri Basson

*Laboratoire d'informatique Signal et Image de la côte d'Opale
Maison de la Recherche Blaise Pascal 50, rue Ferdinand Buisson
BP 719 62228 calais cedex
{kherbouche, bouneffa, ahmad, basson}@lisic.univ-littoral.fr*

RÉSUMÉ. L'évolution continue des systèmes d'information est une démarche essentielle pour les entreprises afin de mieux répondre aux attentes de leurs différents interlocuteurs. Cela a pour conséquence, la nécessité d'adaptation des modèles de données, des processus et des traitements applicatifs. A cet égard, les processus métiers jouant un rôle majeur dans la définition et la gestion des systèmes d'information doivent être dotés d'un mécanisme de contrôle et de gestion des effets de cette évolution. Ceci exige une compréhension exhaustive des causes des changements d'un processus métier, des niveaux d'application de ces changements, et de leurs impacts sur les autres processus ainsi que sur les instances en cours d'exécution. L'objectif de cet article s'inscrit dans le cadre de la gestion du changement des processus métiers. Plus précisément, nous présentons une approche à base de règles permettant l'analyse a priori de l'impact du changement affectant les processus métiers. Le but étant d'aider les concepteurs et les expert métiers à prévoir les risques liés à de tels changements ainsi que l'effort nécessaire pour leurs mises en œuvre.

ABSTRACT. The permanent evolution of information systems is essential for the companies to better fulfill the requirements of their different stakeholders. This necessitates the new changes to be adapted in the existing data, application and process models. In this context, business processes play a critical role in the definition and management of information systems. These must be equipped with a mechanism monitoring and managing the effects of the changes induced by this evolution. This requires an exhaustive understanding of concerned changes, their causes, the application levels of these changes, and their impact on other processes along with current running instances. The major objective of this paper is a part of the change management of business processes. More precisely, we propose a rule based approach to allow an a priori change impact analysis of business processes. The goal is to help designers and business experts to predict the risk of intended changes and the effort required for their implementation.

MOTS-CLÉS: modèle de processus métier ; analyse de l'impact des changements ; relation de dépendance

KEYWORDS: Business process model; change impact analysis; dependency relationship

1. Introduction

Dans le but de préserver leurs avantages concurrentiels, les entreprises doivent être en mesure de réagir rapidement aux mouvements du marché, aux inflexions des besoins de leurs clients et aux transformations métiers des différents interlocuteurs. Dans ce cadre, les processus métiers jouant un rôle majeur dans la définition et la gestion des systèmes d'information, doivent être flexibles pour faire face aux exigences de changement. En d'autres termes, la gestion de l'évolution des processus métiers est essentielle pour permettre aux organisations concernées d'être en phase avec les besoins et l'environnement de leurs interlocuteurs. Ainsi, la capacité de flexibilité et d'adaptabilité des processus métiers a été identifiée comme étant l'un des facteurs critiques de succès pour tout modèle de processus métier appelé Business Process Model ou BPM (Heinl *et al.*, 1999 ; Reijers, 2006). En dépit de la diversité des approches et techniques proposées dans la littérature ayant pour but la gestion des changements dynamiques des processus métiers (Reichert *et al.*, 1998 ; Sun P *et al.*, 2008 ; Van der Aalst *et al.*, 2005; Weske *et al.*, 1998), rares sont les outils de modélisation des processus métiers qui proposent une analyse exhaustive de l'impact du changement de ces processus. Nous proposons dans cet article, une approche voulue évolutive et exhaustive dans les informations qu'elle fournit pour aider les concepteurs et les experts métiers à analyser *a priori* la propagation de l'impact des changements et de ce fait estimer l'effort de leurs mises en œuvre.

Dans la section 2 du papier, nous proposons brièvement une typologie des changements ainsi que l'importance de l'analyse de l'impact de ces derniers. La définition et la formalisation de la notion de dépendance qui existe entre les différents éléments d'un processus métier est détaillée dans la section 3. Dans la section 4 nous proposons un ensemble de règles permettant l'analyse de l'impact des changements des processus métiers. Enfin, nous présentons brièvement les travaux connexes dans la section 5 et nous concluons notre contribution dans la section 6.

2. Typologie des changements et analyse de l'impact

Les types de changements des processus métiers sont multiples, ils émanent pour la plupart d'une correction d'erreurs, d'une exception dans le processus métier, d'une mise en conformité avec de nouvelles normes ou réglementations, d'une évolution du métier et des services de l'entreprise (innovation), ou d'une amélioration des performances des processus existants (reengineering). Toutefois, sans contrôle, ces changements peuvent engendrer des erreurs telles que des blocages, des exécutions infinies, des terminaisons multiples (kherbouche *et al.*, 2012 ; Rinderle *et al.*, 2004), une non-conformité avec la réglementation (Awad *et al.*, 2008), une dégradation du service (temps de réponse, sécurité, taille des messages, etc.), ou une mauvaise qualité générale des applications. Un changement

peut générer différents types d'impacts qui peuvent être structurels, fonctionnels, comportementaux, logiques et/ou qualitatifs sur le processus métier et ses instances. Par conséquent, une analyse *a priori* du changement est primordiale pour maintenir la cohérence des modèles des processus métiers. Elle est définie dans la littérature, comme le processus d'identification des conséquences potentielles d'un changement, et aussi l'estimation de ce qui doit être modifié pour accomplir ce changement (Bohner et al, 1996). Un changement peut être considéré comme une entité qui s'applique soit au niveau du schéma du processus métier "Process Type Level", ou à celui de ces instances "Process Instances Level". Il peut causer différents types d'impact et nécessite une analyse consistant à les décrire ou à les définir. Un processus métier est composé de différents types d'éléments ou entités en interaction les uns avec les autres. L'une des relations les plus couramment identifiée comme un fil conducteur dans le processus d'analyse de l'impact est la relation générique dite de dépendance (Bohner et al, 1996). Notre travail dans cet article traite l'analyse de l'impact du changement des processus métiers à travers une approche basée sur une définition de la relation de dépendance tenant compte des spécificités des modélisations de ces processus.

3. Analyse des dépendances

Dans le but de construire le modèle le plus exhaustif possible pour l'analyse *a priori* de l'impact du changement des processus métier, nous proposons une relation de dépendance tridimensionnelle prenant en compte la dépendance des activités, la dépendance des données et la dépendance des rôles.

3.1. Dépendances d'activités (routage)

La relation de dépendance d'activités ou dépendance de routage permet de traduire l'ordre d'exécution des différentes activités d'un processus métier par le biais des flux de contrôle (flux de séquences "sequence flows" et flux de messages "message flows"). Ces dépendances définissent non seulement l'ordre d'exécution mais également une certaine sémantique associée à cet ordre. Ainsi, pour une passerelle *AND-Join* reliant deux activités A et B à une activité C , l'activité C ne peut être exécutée que si A et B ont achevé leurs exécutions qui se déroulent en parallèle. Une dépendance d'activités peut être définie de façon formelle par $D_a = (\mathcal{D}_p(a), \Omega)$ sur un ensemble d'activités $A = \{a_1, \dots, a_n\}$ et un ensemble de flux de contrôles $T = \{t_1, \dots, t_n\}$, où $\mathcal{D}_p(a) = \mathcal{D}_{pi}(a) \cup \mathcal{D}_{po}(a)$ avec $a \in A$ et $\Omega = \Omega_i \cup \Omega_o$. $\mathcal{D}_{po}(a)$ représente l'ensemble des activités $a_i \in A$ qui succèdent à l'activité a (on notera cela par $a_i \rightarrow a$) et dont l'exécution dépend *a fortiori* de a . De la même manière, $\mathcal{D}_{pi}(a)$ représente l'ensemble des activités $a_i \in A$ qui précèdent l'activité a (on notera cela par $a \rightarrow a_i$). En d'autres termes l'exécution de a dépend de celles des activités appartenant à $\mathcal{D}_{pi}(a)$. Ω_i est l'ensemble des flux de contrôles $t_i \in T$

reliant chaque activité de $\mathcal{D}_i(a)$ à a . Autrement dit, l'ensemble des arcs entrants ($\mathcal{D}_i(a), a$) de a . Ω_o est l'ensemble des flux de contrôles $t_i \in T$ reliant l'activité a à toute activité appartenant à $\mathcal{D}_o(a)$. Ce sont donc les arcs sortant ($\mathcal{D}_o(a), a$) de a .

3.2. Dépendances de données

La dépendance de données met en avant les ressources communes ou les données liées à plusieurs activités. Il existe trois grands types de dépendances de données (cf. Malone *et al*, 1999) : la dépendance par flux ou "Flow dependency", la dépendance par partage ou "Sharing dependency" et la dépendance par mutualisation ou "Fit dependency". Une dépendance par flux existe si une activité produit une ressource ou donnée, utilisée par une autre activité. Une dépendance par partage représente le fait que plusieurs activités utilisent ou partagent la même ressource ou donnée. Et enfin une dépendance par mutualisation existe si plusieurs activités produisent collectivement une ressource ou donnée.

Dans la figure 1, l'activité A produit une donnée D_1 qui est utilisée par une autre activité F . Cela nous permet de conclure que l'activité F a une relation de dépendance de données avec l'activité A . De façon formelle, on définira l'ensemble des données D échangées entre les activités (cf. Reichert *et al*, 1998) par $D = \{d_1, d_2, \dots, d_n\}$. Chaque activité $a \in A$ possède des paramètres d'entrée et des paramètres de sortie, notés respectivement $InPARs(a)$ et $OutPARs(a)$. On définira une connexion de donnée notée dc par $dc = \{d, a, pars, mode\}$. Où, $d \in D$, $a \in A$, $pars \in InPARs(a) \cup OutPARs(a)$, et $mode \in \{lecture, \acute{e}criture\}$. Cela signifie que l'activité $a \in A$ utilise (en lecture ou en écriture) une donnée $d \in D$ fourni comme paramètre d'entrée ou/et de sortie. Une activité $a_i \in A$ dépend d'une autre activité $a_j \in A$ en termes de données, noté par la relation $a_i \xrightarrow{(R)} a_j$ ssi: $\exists dc_x, dc_y \in DC$, où $DC = \{dc_1, dc_2, \dots, dc_n\}$ est l'ensemble de toutes les connexions de données tel que $dc_x = \{d, a_j, pars, \acute{e}criture\}$ et $dc_y = \{d, a_i, part, lecture\}$, où, $d \in D$, $part \in InPARs(a_i)$, $pars \in OutPARs(a_j)$ et a_j précède a_i dans le schéma du processus métier.

3.3. Dépendances de rôles

Dans (Kim, 2003), l'auteur définit des relations de dépendance entre les rôles en remplaçant l'activité par le rôle associé à cette dernière. Autrement dit, le diagramme d'activités devient un diagramme de rôles. Cependant, ce diagramme de rôle est basé sur la structure hiérarchique de l'organisation (Dai *et al*, 2005). La relation de dépendance entre les rôles est définie entre deux rôles A et B si A est affecté à la même activité a exécutée par B . Par exemple, l'activité *prise de radio* dans un processus de l'examen d'imagerie médicale peut être effectuée par un *technicien* ou un *radiologue*. En l'occurrence, le rôle (*Technicien, prend une radio, l'examen d'imagerie médicale*) et le rôle (*Radiologue, prend une radio, l'examen d'imagerie médicale*). De façon formelle, soit $R = \{r1, \dots, rn\}$ un ensemble de rôles

et $A = \{a_1, \dots, a_n\}$ un ensemble d'activités. Une dépendance de rôles peut être définie par $D_r = (\mathcal{O}, \Psi)$, où $\mathcal{O}(r) = \mathcal{O}_i(r) \cup \mathcal{O}_o(r)$ avec $r \in R$ et $\Psi = \Psi_i \cup \Psi_o$. $\mathcal{O}_o(r)$ représente l'ensemble des rôles qui sont des successeurs immédiats à r , c'est-à-dire affectés à des activités $\alpha_i \in A$ qui succèdent à l'activité a dont r est associé. $\mathcal{O}_i(r)$ représente l'ensemble des rôles prédécesseurs immédiats de r . En d'autres termes les rôles affectés aux activités $\alpha_i \in A$ qui précèdent l'activité a et dont r est associé. Ψ_i est l'ensemble des flux de contrôles $t_i \in T$ ou arcs reliant chaque rôle de $\mathcal{O}_i(r)$ à r . Ψ_o est l'ensemble des flux de contrôles $t_i \in T$, reliant le rôle r à tout rôle appartenant à Ψ_o .

4. Règles d'analyse de l'impact

Nous décrivons dans ce qui suit, le processus d'analyse *a priori* de l'impact du changement en définissant un ensemble de règles qui se basent sur la relation de dépendance. Ces règles seront définies de façon algorithmique. Pour des raisons d'espace la règle d'analyse de dépendance de rôles n'est pas présentée. Dans ce qui suit, un modèle de processus métier est représenté par un graphe G_p dans lequel l'ensemble N des nœuds typés représente les activités, les événements et les passerelles dans le sens de la notation BPMN¹. L'ensemble F des arcs représente les flux de contrôle entre les nœuds. Nous associons aux nœuds une fonction $Status \rightarrow \{init, added, deleted, modified\}$ retournant le statut courant de chaque nœud N dans le graphe G_p (traces de changement). L'algorithme 1, présente la première règle qui permet de propager l'impact d'un changement à travers l'analyse des dépendances d'activités. Cette règle est déclenchée lors d'un ajout, suppression ou modification affectant un flux d'objets FO_x . La règle consiste à marquer le nœud correspondant à FO_x dans le graphe. Les ensembles $\mathcal{D}_{p_o}(FO_x)$ et $\mathcal{D}_{p_i}(FO_x)$ ainsi que les flux de contrôle associés sont à leurs tours marqués pour déterminer la profondeur de l'impact.

Algorithme 1: Règle d'analyse de l'impact (analyse des dépendances d'activités)

Entrée : N // l'ensemble des nœuds

Init : $\mathcal{D}_{p_o} \leftarrow \{\}, \mathcal{D}_{p_i} \leftarrow \{\}$

Début

Si $Status(FO_x) = \text{"added" || "deleted" || "modified"}$ **Alors**

/* marquer FO_x et les flux de contrôles associés */

marquer(FO_x);

marquer($FC \in \{\Omega_i(FO_x) \cup \Omega_o(FO_x)\}$);

/* \mathcal{D}_{p_o} est l'ensemble des activités succédant à FO_x (activités dépendant de FO_x) dans N */

Pour tout $a_i \in N$ ($i = 1 \dots n$) **faire**

Si $a_i \rightarrow FO_x$ **Alors** // Si a_i dépend de FO_x pour son exécution

$\mathcal{D}_{p_o} \leftarrow \mathcal{D}_{p_o} \cup \{a_i\}$

FinSi

FinPour

Pour tout $a_i \in \mathcal{D}_{p_o}$ ($i = 1 \dots n$) **faire**

/* marquer chaque élément de \mathcal{D}_{p_o} et les flux de contrôles associés */

1 <http://www.omg.org/spec/BPMN/2.0/PDF/>

```

    marquer(ai);
    marquer(FC ∈ {Ωi(ai) ∪ Ωo(ai)});
FinPour
    /* Dpi l'ensemble des activités qui précèdent FOx (FOx dépend des activités ai) dans N */
    Pour tout ai ∈ N (i = 1... n) faire
    Si FOx → ai Alors // Si FOx dépend de ai pour son exécution
        Dpi ← Dpi ∪ {ai}
    FinSi
FinPour
Pour tout ai ∈ Dpi (i = 1... n) faire
    /* marquer chaque élément de Dpi et les flux de contrôles associés */
    marquer(ai);
    marquer(FC ∈ {Ωi(ai) ∪ Ωo(ai)});
FinPour
FinSi
Fin

```

L'algorithme 2, présente la deuxième règle qui permet de propager l'impact d'un changement à travers l'analyse des dépendances de données et qui est complémentaire à la première règle. Dans cet algorithme, *OutputDs* représente l'ensemble des données *D* produites par *FO_x*, *AllOutDep* représente l'ensemble des activités qui dépendent de *D*. Tandis que *InputDs* est l'ensemble des données *D* utilisées par *FO_x* et *AllInDep* est l'ensemble d'activités dont les données qui sont produites sont utilisées par *FO_x*.

Algorithme 2: Règle d'analyse de l'impact (analyse des dépendances de données)

```

Entrée : N // l'ensemble des nœuds
Init : OutputDs ← {}, AllOutDep ← {}, InputDs ← {}, AllInDep ← {}
Début
    Si Status (FOx) = "added" || "deleted" || "modified" Alors
        /* marquer FOx et les flux de contrôles associés */
        marquer (FOx);
        marquer (FC ∈ {Ωi(FOx) ∪ Ωo(FOx)});
    Pour tout Di ∈ OutPARs(FOx) (i = 1... n) faire
        OutputDs ← OutputDs ∪ {Di};
    FinPour
    Pour tout Di ∈ OutputDs (i = 1... n) faire
        /* marquer les éléments de OutputDs et les flux de contrôles associés */
        marquer(Di);
        marquer(FC ∈ {Ωi(Di) ∪ Ωo(Di)});
        /* AllOutDep retourne l'ensemble des activités qui dépendent de la donnée D produite par FOx */
        Pour tout aj ∈ N (j = 1... n) faire
            /* aj dépend de la donnée D produite par FOx */
            Si aj  $\xrightarrow{D}$  FOx Alors
                AllOutDep ← AllOutDep ∪ {aj}
            FinSi
        FinPour
        Pour tout ak ∈ AllOutDep (k = 1... n) faire
            marquer(ak);
            marquer(FC ∈ {Ωi(ak) ∪ Ωo(ak)});
        FinPour
        FinPour
        Pour tout Di ∈ InPARs(FOx) (i = 1... n) faire
            InputDs ← InputDs ∪ {Di};
        FinPour
    Pour tout Di ∈ InputDs (i = 1... n) faire
        /* marquer InputDsi et les flux de contrôles associés */

```

```

marquer(Di);
marquer(FC ∈ {Ωi(Di) ∪ Ωo(Di)});
/* AllInDep retourne l'ensemble des activités produisant des données D utilisées par FOx */
Pour tout aj ∈ N (j = 1... n) faire
/* FOx dépend de la donnée D produite par aj */
Si FOx (D) aj Alors
    AllInDep ← AllInDep ∪ {aj}
FinSi
FinPour
Pour tout ak ∈ AllInDepk (k = 1... n) faire
    marquer(ak);
    marquer(FC ∈ {Ωi(ak) ∪ Ωo(ak)});
FinPour
FinPour
FinSi
Fin

```

L'exemple ci-dessous (cf. figure 1) illustre l'utilisation des règles évoquées.

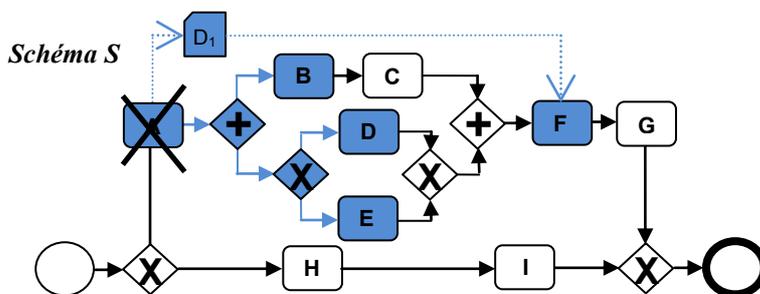


Figure 1. Suppression de l'activité A du schéma S

Dans cet exemple l'activité A est supprimée. Cela a un impact direct sur l'activité B et (D OU E), puisque ces dernières dépendent directement de l'exécution et l'achèvement de l'activité A. Ainsi que sur l'activité F qui utilise la donnée D₁ produite par l'activité A.

6. Travaux connexes

Les travaux de recherche portant sur la gestion des changements des processus métiers continuent à susciter un intérêt croissant de l'industrie et de la communauté scientifique et cela depuis plus de vingt ans. Cependant, la majorité de ces recherches se focalisent sur les changements dynamiques. Autrement dit, la façon dont on peut intégrer les changements avec cohérence sans affecter les instances en cours d'exécution en négligeant l'analyse *a priori* de l'impact de ces changements. En effet, dans (Weske, 1998), l'auteur suggère une modélisation et une exécution flexibles des activités du workflow basées sur un méta-modèle d'activités. Cette approche prend en charge les changements dynamiques comme l'addition ou la

suppression d'activités, mais nécessite que l'activité ne soit pas en état d'exécution lors de l'incorporation du changement. Une autre approche proposée par (Sun et al, 2008) est basée sur l'utilisation des réseaux de Petri pour calculer la région minimale affectée par les changements. ADEPTflex (Reichert et al, 1998) est un modèle graphique de workflow permettant l'intégration des changements dynamiques même pendant l'exécution des instances de ce modèle sans pour autant perdre le contrôle et la cohérence structurelle. Hormis ces travaux, les seules tentatives qui ont permis d'analyser la relation de dépendance qui existe au sein d'un workflow ont été identifiées dans (Chountas et al, 2003 ; Chun et al, 2002 ; Eder J 1999 ; 1999). Cependant, les auteurs de ces publications mettent l'accent sur la modélisation du workflow plutôt que sur l'analyse de l'impact, et la plupart des relations de dépendance se limitent à traiter la dépendance structurelle, à savoir, la dépendance d'activités ou de routage appartenant au niveau *intra-dépendance*. La seule étude qui a utilisé cette analyse de dépendance à des fins similaires aux nôtres est celle des auteurs (Dai et al, 2005). Cette étude propose d'utiliser un ensemble de requêtes définies sous PROLOG² afin d'aider les concepteurs à comprendre les dépendances qui existent entre les différents éléments du modèle de processus métier.

7. Conclusion

Dans cet article, nous avons présenté une approche à base de règles pour l'analyse de l'impact de changement des processus métiers. La méthode consiste à analyser les différents types de dépendances en prenant en compte les concepts dédiés à la description des processus métiers et la sémantique associée. Nous avons également explicité la notion de changement des processus métiers et introduit une typologie de ces changements. Dans des travaux futurs, nous allons proposer d'étendre notre analyse de dépendance à d'autres aspects relationnels, tels que les ressources, les acteurs, etc. et de former ainsi un modèle multidimensionnel de dépendance qui servira comme base pour une analyse de l'impact de changement de processus métier qui soit plus fine et encore plus pertinente.

8. Bibliographie

Awad A. Decker G. and Weske M., "Efficient Compliance Checking Using BPMN-Q and Temporal Logic", in *Proceedings of the 6th International Conference on Business Process Management*, pp. 326-341 Springer-Verlag Berlin, Heidelberg 2008

Bohner S. A. and Arnold R. S., "An Introduction to Software Change Impact Analysis", in S. A. Bohner and R. S. Arnold, Eds., *Software Change Impact Analysis*, IEEE Computer Society, 1996, pp. 1-25.

Chountas P., Petrounias I., Kodogiannis V., "Temporal Modelling in Flexible Workflows",

2. <http://www.gprolog.org>

In Computer and Information Sciences - ISCIS 2003, A. Yazici, C. Sener (Eds.), LNCS 2869, pp. 123 - 130, 2003.

- Chun S.A., Atluri V., Adam N. R., "Domain Knowledge-Based Automatic Workflow Generation", *Proceedings of Database and Expert Systems Applications: DEXA 2002*, R. Cicchetti, et al. (Eds.), LNCS 2453, pp. 81 - 93.
- Dai W. and Covvey H. Dominic. "Query-Based Approach to Workflow Process Dependency Analysis". Technical Report 01, School of Computer Science and the Waterloo Institute for Health Informatics Research, Waterloo, Ontario, Canada, 2005.
- Eder J., Panagos E., Rabinovich M., "Time Constraints in Workflow Systems", *Proceedings of Advanced Information Systems Engineering: CAiSE'99*, LNCS 1626, pp. 286 - 300, 1999.
- Eder J., Panagos E., Pozewaunig H. and Rabinovich M., "Time Management in Workflow Systems", *BIS'99. 3rd International Conference on Business Information Systems*, pp. 265 - 280, 1999.
- Heinl P., Horn S., Jablonski S., Neeb J., Stein K., and Teschke M., "A Comprehensive Approach to Flexibility in Workflow Management Systems", *In WACC '99: Proceedings of the international joint conference on Work activities coordination and collaboration*, pp. 79-88, New York, NY, USA, 1999. ACM
- Kherbouche, O.M., Ahmad, A., Basson, H., "Detecting structural errors in BPMN process models", *In 15th IEEE International Multitopic Conference (INMIC)*, Islamabad, Punjab, Pakistan pp. 425-431, 13-15 Dec 2012
- Malone T. W. and al., "Tools for inventing organizations: Toward a handbook of organizational processes", *Published in Management Science by Center for Coordination Science Massachusetts Institute of Technology*. 45(3) pp 425-443, March, 1999.
- Reichert M. and Dadam P., "ADEPTflex - Supporting Dynamic Changes of Workflows without Losing Control", *Journal of Intelligent Information Systems, Special Issue on Workflow Management*, Vol. 10, pp. 93-129, 1998.
- Reichert M., Dadam P., Bauer T., "Dealing with forward and backward jumps in workflow management systems", *Software and System Modeling 1 (2)*, pp. 37-58, 2003 .
- Reijers. H.A. , "Workflow Flexibility: The Forlorn Promise", *In 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006)*, 26-28 June 2006, Manchester, United Kingdom, , pp. 271-272. IEEE Computer Society, 2006.
- Rinderle S., Reichert M., Dadam P., "Correctness criteria for dynamic changes in workflow systems – a survey", *Data and Knowledge Engineering 50 (1)* , pp. 9-34, 2004.
- Sun P. and Jiang C., "Analysis of workflow dynamic changes based on Petri net," *in Information and Software Technology*, 2008.
- Weske M., "Flexible modeling and execution of workflow activities," *in Proceedings of the Thirty-First Hawaii International Conference* , vol. 7, pp. 713-722, Jan. 6-9 1998.
- Weske M., "Workflow Management Systems: Formal Foundation, Conceptual Design, Implementation Aspects", *University of Münster, Habil Thesis*, 2000.