

Spécification des besoins de performabilité dans les modèles BPMN

Une revue de la littérature

Fatima-Zahra Belouadha

Equipe Siweb, École Mohammadia d'Ingénieurs, Université Mohammed V - Rabat, BP. 765, Agdal-Rabat, Maroc

belouadha@emi.ac.ma

Résumé

La performance du système d'information dépend de l'efficacité opérationnelle des processus métier implémentés, mais aussi de leur capacité à répondre aux besoins non-fonctionnels, en particulier, celui de performabilité (performance et fiabilité). En effet, la phase de modélisation des processus métier est une phase primordiale du cycle BPM (Business Process Management). Néanmoins, la notation BPMN (Business Process Model and Notation) qui est le standard de facto de la modélisation des processus métier, se limite à la modélisation contemplative de leur comportement, permettant aux développeurs de capturer leurs besoins fonctionnels. Dans le cadre d'une approche de modélisation productive, l'intégration des besoins de performabilité dès la conception des processus métier est primordiale pour permettre leur analyse automatique et prédire leur qualité de service. Cet article classe et compare des contributions parmi les plus référencées ou récentes dans ce domaine. L'objectif est de mettre en évidence leurs limites et d'identifier les critères d'une approche efficace.

Abstract

The information system performance depends on the operational efficiency of the implemented business processes, but also on their ability to meet the non-functional requirements, especially those of performability (performance and reliability). Indeed, the modelling phase of business processes is a critical phase within the BPM (Business Process Management) lifecycle. Nevertheless, the notation BPMN (Business Process Model and Notation) which is the de facto standard for business process modelling, is limited to the contemplative modelling of their behavior enabling developers to capture their functional requirements. As part of a productive modelling approach, integrating performability requirements of business processes at design-time is essential to enable their automatic analysis and to predict their quality of service. This article classifies and compares contributions among the most recent or referenced in this field. The aim is to highlight their limitations and to identify the criteria of an efficient approach.

Mots-clés

BPMN, Modélisation des processus métier, Besoins de performabilité, QoS, Annotation

Keywords

BPMN, Business Process modeling, Performability Requirements, QoS, Annotation.

1. Introduction

L'efficacité des processus métier dépend de leur efficacité opérationnelle (capacité de répondre aux besoins fonctionnels), elle-même optimisée par leur efficacité non-fonctionnelle (capacité de répondre aux besoins non-fonctionnels). Rappelons que les besoins fonctionnels expriment ce que le processus métier doit faire alors que les besoins non-fonctionnels expriment la manière de le faire. En fait, il n'y a pas de définition exhaustive des besoins non-fonctionnels dans la littérature. Ils correspondent à titre d'exemple, à des critères tels que la performance, la fiabilité, la sécurité, la disponibilité, la maintenabilité et la portabilité. Nous pouvons constater qu'ils traduisent la ou les politiques adoptées pour implémenter ou configurer un processus métier, y compris les propriétés techniques de son implémentation et le contexte de son utilisation. Ces politiques jouent un rôle crucial lors du développement du système d'information. Elles guident les choix et les décisions prises lors de son implémentation. Comme le confirment Mylopoulos, Chung *et al.* (Mylopoulos, Chung *et al.*, 1992), toute omission ou erreur au niveau des besoins non-fonctionnels est généralement reconnue parmi les erreurs les plus coûteuses et difficiles à corriger, une fois que la mise en place du système d'information est achevée. S'ajoute à cela que la gestion des processus métier (BPM pour Business Process Management) doit de préférence être réalisée au niveau de la conception métier et non pas de l'implémentation technique (Smith *et al.*, 2002). Aussi, la description des propriétés non-fonctionnelles des processus au moment de leur modélisation s'avère-t-elle nécessaire pour contribuer à assurer leur efficacité.

La spécification des propriétés relatives à la performabilité (performance et fiabilité) des processus métier au moment de la conception est en fait importante pour aligner les objectifs de performance et de fiabilité avec les objectifs métier. Elle s'inscrit dans le cadre d'une approche de modélisation productive qui pourrait conduire l'analyse automatique et anticipée des processus. Néanmoins, la notation BPMN (*Business Process Model and Notation*) (OMG-BPMN, 2013), tout comme d'autres langages de modélisation de processus, s'inscrit dans le cadre de l'approche de modélisation contemplative, plutôt que productive (Belouadha, Omrana *et al.*, 2012) ; (Belouadha, Omrana *et al.*, 2014). Elle se limite à la description des besoins fonctionnels des processus métier à des fins de documentation et de communication.

Pour pallier cette limite, des approches d'extension du standard BPMN pour l'intégration des propriétés de performabilité ont été proposées dans la littérature. Cet article présente une revue de ces approches. Il rappelle les concepts et les apports du standard BPMN avant d'exposer les fondements et les principes des approches étudiées et d'en présenter une analyse comparative. L'objectif est d'en dégager les limites et de tirer des conclusions quant aux qualificatifs d'une approche adéquate pour l'extension du BPMN avec des aspects relatifs à la performabilité.

2. Standard BPMN : concepts et apports

La notation BPMN est le standard de facto de modélisation des processus métier. Cette section décrit ses principaux concepts, les illustre par un exemple et présente ses apports.

2.1. Principaux concepts

Le standard BPMN est une notation graphique dédiée à la modélisation des processus métier. Il a été développé à l'initiative de l'organisation BPMI (*Business Process Management Initiative*) en 2004 et adopté comme standard OMG (*Object Management Group*) en 2006. La version 2 du standard BPMN (OMG-BPMN, 2013) considère cinq concepts principaux pour la modélisation des processus : les *swimlanes*, les objets de flux (*flow objects*), les connecteurs (*connecting objects*), les artefacts et les objets de données (*data objects*).

Les *swimlanes* (*pools et lanes*). Ils permettent de créer des partitions d'activités pour un objectif d'organisation des processus.

- Un *pool* constitue un conteneur d'activités réalisées par un participant intervenant dans un processus collaboratif.
- Les *lanes* sont des couloirs qui permettent d'organiser ces activités à l'intérieur du *pool*.

Les objets de flux. Ce sont des nœuds du graphe qui comportent les activités (tâches ou sous-processus), les événements et les branchements (*gateways*).

- Les tâches sont des activités atomiques et les sous-processus sont des activités composées.
- Un sous-processus est noté par un objet graphique qui en représente une vue réduite (*collapsed*) ou développée (*expanded*). À l'encontre de la vue réduite qui cache les détails de modélisation d'un sous-processus, la vue développée montre ces détails en mettant en relief des activités, des branchements, des événements et des flux de séquence. C'est un conteneur d'objets de flux qui peut être constitué de 0 ou plusieurs ensembles de couloirs (*lanes*).

- Un évènement représente quelque chose qui se produit et affecte le cours du processus. Il a souvent une cause et un résultat et peut débuter, interrompre ou terminer un flux.
- Les branchements (*gateways*) sont des éléments de contrôle de flux des activités. Ils sont répartis en branchements exclusif (*exclusive gateway*), inclusif (*inclusive gateway*), parallèle (*parallel gateway*), à base d'évènement (*event-based gateway*) et complexe (*complex gateway*) pour représenter différents branchements conditionnels tels que les fork, join et merge.

Les connecteurs. Ils représentent des flux d'information qui constituent des flux de séquence, des flux de message, des associations ou des associations de données (*data associations*).

- Les flux de séquence enchaînent les activités à l'intérieur d'un couloir (*lane*) pour montrer l'ordre dans lequel elles seront exécutées. Ils lient des activités, des évènements et des gateways et peuvent être conditionnels.
- Les flux de message représentent les messages échangés entre des *pools*.
- Les associations lient des données ou des artefacts à des objets du flux. Les associations de données sont utilisées pour modéliser un flux de données entre des objets de données et des activités ou des évènements (ils peuvent, par exemple, associer des entrées ou des sorties à une activité). Les associations d'artefacts, quant à elles, relient à titre d'exemple, une annotation (texte défini par l'utilisateur) à un élément graphique.

Les artefacts. Ils apportent des informations additionnelles à propos du processus. Ils peuvent être des groupes (*groups*), des annotations textuelles ou encore des artefacts définis au besoin.

- Les groupes sont utilisés pour mettre en relief des ensembles d'activités qui peuvent appartenir à différents *lanes* et/ou *pools*.
- Les annotations textuelles sont associées aux éléments du diagramme BPMN pour les décrire.

Les objets de données. Ces objets désignent généralement soit des données nécessaires pour exécuter les activités, soit des données que les activités produisent (*business documents*, e-mails, etc.). Ainsi, ils peuvent être de types *data inputs*, *data outputs*, *data stores* ou messages.

- Les *data inputs* et *data outputs* représentent respectivement les entrées/sorties d'un processus et peuvent être associés à ses activités par des associations de données.
- Les *data stores* désignent des emplacements où les activités peuvent chercher ou mettre à jour des données stockées (par exemple, des bases de données.).
- Les messages décrivent le contenu d'une communication entre deux participants.

La figure 1 illustre une vue d'un exemple abstrait de modèle de processus métier en BPMN.

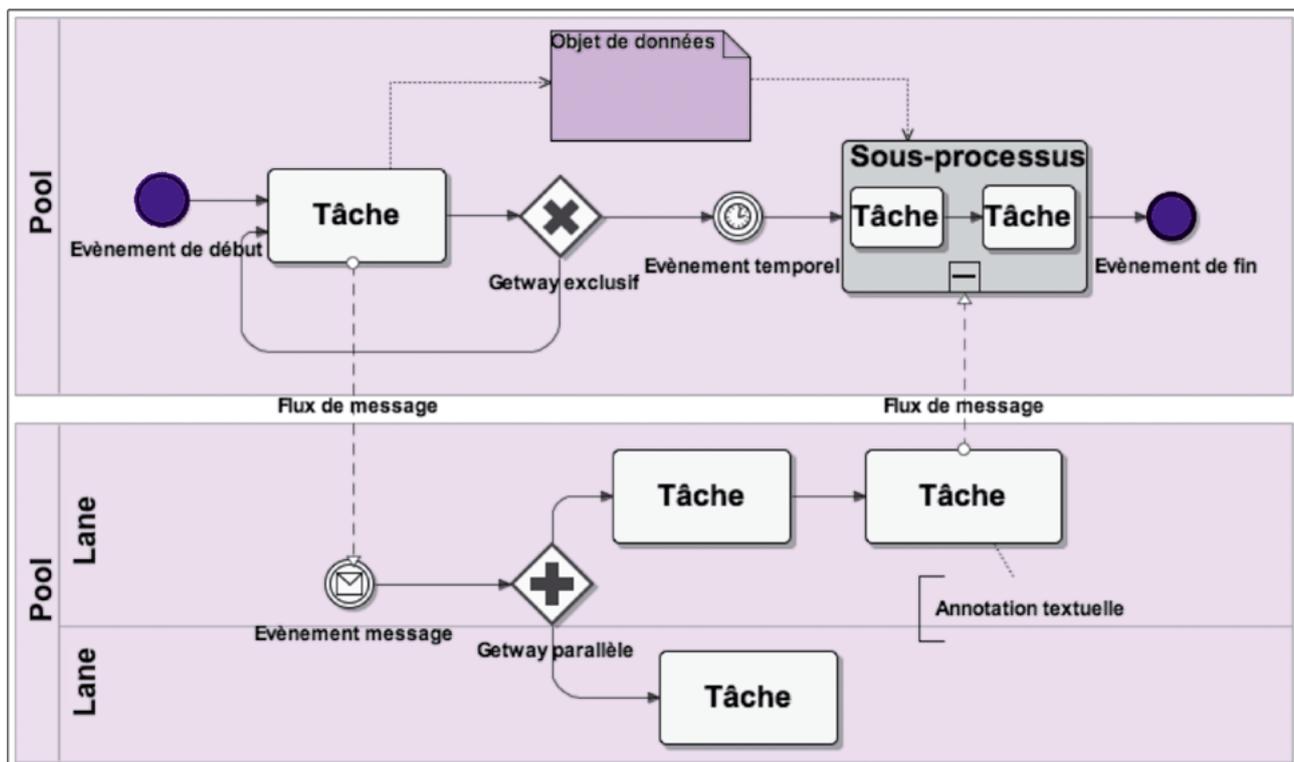


Figure 1. Vue d'un exemple abstrait de modèle de processus métier en BPMN

2.2. Exemple de processus BPMN

La figure 2 illustre un exemple concret de processus modélisé en BPMN. Ce processus d'achat d'articles (produits) est déclenché par le client qui commence par choisir autant d'articles qu'il le souhaite. Le sous-processus *Choisir un article* est en fait un sous-processus *collapsed* qui constitue une boucle consistant en deux tâches : *Sélectionner un article* et le *Rajouter au panier*. Après avoir choisi les articles désirés, le client doit s'authentifier. S'il arrive à le faire avec succès, il devra ensuite payer sa commande, sinon il devra créer d'abord un compte avant de pouvoir effectuer le paiement. La tâche *Créer un compte* a comme entrée les informations du compte à créer (*login*, mot de passe et autres informations y compris le numéro de compte bancaire). Une fois que le paiement est effectué, la date de livraison est retournée et le magasinier de l'agence de vente concernée reçoit la référence du paiement. Suite à cela, il devra préparer la commande. Ensuite, il devra mettre à jour l'inventaire des produits parallèlement à la livraison de la commande. Cette opération est effectuée par le responsable de transport et se concrétisera par l'envoi du bon de livraison au client. Celui-ci, ayant été en attente de la date de livraison, devra signer le bon de livraison une fois qu'il le reçoit à la date prévue. Ainsi, le processus d'achat d'articles prendra fin.

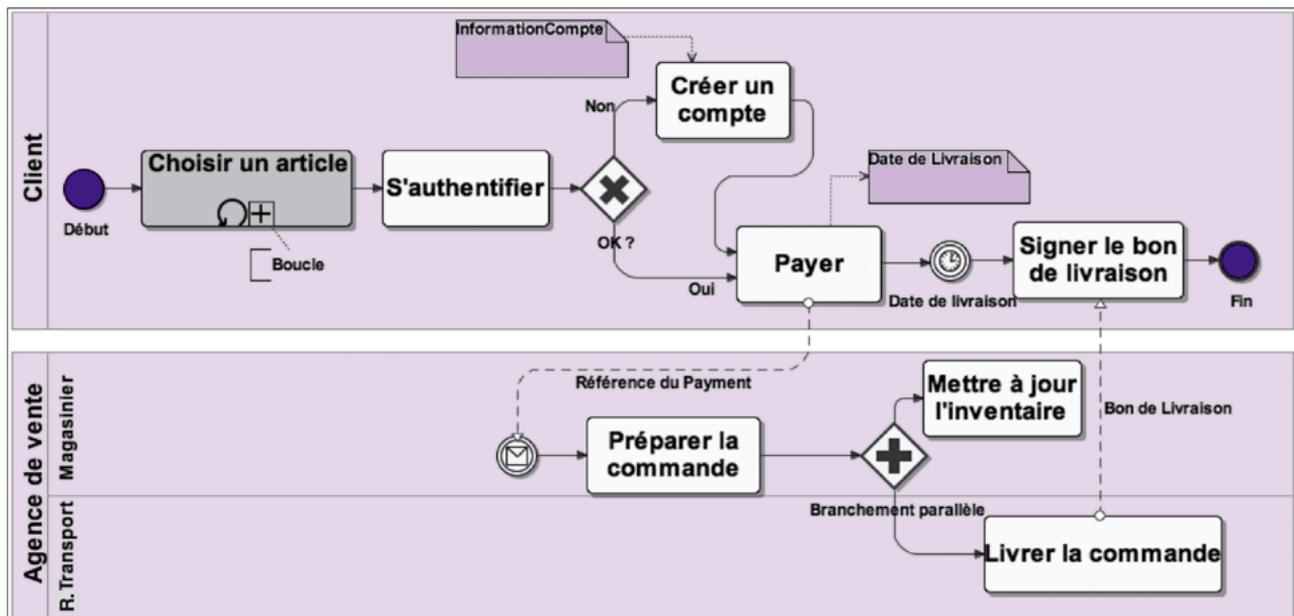


Figure 2. Exemple de processus modélisé en BPMN

2.3. Apports de la notation BPMN

La notation BPMN présente d'importants apports (Owen et Raj, 2003). Parmi lesquels figurent :

La modélisation collaborative. BPMN pallie le problème de la difficulté de communication entre les analystes métier et les experts de technologies de l'information qui avaient à traiter les processus, moyennant des outils et des concepts différents. Il est conçu pour être utilisé par toutes les parties impliquées dans le cycle de vie des processus, à savoir les analystes métier qui modélisent les processus, les experts en sécurité qui définissent les mécanismes adéquats pour les sécuriser, les développeurs techniques qui prennent en charge leur implémentation et l'équipe métier qui gère et contrôle ces processus.

Le mapping automatique à BPEL. BPMN permet la transition directe et standardisée de la conception des processus métier à leur implémentation. Il est automatiquement mappable au standard d'exécution BPEL (*Business Process Execution Language*) (OASIS-BPEL, 2007). En plus, il est compatible avec le langage de requêtes BPQL (*Business Process Query Language*) (Owen et Raj, 2003) qui constitue une interface de gestion standard pour le déploiement et l'exécution des processus métier.

La simplicité et l'expressivité. BPMN est conçu pour être simple et facile à utiliser et à comprendre. Il est cependant expressif et permet de décrire des processus métier complexes. Il propose, entre autres, des notations spécifiques pour décrire en particulier les événements à base de messages et la transmission de messages entre des organisations.

La description consolidée. BPMN représente un processus métier par un seul diagramme. Toutes les spécifications décrivant un processus ne se trouvent pas fragmentées dans plusieurs diagrammes à l'instar de UML (*Unified Method Language*).

3. Spécification des besoins de performabilité dans les modèles BPMN

De nombreux chercheurs se sont intéressés à l'extension des modèles BPMN avec des propriétés de performabilité (Korherr et List, 2007) (Magnani et Montesi, 2007) (Pavlovski et Zou, 2008) (Sampath et Wirsing, 2009) (Saeedi, Zhao *et al.*, 2010) (Bartolini *et al.*, 2013) (Bocciarelli et d'Ambrogio, 2014). Les approches proposées pour l'intégration des propriétés de performabilité dans les modèles de processus BPMN peuvent être classées en quatre catégories selon l'approche d'annotation adoptée : approches orientées artefacts, approches orientées tables, approches orientées annotation textuelle et approches orientées instances UML. Cette section décrit les fondements et les principes de chacune de ces classes d'approches à travers quatre contributions parmi les plus référencées ou les plus récentes dans le domaine. Une analyse comparative de ces contributions est présentée à la fin de la section.

3.1. Approche d'annotation orientée artefacts

La contribution proposée par (Pavlovski et Zou, 2008) constitue une référence dans le domaine. Elle adopte une approche d'annotation fondée sur l'utilisation des artefacts pour annoter les processus avec des propriétés non-fonctionnelles dont les celles de performabilité.

Principe général. Cette contribution est fondée sur deux principes :

- Enrichir les modèles BPMN avec les besoins non-fonctionnels à capturer au moment du développement des systèmes.
- Proposer des éléments de description contribuant à atténuer les risques associés aux contraintes non-fonctionnelles sans toutefois compliquer les modèles de processus.

Aspects considérés. Elle considère des propriétés non-fonctionnelles telles que les politiques de sécurité, les propriétés de performabilité, les politiques organisationnelles et les contraintes réglementaires.

Mécanisme d'annotation. Les propriétés considérées servent à annoter des tâches de processus.

Les auteurs étendent BPMN avec deux nouveaux artefacts (figure 3) :

- *Operating Condition* utilisé pour introduire une contrainte. Il est décrit par un attribut indiquant le type de cette contrainte (contrainte organisationnelle, politique de sécurité, contrainte de performance, etc.).
- *Control Case* utilisé en vue de définir les contrôles métier à mettre en place pour atténuer le risque associé à une contrainte. Il est décrit au moyen d'un ensemble d'attributs catalogués sous format texte sans être introduits dans les diagrammes de processus. Ces attributs sont : le nom de l'artefact *Operating condition* auquel l'artefact *Control Case* est lié, sa description textuelle, la contrainte métier ou technique qu'il exprime, le risque métier qui lui est associé et les contrôles métier à appliquer.

La figure 4 et le tableau 1 montrent l'exemple d'une tâche annotée, tiré d'un exemple donné par (Zou et Pavlovski, 2006). La tâche *FlightReservation* est annotée par une politique de performabilité en vue d'éviter un temps de réponse inacceptable. L'objectif est d'atténuer le risque de perte des clients utilisateurs du système proposant la réservation de billets d'avion.

Cette politique recommande trois mesures (l'utilisation des techniques d'équilibrage de charge (load balancing) et de clustering, l'utilisation d'un cache et la négociation du contrat de la bande passante pour le réseau).

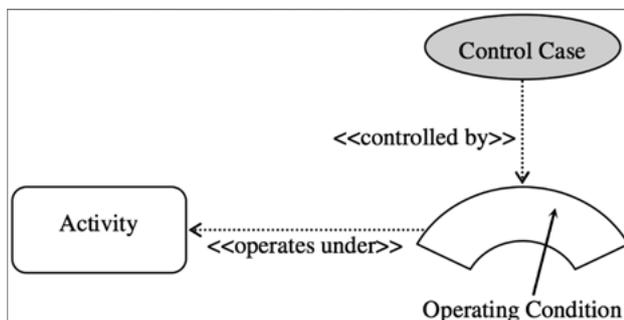


Figure 3. Artefacts *Operating Condition* et *Control Case* (Pavlovski et Zou, 2008)

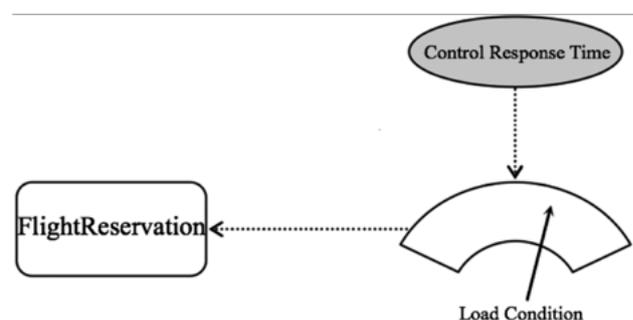


Figure 4. Exemple d'une tâche annotée par une politique de performabilité

Control Case: Temps de réponse	
Operating Condition :	Surcharge due à un accès concurrent (simultané) des utilisateurs.
Description :	Suite à une surcharge due à un accès concurrent de plusieurs utilisateurs à la fois, le temps de réponse du système peut devenir inacceptable. Ce <i>control case</i> définit le temps de réponse maximum des transactions afin d'atténuer le risque de perte des clients.
Contrainte technique :	La durée de la transaction dépend de la bande passante Internet et de l'infrastructure du réseau ISP.
Risque métier :	Satisfaire 90% des utilisateurs en un temps de réponse dépassant 10 secondes exposera l'entreprise à une perte de 20% de ses clients en faveur des entreprises concurrentes.
Contrôles métier :	Pour atténuer le risque, il faut utiliser les techniques d'équilibrage de charge (<i>load balancing</i>) et de <i>clustering</i> , utiliser un cache et négocier le contrat de la bande passante pour le réseau ISP.

Tableau. 1 Exemple de spécification des attributs d'un artefact Control Case

Conclusion. Dans le cadre d'une approche de modélisation contemplative, l'approche proposée adopte une représentation simple des besoins non-fonctionnels dont les propriétés de performabilité. Pour fournir plus de détails sur ces besoins, elle les complète par des informations additionnelles. Toutefois, ces informations sont représentées de façon informelle.

3.2. Approche d'annotation orientée tables

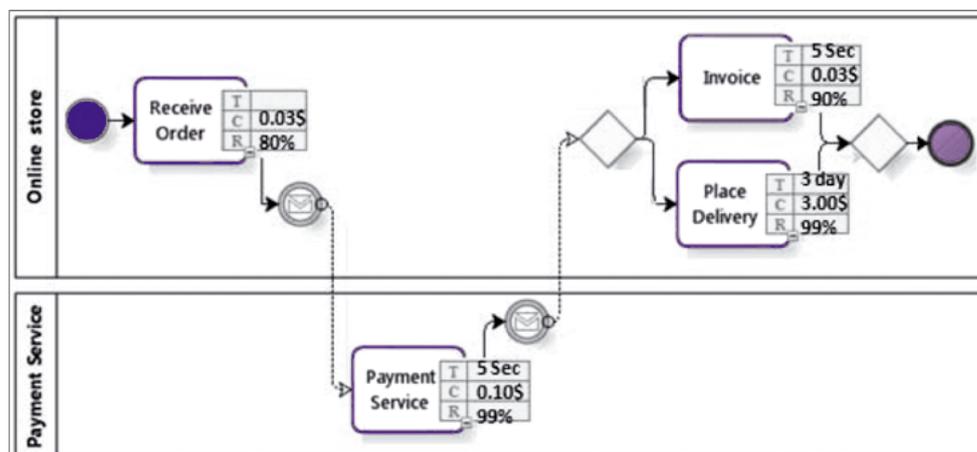
La contribution proposée par (Saeedi, Zhao *et al.*, 2010) adopte une approche d'annotation fondée sur l'utilisation des tables pour l'annotation des propriétés de types temps de réponse, coût et fiabilité.

Principe général. Cette contribution consiste à :

- Décrire les processus BPMN par des propriétés clés de performabilité, souvent exprimées comme préférences d'utilisateur et abordées dans des contrats de service SLA (*Service Level Agreement*).
- Exploiter les propriétés spécifiées pour évaluer la qualité des processus, en utilisant une méthode fondée sur un modèle analytique à base des règles de réduction.

Aspects considérés. Elle considère trois propriétés de performabilité : le temps de réponse, le coût et la fiabilité.

- La notion de temps de réponse réfère à trois composants : le temps de calcul, le temps d'attente (dû par exemple, au délai d'attente d'allocation du processeur selon le rang dans une file d'attente) et le temps de transfert de données qui devient important quand les données transmises sont volumineuses.
- Le coût correspond au coût financier d'installation (*setup*), de déploiement, de gestion, de surveillance et d'exécution du processus.
- La fiabilité est la capacité d'effectuer la tâche attribuée au processus dans une durée donnée. Elle est mesurée en fonction du taux de pannes du système SF (*System Failure rate*) et du taux d'échec du processus PF (*Process Failure rate*).

Figure 5. Exemple d'un processus annoté (Saeedi, Zhao *et al.*, 2010)

Mécanisme d'annotation. Les propriétés considérées sont utilisées pour annoter des éléments de type *SubProcess*, *CallActivity* et *Task*.

Pour annoter une activité, les auteurs utilisent une table qui contient les valeurs correspondant à son temps de réponse, son coût et à sa fiabilité. La figure 5 montre un exemple de processus dont les tâches sont annotées. À titre d'exemple, le temps de réponse de la tâche *Payment Service* est de 5 secondes, son coût est de 0,10 \$ et son taux de fiabilité est de 99%.

Conclusion. Cette approche adopte un mécanisme d'annotation simple pour l'intégration de trois propriétés de performabilité: le temps de réponse, le coût et la fiabilité. Elle s'inscrit dans une perspective de modélisation productive permettant la vérification de la qualité des processus à partir des modèles BPMN. Cette approche a été mise en œuvre à travers l'extension de l'outil *Eclipse Process Modeller*.

3.3. Approche d'annotation orientée instances UML

La contribution proposée par (Bartolini *et al.*, 2013) adopte une approche d'annotation fondée sur un profil UML. Ce profil est utilisé pour l'annotation des processus avec des propriétés non-fonctionnelles dont les propriétés de performabilité.

Principe général. Cette approche permet de surveiller l'exécution des processus en vue de détecter les éventuels défauts de qualité. Elle est fondée sur:

- L'extension Q4BPMN (Quality for BPMN) (Bartolini *et al.*, 2012) qui étend la notation BPMN conformément au métamodèle de propriétés PMM (Property Meta-Model) (di Marco *et al.*, 2011) pour enrichir les spécifications des chorégraphies (celles-ci désignent des interactions entre processus en mettant l'accent sur l'échange entre les participants).
- Les techniques de l'ingénierie dirigée par les modèles pour transformer les modèles Q4BPMN en modèles de qualité conformes à la notation Klaper (*Kernel Language for Performance and Reliability analysis of component-based systems*) et les analyser.

Aspects considérés. Elle considère trois classes de propriétés non-fonctionnelles:

- Les propriétés de performance reliées au nombre de ressources du système, en particulier matérielles.
- Les propriétés de sécurité qui déterminent le niveau de confidentialité (*privacy*) garanti par le système lors de l'échange de données privées.
- Les propriétés de fiabilité qui déterminent le niveau de fiabilité du système.

Mécanisme d'annotation. Les propriétés considérées sont aussi utilisées pour annoter des tâches (*Q4Task*), des acteurs associés à des tâches (*Q4Participants*) et des chorégraphies (*Q4Choreography*).

Des annotations textuelles sont utilisées pour annoter des éléments des diagrammes BPMN avec des instances de classes UML, décrites dans des paquetages qui ne sont pas intégrés à ces diagrammes.

Chaque instance est définie par les attributs suivants:

- *IsHard* pour indiquer si la propriété est obligatoire (contrainte) ou optionnelle (préférence).
- *Metrics* pour spécifier la métrique à laquelle correspond la propriété (par exemple, la durée maximale par défaut).
- *Nature* pour indiquer si la propriété est descriptive (représentant une propriété garantie par le système) ou prescriptive (constituant une propriété requise par le système).
- *PropertyClass* pour spécifier le type de la propriété (performance, sécurité ou fiabilité).
- *Operator*, *Value* et *Unit* pour exprimer une propriété mesurée au moyen d'un opérateur, une valeur et une unité de mesure.

La figure 6 illustre l'exemple d'une tâche annotée par des besoins en performance. La tâche *FlightReservation* est annotée par une propriété qui indique qu'il est préférable (*isHard=false*) que le temps de réponse soit égal à 5000 ms.

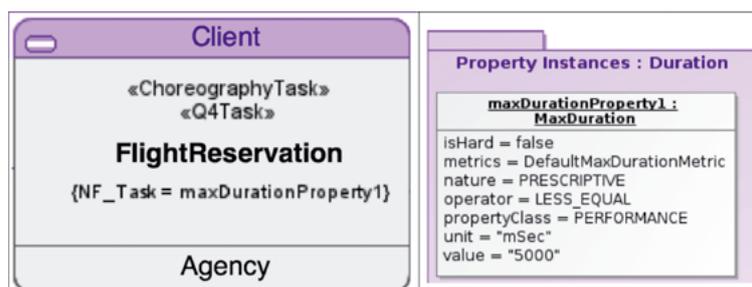


Figure 6. Exemple d'une tâche annotée selon l'extension Q4BPMN

Conclusion. L'extension Q4BPMN s'inscrit dans le cadre d'une approche de modélisation productive visant à détecter les défauts de performabilité et de sécurité à partir des modèles de processus. Elle est implémentée au niveau de l'outil MagicDraw et utilise les outils d'analyse fournis par l'environnement KLAPERSUITE (Ciancone *et al.*, 2011). Néanmoins, cette approche propose une représentation fragmentée des propriétés non-fonctionnelles, en décrivant les modèles BPMN par des instances décrites dans des paquets UML.

3.4 Approche d'annotation orientée annotations textuelles

La contribution proposée par (Bocciarelli et d'Ambrogio, 2014) adopte une approche d'annotation fondée sur l'utilisation des annotations textuelles pour annoter les processus avec des propriétés de performance et de fiabilité.

Principe général. Cette contribution orientée MDA (*Model Driven Architecture*) propose une extension de BPMN, dénommée PyBPMN (*Performability-enabled BPMN*) (Bocciarelli et d'Ambrogio, 2011). Elle est fondée sur deux principes qui consistent à :

- Enrichir le métamodèle BPMN conformément au profil UML MARTE (*Modeling and Analysis of Real-Time and Embedded systems*) (OMG-MARTE, 2011) et ses extensions proposées par (Bernardi, Merseguer *et al.*, 2008) et (Petriu, Bernardi *et al.*, 2008).
- Transformer les modèles *PyBPMN* en modèles d'analyse de performance et de fiabilité pour prédire la performance des processus et sélectionner les meilleures configurations de services Web pour les implémenter.

Aspects considérés. L'approche proposée permet de spécifier les propriétés de performabilité des processus BPMN. Ces propriétés sont des instanciations de quatre principales méta-classes qui étendent le métamodèle BPMN :

- *PaQualification* permettant de décrire la durée d'exécution (*execution demand*), le rendement (*throughput*) et le temps de réponse.
- *DaError* utilisée pour décrire la fiabilité en termes de latence (temps écoulé entre l'occurrence de l'erreur et sa détection) et de probabilité d'occurrence d'une erreur.
- *DaFault* permettant de spécifier la fiabilité en termes de fréquence d'anomalies (*fault occurrence rate*), de probabilité d'occurrence d'une anomalie (*fault*) et de distribution de probabilité des occurrences des faults.
- *DaFailure* utilisée pour décrire la fiabilité à travers trois attributs: la fréquence des pannes (*failure occurrence rate*), la durée moyenne de fonctionnement sans panne MTTF (*Mean Time To Failure*) et la durée moyenne nécessaire pour la réparation d'une panne MTTR (*Mean Time To Repairation*).

Mécanisme d'annotation. Les propriétés considérées sont utilisées pour annoter les objets BPMN de type *Collaboration* (interaction entre plusieurs processus), *Process* et *task*.

Les auteurs utilisent des annotations textuelles qui spécifient les besoins de performabilité au moyen des attributs/valeurs. L'un des attributs exprime une valeur qui correspond à une fréquence (de panne ou d'exécution, etc.) ou une durée. Les autres attributs fournissent des informations sur cette valeur telles que l'expression de calcul utilisée pour l'obtenir, son origine (calculée, mesurée, requise ou estimée) et son type (maximum, minimum, moyenne, etc.).

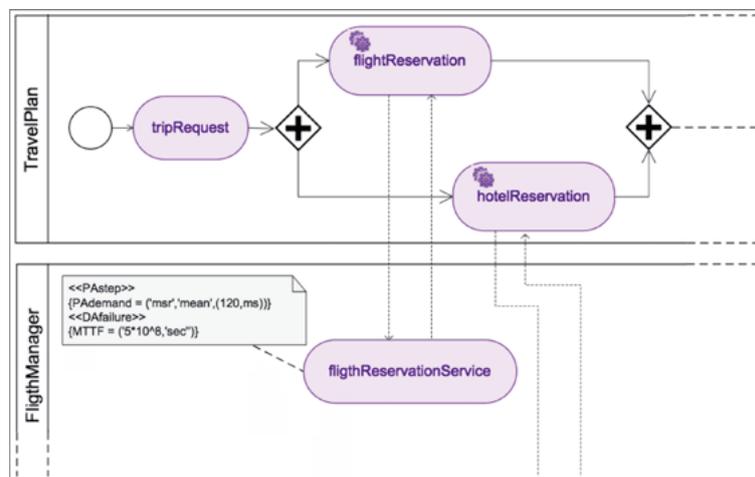


Figure 7. Exemple de processus PyBPMN

La figure 7 montre l'exemple d'une tâche d'un processus, annotée selon *PyBPMN* (Bocciarelli et d'Ambrogio, 2011). Comme spécifié, la durée moyenne d'exécution mesurée de la tâche *flightReservationService* doit être égale à 120 ms et la durée de panne doit être de l'ordre de 5. 108 secondes.

Conclusion. L'approche proposée adopte une représentation des besoins en performabilité, fondée sur l'extension du standard MARTE. Son utilisation demeure compliquée pour des analystes métier qui ne sont pas familiarisés avec ce standard. Par ailleurs, cette approche s'inscrit dans une perspective de modélisation productive. Elle permet l'analyse et la prédiction de la qualité des processus au moment de la conception ainsi que leur implémentation automatique.

4. Analyse comparative

Avant d'évaluer les différentes approches présentées auparavant, cette section expose d'abord les critères sélectionnés pour l'évaluation de leur pertinence.

4.1. Critères d'évaluation

Trois catégories de critères d'évaluation de la pertinence extraits à partir des approches d'annotation des propriétés non-fonctionnelles sont intéressantes pour analyser les approches étudiées.

Ces catégories de critères se résument comme suit :

- **L'objectif recherché.** Il met en relief trois critères : le domaine ciblé, le contexte d'utilisation et les objectifs spécifiques. Cela devrait permettre d'une part d'évaluer la portée des approches analysées en termes de propriétés ciblées et leur adaptation au contexte des services Web et, d'autre part de distinguer les approches de modélisation contemplatives de celles productives. La prise en considération du critère d'adaptation au contexte des services Web est justifiée par la tendance industrielle. Celle-ci consiste à opter pour le couple BPM/Architecture orientée services (SOA pour *Service Oriented Architecture*), en particulier les services Web et leurs standards W3C comme socle des systèmes d'information adaptables et collaboratifs. En plus, notons que les standards W3C des services Web présentent de nombreux apports par rapport aux modèles sémantiques proposés dans la littérature (Omrana, El Bitar *et al.*, 2013) ; (El Bitar, Belouadha *et al.*, 2013).
- **Le modèle de description utilisé.** Il met en évidence le modèle exploité pour l'extension du BPMN ainsi que le vocabulaire utilisé pour décrire les propriétés de performabilité. L'objectif est d'évaluer les approches étudiées en termes de leur conformité aux standards ainsi que leur niveau d'abstraction.
- **Le mécanisme d'annotation.** Il fait référence à l'objet annoté et aux éléments d'annotation. Ces critères devraient permettre d'évaluer la capacité de couvrir les besoins d'annotation des processus par des propriétés de performabilité, aussi bien que la complexité des modèles obtenus.

4.2. Évaluation des approches étudiées

Le tableau 2 en page suivante donne une synthèse de l'évaluation des approches présentées dans cet article.

Par objectif recherché. A l'exception des deux approches (Pavlovski et Zou, 2008) ; (Bartolini *et al.*, 2013), les contributions proposées dans la littérature y compris celles présentées dans cet article, ciblent exclusivement les besoins de performabilité ou de sécurité et non pas les deux à la fois. En plus, très peu d'approches (Bocciarelli et d'Ambrogio, 2014) sont dédiées à un contexte d'utilisation spécifique qui est l'architecture SOA et les services Web en particulier.

Sur un autre plan, les objectifs spécifiques visés par ces approches varient de la capture des besoins, la prédiction et l'analyse de la qualité, à la surveillance des processus ou encore l'automatisation de leur implémentation fondée sur la découverte des services Web. Elles se répartissent ainsi en approches contemplatives et productives.

Par modèle de description utilisé. Certaines approches analysées s'appuient sur des modèles de propriétés. Notons que la proposition de (Bocciarelli et d'Ambrogio, 2014) est fondée sur un modèle étendu du standard MARTE qui est un standard de description des propriétés non-fonctionnelles dans les systèmes temps réel. Néanmoins, cette approche, qui est dédiée au contexte des services Web, ne s'aligne pas avec les standards de cette technologie

pour prendre en considération les besoins de description des propriétés non-fonctionnelles (dont les propriétés de performabilité) dans ce contexte.

En ce qui concerne le vocabulaire utilisé pour spécifier les propriétés de performabilité des processus, il n'est pas unifié. Cela est dû au non-respect des standards. Notons également que ce vocabulaire reflète le niveau d'abstraction ou de détail de la représentation fournie par une approche. Cependant, les approches proposant des représentations détaillées telles que la contribution de (Bocciarelli et d'Ambrogio, 2014) n'est pas dédiée aux analystes métier. Son niveau de complexité requiert la collaboration avec des experts dans le domaine, voire la connaissance du modèle de description considéré.

Critères	Pavlovski et Zou, 2008	Saeedi, Zhao <i>et al.</i> , 2010	Bartolini <i>et al.</i> , 2013	Bocciarelli et d'Ambrogio, 2014	
Objectif recherché	Domaine ciblé	Performabilité/ Sécurité: • Performance, réglementation, Organisation, Sécurité...	Performabilité: • Performance, Fiabilité	Performabilité/ Sécurité: • Performance, Fiabilité, Sécurité	Performabilité: • Performance, Fiabilité
	Contexte des services Web	Non	Non	Non	Oui
	Objectifs spécifiques	• Capture des besoins	• Analyse	• Monitoring	• Analyse • Implémentation
Modèle de description utilisé	Modèle			• PMM	• MARTE
	Vocabulaire	• Contrainte • Risque • Contrôle	• Temps de réponse • Coût • Taux de fiabilité	• Valeur • Opérateur • Nature	• Durée • Fréquence • Origine • Type...
Mécanisme d'annotation	Objets annotés	• Tâche	• Tâche, <i>SubProcess</i> , <i>CallActivity</i>	• Tâche, Acteur, Chorégraphie	• Tâche, Processus, Collaboration
	Éléments d'annotation	• Artefacts • Catalogues	• Table de valeurs	• Instances • Attributs	• Annotations textuelles, Attributs

Tableau 2. Synthèse comparative des approches étudiées

Par mécanisme d'annotation. Les approches analysées se restreignent à l'annotation d'un objet BPMN spécifique comme la tâche (Pavlovski et Zou, 2008) ou tentent en plus d'annoter d'autres objets BPMN. Ce choix dépend du niveau d'abstraction de l'approche et ne respecte pas de norme. Dans le cas de la proposition de (Bocciarelli et d'Ambrogio, 2014) dont le contexte d'utilisation est les services Web, nous notons que le mécanisme d'annotation adopté n'est pas conduit par les exigences d'annotation sémantique et de description des propriétés non-fonctionnelles définies par les standards des services Web.

Concernant les éléments d'annotation, certaines approches étudiées utilisent indifféremment des symboles graphiques et/ou des annotations textuelles qui donnent lieu à des représentations informelles. Pour pallier ce problème, d'autres propositions telles que les contributions de (Bocciarelli et d'Ambrogio, 2014) et (Saeedi, Zhao *et al.*, 2010) utilisent des attributs pour décrire les annotations textuelles ou encore une table de valeurs pour décrire les propriétés.

Nous notons également que le fait d'introduire différents symboles et annotations dans les modèles de processus impacte leur lisibilité. Dans ce contexte, la proposition de (Bartolini *et al.*, 2013) prévoit de décrire le détail des annotations par des instances définies dans des paquetages UML séparés des modèles de processus. Néanmoins, la représentation proposée s'avère ainsi fragmentée.

5. Conclusions et perspectives

Le standard BPMN est le standard de facto de spécification des processus métier. Il assure la simplicité et l'expressivité de description du comportement fonctionnel des processus ainsi que le mapping automatique avec le langage d'exécution BPEL. En outre, il favorise la modélisation collaborative et garantit la description consolidée (compacte et non fragmentée) des processus. Toutefois, ce standard s'inscrit dans le cadre d'une approche de modélisation contemplative à des fins de documentation et de communication.

Des approches d'extension des modèles BPMN avec les besoins en performabilité ont été proposées dans la littérature pour permettre l'analyse automatique et anticipée des processus. Ces approches peuvent être classées selon l'approche d'annotation adoptée, en approches orientées artefacts, tables, annotation textuelle ou instances UML. Le présent article analyse quatre contributions parmi les plus récentes ou référencées dans ce domaine et dégage leurs limites. L'analyse effectuée permet de conclure que ces approches sont :

- **Spécifiques.** Elles ciblent généralement le domaine de performabilité (Korherr et List, 2007) (Magnani et Montesi, 2007) ; (Sampath et Wirsing, 2009) ; (Saeedi, Zhao *et al.*, 2010) ; (Bocciarelli et d'Ambrogio, 2014) sans s'étendre à d'autres domaines non-fonctionnels tels que la sécurité. En plus, elles ne prennent pas en compte l'aspect sémantique. Aucune des approches ne permet d'annoter les processus avec des propriétés non-fonctionnelles et sémantiques à la fois.
- **Productives mais axées sur l'analyse plutôt que sur l'implémentation.** Elles s'inscrivent, en général, dans le cadre d'une approche de modélisation qui n'est pas juste contemplative, mais qui conduit surtout l'analyse et la surveillance des processus. Néanmoins, très peu d'approches s'intéressent à l'implémentation automatique des processus.
- **Non conformes aux standards.** Elles ne s'alignent généralement pas avec les standards. Celles utilisées en particulier dans le contexte des services Web ne sont pas conformes aux standards de description des aspects sémantique et non-fonctionnel des services Web.
- **Dédiées exclusivement aux analystes métier ou aux experts.** Elles proposent en général un niveau élevé d'abstraction ou de complexité pour la description des propriétés de performabilité et sont ainsi, destinées soit aux analystes métier, soit aux experts du domaine.
- **Inadaptées au besoin d'évolutivité ou passage à l'échelle (*scalability*).** Elles convergent par l'introduction explicite des annotations de performabilité au niveau du diagramme BPMN. Cela conduit à une densité élevée d'informations au sein du modèle de processus et risque, ainsi, de le rendre illisible et de compliquer son interprétation.

De ces constats, une approche qui contribue à pallier ces limites semble importante. Elle serait encore plus intéressante si elle s'inscrivait dans le cadre d'une approche de modélisation qui pourrait conduire l'analyse, mais aussi l'implémentation automatique des processus. Notons que le couple BPM/SOA consistant à implémenter des processus par des services, en particulier des services Web, est considéré dans l'industrie logicielle comme socle des systèmes d'information adaptables et collaboratifs. Vu l'intérêt de ce couplage, une approche centrée sur le respect des standards W3C des services Web qui normalisent la spécification des aspects fonctionnel, non-fonctionnel et sémantique des services, serait intéressante. Elle devrait permettre de produire des modèles BPMN de processus sémantiques annotés avec les besoins non-fonctionnels tels que les besoins de performabilité et de sécurité, tout en palliant les limites des approches connexes. Cela contribuerait également à l'implémentation automatique de processus sémantiques, performants et sécurisés à travers la découverte et la composition des services Web en alignement avec les standards W3C (Belouadha, Omrana *et al.*, 2010) ; (Omrana, Belouadha *et al.*, 2012). C'est dans ce contexte que nous proposons l'approche IBPM (*Intelligent Business Process Manufacturing*) (Belouadha, 2015). Cette approche est fondée sur eBPMN (*enhanced BPMN*), une extension de BPMN conforme aux standards des services Web. Elle permet à la fois la modélisation des aspects sémantique et non-fonctionnel des processus métier et leur implémentation automatique par réutilisation des services Web.

6. Références

- Bartolini, C., Bertolino, A., De Angelis, G., Ciancone, A., Mirandola, R. (2012, June). *Non-functional analysis of service choreographies*. in *Principles of Engineering Service Oriented Systems (PESOS)*, ICSE Workshop, Zurich, June, pp. 8-14.
- Bartolini, C., Bertolino, A., De Angelis, G., Ciancone, A., Mirandola, R. (2013, March). *Apprehensive QoS monitoring of service choreographies*. in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, Coimbra, Portugal, March 18 - 22, pp. 1893-1899. ACM.
- Belouadha, F.-Z., Omrana, H., Roudiès, O. (2010). *A MDA approach for defining WS-Policy semantic non-functional properties*. In *International Journal of Engineering Science and Technology (IJEST)*, Vol. 2, Issue 6, juin 2010, pp. 2164-2171.

- Belouadha, F.Z., Omrana, H., Roudiès, O. (2012). *Web services-enhanced agile modeling and integrating business processes*. In E-Business - Applications and Global Acceptance, Prncelo Ifinedo (ed.), InTech, pp. 73-98.
- Belouadha, F.Z., Omrana, H., Roudiès, O. (2014). *Modeling and matching business processes using Web services*. ICMCS'14 International Conference, Marrakech, April 14-16, pp. 617-621.
- Belouadha, F.Z. (2015). *eBPMN pour la fabrication intelligente des processus métier*. Thèse pour l'obtention de doctorat en Génie Informatique, Centre d'études doctorales Sciences et techniques pour l'ingénieur, École Mohammadia d'Ingénieurs, juin 2015.
- Bernardi, S., Merseguer, J., Petriu, D. C. (2008). *Adding dependability analysis capabilities to the MARTE profile*. In Model Driven Engineering Languages and Systems (MoDELS '08), pp. 736-750, Springer Berlin Heidelberg.
- Bocciarelli, P., d'Ambrogio, A. (2011). *A BPMN extension for modeling non functional properties of business processes*. In proceedings of the 2011 Symposium on Theory of Modeling and Simulation: DEVS Integrative M&S Symposium, pp. 160-168, Society for Computer Simulation International.
- Bocciarelli, P., d'Ambrogio, A. (2014). *A model-driven method for enacting the design-time QoS analysis of business processes*. Software & Systems Modeling, Vol. 13 (2), pp. 573-598.
- Ciancone, A., Filieri, A., Drago, M. L., Mirandola, R., Grassi, V. (2011). *KlaperSuite: an integrated model-driven environment for reliability and performance analysis of component-based systems*. In Objects, Models, Components, Patterns, TOOLS'11, Zurich, Switzerland, June 28-30, LNCS, Vol. 6705, pp. 99-114, Springer Berlin Heidelberg.
- Di Marco, A., Pompilio, C., Bertolino, A., Calabrò, A., Lonetti, F., Sabetta, A. (2011). *Yet another meta-model to specify non-functional properties*. In proceedings of the International Workshop on Quality Assurance for Service-Based Applications, pp. 9-16, ACM.
- El Bitar, I., Belouadha, F. -Z., and Roudiès O. (2013). *Review of Web Services Description approaches*. 8th International Conference on Intelligent Systems: Theories and Applications (SITA'13), May 8-9, Rabat, Morocco.
- Korherr, B., List, B. (2007). *Extending the EPC and the BPMN with Business Process Goals and Performance Measures*. In Ninth International Conference on Enterprise Information Systems (ICEIS'07), Madeira, Portugal, pp. 287-294.
- Magnani, M., Montesi, D. (2007). *Computing the cost of BPMN diagrams*. Technical Report. University of Bologna. Available at: <http://www.cs.unibo.it/pub/TR/UBLCS/2007/2007-17.pdf>.
- Mylopoulos, J., Chung, L., Nixon, B. (1992). *Representing and using nonfunctional requirements: A process-oriented approach*. IEEE Transactions on Software Engineering, Vol. 18, N. 6, pp. 483-497.
- OASIS-BPEL (2007). *Web Services Business Process Execution Language*. Version 2.0. Primer, May 9. Available at: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel.
- OMG-BPMN (2013). *Business Process Model and Notation (BPMN)*, Version 2.0.2, December 2013. Available at: <http://www.omg.org/spec/BPMN/2.0.2/PDF/>.
- OMG-MARTE (2011). *UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems*, Version 1.1, June, Available at: <http://www.omg.org/spec/MARTE/1.1/PDF/>.
- Omrana, H., Belouadha, F.-Z., Roudiès, O. (2012). *Template-based Matching Algorithm for Dynamic Web Services Discovery*. International Journal of Information and Communication Technology (IJICT), Vol.4, N°. 2, Inderscience Publisher.
- Omrana, H., El Bitar, I., Belouadha, F. Z., and Roudiès, O. (2013). *A Comparative Evaluation of Web Services Description Approaches*. In Proceedings of the IEEE 10th International Conference on Information Technology: New Generations (ITNG '13), April 2013, Las Vegas, Nevada, USA, pp. 60-64.
- Owen, M., Raj, J. (2003). *BPMN and Business Process Management: Introduction to the New Business Process Modeling Standard*. Popkin Software, available at: http://www.omg.org/bpmn/Documents/6AD5D16960.BPMN_and_BPM.pdf.
- Pavlovski, C. J., Zou, J. (2008). *Non-functional requirements in business process modeling*. In proceedings of the fifth Asia-Pacific conference on Conceptual Modelling, Vol. 79, pp. 103-112, Australian Computer Society, Inc.
- Petriu, D., Bernardi, S., Merseguer, J. (2008). *An UML profile for dependability analysis and modeling of software systems*. Technical report, Universidad de Zaragoza, Spain.
- Saeedi, K., Zhao, L., Sampaio, P. (2010). *Extending BPMN for supporting customer-facing service quality requirements*. In Web Services (ICWS'10), pp. 616-623.
- Sampath, P., Wirsing, M. (2009). *Computing the cost of business processes*. In Third International United Information Systems Conference (UNISCON'09), April 21-24, Sydney, Australia. Published in Information Systems: Modeling, Development, and Integration, Lecture Notes in Business Information Processing, Vol. 20, pp. 178-183, Springer Berlin Heidelberg.
- Smith, H., Neal, D., Ferrara, L., Hayden, F. (2002). *The emergence of business process management*. White paper, Computer Sciences Corporation's Research Services., Available at: http://itexpert.ir/download/ebooks/ITExpert.ir_BPM.pdf.
- Zou, J., Pavlovski, C. J. (2006). *Modeling architectural non functional requirements: from use case to control case*. In IEEE International Conference on e-Business Engineering (ICEBE'06), Shanghai, 24 October, pp. 315-322.