



HAL
open science

Business Process Models Quality : evaluation and improvement

Sarah Ayad

► **To cite this version:**

Sarah Ayad. Business Process Models Quality : evaluation and improvement. Business administration. Conservatoire national des arts et metiers - CNAM, 2013. English. NNT : 2013CNAM0922 . tel-01127026

HAL Id: tel-01127026

<https://theses.hal.science/tel-01127026>

Submitted on 6 Mar 2015

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.

**ÉCOLE DOCTORALE Informatique, Télécommunications et
Électronique (Paris)**

**Centre d'étude et de recherche en informatique et communications
(EA 4629)**

Ingénierie des Systèmes d'Information et de Décision (ISID)

THÈSE présentée par :

Sarah AYAD

Soutenue le 18 decembre 2013

pour obtenir le grade de : **Docteur du Conservatoire National des Arts et
Métiers**

Discipline/ Spécialité : Informatique

Business Process Models Quality: Evaluation and Improvement

THÈSE dirigée par :

Mme. Isabelle Comyn-Wattiau Professeur au CNAM

Mme. Samira Si-Said Cherfi Maître de Conférences- CNAM

RAPPORTEURS :

Mr. Panetto Hervé

Professeur à l'université de Lorraine

Mr. Poels Geert

Professeur à l'université de Ghent

JURY:

M. Hervé Panetto

Professeur à l'Université de Lorraine

M. Geert Poels

Professeur à l'Université de Ghent

Mme. Camille Rosenthal-Sabroux

Professeur à l'Université Paris-Dauphine

Mme. Selmin Nurcan Lakhdari

Maître de Conférences à l'Université Paris1-Sorbonne

Mme. Kathia De Oliveira

Maître de Conférences à l'Université de Valenciennes

Mme. Isabelle Comyn-Wattiau

Professeur au CNAM

Mme. Samira Si-Said Cherfi

Maître de Conférences au CNAM

Acknowledgements

This thesis is the result of three years of devoted work which would not have been possible without the support of many. Here, I would like to express my thanks to people who have been very helpful to me during the course of this work.

First of all, I would like to thank my two supervisors, **Isabelle Comyn Wattiau** and **Samira Si Said Cherfi** for their patience, guidance, encouragement, and their advices throughout my thesis. Thank you for your confidence in me manifested in agreeing to supervise my doctoral work, for all your advices and all the time devoted in directing this research. I also would like to take the opportunity to thank all the members of the research group ISID starting with the director of the research group ISID, **Prof. Jacky Akoka**, as well as all the group members.

My special gratitude goes to my spervisor **Samira Si Said Cherfi**. You were more than a supervisor, you were like a mother for me in a foreign country. Because of your patience, I learned how to be rigorous not only in my professional life but also in my personal life. You were always there for me, day, night, and even on weekends. I know that I am so lucky to have you as a supervisor. You are a role model to me both as a researcher and a person. Thank you for everything.

I would like to thank **Prof. Geert Poels** and **Prof. Herve Panetto** for accepting to review my dissertation. Also many thanks to **Prof. Camille Rosenthal-Sabroux**, **Dr. Selmin Nurcan** and **Dr. kathia Oliveira** for accepting to be a part of jury to evaluate my work.

There are a numerous people with whom I shared the best moments and to whom I am greatly indebted. Dana Azar, you were more than a friend. Rima Daezli, Zeinab Hmedeh, Amina Souag, Manal khalife, Lydia Khalifa, Ryadh Dahimen, Rodney Medina and Sally El Jam. You turned a hard experience into unforgettable memories.

A special thanks are due to Nesrine Baltahji and Basem Alameddine, without whom I couldn't came to france in the begining.

And finally, to my parents who believed in me and supported me, not only morally but also financially.

To father, Mohamad Ayad. The only purpose of my life is to make you proud of me. Thanks for believing in me.

To mother, Mona El kheir, without her continuous support and encouragement I never would have been able to achieve my goals. This one is for you mom!

To Lara Elias, thank you for being there for me. Thanks for being my guardian angel. Sometimes i forget to tell you how much i appreciate you for being an important part of my life. Thank you sis.

To my brother, Ashraf, you are someone to lean on, someone to count on, someone to tell on.

Thank you.

Résumé

La problématique scientifique abordée correspond à la modélisation et à l'amélioration des processus métiers. Ce problème est d'un intérêt croissant pour les entreprises qui prennent conscience de l'impact indéniable que peuvent avoir une meilleure compréhension et une meilleure gestion des processus métiers (PM) sur l'efficacité, la cohérence et la transparence de leurs activités. Le travail envisagé dans le cadre de la thèse vise à proposer une méthode et un outil pour mesurer et améliorer la qualité des modèles de processus métier. L'originalité de l'approche est qu'elle vise non seulement la qualité syntaxique mais aussi la qualité sémantique et pragmatique en s'appuyant notamment sur les connaissances du domaine.

1. Introduction

Une enquête récente de Gartner de plus de 1400 directeurs de l'information a révélé que la priorité de l'entreprise identifiée par leur compagnie était l'amélioration des processus métier (Camp and DeBlois 2007). Et un des éléments clé de la gestion des processus métier (PM) est de rendre le processus explicite pour différents acteurs par la modélisation de ces processus métier (Sinur and Hill 2010).

Ainsi, les avantages de la modélisation des PM ont été étudiés par (Havey 2005), dont comprennent:

a) la formalisation des processus existants, b) la facilitation du flux des processus, c) l'augmentation de la productivité, et enfin d) la simplification de la réglementation et les questions de conformité.

L'amélioration de la compréhension, la fiabilité et la réutilisation des modèles de PM exigent qu'on leur confère une certaine qualité. Diverses approches pour l'amélioration de la qualité des modèles de PM ont été proposées dans la littérature. Notre étude nous a permis de les classer en trois catégories :

1) Les approches centrées sur les méthodes dont l'hypothèse est que l'amélioration du processus de développement soutenu par méthode permettront d'améliorer le résultat de ce processus. Les approches centrées sur les méthodes couvrent toutes les recherches qui proposent des guides méthodologiques et des bonnes pratiques pour assurer la qualité des modèles de PM produits. Dans (Becker, Rosemann, and Uthmann 2000) les auteurs proposent un ensemble de guides pour améliorer certaines caractéristiques telles que l'exactitude, la compréhension etc. des modèles de PM. Une autre approche proposée dans (Mendling, Recker, and Reijers 2010) discute de l'impact d'une bonne documentation, de règles de nommage et d'icônes graphiques adéquates sur l'amélioration de la compréhension et de l'adoption des modèles de PM. D'autres auteurs proposent des motifs de conception réutilisables qui aident à produire des modèles de PM (Van Der Aalst et al. 2003).

2) les approches d'évaluation de la qualité des PM : Les approches d'évaluation de la qualité des PM s'intéressent à ces derniers au niveau de leur exécution et de leur contrôle. Dans (Jansen-vullers and Netjes 2006) les auteurs présentent plusieurs techniques pour la vérification, la validation et l'amélioration des performances des PM.

3) les approches d'évaluation de la qualité des modèles de PM. Cette évaluation passe souvent par la mesure de la qualité à travers la définition de métriques de qualité. Les approches qui s'inscrivent dans ce courant mentionnent les similitudes entre un processus logiciel et un processus métier. Ils appliquent les métriques logicielles pour évaluer les modèles de processus métier. Dans (Aguilar et al. 2006) un ensemble de mesures pour l'évaluation de la maintenabilité des modèles des modèles de PM est défini. L'applicabilité des métriques de qualité dans la gestion des processus métier est étudié dans (Cardoso et al. 2006), (Mendling, Reijers, and Cardoso 2007) and (Van Belle 2004) décrit un cadre permettant l'évaluation et la comparaison des modèles de PM en se basant sur une analyse syntaxique, sémantique et pragmatique. Ils mentionnent cinq des mesures qu'ils considèrent importantes : couplage, cohésion, complexité, modularité, et enfin taille. Dans (Gruhn and Laue, 2006) les auteurs ont présenté un ensemble d'enquêtes où ils ont étudié l'importance de l'exactitude des modèles de PM d'un point de vue empirique. Ils ont défini deux métriques qui quantifient la bonne organisation structurelle (structuredness). Enfin, les auteurs dans (Vanderfeesten et al. 2008) proposent des métriques pour mesurer la cohésion des modèles de PM. En conclusion, nous avons étudié cette catégorie en termes de qualité syntaxique, sémantique et pragmatique qualité emprunté (Lindland et al. 1995)

Mais nous avons constaté certaines lacunes liées à la littérature. En fait,

- Très peu de méthodes fournissent des instructions et des directives pour les défauts de notation. Mais aucune de ces méthodes ne fournissent des instructions correctives pour corriger les défauts sémantiques et pragmatiques.
- D'autre part, la plupart des méthodes ne vérifient pas l'application des normes de notations, ni les outils de modélisation détectent de tels défauts. Par exemple, dans BPMN, l'une des normes de base est que chaque modèle devrait avoir au moins un 'end event', mais aucun des outils de modélisation que nous avons testés, ne détecter un tel défaut.

L'un des principaux obstacles à l'évaluation de la qualité des modèles de processus métier est le manque de mesures de qualité. La plupart des approches d'évaluation existantes se concentrent sur l'évaluation de la qualité syntaxique des modèles de PM: la complexité, la taille, la cohésion, etc. En outre, il existe des approches pour évaluer le comportement des modèles BP (blocages, manque de synchronisation, etc.). Mais il y a très peu d'approches qui fournissent des méthodes d'évaluation de la qualité pour l'évaluation de la qualité sémantique et pragmatique des modèles.

D'autre part, l'évaluation de la qualité n'est qu'une étape pour améliorer la qualité des modèles de processus métier. La plupart des cadres de qualité se concentrent exclusivement sur la détection des défauts (évaluation de la qualité) et ignorer l'aspect de la correction de défaut (amélioration de la qualité).

La modélisation des PM comme toute activité de modélisation consiste à créer un modèle. Au cours du processus de modélisation, le modelleur est préoccupé par l'identification, l'analyse et la

conceptualisation des notions essentielles et les contraintes du domaine d'application. En fait, lorsque les états initiaux et objectifs du modèle sont vaguement définis ou peu claires, et quand il a de multiples solutions et si elle ne contient pas toutes les informations nécessaires à leur solution; par conséquent, il n'est pas clair quelles actions sont nécessaires pour les résoudre, il est un modèle de mauvaise structure.

La question de recherche qui se pose est : Comment aider les modélisateurs à améliorer la qualité de leurs modèles de processus?

2. Etat de l'art

La modélisation est une activité difficile, particulièrement la modélisation des PM.

Et afin d'aider les modélisateurs de PM à améliorer la qualité de leurs modèles, nous avons étudié les problèmes de modélisation.

Nous avons constaté que la maturité des experts de modélisation de processus est faible.

En outre (Coughlan et al. 2003) déclare que l'extraction des exigences nécessite une communication intense pour surmonter tout écart de culture ou des différences sémantiques qui peuvent exister entre les utilisateurs et les développeurs.

En outre, différents chercheurs ont étudié la compréhension d'un schéma conceptuel utilisé dans les premiers stades de développement de systèmes. Les résultats de ces études indiquent que les connaissances de domaine (CD) peuvent compenser un manque de clarté et d'information manquante dans la représentation d'un problème.

(Antón and Potts 1998), qui rapportent une étude de terrain visant à identifier les facteurs qui conduisent à des malentendus concernant les exigences conclut que les facteurs dominants que les utilisateurs et les développeurs indiquent notamment la manque de compréhension de l'entreprise et des difficultés liées à la langue et la terminologie commune de développeurs. Ces deux facteurs proviennent de développeurs possédant un bas niveau de CD.

Sur la base de (Khatri et al. 2006), il y a de deux types de connaissances:

- IS domain knowledge
- Application domain knowledge

Dans le cadre de cette thèse, nous proposons une solution qui capture ces deux types de connaissances.

Par conséquent les modèles de PM sont créés tenant compte, l'exigence de domaine, sous des contraintes spécifiées, en s'appuyant sur les connaissances à la fois du domaine et de la notation utilisée.

La question qui se pose : Comment exploiter les connaissances de domaine afin d'aider les modélisateurs de PM dans leur activité de modélisation?

3. L'approche proposée

Figure 1: Vue d'ensemble de l'approche proposée

La modélisation des processus métiers permet notamment de mieux les comprendre pour pouvoir détecter les axes d'amélioration, les redondances, les tâches sans valeur ajoutée, etc. Analyser les modèles sur le seul plan structurel limite considérablement les possibilités d'amélioration. C'est pourquoi, nous proposons de nous appuyer sur des ontologies de domaine pour améliorer la qualité sémantique des modèles de PM. Notre travail comprend le développement d'un prototype qui met en oeuvre la méthode développée, laquelle s'appuie sur des ontologies de domaine, sur des modèles de processus métier et sur une approche de mesure de la qualité. L'expérimentation et la validation des hypothèses de recherche seront faites sur des cas réels.

- La première étape de ce travail est d'abord de définir un métamodèle pour les processus métier et un autre pour les ontologies de domaine afin d'avoir une approche indépendante de la notation.
- La détection des défauts s'appuie sur la découverte de similitudes entre les éléments du modèle de processus et les concepts de l'ontologie de domaine.
- Nous proposons un ensemble de métriques pour mesurer la qualité des processus métier, allant au delà des considérations syntaxiques en intégrant aussi le sens des modèles avec les besoins et exigences du domaine. Ainsi nous visons la définition d'une méthode de conception des processus métier intégrant la dimension de la qualité.
- Nous proposons un ensemble des actions correctives pour assurer aux modélisateurs des corrections pour chaque défaut de qualité détecté par les métriques.

3.1. IS Domain knowledge

IS domain knowledge représente la connaissance fournie par les méthodes, les notations, les pratiques et les outils.

La solution proposée prend en compte les principaux obstacles à ce genre d'exploitation des connaissances. Les principaux obstacles que nous avons identifiés sont:

- Manque de centralisation: il existe dans la littérature, ainsi que dans les manuels de notations, des dizaines de règles et conseils sur le bon usage des concepts. Cette connaissance est généralement étalée partout, dans les documentations d'outil, sur les sites Web, dans les documents de recherche et dans une variété d'autres sources faisant son exploitation difficile. Notre solution proposée rassemble ces connaissances et propose un outil automatisant son exploitation pour prendre en charge la détection et la correction des défauts liés à la conformité de notation.

- Absence de formalisme: Pendant notre analyse de la connaissance du domaine, se révèle l'hétérogénéité de formalisation. Alors que certains guides sont exprimées à l'aide des langages formels, d'autres sont textuelles et vague et certains sont tout simplement liés à l'expertise de modélisation et ne sont jamais exprimés explicitement. Nous proposons d'utiliser (Object Constraint Language OCL) pour exprimer les règles et les conseils définissant la «meilleure» utilisation de concepts. Ces contraintes, formellement exprimé, sont donc faciles à corriger et à utiliser dans un support automatisé.

- Hétérogénéité des notations: il existe une variété de notations pour la modélisation des PM ayant chacun son propre vocabulaire et ses règles spécifiques. Pour assurer la généricité de notre approche, nous construisons notre raisonnement au niveau d'un métamodèle de PM. Les contraintes OCL qui sont communes et valable dans toutes les notations sont généralisées et donc écrites au niveau métamodèle qui les rend indépendants de notations spécifiques. Il reste cependant quelques concepts spécifiques qui ne vaut que pour quelques notations. Notre approche permet également la définition des contraintes au niveau de la notation de gérer des situations spécifiques. Nous proposons de définir un métamodèle pour aligner les modèles avec les concepts du métamodèle. Deux niveaux de constraints sont définis: générique (valable pour toutes les notations) et des contraintes spécifiques de notation.

Figure 2: Métamodèle pour les modèles de PM

Le métamodèle des modèles de PM est le résultat d'un travail de fédération de travaux existants.

L'extrait du métamodèle est présentés dans la Figure 2.

Un processus métier est composé de flux d'objets, d'artefacts et de connecteurs. Un flux d'objet peut être un branchement conditionnel, un événement ou une activité. Les connecteurs peuvent être des associations, séquences ou flux de message. Les activités exigent des ressources. Ces dernières peuvent être des informations ou des objets.

Il existe aussi de nombreuses associations entre les différents connecteurs et les flux d'objets qui ne sont pas présentés dans ce métamodèle. Mais ils sont définis par les règles OCL qui font partie du métamodèle.

3.2. Application domain knowledge

Application domain knowledge fait référence à des problèmes du monde réel. Basé sur (J. Buchan 2009) deux des obstacles les plus importants identifiés sont : la diversité entre les groupes et le manque d'un vocabulaire commun.

- La diversité dans le groupe concerne les différents niveaux de connaissances et des techniques.
- Un manque d'un vocabulaire commun, qui peut être dérivé de CD, peut entraîner des interactions et à la confusion et les malentendus.

Notre solution repose sur l'existence de la connaissance du domaine décrivant une connaissance et un vocabulaire partagé et accepté dans des domaines spécifiques. Cette connaissance est capturée par les moyens d'ontologies de domaine.

Notre objectif est d'utiliser les connaissances de domaine pour aider à l'amélioration d'un modèle de processus métier en cours de construction.

La définition de ces ontologies n'est pas dans notre champ. Ontologies de domaine sont disponibles dans la littérature.

Figure 3: Métamodèle pour les ontologies (Purao and Storey 2005)

Une ontologie est composée de classes et de relations. Une classe représente un concept identifiable. Une classe peut avoir des liens structurels et/ou sémantiques avec d'autres classes. Pour préciser le concept de classe nous avons adopté la classification proposée par (Purao and Storey 2005). Une classe peut être un acteur, une action ou un artefact. Un acteur est une entité capable d'accomplir une action. L'action représente l'accomplissement d'un acte sur un objet. Enfin, l'artefact est un objet inanimé incapable d'accomplir une action.

3.3. Exploitation de IS domain knowledge

Figure 4: Exploitation de IS domain knowledge

Pour exploiter les connaissances du domaine, en se basant sur les règles d'alignement déjà définies, on annote les modèles de processus métier afin d'aligner les modèles avec les concepts du métamodèle et d'appliquer des contraintes OCL.

On aura un modèle de PM exporté dans un fichier XML ou XPDL, dont on applique la dessus les contraintes.

- Processus d'annotation : Des correspondances entre chaque concept d'une notation et la metamodel des PM sont définis. Par exemple, nous avons dans BPMN un 'Lane' qui sera annoté par un 'acteur' dont représente le responsable de l'exécution d'une activité. Dans Penker-Eriksson, un ressource humaine, responsable de l'exécution d'un processus, sera également annotés par un acteur. Toutes les contraintes de notation identifiés sur l'acteur dans le métamodèle seront appliqués a la fois sur le 'Lane' et sur le ressource humaine dans chacune de ces notations.

3.4. Exploitation de Application domain knowledge

Dans une première étape, nous avons d'abord défini des alignements entre les concepts des métamodèles. Grâce à la catégorisation précise des concepts nous sommes en mesure d'établir des correspondances entre les concepts. Un extrait des correspondances entre les concepts des deux métamodèles est donné dans le Table 1.

Table 1: Correspondances entre les concepts des deux métamodèles

BP metamodel concept Domain Ontology metamodel concept

Actor Actor

Abstract resource Abstract

Information resource knowledge

Process / activity Action

Sequence flow Temporal

Message flow Communication

Transfer

Role Execution

Manipulation

Observation

Influence

Dans une deuxième étape, en nous fondant sur la sémantique des concepts, nous avons développé un ensemble de règles de correspondance, permettant la réconciliation de l'ontologie de domaine avec les concepts des modèles de processus.

1. La similitude basée sur le nom: renvoie les concepts d'ontologie qui sont syntaxiquement équivalent au concept du processus métier. L'équivalence syntaxique est calculée en se basant sur la distance de Levenshtein ("Distance de Levenshtein" 2014).
2. La similitude partielle base sur le nom.
3. Synonymie: renvoie les concepts d'ontologie qui sont des synonymes du concept de processus métier. La synonymie est calculée en se basant sur des distances de Wordnet : Resnik information content (Resnik 1995), Jiang et Conrath (Jiang and Conrath 1997), Lin (Lin 1998), Wu and Palmer

path length (Wu and Palmer 1994), Purandare & Pedersen context vectors (Purandare and Pedersen, 2004).

4. Niveau plus général: Cette fonction retourne la liste des concepts de l'ontologie qui représentent un niveau plus général du concept du PM. Ce qui veut dire, il existe une relation de supériorité (également appelé hyperonymes ou IS- relation) entre le concept de processus métier en paramètre d'entrée et les concepts de l'ontologie en retour. En d'autres termes, il renvoie le point d'arrivée (Range) de la relation IS-A.
5. Niveau plus spécifique: Cette fonction renvoie les concepts de l'ontologie ayant une relation d'infériorité avec un concept de l'ontologie détecté comme synonymes ou syntaxiquement équivalent à l'élément de modèle BP.
6. Les concepts associés: Cette fonction renvoie un ensemble de concepts de l'ontologie liés aux résultats obtenus par la fonction similitude basée sur le nom ou par la similitude partielle basée sur le nom par une relation "temporelle" ou "influence" à une autre ontologie notion.

4. Mesure et évaluation de la qualité

Nous évaluons la qualité des modèles en termes de syntaxique, sémantique et pragmatique sur la base de Lindland et al. (Lindland, Sindre, and Solvberg 1994).

Figure 5 : Lindland et al. (Lindland, Sindre, and Solvberg 1994)

Qualité syntaxique: Concerne la relation entre le modèle et la langage de modélisation en décrivant les relations entre les constructions de langage sans tenir compte de leur signification. Il n'y a qu'un seul objectif syntaxique: l'exactitude syntaxique, toutes les déclarations dans le modèle sont conformes à la syntaxe.

Qualité sémantique: Concerne le modèle du domaine en considérant non seulement la syntaxe, mais aussi les relations et leur signification entre les assertions.

Il y a deux objectifs sémantiques: la validité et la complétude. La validité signifie que toutes les déclarations faites par le modèle sont correctes et pertinentes au problème, tandis que, la complétude signifie que le modèle contient toutes les déclarations sur le domaine qui sont correctes et pertinentes.

Qualité pragmatique: Concerne la participation du public en tenant compte non seulement la syntaxe et la sémantique, mais comment le public (toute personne impliquée dans la modélisation) les interprétera.

4.1. Evaluation de la qualité syntaxique

Afin d'évaluer la qualité syntaxique, nous avons identifié un ensemble de mesures de la qualité syntaxique basé sur le métamodèle BP.

4.2. Evaluation de la qualité sémantique

Selon Guizzardi et al. (Guizzardi, Ferreira Pires, and Sinderen van 2002) , deux positions de la sémantique d'un modèle doivent être séparés:

En premier lieu, la sémantique comprend les règles d'interprétation automatique d'un modèle, c'est-à-dire les concepts sémantiques et son rôle dans le modèle. Deuxièmement, chaque construction de modélisation représente certaines entités du monde réel, ce qui est un autre aspect de la sémantique de modèle et est exprimé par les étiquettes des fonctions et des événements.

Pour évaluer la qualité sémantique, nous avons identifié un ensemble de ce que nous appelons défauts d'expressivité tels que les concepts dépourvus de sens, des représentations ambiguës et incomplet. Ces défauts correspondent à des choix de modélisation qui pourraient diminuer l'expressivité des modèles, conduisant à des erreurs dans leur interprétation et leur mise en oeuvre. Ces modèles conduisent à des systèmes inadéquats en raison du caractère incomplet ou à des malentendus au cours de leur mise en oeuvre.

- Défaut de clarté: désigne les ambiguïtés qui conduisent souvent à des erreurs d'interprétation. Une ambiguïté survient lorsque plusieurs termes dans le modèle sont utilisés pour désigner des concepts distincts alors que, d'après l'ontologie du domaine, ces termes sont identifiés comme synonymes.
- Défaut d'abstraction: caractérise le choix d'un niveau d'abstraction non approprié. En effet, dans certains cas, l'utilisation de concepts généraux au lieu de concepts spécifiques peut créer une confusion et avoir un impact sur l'efficacité de l'exécution du processus. Au contraire, dans d'autres situations, l'utilisation de termes précis peut, lors de l'exécution, empêcher par exemple la substitution de ressources ou la délégation de tâches etc. Le choix pertinent d'un niveau d'abstraction dépend de plusieurs facteurs parmi lesquels nous pouvons mentionner l'usage du modèle de PM (les développeurs ou les utilisateurs), l'objectif du modèle (explication ou mise en oeuvre), etc.
- Défaut de complétude: se produit quand un concept est complexe et quand une partie de ses composantes seulement est représentée dans le modèle. Il est nécessaire de vérifier si la couverture partielle des composants est voulue ou est due à une méconnaissance du domaine du problème.

4.3. Mesure des défauts d'expressivité

Nous proposons, en premier lieu, un métrique de qualité permettant de mesurer la clarté sémantique. La clarté calcule le rapport de 1 au nombre total de synonymes détectés dans l'ontologie.

Clarity (o) = $1 / |C_synonyms(o)|$

Si $|C_synonyms(o)| = 0$ le concept est dépourvu de sens.

Pour évaluer le niveau d'abstraction, on a défini deux métriques de qualité :

- Généralité: calcule le rapport de 1 au nombre total de hyponymes détectés dans l'ontologie.

Generality (c) = $1 / |hyponyms(c)|$

Si $|hyponyms(c)| = 0$ et inconnu sinon

- Spécialisation: calcule le rapport de 1 au nombre total de hyperonymes détectés dans l'ontologie.

Specialization (c) = $1 / |hypernyms(c)|$

Si $|hypernyms(c)| = 0$ et inconnu sinon

Pour évaluer l'incomplétude, on a défini quatre métriques :

- Input incompleteness: Chaque activité dans le PM a besoin de consommer/produire des ressources. Donc l'incomplétude est détecté si les entrées et / ou sorties sont manquantes pour une activité.
- Actor allocation incompleteness: Chaque activité est attribué à un acteur responsable de son exécution, nous pouvons compter sur les connaissances fournis par l'ontologie pour compléter le PM.
- Model richness: la richesse du modèle est liée à la couverture totale du domaine.

Pour évaluer le modèle richesse, nous avons défini deux mesures: Model activity richness et model resource richness.

Model Activity Richness (c) = $Nb\ of\ activities\ in\ the\ model / Nb\ of\ actions\ in\ the\ ontology$

Model Resource Richness (c) = $Nb\ of\ resources\ in\ the\ model / Nb\ of\ artefacts\ in\ the\ ontology$

4.4. Evaluation de la qualité pragmatique

Afin d'évaluer la compréhensibilité, nous avons identifié un ensemble de métriques de qualité pragmatiques.

5. Amélioration des défauts d'expressivité

Le processus d'amélioration comprend trois étapes.

-
1. La première étape réalise les améliorations dans le but de corriger les défauts captés par l'application des contraintes.
 2. La deuxième étape consiste à étiquette refactoring.
 3. La troisième étape met en œuvre l'amélioration qui corrige les défauts capturés par l'ontologie.

L'activité d'amélioration de la qualité fournit à l'analyste ou à l'expert qualité un ensemble de directives d'amélioration afin de corriger les défauts. Dans tous les cas, les améliorations sont à confirmer par l'expert.

- Les contraintes sont violées quand un défaut de qualité est capturé. Le but est de résoudre les défauts de qualité liés à la façon dont les concepts sont utilisés dans le modèle c.a.d l'utilisation du concept correctement et la façon dont ce concept est élaboré, ainsi ces contraintes soulignent le problème, mais ne les résout pas. Nous visons à proposer l'amélioration de chacune de ces défauts.
- Refactoring des étiquettes d'activité consiste à catégoriser les étiquettes dans des classes et l'extraction informations de ces étiquettes.
- Base sur les valeurs des métriques de qualité, des défauts de qualité sont détectés. En fait, la qualité sémantique dépend de la connaissance fournie par l'ontologie. De même, les résultats des métriques signalent les problèmes, mais ne les résout pas. Nous visons à proposer des améliorations à chacune de ces défauts.
 - Correction des défauts de clarté: consiste à remplacer les concepts synonymes, si cette synonymie est confirmée par l'expert ou l'analyste, par un nom de concept unique. Une fois encore, l'ontologie fournit la liste des synonymes pour aider l'analyste à choisir le terme approprié.
 - Correction des défauts d'abstraction: en fonction de la situation, les concepts pourraient être remplacés par d'autres plus génériques si plus de généralité est nécessaire. Cependant, si des descriptions plus détaillées sont nécessaires, le concept général dans le modèle est remplacé par d'autres plus spécifiques.
 - Correction des défauts d'incomplétude: l'amélioration de la complétude consiste à s'appuyer sur les connaissances fournies par l'ontologie pour compléter les parties manquantes du modèle. Par exemple, si une ressource manquante est détectée cela devrait conduire à l'ajout de cette ressource et éventuellement du fragment du processus qui la manipule.

6. Validation

L'objectif principal de notre validation est d'évaluer la démarche et de démontrer sa valeur. Nous avons effectué deux expériences pour valider notre approche à différents stades. Ainsi, une première expérience a été effectuée afin de vérifier que les contraintes de qualité définies servent à détecter un grand nombre des défauts de qualité dans l'évaluation des modèles de PM. Cet exercice de validation intermédiaire a été réalisé sur une collection de modèles de processus d'affaires de l'initiative BPM académique (« Academic Initiative BPM | Signavio » 2013) .

Les principaux objectifs derrière cette première validation étaient les suivants:

- i. Pour s'assurer que ces contraintes détectent un grand nombre des principaux défauts de qualité
- ii. Pour s'assurer que les concepts sont bien utilisés dans le modèle.
- iii. Pour étudier le pourcentage des défauts les plus courants.

Les résultats des expériences ci-dessus nous ont aidé à valider la sélection de nos contraintes de qualité et à corriger l'utilisation des concepts.

Nous avons effectué la deuxième expérience pour valider l'efficacité de notre approche complète.

Nous avons soigneusement sélectionné les répondants de l'expérience de validation de telle sorte que tous les répondants ont une connaissance préalable sur les modèles conceptuels et une certaine expérience de la modélisation.

Les principaux objectifs derrière cette deuxième validation sont les suivantes:

- i. Pour vous assurer que les paramètres de qualité définis sont utiles pour évaluer la qualité des modèles de PM.
- ii. Pour veiller à ce que les connaissances fournies par l'ontologie de domaine peut apporter des améliorations à la modèle.
- iii. Pour étudier les améliorations les plus utilisés.

La deuxième expérience nous a aidés à valider les points forts et les avantages de l'emploi de notre approche pour évaluer et améliorer le modèle de PM. Dans les deux expériences, nous avons utilisé plusieurs graphiques et des tableaux pour analyser les défauts les plus épannage. Nous étions surtout intéressés à la façon de détecter et corriger ces défauts.

7. Outil

Nous avons conçu et développé un prototype « Business Process Model Quality », qui met en œuvre notre démarche.

Cependant, le but de l'outil est de permettre l'évaluation et l'amélioration semi automatique des modèles de processus métier.

Il est divisé en deux modules: Le module d'annotation et le module de gestion basée sur la connaissance de la qualité de domaine.

Le module d'annotation où l'expert de la qualité est chargé de définir les mappages Modèle - métamodèle. Le Module de gestion basée sur la connaissance de la qualité où l'expert de la qualité est responsable de la gestion de la qualité en définissant les mesures de qualité, des guides d'amélioration de la qualité, etc. De même, il / elle est également responsable de la gestion des connaissances de domaine en définissant les métamodèles (métamodèle de PM et métamodèle d'ontologie), la définition des règles de mappage, etc et où le modélisateur est responsable de l'exécution des contraintes, des mesures et des améliorations de qualité.

Les deux modules accèdent une base de connaissances commune. En fait "BPM-quality» contient une base de connaissances pour stocker les règles de notations, les metamodels de PM et l'ontologie, etc. En outre, la base de connaissances stocke également les sessions de la qualité des modèles de domaine BP.

Figure 6: Architecture de l'outil

8. Conclusion

La recherche présentée dans cette thèse est un pas en avant dans l'évaluation sémantique et amélioration de la qualité des PM en se basant sur les connaissances de domaine. Afin de formuler une solution complète, nous avons exploiter les connaissances de la sémantique de notations et de l'ontologie de domaine. L'approche prend en compte la variété des notations des modèles de PM en utilisant un métamodèle. L'ontologie de domaine est représenté par une ontologie où la sémantique est enrichie par l'utilisation de plusieurs types de relations entre les concepts. Le processus d'alignement peut être appliquée à chaque domaine pour lequel il existe une telle ontologie. Il peut très faciliter la tâche des modélisateurs de PM et conduire à une amélioration significative de modèles. L'autre avantage est qu'il encourage la capitalisation des compétences. En effet, dans de nombreux domaines, il y a un effort de définition de la connaissance structurée et partagée dans plusieurs domaines : médical pratiques, les processus RH, E-learning, etc. L'approche proposée est une utilisation réelle de ces connaissances.

La plupart des cadres de qualité existants sur les models de PM proposent des mesures de qualité en s'appuyant sur la structure du modèle. En revanche, nous avons proposé:

- Quinze mesures de la qualité syntaxique: chaque métriques a une valeur de retour qui peut être un booléen ou un nombre entier. Nous avons expliqué chacune de ces mesures par la définition de leur rôle, le défaut détectent et l'analyse de leur valeur de retour.
- Six métriques pour évaluer la qualité sémantiques: tous ces paramètres ont une valeur de retour entière et s'appuient sur l'ontologie de domaine pour être calculé.

- Trois métriques pour évaluer la qualité pragmatiques: chaque métrique a une valeur de retour, deux d'entre eux comptent sur WordNet pour être calculé.

Notre approche souffre de quelques limites. Le principal est le fait que la connaissance de domaine en main n'est pas nécessairement complète. C'est la raison pour laquelle nous proposons d'enrichir les sources des connaissances de domaine avec les contraintes. Par conséquent, la qualité de l'ontologie de domaine affecte la fiabilité des résultats des métriques et l'efficacité des améliorations proposées.

Une autre limite est la complétude des contraintes et des paramètres de qualité. Nous ne pouvons pas assurer qu'on peut détecter tous les défauts d'un modèle, mais nous avons regroupé autant de défauts de la littérature que possible et on a définis de nouveaux défauts sur la base des expériences faites.

Enfin, notre approche nécessite d'autres modèles de validation pour prouver la robustesse de notre processus d'amélioration. En fait, nous avons appliqué notre approche sur huit modèles permettant de calculer et d'illustrer les résultats des métriques sur chaque concept dans les modèles de PM et à afficher les différents améliorations proposés pour chaque défaut.



Abstract

In recent years the problems related to modeling and improving business processes have been of growing interest. Indeed, companies are realizing the undeniable impact of a better understanding and management of business processes (BP) on the effectiveness, consistency, and transparency of their business operations. BP modeling aims at a better understanding of processes, allowing deciders to achieve strategic goals of the company. However, inexperienced systems analysts often lack domain knowledge leading and this affects the quality of models they produce.

Our approach targets the problem related to business process modeling quality by proposing an approach encompassing methods and tools for business process (BP) models quality measurement and improvement. We propose to support this modeling effort with an approach that uses domain knowledge to improve the semantic quality of BP models.

The main contribution of this thesis is fourfold:

1. Exploiting the IS domain knowledge: A business process metamodel is identified. Semantics are added to the metamodel by the mean of OCL constraints.
2. Exploiting the application domain knowledge. It relies on domain ontologies. Alignment between the concepts of both metamodels is defined and illustrated.
3. Designing of the guided quality process encompassing methods and techniques to evaluate and improve the business process models. Our process propose many quality constraints and metrics in order to evaluat the quality of the models and finally the process propose relevant recommendations for improvement.
4. Development of a software prototype “BPM-Quality”. Our prototype implements all the above mentioned artifacts and proposes a workflow enabling its users to evaluate and improve CMs efficiently and effectively.

We conducted a survey to validate the selection of the quality constraints through a first experience and also conducted a second experiment to evaluate the efficacy and efficiency of our overall approach and proposed improvements.

Keywords: Business Process Modeling, model quality, semantic quality, domain knowledge, domain ontology, quality evaluation, quality improvement.

Résumé

Au cours des dernières années, les problèmes liés à la modélisation et l'amélioration des processus métier ont reçu un intérêt croissant. En effet, les entreprises prennent conscience de l'impact indéniable d'une meilleure compréhension et gestion des processus métier (PM) sur l'efficacité, la cohérence et la transparence de leurs opérations commerciales. La modélisation des PM vise à une meilleure compréhension des processus, permettant aux décideurs d'atteindre les objectifs stratégiques de l'entreprise. Cependant, les analystes manquent souvent de connaissances du domaine, ce qui affecte la qualité des modèles qu'ils produisent.

Notre approche vise le problème lié à la qualité des modèles de processus métier en proposant une approche globale: des méthodes et des outils) de mesure et d'amélioration de la qualité des modèles de processus métier (PM. Nous proposons de soutenir cet effort de modélisation avec une approche qui utilise les connaissances de domaine pour améliorer la qualité sémantique des modèles de PM. La principale contribution de cette thèse est quadruple:

1. L'exploitation des connaissances du domaine d'u système d'information (IS domain knowledge): Un métamodèle des processus métier est identifié. Des règles sémantique ajoutés au métamodèle par des contraintes OCL.
2. Exploiter des connaissances du domaine d'application (application domain knowledge). Elle s'appuie sur des ontologies de domaine. Des alignement entre les concepts des deux métamodèles sont défini et illustré.
3. La conception du processus de qualité englobant des méthodes et des techniques pour évaluer et améliorer les modèles de processus métiers. Notre processus propose de nombreux contraintes et métrologie de la qualité afin d'évaluer la qualité des modèles et enfin le processus propose des recommandations pour l'amélioration de ces modèles.
4. Développement d'un prototype "BPM-Quality". Ce prototype met en œuvre tous les objets mentionnés ci-dessus et propose un flux de travail permettant à ses utilisateurs d'évaluer et d'améliorer les PM.

Nous avons mené une enquête visant à valider la sélection des contraintes de qualité à travers une première expérience et aussi réalisé une seconde expérience pour évaluer l'efficacité de notre approche globale et des améliorations proposées.

Mots-clés: modélisation des processus métier, la qualité des modèles, la qualité sémantique, les connaissances de domaine, l'ontologie de domaine, évaluation de la qualité, amélioration de la qualité.

Table of Contents

CHAPTER 1 INTRODUCTION	29
1.1 Domain of the thesis	29
1.2 Problem statement	31
1.3 Objective of the thesis	32
1.4 Organization of the thesis	34
CHAPTER 2 STATE OF THE ART	36
2.1 Business process modelling.....	36
2.2 Business process modelling notations	39
2.2.1 Business process modelling and notation	40
2.2.2 Eriksson-Penker notation	41
2.2.3 Event-driven process chains	41
2.2.4 UML activity diagrams	43
2.2.5 Petri-nets.....	43
2.3 Comparison of business process modelling notations	44
2.3.1 Business process modelling notations comparison frameworks.....	44
2.3.2 Results of business process modelling notations comparison	45
2.4 Business process modelling tools.....	49
2.4.1 Bizagi	49
2.4.2 Star UML.....	50
2.4.3 Visual Paradigm	50
2.4.4 ARIS.....	51
2.4.5 MEGA	51
2.4.6 Woflan.....	52
2.4.7 Tools comparison	53
2.5 From modelling to quality	54
2.5.1 Quality of business processes	55
2.5.2 Dimensions of BP quality (frameworks and standards)	56
2.5.2.1 Lindland/Sindre/Solvberg's framework	58
2.5.2.2 <i>Krogstie, Lindland et Sindre's framework</i>	59
2.5.2.3 <i>Moody et Shanks's framework</i>	60
2.6 Process model quality measurement.....	61
2.6.1 Syntactic quality	61
2.6.1.1 Size of the model	61
2.6.1.2 Complexity of the model	62
2.6.1.3 Structure of the model	63
2.6.1.4 Modularization of the models	65
2.6.2 Semantic quality	65
2.6.2.1 Validity of the models	66
2.6.2.2 Completeness of the models	67
2.6.3 Pragmatic quality.....	67
2.7 Quality approaches	69
2.7.1 Approaches dedicated to conceptual models quality	69
2.7.2 Approaches that evaluate business process model's quality	70

Table of contents

2.7.3 Approaches for business process improvement.....	73
2.7.4 Business process improvement patterns	77
2.7.5 Comparison of quality approaches	78
2.8 Conclusion.....	80
CHAPTER 3 IS AND APPLICATION DOMAIN KNOWLEDGE.....	81
3.1 Approach overview	81
3.2 Capturing IS domain knowledge	84
3.2.1 Business process meta-model.....	85
3.2.1.1 Flow objects.....	87
3.2.1.2 Connectors	88
3.2.1.3 Resources.....	88
3.2.2 Business process model - Business process metamodel alignments	89
3.2.3 BPM notations' semantics formalization	93
3.2.3.1 Syntactic constraints	95
3.2.3.2 Notations semantics constraints.....	97
3.2.3.3 Pragmatic constraints.....	106
3.3 Capturing application domain knowledge.....	107
3.3.1 Ontology meta-model.....	108
3.3.2 Business Process metamodel - Ontology metamodel mappings.....	111
3.3.2.1 Type-based mapping rules	111
3.3.2.2 Concept based mappings	111
3.3.2.3 Relation type based mappings	112
3.3.2.4 Instance based mapping rules	112
3.4 Conclusion.....	119
CHAPTER 4 EVALUATION AND IMPROVEMENT OF BUSINESS PROCESS MODELS QUALITY	120
4.1 Syntactic quality evaluation	122
4.2 Semantic quality evaluation	124
4.2.1 Quality defects detection	125
4.2.1.1 Meaningless state defects	125
4.2.1.2 Ambiguous representation defects:.....	125
4.2.1.3 Incomplete representations defects.....	127
4.2.1.4 Model- richness	128
4.3 Pragmatic quality evaluation	128
4.4 BPM quality improvement.....	129
4.4.1 Business process improvement initiative	131
4.4.1.1 Improvement process.....	131
4.4.2 Label refactoring	132
4.4.3 Quality improvement categorization	136
4.4.3.1 Textual guidelines.....	136
4.4.3.2 Textual improvements for syntactic constraints	137
4.4.3.3 Textual improvements for semantic constraints	139
4.4.3.4 Textual improvements for pragmatic constraints.....	139
4.4.3.5 Corrective actions	140
4.4.3.6 Preventive actions.....	143
4.4.3.7 Quality improvement patterns	145

CHAPTER 5 BPM-QUALITY: SOFTWARE PROTOTYPE IMPLEMENTING THE PROPOSED APPROACH.....	152
5.1 General architecture	152
5.2 Functional view for quality expert.....	153
5.3 Functional view for modeller	155
5.4 Annotation module.....	155
5.5 Domain knowledge based quality management module	157
5.5.1 Knowledge management submodule.....	158
5.5.1.1 Application domain knowledge management.....	158
5.5.1.2 IS domain knowledge management	160
5.5.2 Quality management subModule.....	160
5.5.2.1 Metric management	161
5.5.2.2 Guides management	161
5.5.3 Knowledge base	161
5.6 Quality evaluation in BPM-Quality.....	162
5.7 Comparison of BPM-quality with other existing software	164
5.8 Conclusion.....	166
CHAPTER 6 VALIDATION	168
6.1 Validating the selected quality constraints	169
6.2 Quality evaluation	176
6.3 Validating the proposed quality approach.....	179
6.3.1 Samples	179
6.3.2 Evaluation process.....	181
6.3.3 Improvements.....	183
6.4 Quality evaluation	185
6.5 Conclusion.....	188
CHAPTER 7 CONCLUSION	189
7.1 Contributions.....	189
7.2 Future Work – Perspectives.....	191

List of Figures

Figure 1: Van der Aalst BP Management Life Cycle (Wil M. P. van der Aalst, Hofstede, et Weske 2003).....	38
Figure 2: BPMN Meta-Model [http://www.omg.org/spec/BPMN/2.0/].....	40
Figure 3: A basic meta-model of business modelling concepts (Penker et Eriksson 2000).....	41
Figure 4: EPC Meta-model (Seel et Vanderhaeghen 2005).....	42
Figure 5: UML activity diagram Meta-Model (Havey 2005).....	43
Figure 6: Petri net meta-model (Bosilj-Vuksic, Giaglis, et Hlupic 2001).....	44
Figure 7: The framework of the Guidelines of Modelling (Becker, Rosemann, et Uthmann 2000b).....	55
Figure 8: Lindland and al.'s quality framework (Krogstie, Lindland, et Sindre 1995a).....	59
Figure 9: Krogstie et al.'s Quality framework (Krogstie, Lindland, et Sindre 1995a).....	59
Figure 10: Concepts in the framework of Moody and Shanks (1994).....	60
Figure 11: Business process modelling context.....	82
Figure 12: Independence of a given BP modelling notation.....	85
Figure 13: Business Process model Meta-Model.....	86
Figure 14: Mappings between different concepts having the same role: Process.....	90
Figure 15: Mappings between concepts having the same role: Actor.....	90
Figure 16: Connectors in different modeling languages (Gruhn et Laue 2010).....	91
Figure 17: Resources in different notations.....	92
Figure 22: Activity with two outgoing edges.....	96
Figure 20: Sequence flow crosses the process boundary.....	101
Figure 21: XOR Split Gateway.....	102
Figure 22: Activity with two outgoing edges.....	102
Figure 23: Timer intermediate event.....	104
Figure 18: Process with no start event Figure 19: Process with multiple start events.....	106
Figure 24: Erroneous model.....	107
Figure 25: Mappings between Domain Ontology and Business Process Model.....	108
Figure 26: Ontology Meta-model.....	110
Figure 27: The overall approach for quality evaluation and improvement.....	121
Figure 28: SPI Initiatives distribution of the publications (Unterkalmsteiner et al. 2012).....	130
Figure 29: Our improvement process.....	132
Figure 30: Improvements type.....	136
Figure 31: Prototype Architecture.....	153
Figure 32: Use Case Diagram: Define the Quality Concepts.....	154
Figure 33: Use Case Diagram: Evaluate the Quality.....	155
Figure 34: Annotation Process.....	156
Figure 35: BPM annotated.....	157
Figure 36: Quality Management Module architecture.....	158
Figure 37: Activity synonyms interface.....	159
Figure 38: Actor hyponyms interface.....	159
Figure 39: IS knowledge application interface.....	160
Figure 40: Knowledge base Structure.....	162
Figure 41: Defect types percentage.....	171
Figure 42: Notation semantic defects percentage distribution.....	173
Figure 43: Syntactic defects percentage distribution.....	174
Figure 44: Pragmatic errors percentage.....	175
Figure 45: OWL extract.....	183
Figure 46: Doctor hyponyms.....	184
Figure 47: Generality metric variation.....	186
Figure 48: Input completeness metric variation.....	186
Figure 49: Actor allocation metric variation.....	187
Figure 50: Model richness metric variation.....	187
Figure 51: Load interface.....	199
Figure 52: constraints interface.....	200
Figure 53; Activity evaluation and improvement interface.....	201
Figure 54: People Resource evaluation and improvement interface.....	202
Figure 55: Model-1.....	205
Figure 56: Model-2.....	206
Figure 57: Model-3.....	207
Figure 58: Model-4.....	208

<i>Figure 59: Model-5</i>	209
<i>Figure 60: Model-6</i>	210
<i>Figure 61: Model-7</i>	211
<i>Figure 62: Model-8</i>	212
<i>Figure 63: Actor instances</i>	214
<i>Figure 64: Action instances</i>	215
<i>Figure 65: knowledge instances</i>	216
<i>Figure 66: Abstract instances</i>	217
<i>Figure 67: Relation instances</i>	218
<i>Figure 68: Patient relations</i>	219
<i>Figure 69: Doctor relations</i>	220

List of Tables

<i>Table 1: Notation's strengths and drawbacks</i>	47
<i>Table 2: Tools comparison</i>	53
<i>Table 3: Dimensions of BP quality (Heravizadeh, 2009)</i>	57
<i>Table 4: Characteristics, attributes and measures of activity (Heinrich et Paech 2010)</i>	57
<i>Table 5 : Quality approaches of conceptual models</i>	58
<i>Table 6:FMESP set of metrics (Aguilar et al. 2006)</i>	63
<i>Table 7: Possible experimental studies (Soffer, Kaner, et Wand 2012)</i>	68
<i>Table 8: Quality approach comparison</i>	79
<i>Table 9: Workflow defects</i>	98
<i>Table 10: Concept alignment</i>	112
<i>Table 11: Relationships alignment</i>	112
<i>Table 12: Syntactic quality metrics</i>	124
<i>Table 13: Pragmatic quality metrics</i>	129
<i>Table 14: Label classes</i>	133
<i>Table 15: Time conjunction</i>	135
<i>Table 16: Comparing BPM-Quality with existing evaluation approaches</i>	166
<i>Table 17: Categorization of the defects</i>	171
<i>Table 18: Tools comparison</i>	178
<i>Table 19: Respondents information with respect to their occupation</i>	181
<i>Table 20: Extract from the computed Metrics for Model-3</i>	182
<i>Table 21: Improvements occurrence</i>	184
<i>Table 22: Metrics average calculation</i>	185

Chapter 1

Introduction

Business process modelling has been receiving increasing attention from companies as well as the academic community, since it is likely to improve practice and working processes in organizations in the future. Indeed, to keep up and remain competitive, companies must be able to assess the quality of their products and the efficiency of their services. This requires understanding the current system and analyzing the potential benefits of the underlying processes. Using models helps reaching such an objective as they provide an abstract and coherent vision of the business and facilitates understanding and reasoning about alternative solutions to reach an agreement (Nurcan et Edme 2005). Business process models (BPM) can be realized through methodologies, techniques, or software, in a way that helps organizations bring together processes and their context including people, documents, information sources, organizational structures, and applications (Van Herk 2006). As a methodology, BPM help organizations in gaining control over their business processes by modelling, validating, analyzing, and monitoring the processes.

A business process is a collection of activities that takes one or more kinds of inputs and creates output that is of value to the customer (Hammer 1993).

1.1 Domain of the thesis

Business process (BP) modelling is an important tool for understanding and revealing weaknesses of business process (Heravizadeh 2009). It is recognized as a key part of the business process lifecycle. It is during the modelling stage that a conceptual model is produced by collecting business process requirements and representing them with a specific business process notation (Sánchez-González et al. 2013). In order to meet customer demands, companies have to design business processes in an appropriate way.

BP modelling is a prerequisite. It is now considered as an engineering activity aiming at providing the actors with a better understanding of the processes in which they are involved. The resulting

Chapter 1: Introduction

models serve as concrete tools allowing stakeholders to understand i) how processes work, ii) why they may dysfunction, iii) which software could support them efficiently, and so on.

Business processes influence product quality and customer satisfaction, which are fundamental aspects in a market environment (Sánchez-González et al. 2013), and enterprises are therefore constrained to improve their processes in order to improve products and services (Cardoso 2006). Indeed, it is widely agreed that quality of product depends on the quality of the underlying processes and this is why business process quality is considered as an important issue. Quality problems can inflate system development cost and consume scarce resources, in addition to increasing error detection and correction costs (Ravichandran et Rai 1994). Moreover, (Ackoff 1968) found that the existence of defects or deficiencies can hamper IS quality and put its adoption at stake since IS adoption is linked to IS quality, satisfaction, and usage (H. James Nelson 2005).

In recent years research related to modelling and improving business processes has been of growing interest. Thanks to their experience, companies are aware of the undeniable impact of a better tuning of business processes on the effectiveness, consistency and transparency of their business operations. This tuning requires a better understanding and an effective management of BP.

Many research efforts have been devoted to the development of suitable methods and notations for business process modelling. However, to achieve the expected benefits on quality it is necessary to rethink the approach of designing these processes by integrating the quality objectives into design. Indeed, BP modelling is difficult. As it requires a high level of expertise in modelling and a good understanding of the business domain, the quality of produced models is variable and need to be assessed and improved.

Quality can be defined as the total of properties and characteristics of a product or service that are relevant for satisfying specific and obvious requirements (« ISO 8601 »). Following Boehm and other quality precursors, ISO subdivides quality in a number of quality characteristics such that each one addresses a particular aspect of quality.

The business process modelling approaches share many similarities with conceptual modelling activities, but are much more complex. A business process model is a representation at a very high level of abstraction but it has also to integrate non-functional requirements such as flexibility and maintainability. Modelling these processes requires a high degree of pragmatic expertise: designers refer to a set of mainly empirical rules and heuristics. The latter are difficult to formalize and to share. In fact, it's too late to identify the defects and deficiencies when the software is already developed, as the maintenance cost of these defects could be enormous and might require major design or architectural modifications. It has been noticed that majority of the IS change-requests result from deficient functionalities in the information systems such as the lack of desired functionalities within a

Chapter 1: Introduction

system, etc. However, as mentioned above, these change requests will be expensive to fix when the system is already developed. Studies show that defect detection in the early stages of the application development can be thirty three times more cost effective than testing done at the end of development (Walrad et Moss 1993).

Therefore, it is imperative to emphasize the need of introducing quality mechanisms at the earlier stages of development such as during analysis and design. It has now been widely agreed that the quality of the end-system depends on the quality of the system's business process model. As a result, different methodologies propose different methods and guidelines to ensure a certain degree of quality to the process models.

The problems addressed during this thesis are discussed in the next section.

1.2 Problem statement

In an enterprise, different processes are performed. If the enterprise wants to reach its goal and stay in front in the market, every process has to be evaluated and may be changed to support the goals of the organization. The quality of business processes and business process models have been looked upon in this light in a number of partly uncoordinated works.

Various researches focus on how to produce "good" graphical models. But according to (Moody 2007), the design of models using a graphical notation is both ad hoc and unscientific. Thus the quality of the model is not limited to the quality of the graphic model but also on the vocabulary used in the model to communicate.

Lindland proposes a semiotic framework for quality that distinguishes syntactic, semantic, and pragmatic quality (Lindland, Sindre, et Solvberg 1994a). The syntactic quality addresses the structural relationships between modelling elements. The semantic quality is related to the relationship between the model and what it stands for (the domain). Finally, pragmatic quality considers the link between models and their interpreters.

On the other hand, commercial tools for business process modelling activities mainly focus on the accuracy of models based on a set of syntactic criteria and provide little or no guide to guarantee the quality of produced models. Recent research work focusing on the quality of process models concentrates mainly on structural aspects of models taking into account two criteria of quality namely correctness and complexity.

The main problems targeted within the context of this thesis are listed in the following.

- **Lack of business process models evaluation approaches**

One of the major hurdles in evaluating the quality of business process models is the lack of quality metrics. Most of the existing evaluating approaches concentrate on the evaluation of BP models syntactic quality: complexity, size, cohesion, etc. But there are few approaches that provide quality

Chapter 1: Introduction

assessment methods for the evaluation of semantic and pragmatic quality of models. Semantic quality of the model includes: (i) the semantic of the concepts labels i.e. the vocabulary used and (ii) the semantic emplacement of different concepts i.e. the relationship of the concept with its neighbours. The semantic of the concepts depends on the domain. It is presented through the concepts label. The semantic emplacement of the concepts depends on the role of each concept type. The pragmatic quality is related to the understandability and the readability of the model.

Additionally, there are no tools automating the syntactic evaluation process. Most of the existing modelling software tools such as Bizagi, Mega, Blueprint, etc. do not provide a comprehensive evaluation mechanism.

- **Lack of business process models improvement approaches**

Enterprises try to improve their existing processes in order to provide new or better services. But quality evaluation is only a step to improve the business process models. Most of the quality frameworks focus exclusively on defect detection (quality evaluation) and ignore the defect correction (quality improvement) aspects. Thus they may help in identifying the problem but the analysts must rely on themselves for the solution (D. L. Moody 2005). Similarly, the domain of BPM quality lacks a guided process helping modellers to improve their models. Limited research is carried out to represent process knowledge for improvement (Vergidis 2008).

There has been some work done to capture common improvement approaches in the form of business redesign patterns that contribute to the improvement of processes from four main perspectives, namely time, quality, cost, and flexibility (Hajo A. Reijers 2005). Although these redesign patterns could be used as guidelines for the improvement step, most of the improvement methods do not utilize these patterns. On the contrary, these methods rely heavily on human innovation and creativity rather than on rationality.

Furthermore, available process modelling notations such as BPMN, EPC, Eriksson-Penker do not provide preventive guidelines nor improvement guidelines. Therefore, support for effective improvements are very limited. Some of the above notations are complex and require assistance in their use but these guidelines are insufficient.

To conclude, our review of literature leads us to the conclusion that researchers mainly focused on evaluating the syntactic quality aspects of the model. Therefore, an effort on semantic and pragmatic quality definitions needs to be provided. Moreover business process models quality improvement is required. Our approach is a step forward to improve quality of BP modelling along these two directions.

1.3 Objective of the thesis

The business process modelling activity may concern two types of users: i) Technical users that represents modelers and developers as well as ii) Business users that represents domain experts.

Chapter 1: Introduction

The primary objective of a business process model is to provide the modeller with a semi-formal representation of the organization main functions. In particular, a modeller that do not have sufficient domain knowledge or not competent in using a modelling notation.. Although these various formulations can be correct, they might not necessarily be equal in terms of their usage. The core objective of this thesis is to develop a comprehensive quality approach for business process models. This objective can be divided into the following goals.

1. Define the semantic quality property for business process model evaluation

The semantic quality measures the degree of correspondence between the model and the domain. The semantic quality is related to both completeness and validity of the models (Krogstie, Lindland, et Sindre 1995a). We consider semantic quality as the conformance of BP models to the domain knowledge.

According to (Krogstie, Sindre, et Jørgensen 2006), in Information Systems (IS) discipline, domain knowledge comprises two parts:

- IS domain knowledge representing know-how provided by methods, notations, and tools.
- Application domain knowledge referring to real-world problems: application domain knowledge can be extracted from several sources such as user's requirements statements, domain expertise or existing models related to the same problem or to similar problems

To conclude, we aim to capture IS Domain Knowledge through Meta modelling and Application domain knowledge through a domain ontology.

2. Capture the IS domain knowledge

In order to capture the IS knowledge for BP modelling activity.

- We define a business process model meta-model aiming to both structuring BP models to help in their understanding, and ensuring the generality of the approach regarding the variety of BP modelling notations.
- We describe the model semantics by the means of object constraint language (OCL) rules. These rules are classified into semantic, pragmatic, and syntactic depending on the category of model's quality property they aim to ensure.

3. Propose an approach based on domain knowledge to improve semantic quality of business process model

In order to evaluate the semantic quality, we aim to identify a set of what we call quality deficiencies.

The latter are detected based on the knowledge provided by the domain ontology. To do so we aim to:

- Exploit the domain ontology.
- Define mappings between the business process model and the domain ontology in order to extract the adequate knowledge.
- Based on the knowledge extracted, evaluate the semantic quality of the model by proposing semantic metrics.

4. Enrich this approach by proposing guides to help business process modellers to improve their models and thus upgrade their semantic quality

Another problem with existing quality approaches is that they don't propose a guided process. They merely identify and propose evaluation criteria and leave the modeller on his/her own for evaluation. Similarly, majority of the quality frameworks fail to provide post-evaluation recommendations for improvement. Thus once the modeller is done with evaluation, he/she is left without any guidelines for improvement.

It is therefore required that the proposed quality approach should encompass a complete guidance process helping modellers in evaluating the model and guide them in improving the quality of their BPM based on the evaluation results.

5. Develop a software utility to automate the evaluation and improvement process

Quality evaluation in BPM doesn't attract many users as it is difficult to evaluate the model manually. People find it difficult to calculate the metrics by hand. Moreover, semantic metrics are not calculable by hand.

BPM quality evaluation can become much easier and efficient if the proposed quality approach is supported by a software tool able to perform the following:

- i. Exploit a domain ontology
- ii. Calculate metrics automatically on the model
- iii. Maintain a knowledgebase of evaluation metrics
- iv. Define new quality metrics
- v. Provide post-evaluation recommendations for improvement
- vi. Define new quality patterns.

1.4 Organization of the thesis

The thesis is organized in a sequential way starting with a thorough state of the art in Chapter 2. Different existing evaluation methodologies and quality frameworks are categorized and discussed in detail to obtain an overall idea of current state of the target domain.

In Chapter 3, we discuss our proposed solution in detail. It starts with the formulation of a multi-faceted quality approach for business process models. It is followed by the description of our quality model including the descriptions of each of its components such as IS constraints and quality metrics. Quality improvement patterns are one of our major contributions. Therefore we describe the concept of business process quality improvement in details along with all its components such as quality metrics, improvement actions, preventive actions, textual guidelines and recommendations.

Thus in Chapter 3 we discuss the representation of knowledge in two sections:

Chapter 1: Introduction

- IS Domain knowledge: It is divided into two main parts. In the first part, we discuss the business process models meta-model that allow our approach to be independent of any notation. Mappings between concepts of each notation and the meta-model concepts are identified. Whereas in the second part, we describe notation constraints. These constraints are applied on the meta-model level allowing the modeller to detect the modelling defects.
- Application domain knowledge: We define an ontology meta-model. This section is divided into two main parts. In the first part, we discuss the ontology meta-model whereas in the second part, we define different types of mappings between the BPM meta-model concepts and the Ontology meta-model concepts.

In Chapter 4 , we discuss the quality evaluation and quality improvement procedure.

- The evaluation procedure includes the creation of the quality metrics, the definition the quality defects and how to evaluate each defect.
- The improvement process proposed by our approach starts by defining the different types of improvements. It is followed by the description of each step of the improvement process, and finally, by defining the quality patterns.

In Chapter 5, we present our software prototype “*BPM-Quality*” implementing our proposed solution and process. In the first part of the chapter we describe the application architecture at multiple levels of granularities whereas in the second part we present different interfaces available in “*BPM-Quality*” for quality evaluation and quality improvement operations. We have taken small examples to demonstrate the flow of the application.

In Chapter 6, we discuss the two validation experiments conducted for our approach. In the first section, we describe our first experiment and the results aiming at validating the selection of the IS constraints defined. In the second section, we discuss our second experiment that was conducted to validate the efficacy of our complete approach including quality patterns.

In Chapter 7 some perspectives of the work are discussed along with the conclusions of the thesis.

Three appendixes are given at the end. Appendix-A describes the different functions used to mapp the BP model concepts to the domain ontology concepts. It also describe the BPM-Quality tool interfaces.

Appendix-B present the validation scenario and lists the models used in the second experiment.

Appendix-C illustrates the domain ontology used in the second experience.

Bibliography is given after the three appendixes. A list of our publications is provided in the last part of the thesis.

Chapter 2

State of the Art

There is no universal definition of quality for products and services. One of the definitions is that of ISO which is "the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs" (ISO 8402:1994).

Thus quality is a subjective domain. It is highly dependent on multiple view points (consumer, producer, etc.). The notion of quality for an object from one viewpoint might not hold from another viewpoint (Reeves et Bednar 1994).

The question is why to ensure business process quality? As a matter of fact, business processes are designed to execute strategies that aim at achieving organisational goals (Poels et al. 2013). Consequently ensuring its quality means identify and reduce opportunities for defects, which cause reductions in the quality of process outcomes. (BusinessDictionary.com, 2013). To evaluate the quality, we need to identify quality criteria (dimensions, attributes and metrics).

As this thesis is concerned with the quality of business process models, we will be discussing BP modelling notations and tools in this chapter in addition to quality criteria of models. We chose to review literature on business process quality dimensions and metrics. Last of all, we review the quality approaches existing in the literature.

2.1 Business process modelling

In recent years research related to modelling and improving business processes has been of growing interest. Thanks to their experience, companies are aware of the undeniable impact of a better tuning of business processes on the effectiveness, consistency and transparency of their business operations. This tuning requires a better understanding and an effective management of BP.

The Oxford English Dictionary (1999) defines a 'process' as a series of actions or operations conducting to an end, or as a set of gradual changes that lead toward a particular result. Business processes (i.e. processes performed by the 'physical system') are a set of activities intended to

transform system inputs into desired (or necessary) system outputs by the application of system resources (Leondes 2010).

A BP is a structured set of activities designed to produce a specified output for a particular customer or market so that it implies a strong emphasis on how work is done within an organization, in contrast to a product's focus on what (Davenport 1993).

A business process is the combination of a set of activities within an enterprise with a structure describing their logical order and dependencies whose objective is to produce a desired result (Aguilar-Savén 2004a). Business process modelling enables a common understanding and analysis of a business process. A process model can provide a comprehensive understanding of a process. Recently, an empirical study has shown that business processes have become the central objects in many conceptual modelling efforts (Davies et al. 2006). In fact the aim of the research was to determine whether practitioners still embraced conceptual modelling seriously and to find out what are the most popular techniques and tools used for conceptual modelling. The conclusion of their research was that the highest ranked purposes for which modelling was undertaken were business process improvement and business process documentation.

Even if business process models (BPM) are also conceptual models, they have specific aspects related to the fact that the underlying processes must be executed. In fact, business process execution needs standards for capturing the workflow management (Hollingsworth 1995). Also, it should represent an executable behaviour captured by business process patterns such as parallel split and synchronization. These patterns specify the basic functional and behavioural requirements of an execution architecture (Van Der Aalst et al. 2003), in addition to managing shared resources and simultaneous constraints (Malone et Crowston 1994).

Thus BPM evaluation needs specific approaches and tools, different from those of conceptual data models quality approaches (D. L. Moody et Shanks 1994).

To understand the terminologies and features of BPM, one should start from an appreciation of the BP Management life cycle. The latter describes the various phases in support of operational business processes (Weske 2012). In the literature there is no uniform view on the number of phases in this. It varies depending on the chosen granularity for identifying the phases (Wetzstein et al. 2007).

Besides the BP management, the importance of the workflow management is of big interest. Workflow management supports business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents, and other sources of information (Wil M. P. van der Aalst, Hofstede, et Weske 2003a). Van der Aalst and al. restrict BPM to operational processes. They consider four phases to represent the workflow management (Figure 1).

1. Process design - In this stage, business processes are modelled into BPM systems. Graphical standards are dominant in this stage. They allow users to express business processes and their possible flows and transitions in a diagrammatic way.

2. System configuration - This stage configures the underlying system infrastructure e.g. synchronization of roles and organization charts.
3. Process enactment - Business processes once modelled are deployed in business process modelling software engines. Execution standards dominate this stage. They computerize the deployment and the automation of business processes.
4. Diagnosis - Given appropriate analysis and monitoring tools, the BPM analyst can identify and improve on potential loopholes in the business process. The tools provide administrative and monitoring capabilities.

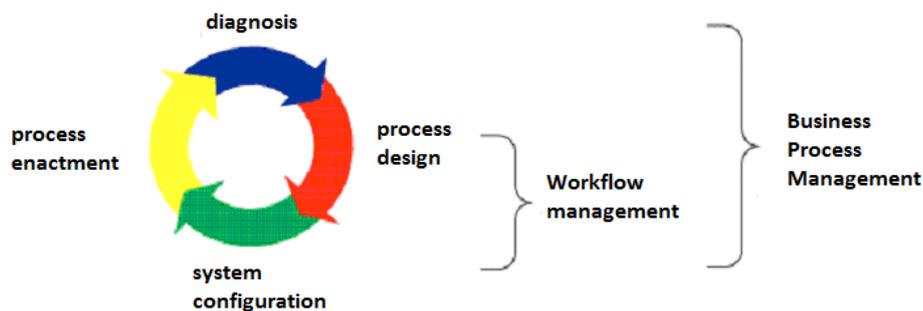


Figure 1: Van der Aalst BP Management Life Cycle (Wil M. P. van der Aalst, Hofstede, et Weske 2003)

The conceptual design phase is a critical stage in the business process management lifecycle, since in this stage, designers and other members of the development team brainstorm product ideas based on research into customer needs (Lilton, 2013). Also, according to Boehm (Boehm 1981), removing the defect during the design stage costs less than removing the same defect during the other stages since an error detected in the further phase will generally require the expensive rework of much of the previous phases.

At the design stage, users are allowed to express the information flow, decision points and the roles of business processes in a diagrammatic way (Ko, Lee, et Lee 2009). One of the primary motivations for developing any model is to increase the understanding of the business and facilitate communication about the business. The business process models should generate descriptions of a complex reality that capture the core functions of the business.

Current initiatives in the field of Business Process Management (BPM) strive for the development of a BPM standard notation. However, BP modelling techniques need to be carefully examined, analyzed and evaluated. In the literature, a number of notations have been proposed to model business processes e.g. BPMN (Business process modelling and notation) (S. A. White 2004), Eriksson-Penker notation (Penker et Eriksson 2000), Event-Driven Process Chains (EPC) (W. M. P. van der Aalst 1999), Petri Nets (Bosilj-Vuksic, Giaglis, et Hlupic 2001), UML activity diagrams (Havey 2005) etc. The common

feature of these languages is their focus on providing a comprehensive, integrated notation for (business) process modelling.

Business process modelling is commonly used to document information about structure and behaviour of a business process. The main goal of a notation is to capture and present the essential concepts of the process (Caetano et al. 2005). Hence, the quality of business process models is affected directly by the notation. In the next section, we discuss some of the existing business process modelling notations.

2.2 Business process modelling notations

Business process modelling has several purposes, we can cite:

1. Use a common representation of the processes in order to help the communication between the business communities (analysts, designers, developers...). They will have the same vocabulary which will facilitate their understanding and communication (Luo et Tung 1999).
2. Modelling business processes requires capturing the essential concepts of the process. These concepts can later be composed, specialized and reused into other process models (Caetano et al. 2005).
3. Business process models also improve model reengineering by facilitating model reuse and making explicit the dependencies between the model elements (Caetano et al. 2005).
4. The business process model acts as the basis for decision making and affects decisions about prioritizing goals, obtaining the right resources. It allows for changes and improvements in the process and enables decision support during process execution, and control (Eriksson et Penker 2012).

Business process modelling represents an important part of information systems (IS) development and evolution within organizations. One of the main issues in business process modelling is the variety of notations.

A lot of conceptual business processes modelling notations are available. The difference between those notations is limited to their abilities to integrate many forms of information into the business process model. Each notation can focus on different facets of the process modelled.

Several languages have been proposed for business process modelling, they emphasize different aspects of processes and related structures, such as organizations, products, and data. Consequently, they are suited for different kinds of processes. We can distinguish between process modeling languages that are hierarchical, flow-oriented, role-oriented, communication-oriented, declarative, goal-oriented, timelines, product and document state machines etc.

Based on Ouyang and al.'s definition (Ouyang, Dumas, et Hofstede), a business process model is a flow-oriented representation of a set of work practices aimed at achieving a goal, such as processing a customer requestor complaint, satisfying a regulatory requirement, etc.

Hence we will review some of the flow oriented languages (BPMN, UML AD, EPC, Petri-net) and the product/data oriented Eriksson-Penker notation whose one of its concern is in the structuring and handling of data, followed by a comparison of the notations based on several references.

2.2.1 Business process modelling and notation

The Business Process Management Initiative (BPMI) has developed a standard Business Process Modelling and Notation (BPMN) (S. A. White 2004). The BPMN meta-model consists of four different categories: Flow objects, Connecting objects, Swim lanes, artefacts (Figure 2). The elements Activity, Process, Sub-Process, Task as well as Events and Gateways are Flow objects, which define the behaviour of a business process. An event is something that “happens” during the execution of a business process. There are three types of events, based on when they affect the flow: Start, Intermediate, and End. A Gateway is used to control the divergence and convergence of a sequence flow. The connecting objects Sequence Flow, Message Flow and Association describe the ways of connecting the flow objects to each other. A message flow illustrates the exchange of messages. A sequence flow shows the order in which activities are performed in a process. An association is used to link information to activities, and relates a Data Object to a flow or connects it to an activity (Omg 2004).

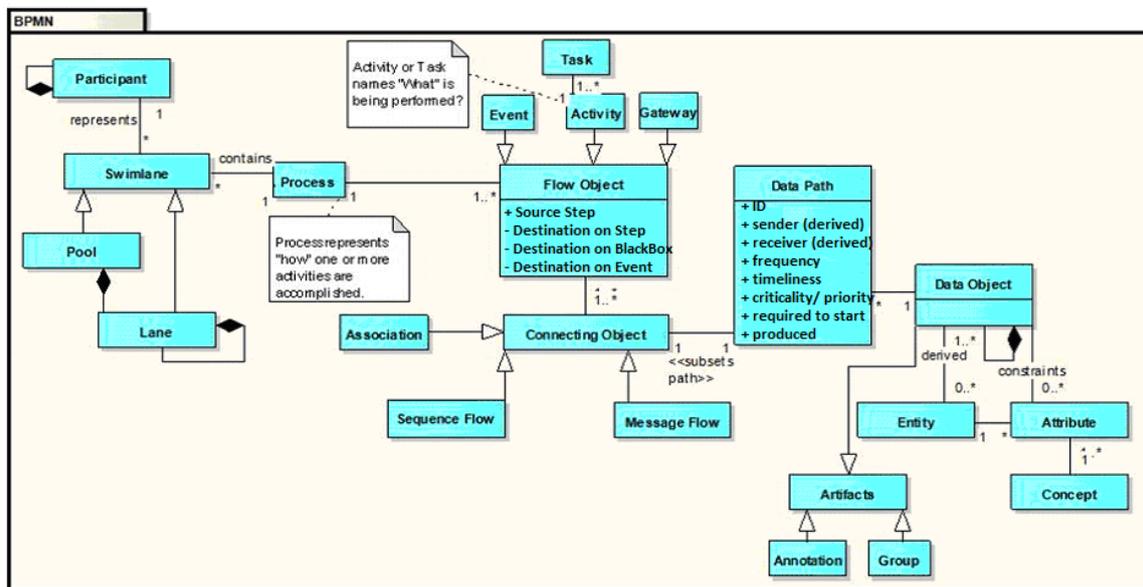


Figure 2: BPMN Meta-Model [http://www.omg.org/spec/BPMN/2.0/]

The BPMN elements are compliant with most flow charting notations but offer much more precise flow control semantics. The strength of BPMN is that it enables roles to be defined at various levels of granularity through pools and swim-lanes. One of the shortcomings of BPMN is that it lacks formal semantics. Additionally, BPMN lacks concepts hierarchy (Wahl et Sindre 2005a). As stated in (S. White 2004), BPMN is not suitable for modelling organizational structures and resources, data and

information models, strategy or business rules. BPMN contains only concepts that are relevant to model business processes so that concepts related to other domains can not be presented.

BPMN has similar notations to other languages such as Activity diagrams, EPC, Petri nets and others, which is helpful as the modellers may be familiar with the other languages. Also, BPMN concepts are defined in a clear way to avoid confusions.

2.2.2 Eriksson-Penker notation

Eriksson-Penker extensions provide a framework for UML business processing model extensions, to which an enterprise architect can add stereotypes and properties appropriate to his/her business (Penker et Eriksson 2000). Eriksson-Penker extensions provide symbols for modelling the processes (i.e. the activity performed within the business), resources (such as people, material, information, and products that are used or produced in the business), rules (statement and constraints), and goals (i.e. the purpose of the business) of a business system (Figure 3).

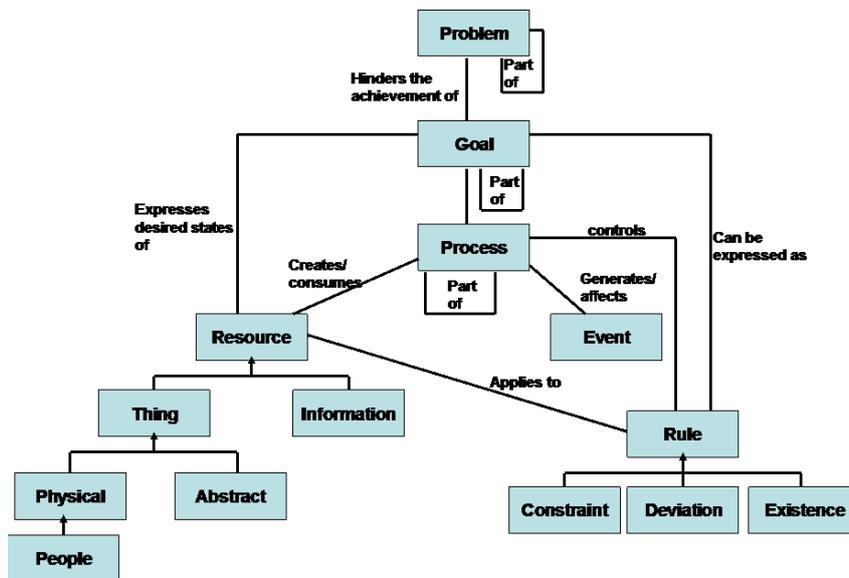


Figure 3: A basic meta-model of business modelling concepts (Penker et Eriksson 2000)

Eriksson-Penker elements are understandable and well defined because of the stereotypes. They describe different types of resources which represent the information flow in the business process.

One of the shortcomings of Eriksson-Penker notation is that it is not sufficiently detailed regarding the synchronization of the activities. In other words, there is only one type of synchronization; it does not include the XOR and OR gateways presented in BPMN for example. In addition, it is not exportable in any other language.

2.2.3 Event-driven process chains

Event-Driven Process chain (EPC) was developed by the Institute for Information Systems (IWi) at the University of Saarland, Germany. It is an intuitive graphical business process description language.

Event-driven process chains have become a widespread modelling technique because of the success of products such as SAP R/3 and ARIS. EPC notation is easy to understand and capable of presenting the information flow in the process. Based on van der Aalst work (W. M. P. van der Aalst 1999) EPC suffers from a serious drawback : neither the syntax, nor the semantics of an event driven process chain are well defined which leads to ambiguous representations of the processes.

2.2.4 UML activity diagrams

OMG's Unified Modelling Language (UML) captures all attributes and behaviours of the objects modelled. The UML activity diagram is one of these diagrams. It is both a flowcharting technique and a special kind of state machine whose activities are states and interactivity links trigger-less transitions (Havey 2005).

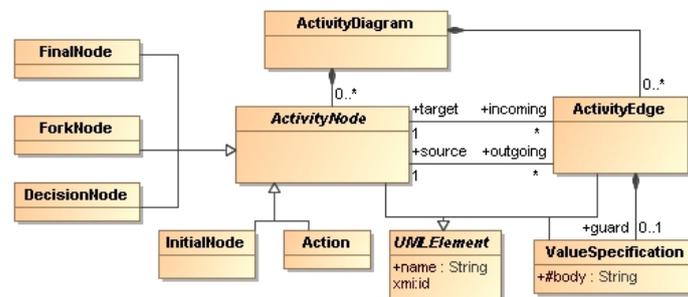


Figure 5: UML activity diagram Meta-Model (Havey 2005)

An activity diagram is used to display the sequence of activities. It shows the workflow from a start point to an end point passing by decision nodes or fork nodes (Figure 5).

UML activity diagram elements are comprehensible. They support the control flow and the information flow. However, they are limited in terms of model resources and of the different synchronization elements.

2.2.5 Petri-nets

Petri net is the notation suited to model the behaviour of the system in terms of "flow". A Petri net is a particular kind of bipartite directed graphs populated by three types of objects: places, transitions and directed arcs (Figure 6). A transition has a number of input places and output places. Places, represented by circles, are the containers of tokens. Tokens represent the information or the thing that flows through the system. Directed arcs connect places to transitions and vice versa. At a given point, a place may contain zero or several tokens and the number of tokens represents the state of the place. A transition is enabled if there is at least one token in each input place. When a transition is executed, a token is removed from the input and added to the output of the current transition.

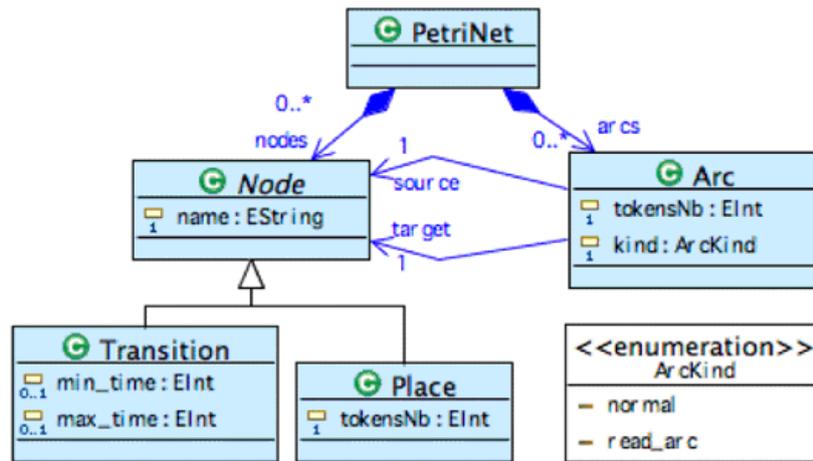


Figure 6: Petri net meta-model (Bosilj-Vuksic, Giaglis, et Hlupic 2001)

Petri nets have a limited number of symbols but have a large power of representation. They are well suited for representing the synchronization of resource communication (Bosilj-Vuksic, Giaglis, et Hlupic 2001). However, these models also have disadvantages. Perhaps the main disadvantage is that the basic Petri net constructs are quite primitive, so that is not only a significant burden placed on the analyst in order to specify complex models, but in addition the graphical representation may become too complex to be useful. Another disadvantage is that the representation of priorities or ordering is hard to manage, although priority queues are important in performability modelling.

We can seize from the literature other notations such as flowcharts, Yet another workflow language (YAWL) (W.M.P. van der Aalst et ter Hofstede 2005), Role Activity Diagramming (RAD) (C. Badica et al. 2005), Integrated Computer-aided Manufacturing Definition (IDEF) (Badica et Fox 2003), etc...

The motive for the choice of the five process modelling notations is that they are widely used in describing business process models and they are commonly used by both practitioners and researchers (Johansson et al. 2008).

2.3 Comparison of business process modelling notations

Researchers always aim to develop new business process notations because of dissatisfaction with the existing notations for presenting information and knowledge on business processes. Thus the notations which use a less total number of commonly acceptable symbols are intuitively more meaningful than the existing arbitrarily assigned symbolic notations (Birkmeier, Klöckner, et Overhage 2010a).

The evaluation and comparison of conceptual modelling languages in general and process modelling languages in particular has frequently been addressed in literature. This evaluation is important to help in selecting the most suitable language according to the needs, and also to improve existing languages.

2.3.1 Business process modelling notations comparison frameworks

Previous efforts are done to study the strengths and drawbacks of BPM notations by proposing frameworks to evaluate them.

In order to compare BPM notations, Aldin et al. (Aldin et Cesare 2009) derive five criteria :

- flexibility, which affirms how easy changes in the business process are possible,
- ease of use, the extent to which the technique can be easily applied by business modellers who are not specialists of the technique,
- understandability of the modelling technique,
- simulation, demonstrates the capability of the modelling technique to simulate a business process,
- scope establishes the definition of the process modelling elements.

On the other hand, Wahl and al. proposed the semiotic framework (Wahl et Sindre 2005b) where five criteria are identified for evaluating the quality of conceptual modelling language:

- Domain appropriateness, conveys how suitable the language is for use within different domains.
- Participant language knowledge appropriateness shows if the language is understandable by the modellers.
- Knowledge externalizability appropriateness, verifies if the language is suitable to express all the knowledge of the modellers.
- Comprehensibility appropriateness, evaluates language comprehensibility.
- Technical actor interpretation appropriateness, evaluates the language reasoning process.

The Wohed and al. comparison framework (Wohed et al. 2006) is based on four criteria: (1) widely used; (2) well accepted; (3) comprehensible to IT practitioners; and (4) sufficiently detailed to provide a comprehensive basis for assessing the capabilities of process modelling languages.

In the next section, we assess some of the modeling notations for business process models. We selected UML activity diagrams, BPMN, EPC, Petri-nets and Eriksson Penker notations. Moreover, we selected Wohed and al's four criteria (Wohed et al. 2006) to evaluate these notations. This evaluation combines our expertise and literature studies related to notations understanding and analysis.

2.3.2 Results of business process modelling notations comparison

Several authors who conducted analytical comparisons have highlighted considerable similarities between the notations: Business Process Model and Notation (BPMN) and Activity Diagrams (AD). Some presumed the superiority of BPMN over AD and others found BPMN to be too complex.

Birkmeier and al. (Birkmeier, Klöckner, et Overhage 2010b), concluded that UML AD was significantly more effective in data handling. Also the usage of BPMN promoted a sequential modelling style in which unrelated activities run one after the other or AD allow the activities to run in parallel, which increases the process flexibility. Similarly, Recker et al. (Recker et al. 2009) used an established approach for evaluating the complexity of conceptual modelling methods and compared BPMN and AD based on their complexity. They found that BPMN has very high levels of complexity when contrasted with UML AD. On the other hand, some authors used a semiotic quality framework

to analyze BPMN such as in Nysetvold and al. (Nysetvold et Krogstie 2005). They found BPMN to be superior on AD with respect to learnability, precision, and its language patterns.

Comparison between BPMN and EPC is also studied. Kunze et al. (Kunze et al. 2011) concluded that BPMN models expose a higher diversity than EPC's in terms of construct heterogeneity due to the greater and more detailed expressiveness of the BPMN language compared to EPC's, which is also leveraged in process models. Similarly (Tscheschner, 2010) , studied how to transform EPC models to BPMN models due to the fact that BPMN has more expressiveness than EPC.

Vuksic et al. (Vesna Bosilj-Vuksic 2000) compared IDEF diagrams (Aguilar-Savén 2004b) and Petri nets (Bosilj-Vuksic, Giaglis, et Hlupic 2001). They concluded that IDEF diagrams are very simple but not available for very complex models on the contrary of Petri nets. Hierarchical structure is possible in both diagrams. Petri nets have a lack of standardization on the contrary of IDEF diagrams.

We borrowed the choice of BP languages evaluation criteria from (Wohed et al. 2006) since it is more adequate to our study for the following reasons:

- (1) Widely used: help us to asses evaluation based on these criteria.
- (2) Well accepted: also help us to identify evaluation criteria.
- (3) Comprehensible to IT practitioners: verify the pragmatic quality and it may be used as pragmatic metrics threshold.
- (4) Sufficiently detailed: it represents almost all the semantics we are searching for as we are interested in the completeness of a notation.

	Widely used	Well accepted	Comprehensible to IT practitioners	Sufficiently detailed
UML Activity diagram	- AD are less used than the other notations since it is more complicated within the gateways (parallel and event base gateways). (White 2004). + The two most widely used graphical notations for business processes are BPMN and AD (GEAMBASU 2012).	+ UML elements are easy to sketch on paper (Wohed et al. 2006).	+ It offers comprehensive support for the control flow and data perspectives (Russell et al. 2006). - Some of the constructs lack a precise syntax and semantics (Russell et al. 2006).	- It does not fully capture important kinds of synchronization (Russell et al. 2006). - It is extremely limited in modelling resource-related or organizational aspects of BP (Russell et al. 2006). - Can not define various role levels of granularity (White 2004). + Supports signal sending and receiving at the conceptual level (Dumas et Hofstede 2001). + Supports both waiting and processing states (Dumas et Hofstede 2001). + Provides a seamless mechanism for decomposing an activity into sub-activities (Dumas et Hofstede 2001).
BPMN	+ BPMN gets popular and the interest in the industry and companies gets	- BPMN elements are hard to sketch on paper (Wohed et al. 2006). + Overall well-	- BPMN is easily learned for simple use (Wahl et Sindre 2005b). + BPMN particularly	- BPMN does not directly support pre- and post condition definitions (Wohed et al. 2006). - BPMN does not support

	growing up (Tscheschner, 2006) + The two most widely used graphical notations for business processes are BPMN and AD (Geambasu 2012).	suited for the domain of business process modelling (Wahl et Sindre 2005b).	excels in terms of comprehensibility appropriateness (Wahl et Sindre 2005b).	concepts hierarchy (S. A. White). -BPMN is not able to model other domains concepts (S. A. White). + Enables roles to be defined at various levels of granularity through pools and swimlanes (White 2004). + Captures the interleaved parallel routing and the synchronising merge patterns (Wohed et al. 2006). + Supports the resource perspective (Wohed et al. 2006).
EPC	+ EPC is widely used in commercial projects from the field of management information systems (Ferdian 2001).	- It is not classified as a Graphical Standard (Kindler 2004). + Easy for non technical users to pick up.	- The semantics and syntax of the EPC are not well defined (Kindler 2004). - Ambiguity caused by the not well defined presentation of concepts (W. M. P. van der Aalst 1999).	+ EPC does present the information inputs and outputs of the functions - EPC does not present activity or functions hierarchy. .
Petri-nets	+ Petri nets variants are widely used as a workflow modelling technique (Eshuis et Wieringa 2002).	+ Petri nets are fairly straightforward (Bosilj-Vuksic, Giaglis, et Hlupic 2001).	+ Petri nets are intuitive and easy to learn (V. D. Aalst 1998a).	- It lacks of standardization (Bosilj-Vuksic, Giaglis, et Hlupic 2001). + Petri nets are suitable for complex models. + Petri nets support all the primitives needed to model a workflow process. It can model all the routing constructs (V. D. Aalst 1998a).
Eriksson-Penker	- It is not widely used.		- It is comprehensible as each concept is defined by a stereotype (Penker et Eriksson 2000).	- It is not detailed regarding events and synchronization. + It is well detailed regarding different types of resources.

Table 1: Notation’s strengths and drawbacks

BPMN is an increasingly important standard for the process modelling and offers a wide range of modelling constructs. Its development has been based on the revision of other notations including UML, IDEF, ebXML and Event-driven Process Chains (OMG Final Adopted Specification 2006). Muehlen et al. (Muehlen et Recker 2008) concluded that not all the BPMN constructs are equally used, less than 20% of the vocabulary is used regularly. Unified Modelling Language (UML) has emerged as the software industry’s dominant modelling language (Keng Siau et Cao 2001). UML and BPMN contain a large set of constructs in contrast to competing languages, and offer a multitude of options for conceptual modelling. Both have been found in analytical studies to be not only semantically richer but also theoretically more complex than other modelling languages (Keng Siau et

Cao 2001). Similarly, related studies found that frequently less than 20% of the constructs are used (Siau, Erickson, and Lee 2005), (Kobryn 1999).

Muehlen found that the practical complexity of BPMN is remarkably less than its theoretical complexity (Muehlen et Recker 2008). Practical complexity is measured by the number of constructs actually used in a model while theoretical complexity its measured by the number of constructs originally specified (Keng Siau, Erickson, et Lee 2005) .

On the other hand Siau et al. (Keng Siau et Cao 2001) evaluate the complexity of UML using complexity metrics. They concluded that UML is not distinctly more complex than techniques in other object oriented methods. But as a whole, UML is very complex.

Other studies evaluate process oriented modelling techniques such as (Lars-Olof, Magnus, et Sven 2009) where they evaluate BPMN, Flowcharts, EPC and UML activity diagrams using Moody's quality criteria: discriminability, perceptual and cognitive limits, emphasis, cognitive integration, perceptual directness, structure, identification, expressiveness and simplicity. Based on the used quality criteria, this indicates that there are problems to be solved when it comes to modelling techniques. BPMN reached a high score but it is far from the maximum sum which indicates that even BPMN could be improved. All four process modelling techniques have a negative score on emphasis and expressiveness, a neutral value on perceptual directness and identification, and a positive value on simplicity. All these findings indicate that if there is a need to choose which notation to use, the decision can not be done based on the representation power. It is better to base the choice between the notations on other matters such as what I want to present in my model. If I want to discuss the process workflow I have to use EPC notation. If I want to discuss the different types of resources, I need to use Eriksson-Penker notation, etc.

2.4 Business process modelling tools

Business process modelling tools provide business users with the ability to model their business processes, implement and execute those models. As our thesis aims to improve the quality of business process models, we are interested in the BP modelling tools. These tools are developed to describe and analyze business processes.

There exist numerous modelling tools for designing business process models based on different notations such as BPMN, Petri nets, and EPC, etc. However, only some of them provide means to evaluate the quality of the models. And even fewer provide improvement guidelines. In this section we will assess some of these modelling tools.

2.4.1 Bizagi

Bizagi is a business process modelling and documentation tool that allows to model and document business processes in BPMN (Business Process Model and Notation). It offers the capability to export the BP diagrams into: Microsoft Visio, Image files (png, bpm, svg or jpg format) and to XPD. L.

Bizagi provides powerful simulation capabilities that enable organizations to make better decisions by visualizing the impact of proposed ideas. Bizagi simulation is robust and easy to use to better illustrate the advantages of analyzing your business operations using simulation in Bizagi.

Bizagi Simulation comprises four levels:

- Process Validation: checks if the process flow works as expected insuring calls and flow rates followed according to the scenario. This step will prevent modelling errors showing up in the results of the excessive levels.
- Time Analysis: measures the end-to-end process time by defining the time required by each activity. It calculates the processing time under the current conditions.
- Resource Analysis: predicts how the process will perform different levels of resources. It will show the resource utilizations and highlight the possible delays in the process.
- Calendar Analysis: reflects the process performance over dynamic periods of time as shift works, weekends and holidays that may affect the real performance of the process.

To conclude, Bizagi Process modeller does not validate the notation and the logic used in your diagram. However, it validates the elements locations and connections. It also highlights the real world problems over utilization of resources and significant process delays. Moreover, it provides the capability to compare two process models via the resource utilization and the process time.

2.4.2 Star UML

StarUML is an open source modelling tool designed to replace commercial applications such as Objectteering and Rational Rose. It supports UML diagrams i.e. activity diagrams and business process models based on Eriksson-Penker notation by adding the Eriksson-Penker Extensions (EPBE) v1.0 module. To start with the Eriksson-Penker Extensions diagrams we require the EPBE module. This module is developed by Hans-Erik Eriksson and Magnus Penker. The Eriksson-Penker Business Extensions form a basic framework for business extensions to UML to which a business architect can add stereotypes or properties suitable to his or her line of business. Similar to Bizagi, StarUML validates the elements locations and connections.

Secondly, StarUML allows creating Activity diagrams and verification of software models by applying the basic UML regulations. By default there are thirty-eight regulations defined for verifying models. These definitions are mostly adaptations of the Well-formedness Rule in the UML specification. StarUML doesn't provide any evaluation report.

2.4.3 Visual Paradigm

It allows to draw the following kinds of diagrams to aid in business process modelling: Business process diagrams with Business process modelling notation (BPMN) 2.0 standard, Data flow diagram which represents flows of data, Event-driven process chain diagram supported by EPC notation, Process map diagram which provides a high level view of business process. The process in the map can drill down using business process diagrams and the process link shows relationships between different business processes and the organization chart that shows the structure of an organization.

From the evaluation point of view, the business process simulation in Visual Paradigm simulation edition helps us to:

- identify bottlenecks,
- quantify candidates for task automation,
- quantify benefits from possible process change for comparison.

It analyzes numerically how much time and money will be spent under the model conditions.

Visual also defines scenarios for load testing i.e. a number of cases of a possible path.

The analysis results can be shown as charts that present the resource usage, queue time, cost per flow object, cost per input and time cost.

Visual paradigm allows the modeller to define and describe business operations, definitions and constraints. Moreover, it detects syntactic errors and suggests corrections. For example it detects an invalid connection and suggests another type of connection which applies the BPMN specifications.

2.4.4 ARIS

ARIS (Architecture of Integrated Information Systems) (Scheer et Nüttgens 2000) is a professional tool for the dynamic analysis of business processes. It is developed by IDS Scheer AG (www.ids-scheer.nl). It uses the modelling language Event-driven Process Chains (EPC).

ARIS provides a business process analysis platform which is ideal for organizations that want to document, analyze, standardize, and improve their processes. They can respond faster to changing business and market requirements. It offers methods for analyzing processes and taking a holistic view of process design, management, work flow, and application processing by analyzing the process throughput times, dynamic wait time and costs rates.

Based on process models and organizational structural simulation, comparisons of actual and targeted processes are enabled in terms of execution ability and efficiency. The focus can also be lading on:

- costs,
- execution time,
- resource usage.

These simulations provide answers about throughput times, weak points, bottlenecks and resource requirements. The results can be provided through statistics, animations or charts. Thus, the simulation's statistics can be used to process analysis and optimization.

ARIS is intended to serve various purposes: documentation of existing business process types, blueprint for analyzing and designing business processes, and support for the design of information systems (Lankhorst 2013).

2.4.5 MEGA

MEGA Process (« MEGA Process » 2013) provides capabilities for modelling and documenting business processes and organizational structures. This business architecture serves as a reference for decision making and impact analysis for organizational choices at the business, the operational, and the IT levels. Coupled with MEGA Simulation, MEGA Process helps compare and optimize business processes by testing multiple scenarios and their potential outcomes.

MEGA main features are:

- process modelling,
- organizations modelling,
- analysis of the information flows,
- link business process to systems currently in use,

- support quality management initiatives.

Mega tool enforces rules to make sure the main modelling standards are respected when the modeller describes his/her process. It detects the syntactic errors by activating the BPMN constraints and pointing out the defect with a red circle. Moreover it can show which BPMN rule has been broken.

The modeller can also identify where risks can occur in the process and the severity of the risk. It allows the modeller to design an alternative version of a process. Thus he/she can analyze the impact of changes by generating an impact analysis report.

2.4.6 Woflan

Woflan (WorkFLow Analyzer) is a Petri-net-based Workflow Diagnosis Tool (Verbeek, Basten, et Aalst 1999) which can be used to verify the correctness of a workflow procedure. It is addressed to analyze workflow process definitions specified in terms of Petri nets. It uses state-of-the-art techniques to find potential errors in the definition of a workflow procedure and gives workflow designers a handle to construct correct workflows. It guides the modeller of a workflow process definition towards finding and correcting possible errors. It can be used to verify a process definition i.e. it checks the syntactic and the behavioural properties.

Two of the key concepts on which Woflan is based, are the definition of a workflow net and the soundness property (W. M. P. van der Aalst 1999). For a given workflow net, Woflan is able to decide whether it is sound. Finally, Woflan guides the user in finding and correcting the error. To assist the user in repairing the error, Woflan offers an on-line help facility.

Business process modelling activity should not be restricted to the drag and drop of different concepts. It should analyze the semantics of the concepts and detect the errors provided by the modeller. Thus, business process modelling tools should propose guidelines to correct the defects. (Table 2) summarizes the above mentioned points about the six evaluated tools.

2.4.7 Tools comparison

Product Criteria	Bizagi	Star-UML	Visual Paradigm	ARIS	MEGA	Woflan
For use by business users or engineers?	Business	Engineers	Engineers	Business and engineers	Business and engineers	Business and engineers
Drag and drop process mapping?	Yes	Yes	Yes	Yes	Yes	Yes
Supports actors?	Yes	Yes	Yes	Yes	Yes	Yes
Support process hierarchies and sub process?	No	No	No	No	Yes	No
Business rules documentation?	Yes	Yes	No	No	Yes	No
Export to XML/XMI/XPDL?	Yes	No	Yes	No	Yes	No
Available for free trial?	Yes	Yes	Yes	Yes	No	Yes
Does it support BPM simulation?	Yes	No	Yes	Yes	Yes	Yes
Does it detect BPM defects?	Yes	No	No	No	Yes	Yes
Does it provide improvement guidelines?	No	No	No	No	Yes	No
Adopt which notation?	BPMN	Eriksson-Penker And AD	BPMN, EPC, DEF	EPC	BPMN	Petri nets

Table 2: Tools comparison

To summarize, all modelling tools support actors and represent in different ways the actors participating in the model. In contrast to the process hierarchies, EPC for example does not support

functions hierarchy. Thus all the EPC modelling tools do not present functions hierarchy, similarly to Petri nets.

Bizagi, StarUML and Mega allow the modeller to define business rules documentation. Different categories of business rules (White 2009) are possible:

- Business terms : documented in a glossary or entities in the model as in Bizagi and Mega.
- Facts relating terms to each other that can be documented as relationships, attributes etc, such as in Mega
- Constraints as in StarUML.

Almost all the modelling tools provide the feature to support BPM simulation, and study the variation of cost, time and resource allocation.

Regarding the business process model defects detection, Bizagi detect the very basic defects such as bad emplacement of connectors. On the other hand, Mega does detect BPMN standards defects, as lack of start event, end events, edges conditions ...etc. And Woflan does detects behavioural defects as deadlocks, live locks ... etc. None of the tools detects syntactic and behavioural defects in the same time, nor semantic and behavioural defects.

Moreover, none of the modelling tools do provide improvements in order to correct these defects, except Mega which points out the defects and suggests some corrections.

2.5 From modelling to quality

Business process models are a direct input into the software development process (« Business-Driven Development » 2005). Consequently, the impact of the process models on the operational efficiency of an enterprise is increasing. However empirical studies show that more than half of the errors which occur during system development are requirements errors (Endres et Rombach 2003). Zultner (Zultner 1993) suggests that it would be more effective to concentrate quality insurance efforts in the requirements analysis stage.

The choice of the best representation of data is one of the most crucial tasks in information systems (IS) since it impacts the quality of the final system. It does not only determine the costs, the flexibility of the system but also organizes the work of a developer and gives the ability to the system to meet the user requirements (Simsion 1994). Thus, in the work of Zultner (Zultner 1993), the authors argue that the total quality management approaches suggest that it is faster and cheaper to concentrate effort during the early development phases of a product in order to detect and correct defects as early as possible. Also according to Boehm (Boehm 1981), the cost of removing a requirement defect at the design phase is 3.5 times more than removing the same defect at the analysis phase, and removing it at the implementation phase is fifty times more. In order to have a reliable development phase, we have to ensure its quality.

When exploring related work on the business process model quality, we found only very few papers that address the measurement of quality for process models. We can cite the work of Koehler and al.

(Koehler et Vanhatalo 2007) who introduced the anti-patterns for process models that allow us to measure an important aspect of the quality of process models. Anti-patterns capture typical design errors in a process model, which make the process model incorrect. On the other hand, Becker et al. (Becker, Rosemann, et Uthmann 2000b) present a framework to structure factors for the evaluation of process models: Guidelines of Modelling (GOM) which structures different quality criteria as correctness, relevance, economic efficiency, clarity, comparability, and systematic design. GOM aims to develop specific design recommendations in order to increase the quality of models beyond the fulfilment of syntactic rules.

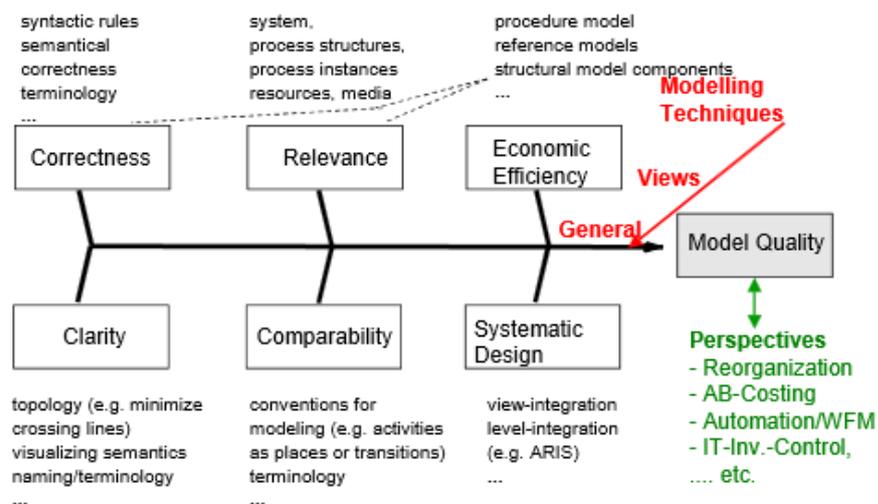


Figure 7: The framework of the Guidelines of Modelling (Becker, Rosemann, et Uthmann 2000b)

2.5.1 Quality of business processes

Goals of business process modelling are: to facilitate the understanding of the key mechanisms of an existing business, to serve as the basis for the creation of appropriate information systems supporting the business, to improve the current business structure and operation, to show the structure of an innovated business, to identify outsourcing opportunities, and to facilitate the alignment of business specifications with the technical framework that IT development needs (Aguilar et al. 2006).

In order to meet customer demands, companies have to design business processes in an appropriate way. In particular, four essential process competencies have been discussed in business process management: process cost, process flow time, process flexibility, and process quality (Anupindi et al. 2011). Each of these points has been subject to dedicated research. We are interested in the business process quality.

Measuring metrics enables the organizations to improve their business processes: planning, monitoring and control software projects and evaluate software quality. Using these parameters can produce artefacts (software, components, etc.) that are likely to be less error prone, easy to understand, maintain and manage (Cardoso 2007). We are interested in the business process quality.

Quality of business processes refers to the ability of a process to produce and deliver quality products (Anupindi et al. 2011). We aim in the next sections to seize from the literature the existing frameworks and standards of BP quality.

2.5.2 Dimensions of BP quality (frameworks and standards)

We can start with Heravizadeh and al. study (Heravizadeh, Mendling, et Rosemann 2009). They have introduced the QoBP which is a framework for capturing the quality dimensions of a process. They have defined four categories of the business process quality:

- Function quality: Concentrates on the quality of functions that correspond to an activity and need to be executed. Thirteen quality dimensions have to be studied in order to evaluate activities quality.
- Input and output quality: They put a particular emphasis on data and information quality as they capture the physical and the informational objects that are consumed and produced by the activities. Eleven quality dimensions are adopted and gathered from different studies.
- Human Resource quality: The quality of business process is influenced by the competency required to execute a function within a business process. As a consequence, they build their own six human resource quality attributes.
- Non Human resource quality: The non-human resources also influence the quality of functions and the business process as a whole.

For each of these categories, metrics are defined in order to evaluate them. (Table 3) summarizes these metrics.

Function	Input/Output	Non Human Resource	Human Resource
- Suitability	- Accuracy	- Suitability	- Domain Knowledge
- Accuracy	- Objectivity	- Accuracy	- Qualification
- Security	- Believability	- Security	- Certification
- Reliability	- Reputation	- Reliability	- Experience
- Understandability	- Accessibility	- Time efficiency	- Time Management
- Learn ability	- Security	- Resource Utilization	- Communication Skills
- Time efficiency	- Relevancy	- Effectiveness	
- Resource Utilization	- Value Added	- Safety	
- Effectiveness	- Timeliness	- User Satisfaction	
- Productivity	- Completeness	- Robustness	
- Safety	- Amount of Data	- Availability	
- User Satisfaction			
- Robustness			

Table 3: Dimensions of BP quality (Heravizadeh, 2009).

The QoBP approach has been presented in more details in (Recker et al. 2009). In this context, quality has been defined as non-functional but distinguishing characteristics of a business process. They provide a framework for organizations to manage the quality of business processes during different phases of the BPM lifecycle.

On the other hand, Heinrich and Paech proposed a business process quality framework based on software quality (Heinrich et Paech 2010). By looking at typical process problems they derived attributes and measures for the characteristics. They discuss these attributes and measures based on QoBP categories. The characteristics, attributes and measures for the activities are presented in (Table 4).

Characteristic / Attribute	Base Measure
Mean occurrence of errors	Number of error messages and failures and number of evaluations
Capacity of the resource with respect to activity	Number of cases in which a resource is not available
Amount of resources	Number of resources involved
Adequate resource usage	
Activity time	
Opinion of the actor	Number of complaints by the actors
Opinion of the customer	Number of complaints by the customers

Table 4: Characteristics, attributes and measures of activity (Heinrich et Paech 2010)

Finally, by looking at typical process problems, they concluded that only part of the characteristics and the ISO-derived measures are really relevant for business processes. Accordingly, the notion of quality for conceptual models (CM) in general and business processes in particular is not well established (Krogstie 1998). Attempting to describe it, studies have mainly concentrated on providing lists and collections of features and properties of CM. Comprehensive frameworks have been proposed which attempt to organize and structure the key concepts and features of quality in conceptual modelling (Table 5).

Other frameworks in the literature, as the conceptual modeling quality framework (CMQF) (Nelson, Poels et al. 2012) which is a combination and an extension of both the LSS (Krogstie et al. 2000) and the BWW (Wand et Wang 1996) quality frameworks. Their framework contains eight quality cornerstones and four “layers” : Physical, knowledge, Learning and development layer.

Approach	Purpose	Characteristics
----------	---------	-----------------

Lindland, Sindre et Solvberg (1994)	Understanding quality in CM	Linguistic base separation of goals from means
Pohl (1994)	Designing goals and process dimensions for requirements modelling	Specification, representation and agreement dimensions
Moody et Shanks (1994)	Evaluating the quality of entity relationship (ER) models	Quality factors, strategies and evaluation methods
Krogstie, Lindland et Sindre (1995)	Understanding quality in CM	Lindland, Sindre and Solvberg (1994) with agreement goal and social construction theory
Cherfi and al. (S. S. Cherfi, Akoka, et Comyn-Wattiau 2006)	Use case refinement process hat combines quality metrics with use case transformation rules.	Transformation rules and quality metrics.
Nelson , Poels and al. (Nelson, Poels et al. 2012)	Conceptual Modeling Quality	Organizing the various quality cornerstones and then defining the many quality dimensions that connect one to another.

Table 5 : Quality approaches of conceptual models

2.5.2.1 Lindland/Sindre/Solvberg's framework

The basic idea of Lindland et al. 's framework (Krogstie, Lindland, et Sindre 1995b) (Lindland, Sindre, et Solvberg 1994b) is to evaluate the quality of models along three dimensions: syntax, semantics, and pragmatics.

- The model is the set of all the statements explicitly or implicitly made in the model.
- The language is the set of all statements which are possible to make according to the vocabulary and grammar of the modelling language used.
- The domain is the set of all statements which would be correct and relevant about the problem at hand.
- The audience interpretation, i.e., the set of all statements that the audience (i.e., various stakeholders of the modelling process) thinks the model consists of.

The primary sources for model quality are defined using the relationships between the model and the three other sets:

- Syntactic quality is the degree of correspondence between model and language.
- Semantic quality is the degree of correspondence between model and domain. It allows detecting the existence of invalid statements or the incompleteness of models.
- Pragmatic quality is the degree of correspondence between model and audience interpretation (i.e., the degree to which the model has been understood). Usually, it is neither necessary nor possible that all stakeholders understand the entire conceptual model. Instead each member of the audience should understand the part of the model which is relevant to him. The main structure of the framework by Lindland et al. is illustrated in (Figure 8).

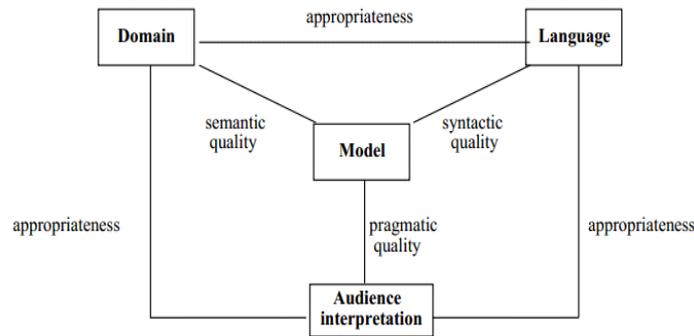


Figure 8: Lindland and al.'s quality framework (Krogstie, Lindland, et Sindre 1995a)

2.5.2.2 Krogstie, Lindland et Sindre's framework

Krogstie, Lindland and Sindre's framework (Krogstie, Lindland, et Sindre 1995a) is based on semiotic theory. It is composed of five components: model, language, domain, audience participation and perceived knowledge. Model quality is defined by relationships between the model and the other four components. These relationships use four semiotic levels: syntactic, semantic, pragmatic and social. They enrich Lindland et al's framework with a new component, the social quality.

- Social quality has the goal of feasible agreement between the actors, where inconsistencies between the various actors' interpretations of the model are resolved. Means of achieving feasible agreement include model comprehension, conflict resolution, and model merging.
- Syntactic quality goal is syntactic correctness and the means of achieving it is by syntax checking.
- Semantic quality goal is the model completeness and perceived validity. The means are consistency checking and audience interpretation.
- Pragmatic quality goal is the model comprehension (understandability). Means to achieve include structuredness and executability.

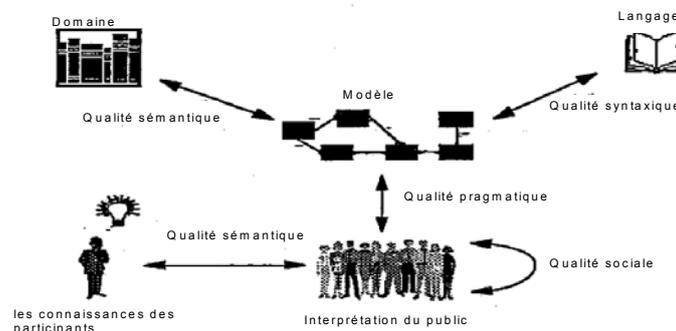


Figure 9: Krogstie et al.'s Quality framework (Krogstie, Lindland, et Sindre 1995a)

2.5.2.3 *Moody et Shanks's framework*

Moody and Shanks propose a framework for evaluating the quality of conceptual data models (D. L. Moody et Shanks 1994). It is intended to provide practitioners with a coherent approach to resolving two key problems which arise in the practice of data modelling: the need to choose between a number of alternative data models and the need to understand and accommodate the different views of the various stakeholders in the data modelling process.

The first problem focuses on what makes one model of higher quality than another, and the second focuses on how the quality of a data model can be improved to meet the needs and expectations of all stakeholders. This framework provides a systematic basis for the evaluation of data models in practice. It consists of seven main components: model, quality factor, stakeholder, evaluation method, weighting, rating, and strategy.

- A quality factor is a desirable property of a data model. Five quality factors are proposed: correctness, completeness, understandability, flexibility, and simplicity.
- A stakeholder is a participant involved in the data modelling process (business users, data analysts, application developers, and data).
- A strategy is a process or activity which can be used to increase the value of a data model with respect to one or more quality factors.
- An evaluation method or metric is a systematic way of measuring a quality factor.
- A weighting is a value assigned to a quality factor which represents its relative importance in the context of the project.
- A rating is a value assigned to a quality factor representing its validation in a particular model by a stakeholder.

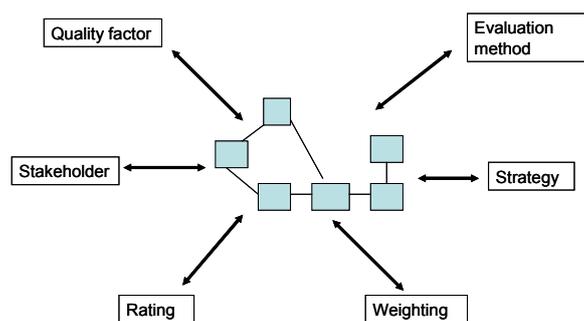


Figure 10: Concepts in the framework of Moody and Shanks (1994)

Lindland, Sindre and Solvberg's framework purpose is to understand quality in conceptual modelling based on the separation of goals from means. Similarly Krogstie, Lindland and Sindre's framework purpose is to understand quality in conceptual modelling by adding the agreement goal and the social construction theory. Both frameworks focus on the theory. On the other hand Moody and Shanks framework purpose is to evaluate the quality of entity relationship models based on evaluation methods. It focuses on the practice.

Business process modelling is an important tool for understanding and revealing weakness of business processes. The quality of a business process builds on the quality of its functions, input and output, non human resource and human resource participating in the process (Heravizadeh, Mendling, et Rosemann 2009). As all these concepts are present when modelling the business process, the quality of business process is directly influenced by the quality of its model.

2.6 Process model quality measurement

Quality measurement procedures aims to quantify the quality level (Overhage, Birkmeier, et Schlauderer 2012). Abd Ghani et al. (Abd Ghani et al. 2008) proposed a Goal-Question-Metric (GQM) framework for measuring the understandability and maintainability of BPMs. The Goal-Question-Measure approach consists in identifying customers goals, and then identifying a set of target attributes. In other words a GQM defines project goals and a set of questions to achieve each goal. Then the team develops metrics to address each question. One of the questions asked is: how easy is it to read the model? The answer would be the number of symbols and formulas used and type of structures used.

We choose to be based on Lindland's framework. In fact, based Moody et al. (D. L. Moody et al. 2002) calculated the Completeness (Sufficiency) of quality Categories: syntactic, semantic and pragmatic by measuring the proportion of the variance of the dependent variable explained or predicted by the independent variables. They found that Lindland's framework categories account for more than 93% of the variance in Overall Quality. This provides strong evidence that the set of quality categories is rather complete. Moreover, they studied the independence of the quality Categories, there was no evidence of multicollinearity between the quality categories.

2.6.1 Syntactic quality

The syntactic analysis deals with the purely structural aspects of the model. There is only one syntactic goal, syntactic correctness, meaning that all statements in the model are compliant to the syntax of the language. All the articles proposing quality metrics for business process models mention the similarities between a software process and a business process. Basing on this hypothesis, they apply software metrics to evaluate business process models. In this section, we will analyze and summarize the syntactic metrics existing in the literature into four categories: size, complexity, structure, and modularization.

2.6.1.1 Size of the model

In an attempt to minimize the influence of “size”, there are hundreds of software complexity metrics that have been described and published by a significant number of researchers. Three size measures are reported in (J.-P. Van-Belle 2004): 1) entity counts the total number of entities (classes) in the

model, 2) CASE size (or concept counts the number of entities, relationships and group elements) and 3) the number of lines of code (LOC) which simply counts the lines of executable code, data declarations, comments, and so on. While these measures are extremely simple, they have been used successfully for the purposes like predicting the error rate, estimating development and maintenance costs (J.-P. Van-Belle 2004).

2.6.1.2 Complexity of the model

We should distinguish between the structural complexity and the workflow complexity of the model. Aguilar et al (Aguilar et al. 2006) suppose the adaptation and extension of the FMESP(framework for the modelling and evaluation of software processes) (García et al. 2006). They measure the structural complexity of software process models. The aim is to evaluate the influence of the structural complexity of the software process models on their maintainability.

Metric	Definition and formula
NA	Number of activities of the software process model.
NWP	Number of work products of the software process model.
NPR	Number of roles which participate in the process.
NDWPin	Number of input dependencies of the work. Products with the activities in the process.
NDWPOut	Number of output dependencies of the work products with the activities in the process.
NDWP	Number of dependencies between work products and activities. $NDWP(PM) = NDWPin(MP) + NDWPOut(MP)$
NDA	Number of precedence dependencies between activities.
NCA	Activity coupling in the process model. $NCA(PM) = NA(PM) / NDA(PM)$
RDWPin	Ratio between input dependencies of work products with Activities and total number of dependencies of work products with activities. $RDWPin(PM) = NDWPin(PM) / NDWP(PM)$
RDWPOut	Ratio between output dependencies of work products with activities and total number of dependencies of work products with activities. $RDWPOut(PM) = NDWPOut(PM) / NDWP(PM)$
RWPA	Ratio of work products and activities. Average of the work products and the activities of the process model. $RWPA(PM) = NWP(PM) / NA(PM)$

RRPA	Ratio of process roles and activities. $RRPA(PM) = NPR(PM) / NA(PM)$
------	---

Table 6:FMESP set of metrics (Aguilar et al. 2006)

On the other hand, the authors in (Cardoso 2007) began with the hypothesis that high complexity in workflows may result in poor understandability, errors, defects, and exceptions leading processes to need more time to develop, test, and maintain. Therefore, they describe a quality metric to analyze the complexity of workflow patterns from a log-based perspective. They define log-based complexity metric as follows:

- $LBC = 1$ if we have a single activity, since it only generates one entry in the process log.
- $LBC = \prod LBC$ if we have an ordered series of tasks, with one task starting after a previous task has been completed
- $LBC = \sum pi * LBC$ if we have an exclusive choice and deferred choice where pi is the probability of following a specific path at runtime.

In (Cardoso, 2007) the Control-Flow Complexity (CFC) metric evaluates the complexity introduced in a process by the presence of XOR-split, OR-split, and AND-split constructs. For XOR-splits, the control-flow complexity is simply the fan-out of the split, i.e. $CFC_{XOR-split}(a) = fan-out(a)$. For OR-splits, the control-flow complexity is $2n-1$, where n is the fan-out of the split. i.e. $CFC_{OR-split}(a) = 2fan-out(a)-1$. For an AND-split, the complexity is simply 1, i.e. $CFC_{AND-split}(a) = 1$.

It is necessary to be able to evaluate the quality of a business process model, which in turn requires a set of quality metrics. The latter provide guidance to improve business process models by giving the modeler information about different quality criteria of business processes such as complexity, cohesion, etc.

2.6.1.3 Structure of the model

One specific category of BP syntactic quality metrics is coupling, which measures the functional and informational dependencies between the tasks/processes in a business process model. Coupling in business process models (BPM) focuses on how strongly the activities in a business process are related, or connected, to each other. The application of these measurements is straight forward if the process model is available in a graph-based notation (Khlif et al. 2009). The average degree, also called coefficient of connectivity refers to the average number of connections that a node has with other nodes of the process. In contrast to that, the density metric links the number of available connections to the number of maximum connections for the given number of nodes.

Moreover the definition for coupling used in (Vanderfeesten, 2013) was taken from the definitions found in the software engineering area since it measures the number of interconnections between the

activities in a process model. The degree of coupling depends on how complicated the connections are and also on the type of connections between the activities. So they define a new coupling metric, based on the existing ones and inspired by software metrics, which weights different connections between activities (e.g. AND, OR, XOR). This metric is defined below.

$$CP = \frac{\sum_{t1, t2 \in T} \text{connected}(t1, t2)}{|T| * (|T| - 1)}$$

where $\text{connected}(t1, t2) =$

1	,if $(t_1 \rightarrow t_2) \wedge (t_1 \neq t_2)$
1	,if $(t_1 \rightarrow AND \rightarrow t_2) \wedge (t_1 \neq t_2)$
$\frac{1}{(2^m - 1)(2^n - 1)} + \frac{(2^m - 1)(2^n - 1) - 1}{(2^m - 1)(2^n - 1)} \cdot \frac{1}{m.n}$,if $(t_1 \rightarrow OR \rightarrow t_2) \wedge (t_1 \neq t_2)$
$\frac{1}{m.n}$, if $(t_1 \rightarrow XOR \rightarrow t_2) \wedge (t_1 \neq t_2)$
0	, if $(t_1 = t_2)$

in which $t1$ and $t2$ are activities, m is the number of ingoing arcs to the connector, and n is the number of outgoing arcs from the connector.

Moreover in (Khlif, Zaaboub, et Ben-Abdallah 2010), the author's contribution consists in adapting object oriented software coupling metrics for business process models. This adaptation is based on correspondences they established between concepts of the Business Process Modelling Notation and object oriented concepts.

- ICP (Imported Coupling of a Process): counts, for each (sub-) process, the number of message/sequence flows sent by either the tasks of the (sub-) process or the (sub-) process itself.
- ECP (Exported Coupling of a Process): counts, for each (sub-) process, the number of message/sequence flows received by either the tasks of the (sub-) process or the (sub-) process itself.
- MPC metric: In the business process domain, it is the number of message or sequence flows sent directly from a process task to a task of another process. Note this measure takes into account only the coupling in terms of interaction between tasks. Thus, it helps to estimate the degree of dependency between the process tasks and the tasks of other processes.
- RFC metric: $RS = \{Tj\} \cup \{Ri\}$ is the set of all responses of a process, where $\{Ri\}$ is the set of resources invoked by a task i in the process and $\{Tj\}$ is the set of all tasks j in the process. Note that, the larger the RFC is, the greater the complexity of the process is: Indeed, if a large number of tasks can be invoked in response to a message, then the process becomes complex and requires a greater level of understanding.

- Locality of Data (LD): This metric links data from the activity (process or sub process) to the total data used by this activity.

2.6.1.4 Modularization of the models

Cohesion measures the coherence within the parts of the model. The authors in (Hajo A. Reijers et Vanderfeesten 2004) developed a cohesion metric for workflow processes which looks at the coherence within the activities of the process model. Similar to their coupling metric, this cohesion metric also focuses on the information processing in the process and takes a data oriented view. For each activity in the process model, the total cohesion is calculated by multiplying the information cohesion and the relation cohesion of the activity.

Dividing a BPM in modular sub-models cannot only help to make the BPM easier to understand, it can also lead to smaller, reusable models – if modularization is used in a reasonable way (Gruhn et Laue 2006). They measure the fan-in and fan out for every module, where fan-in is a count of all other modules that call a given module and fan-out is a count of all other modules that are called from the model under investigation. Finally, some research, such as (Jan Mendling, Neumann, et Van Der Aalst 2007a) aims to analyze the connection between formal errors and a set of metrics that capture various structural and behavioural aspects of a process model.

- Size SN, which refers to the number of nodes of the process model graph. An increase in SN should imply an increase in error probability.
- Coupling or density Δ , relates the number of arcs to the maximum number of arcs between all nodes. They presume a positive connection i.e. an increase in Δ implies an increase in error probability.
- Control Flow Complexity CFC, sums up all choices of a process based on the number of splits of each type and its number of outgoing arcs presume a positive connection.

Yet (Ehrig, Koschmider, et Oberweis 2007) have shown that calculating only syntactic and linguistic similarities is insufficient since the instance context is not considered and homonyms can not be discovered.

We shall now move to the second category of Lindland's framework, the semantic quality.

2.6.2 Semantic quality

The primary goal of semantic quality is to improve the correspondence between the model and the domain, but this correspondence can neither be established nor checked directly. To build the model, one has to go through the audience's understanding of the domain. To check the model, one has to compare it with the audience's interpretation of the model. Semantic quality improves model efficiency and effectiveness in the activity semantics, reducing the precision deficit in modelling. It leads to a better understanding and knowledge.

The framework in (Krogstie, Lindland, et Sindre 1995b) contains two semantic goals: validity and completeness.

- Validity: means that all statements made by the model are correct and relevant to the problem.
- Completeness: means that the model contains all the statements which would be correct and relevant about the problem domain. (Krogstie, Lindland, et Sindre 1995b) gives a definition for the degree of validity and completeness:

$$\text{Validity} = 1 - \frac{D}{M} \qquad \text{Completeness} = 1 - \frac{M}{D}$$

Where M represents all the statements made in the model, D represents the statements that are correct and relevant about the domain at hand.

We will analyze and summarize contributions on semantic quality measurement according to these two categories: validity and completeness of models.

2.6.2.1 Validity of the models

Process is aimed at attaining a goal, which is a set of stable states that satisfy a condition over a criterion function. A process model will be called a valid model if and only if there exists at least one successful process path, where a successful process path is a path that leads to a stable state which is in the process goal set. Based on this definition, they distinguish three types of situations when a process model is invalid i.e. (is not guaranteed to reach its goal).

1. Incompleteness in the process definition: It may be that a certain combination of state variables in the domain does not appear in the process definition. And incompleteness can relate to internal events where the step j of the process does not provide all the information needed to trigger the step j+1 of the process. It can be related also to external events.
2. Inconsistency between the functions of the process and the process definition: It is possible that when the process begins progressing, it reaches a state from which it cannot proceed further to reach a goal state. Two possibilities exist. First the process entered into an infinite loop where activities cause transitions without reaching a stable state. Secondly the process reached a stable state not in the goal. For both cases, there is no continuous path from the start state to a goal state.
3. Dependency of the process on external events, where the process is “waiting” for an external event, with no guarantee that this external event will eventually occur.

Validity also can be affected by the expressiveness, the readability and the clarity of the model. (J. V. Belle 2004) defines the expressiveness of the model. An expressiveness score can be calculated as the weighted index of the number of meta-model attributes covered in a model.

Model readability refers to the extent to which the model can be understood or comprehended by the users of the model. The perspicuity (clarity) analysis is based on matching all model element names against common domain vocabulary lists.

In (Soffer et Wand 2004) the authors propose a generic theory-based process modelling framework as well as criteria for validity evaluation of process models. They discuss and characterize causes for process invalidity and they suggest ways to avoid these situations.

2.6.2.2 Completeness of the models

Rozinat et al. in (Rozinat et Van Der Aalst 2013) describe an approach for measuring the compliance between events logs and process models. This compliance measurement indirectly enables the evaluation of BP model completeness. These BP models represent how the system should be used.

Pirro in (Pirró 2009) proposes a new semantic similarity that exploits some notions of the feature-based theory of similarity. In particular, the proposed metric exploits the notion of intrinsic information content (IC) which quantifies IC values by scrutinizing how concepts are arranged in an ontological structure.

- Similarity Metrics based on WordNet: it's a similarity metric between concepts that belong to the information theoretic approach, which can be considered as a measure quantifying the amount of information a concept expresses. Knowing the IC values for each concept, we may then calculate the similarity between two given concepts.
- Similarity Metrics based on ontology approach: two initiatives have been studied. The first referred to a depth based approach and the second to a path based approach.
- Similarity Metrics based on hybrid approach: it combines different sources of information to assess a score of similarity or distance between concepts.

The third and last category of Lindland's framework concerns the pragmatic quality and is summarized below.

2.6.3 Pragmatic quality

Pragmatics relates the model to the audience (those who are involved in modelling). It defines how well the model corresponds to its audience interpretation. Pragmatic model quality (PMQ) refers to the user's interpretation of the model. This is typically understood as the comprehensionability or understandability of a model (Gemino et Wand 2005), (Krogstie 2002). The common measure for comprehension involves asking the user a number of questions that relate to the meaning of the model and that can only be answered right if the model is understood (Rittgen 2010).

Krogstie et al (Krogstie, Lindland, et Sindre 1995b) confirmed that there is only one pragmatic goal: comprehension, that is, how well the model has been understood by the audience.

The authors in (Jan Mendling, Reijers, et Cardoso 2007) focused on the understandability of process models as an enabler of pragmatic quality. They concluded that measurement of the BP model

understandability should include the purpose of the BP model and the field of knowledge that it exploits.

Understandability is defined as the ease with which the concepts and structures can be understood by the users of the model (D. L. Moody et Shanks 1994). They propose three metrics to measure the understandability of a model.

4. Business user rating: The user must be able to understand the model in order to verify that it is a complete and accurate representation of his/her requirements.
5. Data administrator rating: He/She must be able to understand the model.
6. Application developer rating: He/She relies on the model for implementing it. Thus the model must be a complete specification of the user requirements.

Empirical studies of process modelling are aimed at gaining an understanding that can guide the development of higher quality models (Soffer, Kaner, et Wand 2012). Soffer et al. propose to study the process of process modelling based on problem solving theories. (Table 7) represents a summary of the research questions that can be asked with basic features of possible experimental designs.

Research question	Independent variable	Dependent variable	Point of measurement	Comments
Do model practices affect the model?	Modelling practices	Domain understanding	After model construction	Task should be related to domain understanding
How is the model affected by process size and complexity?	Process size and complexity	Domain understanding	Prior Model construction	Relates the domain behaviour, requires suitable process metrics
Are poor syntactic quality attributes due to problems of conceptualization or of mapping?		Correlation of domain understanding and syntactic model quality	After model construction	Test correlation between variables

Table 7: Possible experimental studies (Soffer, Kaner, et Wand 2012)

Unfortunately, most pragmatic criteria involve a substantial degree of subjectivity. Therefore the discussion in (J. V. Belle 2004) is limited to the two pragmatic measures which are believed to be more objective and universally applicable:

- Model authority refers to the acceptance of the model by practitioners in the field.
- Model flexibility is concerned with how well models can be changed or adapted to different situations.

Moreover, flexibility is defined as the ease with which the model can be adapted to changes in requirements. It is a major determinant of the maintenance costs of a system. The senior management,

the industry expert and the expert modeller can rate the flexibility of a model (D. L. Moody et Shanks 1994).

Based on the quality evaluation results, improvements should be applied on the model in order to improve its quality. Thus a quality approaches should contain two main parts: 1) Evaluation of the quality, which is done by different evaluation procedures and 2) Improvement of the quality, which is done by improvement guidelines.

2.7 Quality approaches

Quality approach's evaluation procedure covers the *measurement task* and the defining quality criteria task. Most proposals specify these *quality criteria* without evaluating how they should be measured. As a result, the approach should develop formal metrics for measuring quality of the models to reduce subjectivity. Quality approach's *improvement procedure* should cover the definition of guidelines, improvement patterns, recommendations and other actions that aim to improve the model. Most of the approaches in the literature focus on the evaluation of the model and ignore the issue of improving it. However, quality management approaches have not yet provided an effective solution to the quality evaluation and improvement issues. Progress in conceptual models can only be made by approaches that are able to integrate quality measures and schema improvements.

Graeme Shanks (Shanks 2007) studied the difference between conceptual models designed by domain experts and those designed by novices by evaluating them using a quality metrics synthesized from previous studies. They concluded that models produced by experts are more correct, more complete, more innovative and more flexible than those produced by novices. Similarly, Batra and al. (D. Batra et Davis 2013) found that conceptual models modelled by novices have problems with the domain description but both models are similar in terms of syntactic quality. None of these studies shows how to use this knowledge to improve the quality of models.

2.7.1 Approaches dedicated to conceptual models quality

There are many quality management studies which are not specific to the field of BPM but might be adopted. We will start with Eick's approach. He surveys the back end of a conceptual model called ANNAPURNA (Eick 1991). He introduced quality measures in order to evaluate the validity of the model. To be valid, the model should answer the following quality criteria:

- Completeness: The model is complete if it allows the specification that satisfies the information requirements.
- Rule correctness: The model is called rule correct if the rules defined in the model are not violated.
- High expressive power i.e. it should be possible to describe all rules that hold in a universe of discourse.
- Low complexity in terms of the number of classes and attributes needed for a particular specification.
- Normalized i.e. the equivalent or the similar objects are described in the same way in the model.

Furthermore, in his work, he defined schema transformations in order to improve the model and make it valid. The transformation based on unary and binary relations is possible to specify classes, subclasses and attributes.

Finally, he defined an evaluation function that incorporates the three criteria: complexity, normalizedness and expressiveness:

1. Complexity of the model is measured by the number of classes, subtype-connections, and attributes.
2. Normalizedness is measured by an evaluation function using the number of labels. The models that carry few labels are preferred.
3. The expressiveness of a model can be measured by the number of general functional (expressing at most one constraint) and existence dependencies that hold in the transformed model but which have not been expressed in the model.

In his approach, Eick evaluates the syntactic quality by measuring the complexity of the model, the semantic quality by measuring the normalizedness and finally the pragmatic quality by measuring the expressiveness of the model. As our thesis is oriented towards business process models, we are concentrating on the approaches dedicated the latter.

The quality evaluation procedure of Business process models is not limited to the analysis of the model structure quality. The semantics of the models also should be analyzed. For Dijkman et al. (Dijkman, Dumas, et Ouyang 2008), BPMN is dedicated to the specification of the syntactic constraints of models but it is inconsistent when it comes to defining the models semantics. On the other hand, Petri net is dedicated to study the semantic behaviour of the model. They proposed to map the BPMN to Petri nets in order to fulfill this gap. Each concept in BPMN is represented by a Petri net concept. These mappings have been implemented in a "ProM" tool, which verifies the soundness of BPMN models.

2.7.2 Approaches that evaluate business process model's quality

First of all, before evaluating the models quality, researchers have concentrated on the modelling competence level. As in Mendling and al's paper (Jan Mendling, Reijers, et van der Aalst 2008), they concentrated on the problem of the low modelling competence level and analyzed existing research on relationships between model structure on the one hand and error probability and understanding on the other hand. As a synthesis they proposed a set of seven process modelling guidelines (7PMG).

As a matter of fact these guidelines are defined in order to make the model more comprehensible, more understandable.

G1: Use as few elements in the model as possible

G2: Minimize the routing paths per element.

G3: Use one start and one end event.

G4: Model as structured as possible.

G5: Avoid OR routing elements.

G6: Use verb-object activity labels.

G7: Decompose the model if it has more than 50 elements.

The authors concentrate on several quality criteria that affect the quality of the business process models and propose seven guidelines to improve each of these criteria.

Furthermore, guidelines of Business process modelling are proposed by Becker et al. (Becker, Rosemann, et Uthmann 2000a). Guidelines of modelling (GoM) framework not only provide factors for the evaluation of process models but also guidelines of modelling for workflow management and simulation are presented. It adopts elements from existing approaches for the evaluation of models such as correctness, relevance, economic efficiency, clarity, comparability, and systematic design. But the basic guidelines consist of the guideline of correctness, the guideline of relevance, and the guideline of economic efficiency.

Other studies of Mendling et al, have concentrated on the understandability of the models (Jan Mendling et Strembeck 2008). They discussed the factors that influence the understandability of the business process models. In fact their contribution is based on two steps: first they identified three categories of understandability factors: personal, structural, and textual. Second, they presented the importance of these factors. Furthermore they calculated some model metrics such as size, structuredness, separability, cyclicity, heterogeneity, etc. and some metrics related to the textual labels such as Textlength, correct answer. Finally for the personal metrics they defined a Pscore (It captures the number of correct answers by the person.) and a Mscore (it captures the sum of correct answers for the model given by the participants).

An attempt to develop a "Quality of Business Process framework (QoBP)" was made by Heravizadeh et al. (Heravizadeh, Mendling, et Rosemann 2009). They identified four generic quality categories of business process quality and populate them with quality requirements from related research. The quality dimensions are arrayed along the categories of:

- Function quality: A function is an activity in a BPM which needs to be executed. They mentioned 13 quality dimensions derived from the literature to evaluate the activity and suitability, accuracy, security, reliability, understandability, learn ability, time efficiency, etc
- Input/Output quality: Inputs and outputs of a function or an activity represent the information or objects that are consumed and produced by it. Similarly, they adopted quality metrics from the literature such as accuracy, objectivity, accessibility, etc.
- Non human resource: Functions may be executed by non-human resources such as machines, devices, or software programs. It can be measured by the availability metric.
- Human resource: Functions may be executed by human resources. Metrics such as domain knowledge, qualification, certification, experience, etc may evaluate them.

Henrich and Paech proposed a business process quality framework based on software quality (Heinrich et Paech 2010). They used software quality to derive eight activity characteristics: Functionality, reliability, usability, efficiency, maintainability, portability, quality in use and compliance.

Business process reengineering and optimization is related to the BPM improvement domain. A framework for classifying the best practices was given in Reijers et al's paper (H.A. Reijers et Liman Mansar 2005). This framework described 29 best practices and evaluated qualitatively their impact on the cost, quality, flexibility and time criteria.

Evaluation is not restricted on the evaluation of the model structure. On the contrary, business process models contain control-flow errors such as deadlocks and the lack of synchronization. Such errors affect the code generation and the execution of these models. Therefore detecting and removing control-flow errors becomes a main goal of the Business process quality management studies. In the literature different approaches can be used to check the soundness of business process models such as the IBM WebSphere Business Modeller / SESE approach (Vanhatalo, Völzer, et Leymann 2013), CTL model checking used by the Petri net model checker LOLA (Wolf 2007), and Petri net analysis techniques used by Woflan (Verbeek, Basten, et Aalst 1999). To start with the Petri net analysis techniques, a Petri net model should at least satisfy the following conditions:

- It should always be possible to complete a case i.e. no deadlocks and no live-locks.
- It should not be possible to attend an end event while the process is not finished yet.
- Every task in the model should have a path to be attempted.

To conclude, the verification of a Petri net should guarantee the effectiveness and the efficiency of the model. They also developed a diagnosis process in order to guide the designer in finding and correcting the errors. To do so, they developed the Woflan tool.

Secondly, the single-entry-single-exit (SESE) technique aims to analyze and improve the business process models by decomposing the model into SESE fragments. This approach speeds up the control flow analysis by detecting the structuredness errors and the complexity of the model.

Thirdly, the LoLA tool generates state space reduction techniques for the purpose of verifying a particular property.

Some of the properties that can be verified include:

- Reachability of a given state,
- Existence of deadlocks,
- Existence of home state,
- Liveness of a state predicate.

Fourthly, a novel approach called "causal footprints" (Dongen et Mendling 2006), represents a set of conditions that should be verified by the process model. They identified three kinds of error patterns that affect the soundness of a process model: the deadlock pattern, the multiple termination patterns and the trap pattern.

In short, all these techniques aim to verify the soundness of business process models. Some provide analysis techniques; others propose metric evaluation and improvements. Some are limited to one notation such as SESE and others can be applied on several notations i.e. the Casual Print.

Hallerbach et al. discuss how to ensure soundness for an entire process family (Hallerbach, Bauer, et Reichert 2009). They have described the Provop approach that aims to ensure the soundness of the configurable variants of a whole process family taking into account semantic as well as structural constraints. They extended the ABIS approach of adaptive business process modelling with soundness concepts and with an algorithm to apply them.

An approach aimed to automatically correct errors in an unsound process model presented by (Gamma et al. 1994). Their approach considers three dimensions: the structural distance, behavioural distance and badness of a solution w.r.t. the unsound process model. The approach uses simulated annealing to simultaneously minimize all three dimensions. The edits applied to the process model are aimed to correct the model rather than to balance the five different forces.

Or applying Petri nets to workflow management systems has many advantages. One of the advantages beside their clear semantics, is the existence of Petri net verification techniques (W. M. P. V. D. Aalst 1998). Van der Aalst and al. (W. M. P. van der Aalst 1999), Pankratius et al. (Pankratius et Stucky 2005) formalize EPCs, BPMN and workflow models with Petri nets and provide them with clear behavioural semantics that can be analyzed.

Sabbagh et al. (Eid-Sabbagh et Weske 2013) introduced a novel Petri net based formalization of Business Process architectures, trigger-flow nets, that enable the analysis of business process architectures, trigger-flow nets, with know Petri nets analyzing techniques in regard to their structural and behavioural properties. They found several dead events and a deadlock that could not be identified by a pattern based verification technique.

2.7.3 Approaches for business process improvement

Forster (Forster 2013) defines Business Process Improvement as a systematic approach to help organizations to achieve significant changes in the way they do business. So far, academic research has mainly focused on the description of successful Business Process Improvement implementations. Usually only the situations before and after implementation are outlined, but not the actual act of improvement (Forster 2013).

Harrington (Harrington 1998) states that Business Process Improvement is basically the product of Business Process Reengineering, Redesign, and Benchmarking, depending on the degree of change necessary. Thus literature mentions many different terms relating to the management and improvement of Business Processes, including: Business Process Redesign, Business Process Reengineering and Business Process change.

Process Redesign: Process Redesign is described by Davenport et al. (Davenport 1998) as the analysis and design of work flows and processes within and between organizations. Business Process Reengineering aimed to help organizations fundamentally rethink how they do their work in order to dramatically improve customer service, to cut operational costs, and become world-class competitors

(Jr, Finedore, et Davis 1997). In order to redesign a business process, they defined five major steps involved:

Develop the business vision and process objectives: Business vision depends on the company's management. This vision can involve taking the perspective of the customer and developing systems rather than standalone products. It can also consist of improving product quality. Thus multiple objectives in redesigning processes, for example, set out to improve the cost, time, and quality.

Identify the processes to be redesigned: There are two major approaches to identify the processes to be redesigned: the exhaustive approach attempts to identify all processes within an organization and then prioritize them in order of redesign urgency. The high-impact approach attempts to identify only the most important processes or those most in conflict with the business vision and process objectives.

Understand and measure the existing process: There are two primary reasons for understanding and measuring processes before redesigning them. First, problems must be understood so that they are not repeated. Second, accurate measurement can serve as a baseline for future improvements. Thus measurement is related to the objectives of the redesigning process. For example if the objective is to cut time and cost, the time and cost consumed by the untouched process must be measured accurately.

Identify information technology levers: IT usage has always been to first determine the business requirements of a process, or other business entity, and then to develop a system. Thus the awareness of IT capabilities can influence process design.

Design and build a prototype of the new process: using IT as a design tool, understanding generic design criteria, and creating organizational prototypes.

Process Reengineering: Reengineering means to disregard all the assumptions and traditions of the way business has always been done, and instead develop a new, process-centered business organization that achieves a quantum leap forward in performance (Hammer et Champy 2003). Reengineering is defined as the fundamental rethink and radical redesign of business processes to generate dramatic improvements in critical performance measures such as cost, quality, service and speed. Based on Hammer et al.'s study, reengineering initiatives typically lead to simplified business processes, a multi dimensional description of tasks, a flatter arrangement of the organizational structure, a focuses on the professionals not the managers and an aligned with the end_to_end process. Additionally, Tawlar (Talwar 1993) define business process re-engineering as a way to deliver discontinuous leaps in performance. He mentioned two main approaches to re-engineering. The first-known as Process re-engineering, offers the opportunity to rethink and streamline individual processes. The second-termed Business Re-engineering provides an approach to rethinking and redesigning the entire business behind a more focused, competence based competitive strategy.

The central challenge in re-engineering is to understand where and how we can create value for both customers and shareholders. This requires us to ask fundamental questions about what we do, how we do it, whether it is necessary and how it can be improved.

Business Process Change: Paul Harmon's Business Process Change is an outstanding overview and a comprehensive guide to business process initiative (Harmon 2003).

Harmon compellingly argues that organizations are systems, and consequently, that processes are the optimal vehicle to analyze and manage organizations. Determining how process methodologies interact, what they involve, which ones are appropriate in which situation, etc., is difficult and not often covered by other texts. Business Process Change provides an easily digestible positioning of the interactions and applications of the numerous process approaches. After setting the context, Business Process Change moves into a very practical introduction to process modelling basics. The ability to interpret and interact with process models is at the core of all process initiative. Business Process Change then moves on to the more radical approach of process redesign or new process creation. Lastly, Harmon addresses the large area of business process automation through workflow systems, pre-packaged applications and custom software development.

All these methodologies address the same notion of enhancing the work in organizations by means of Business Processes. However, the level of change, the starting point, the frequency of change, the time and scope differ.

Apart from these methodologies, some researchers came with the hypothesis that they can rely on existing models to complete or improve their models. Chan et al. (Chan, Gaaloul, et Tata 2012) introduced the concept of activity neighbour context. They recommend to the designer the activities that are close to the process under design from existing business processes. Close activities are detected by calculating the "sem" metric, they capture the similar activities and assist the designer in order to improve the process. Also Holschke et al. worked on lining services of business activities (O. Holschke et al. 2008). In fact they analyzed a scenario for business process improvement by applying reference models in a Service Oriented Architecture (SOA) environment. Additionally, they proposed a method for evaluating the implementation effort based on a cost model. They concluded that the use of reference models in business process improvement will contribute to better models. Furthermore, Assenova et al. (Assenova et Johannesson 1996) aim to improve the schema quality by suggesting an approach based on a systematic use of schema transformations. They introduced a set of quality criteria for conceptual schemas (homogeneity, explicitness, size, rule simplicity, rule uniformity, query simplicity, and stability) and they showed how the transformations affect these criteria.

As the models are related to the real world events and demands, there is a big risk of requirements redundancies and of increasing the complexity of the model. Weber et al. developed process model refactoring (Weber et al. 2011). They introduced a set of behaviour-preserving techniques for refactoring large process repositories which help the designers to simplify their models and to reduce redundancy.

Detecting deviations of a process model from the observed behaviour has been researched, among others, by W. van der et al. (W. van der Aalst, Adriansyah, et van Dongen 2012). Given a process model and an event log, deviations are expressed in the form of skipped activities (activities that

should be performed according to the model, but do not occur in the log) and inserted activities (activities that are supposed to happen according to the model, but that occur in the log). A cost is attributed to these operations based on the particular activity being skipped/ inserted. Based on this information, an alignment can be computed between the process model and the log, which indicated how well the process model can describe the recorded behaviour. While this approach provides an effective measure for the replay fitness quality dimension, the approach does not suggest any corrections to rectify the process model's behaviour.

Buijs and al. (Buijs et al. 2013) proposed a novel process mining algorithm that improves a given reference process model using observed behaviour, as extracted from the event logs of an information system. The proposed algorithm is able to improve the model with respect to the four basic quality aspects (fitness, precision, generalization, and simplicity).

While an extensive set of research contributions has been made on the model's concepts structure verification and improvement techniques, others focus on the style of activity labelling.

Recent research revealed the high impact of the labelling style in business process models on the semantic and pragmatic quality of the models: labelling styles in Business Process Models (Leopold, Smirnov, et Mendling 2011) and refactoring the activity labels (Leopold, Smirnov, et Mendling 2012) constitute an approach to improve the semantic of the concepts labels. They started their study by differentiating the activity labelling styles as "Verb phrase", "Noun phrase", "Phrase with coordinating conjunctions"... etc. To recognize each of the labelling style, algorithms are proposed. The goal of these algorithms is to evaluate them and to aid the modellers in labelling their models. Thus, they completed their research by designing a technique for the automatic refactoring of labels with quality issues. More specifically, they developed an algorithm that is able to correct the activity labels.

The refactoring algorithm works in four phases:

1. Recognition of the activity labelling style
2. Analysis of action-noun labels
- 3 Derivation of an action and a business object from activity label
4. Composition of a verb-object activity label.

Finally, to evaluate their approach, they introduced two quality metrics: refactoring gain which represents the share of labels which has been correctly refactored and refactoring effect takes into account the erroneously refactored labels. The results of the refactoring shows the growth of the gain curves i.e. the increasing number of gain. They also shows the maintenance of the effect curves i.e. no effects were noticed.

Business process models can also face semantic deficiencies such as ambiguity issues caused by synonyms or different abstraction levels for process element names. Ehrig et al. (Ehrig, Koschmider, et Oberweis 2007) propose, in order to solve such problems, the use of ontology-based description of process models. They proposed an approach, based on ontology concepts, to measure the similarity

between business process models, by calculating the syntactic, linguistic and structural aspects similarities. Also Galatescu (Alexandra Galatescu 2004) used three ontologies in a software aiming to assist the improvement and analyze business process models in a virtual enterprise. The ontologies provide the team-based work in BPI with a common vocabulary for the team: automatic integration of verbal diagrams and structures with ideas, communication with structures and an extensible user interface.

It should be mentioned that ontologies can be used for several purposes, provide the team-based work in BPI with a common vocabulary (Alexandra Galatescu 2004), for product data management interoperability (Panetto, Dassisti, et Tursi 2012), etc.

McCusker et al. (McCusker, Luciano, et McGuinness 2013) show how using an ontology for conceptual modelling provides a common vocabulary which leads to unambiguous models. Shanks et al (Shanks, Tansley, et Weber 2003) show how ontologies can be used to facilitate the modelling task and how it can be used to evaluate the fit between an organization's needs and the business models. They came with the conclusion that ontology helps ensure that they select a conceptual modelling grammar needed to produce high-quality models of the focal domain. It can also guide the grammar used to generate clear, complete descriptions of the domain and that the can be used to help make sense of ambiguous semantics in conceptual models that need to be validated.

2.7.4 Business process improvement patterns

Riehle and Züllighoven (Riehle et Züllighoven 1995) define pattern as the concrete form which recurs is that of a solution to a recurring problem. While Alexander et al. (C. Alexander, Ishikawa, et Silverstein 1977a) define business process improvement pattern as an abstract form of a recurring instance of a process modification step used in a Business process improvement activity. Research shows that the best way to express creative thoughts is through visual imagery in art (Kosslyn 1988), ergo in graphical patterns that simplify complex circumstances. The idea of trying to identify, analyze, design and organize patterns by describing commonly recurring constructs has been popular in the software community (Gamma et al. 1994). Various kinds of patterns are currently in application.

We can cite from the literature two types of patterns: adaptation patterns and workflow patterns. Adaptation patterns allow structurally changing process models using high-level change operations. They can be applied along to the entire process lifecycle (Weber, Reichert, et Rinderle-Ma 2008). Workflow patterns were introduced for analyzing the expressiveness of process modelling languages. Patterns cover different perspectives like control flow (Van Der Aalst et al. 2003), data (Russell, Hofstede, et al. 2005), resources (Russell, Van der aalst, et al. 2005).

Ayora and al. (Ayora et al. 2013) proposed nine patterns for dealing with changes in process families. They complement existing work on patterns for creating and modifying BP models by introducing a set of generic and language independent patterns that cover the specific needs of process families. They first present the research methodology they employed to identify these patterns. Then they

identified the set of variability specific language constructs frequently used by existing proposals to capture the variability within a process family.

2.7.5 Comparison of quality approaches

(Table 8) summarizes the information about the different approaches presented in the previous sections. The comparison is done based on seven criteria. 1) What models the approach can be applied on? 2) Does the approach define quality criteria? 3) Does it define metrics in order to evaluate each quality criterion? 4) Does the approach provide guidelines or rules in order to improve the models quality? 5) What quality does it evaluate? 6) What notation can be applied? 7) Does the research accomplish the work by implementing a tool?

Approach	Model category	Quality criteria	Metrics definition	Improvement Procedure	Quality type	Notation	Tool
(Jan Mendling, Reijers, et van der Aalst 2008)	BPM	yes	no	Yes	Syntactic and pragmatic	EPC	no
(Becker, Rosemann, et Uthmann 2000a)	BPM	yes		yes	Syntactic and semantic	EPC and Petri nets	no
(Heravizadeh, Mendling, et Rosemann)	BPM	yes	yes	no	Semantic and pragmatic	EPC	no
(Heinrich et Paech 2010)	BPM	yes	yes	no	Semantic and pragmatic	-	no
(H.A. Reijers et Liman Mansar 2005)	BPM	yes	no	no	Pragmatic	-	no
(Eick 1991)	Conceptual models	yes	yes	yes	Syntactic, semantic and pragmatic	UML	yes
(Jan Mendling et Strembeck 2008)	BPM	yes	yes	no	Pragmatic	EPC	no
(Dijkman, Dumas, et Ouyang 2008)	BPM	no	yes	no	Syntactic, semantic	BPMN	yes
(Vanhatalo, Völzer, et Leymann 2013)	BPM	no	yes	no	Syntactic	Petri net	yes
(Wolf 2007)	BPM	no	yes	yes	Syntactic	Petri net	yes
(Verbeek, Basten, et Aalst 1999)	BPM	no	yes	no	Syntactic	Petri net	yes
(Dongen et Mendling)	BPM	no	yes	yes	Syntactic	Petri net	no

2006)							
(Hallerbach, Bauer, et Reichert 2009)	BPM	no	yes	no	Syntactic and semantic	Petri net	yes
(Leopold, Smirnov, et Mendling 2011)	BPM	no	yo	yes	Semantic	BPMN	no
(Leopold, Smirnov, et Mendling 2012)	BPM	no	yes	yes	Semantic	BPMN	no
(Weber et al. 2011)	BPM	no	yes	yes	Syntactic	EPC, BPMN, Petri net	no
(Chan, Gaaloul, et Tata 2012)	BPM	no	yes	yes	Semantic	BPMN	no
(Oliver Holschke et Schröpfer 2013)	BPM	no	yes	yes	Semantic, pragmatic	BPMN	no
(Cherfi et al, 2008)	Conceptual models	yes	yes	yes	Syntactic and Pragmatic	-	no
Cherfi and al. (Cherfi,2006)	Use case diagrams	no	yes	yes	Syntactic	UML	no
(Ehrig, Koschmider, et Oberweis 2007)	BPM	yes	yes	no	Semantic, Pragmatic	BPMN, EPC	no
(Alexandra Galatescu 2004)	BPM	yes	no	yes	Semantic, Pragmatic	-	yes
(McGuinness 2011)	Conceptual models	yes	no	no	Pragmatic	Use case	yes
(Shanks, Tansley, et Weber 2003)	Conceptual models	yes	no	yes	Semantic, Pragmatic	-	no
(Shanks 2007)	Conceptual models	yes	yes	no	Semantic, pragmatic	-	no
(D. Batra et Davis 2013)	Conceptual models	yes	yes	no	Syntactic, Semantic, pragmatic	-	no
(Assenova et Johannesson 1996)	Conceptual models	yes	no	yes	Pragmatic	-	no

Table 8: Quality approach comparison

By comparing these approaches, we can conclude that the majority of business process quality approaches concentrate on either quality evaluation using quality criteria and metrics or on quality improvement. Moreover each of these approaches is applicable on a specific notation.

Anyway, in order to have a complete quality improvement approach, it should:

- Be adaptable to different notations (BPMN, EPC, Petri-nets, etc.).
- Be able to evaluate models quality using metrics defined by defaults.
- Englobe a module to define quality metrics and apply them on the model.
- Define improvement procedures (textual recommendations, constraints, guidelines).

2.8 Conclusion

Most of the existing business process model quality studies are rich in syntactic quality metrics, but there is a lack in pragmatic and semantic metrics.

In the literature, studies that show the role of domain knowledge on the quality improvement does not provide any guidelines on how modellers can use this knowledge to improve business process models. This thesis proposes a step forward in this direction. We aim to explore the domain knowledge to improve the semantic quality of the business process models.

In the following chapters, we present our solution for the above mentioned problems. Our solution includes the following:

- i. We define the semantic quality for business process models.
- ii. We propose an approach based on domain knowledge i.e. application domain knowledge extracted from ontologies and IS domain knowledge represented by semantic constraints to evaluate business process model semantic quality.
- iii. We enrich this approach with proposing guides helping business process modellers to improve their BP model semantic quality.

Chapter 3

IS and Application domain Knowledge

The solution proposed within this thesis aims to provide a method and tool support to help in constructing syntactically, semantically and pragmatically correct process models. Indeed, despite the recent efforts dedicated to improve notations and tools, business process modelling remains difficult since it relies heavily on the modeller's knowledge about the modelled domain and the notation used.

3.1 Approach overview

As sketched in (Figure 11), BP modelling as any conceptual modelling activity consists in creating and/or evolving a model considering a given domain requirements, under specified constraints, relying on knowledge about both the domain and the notation used (method knowledge).

The information elicited during this process has to be interpreted, analyzed and modeled before the analysts can feel satisfactorily confident of the requirements completeness and correctness.

In prior studies, which refer to the importance of application domain knowledge in requirements analysis, it has mostly been argued that it positively impacts the effectiveness of IS (information systems) problem solving (Antón et Potts 1998), (Vessey et Conger 1993).

When discussing problem-solving in general, it is important to differentiate between well and ill structured problems. Based on (Khatri et Vessey 2010b), a well-structured models are those that have a well-defined initial state, a clearly defined goal, a well-defined constrained set of activities. On the other hand, ill-structured models are those which the initial and goal states are vaguely defined or unclear. Further, the problem statement does not contain all the information needed for their solution.

The cognitive psychology literature presents evidence that prior domain knowledge may constrain the analyst's search (Wiley 1998).According to Chiesi and al. (Chiesi, Spilich, et Voss 1979) knowledge in a given domain facilitates the acquisition of new domain requirements. Moreover, different studies confirm these statements, as (Burton-Jones et Weber 1999) (Khatri et al. 2006), they indicate that domain knowledge can compensate for a lack of clarity and missing information in the representation of a problem.

Buchman and Edkadharmawan (Buchman et Ekdaharmawan 2009) report a number of case studies as a result of which barriers to shared domain understanding among stakeholders and development teams were identified and graded.

Two of the most important barriers identified are: i) inter-group diversity relates to different levels of knowledge, both business (domain) and technical and ii) lack of common vocabulary that may lead to delays and to confusion and misunderstandings. Similarly, McAllister (McAllister 2006), who reports

a field study intended to identify the factors that lead to misunderstandings regarding requirements, as perceived by users and by developers. The dominant factors that both users and developers indicate include developer's lack of understanding difficulties related to common language and terminology.

Domain knowledge results from an accumulation of knowledge about a problem domain understanding. Many experiences in academic and non academic topics showed that prior knowledge in a domain has consistently been shown to have positive effects on activity performance (Chiesi, Spilich, et Voss 1979). In conceptual modelling several authors pointed out the impact of lack of domain knowledge on the quality of produced models. Indeed, experiences conducted in (Agarwal, Sinha, et Tanniru 1996) (Shanks 2007) (Dinesh Batra et Davis 1992) (Abd Ghani et al. 2008) suggest that user differences in cognitive abilities, application domain knowledge and method knowledge affect the way that conceptual modelling and information acquisition is conducted. Moreover, authors in (Khatri et al. 2006) (Khatri et Vessey 2010a) concluded through a study that in Information Systems discipline, domain knowledge comprises two parts: 1) IS domain knowledge embedded in methods, notations, and tools and 2) application domain knowledge referring to real-world problems understanding.

In the context of this thesis, we propose a solution that captures the three kinds of knowledge impacting the quality of models namely: (i) IS domain knowledge, (ii) Application domain knowledge and (iii) knowledge related to accumulation of experiences in modelling.

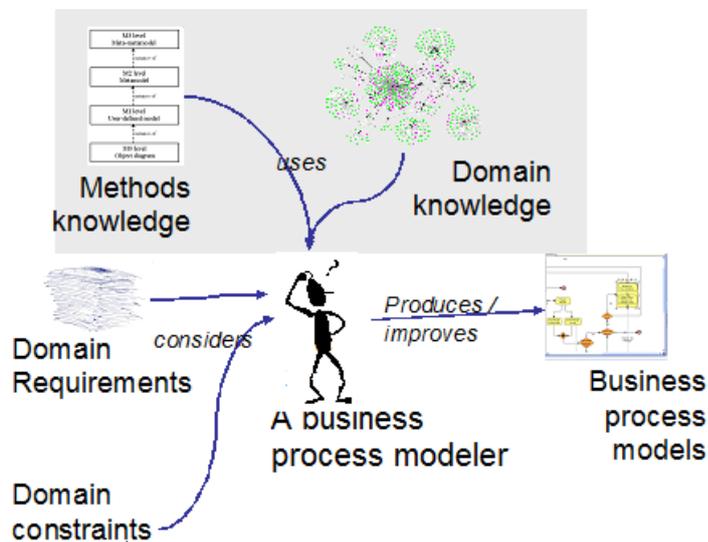


Figure 11: Business process modelling context

(i) The IS domain knowledge represents the knowledge related to BP modelling notations, methods, and practices. Even if these guides exist, modellers still produce models that do not conform to these guides and advices. Our approach proposes to exploit this knowledge capitalized from notations handbooks, research literature and several sources providing such guides. The proposed solution takes into account the main barriers to this kind of knowledge exploitation. The main barriers we have identified are:

a. Lack of centralization: there exists in literature, as well as in notations' handbooks, dozens of rules and advices on the good use of concepts and constructs. However, this knowledge is generally spread out all over tool documentation, websites, research papers and a variety of other sources making its exploitation difficult. Our proposed solution gathers this knowledge and proposes a tool automating its exploitation to support both detection and correction of defects related to notation's conformance.

b. Lack of formality: our analysis of the IS domain knowledge revealed heterogeneity of formalization. While some guides are expressed using formal languages, others are textual and vague and some are just related to modelling expertise and are never explicitly expressed. We propose to use the Object Constraint Language (OCL) to express the rules and advices defining the "right" or the "better" usage of concepts and constructs. These constraints, formally expressed, are thus easy to correct, enrich and use within an automated support.

c. Heterogeneity of notations: there exists a variety of notations for BP modelling each having its own vocabulary and thus specific rules. To ensure the genericity of our approach, we build our reasoning on a BP modelling meta-model. The IS knowledge OCL constraints that are common and valid in all notations are generalized and thus written at the meta-model level making them independent from specific notations. There remain however some specific concepts that holds only for some notations. Our approach allows also the definition of constraints at notation level to handle specific situations. Finally, as models are expressed in different notations we have developed a set of corresponding rules between the meta-model's concepts and those of frequently used notations. An annotation tool uses these corresponding rules to annotate the models making them readable at both notation and meta-model levels.

(ii) The application domain can be extracted from several sources such as user's requirements statements, domain expertise or existing models related to the same problem or to similar problems such as data models, resource models etc.. Within this thesis we do not address the problem of transforming requirements into models as this problem has been extensively covered by literature. Our objective is to use domain knowledge to help in improving a process model under construction. For example, domains such as healthcare, tourism etc. are facing the challenge of delivering reliable and high quality services at affordable costs. To tackle this challenge they develop decision support systems based on clinical pathways for the healthcare domain or services ontologies for the tourism domain. This knowledge domain is either used manually by actors as in the healthcare domain where clinical pathways are generally textual structured definitions of practices. It could also be exploited through an automated tool as in the case of web services composition (Arpınar et al. 2005). We are more interested in an automated usage of domain knowledge. Our solution relies on the existence of

domain ontologies describing concepts and relationships related to the application domain. Such ontology could be the result of a collaborative work in a given domain such as the e-tourism ontology for data exchange in tourism business domain. They also could be constructed by integration of ontology fragments or a mapping process. The construction of such ontologies is out of the scope of this thesis and contributions on this problem can be found in the literature for several applications domains (Danial, Abidi, et Abidi 2009) (Kouneli et al. 2012) (Clark et al. 2012).

We present in this chapter how both IS domain and application domain knowledge are modelled, formalized in our solution. The next chapter is devoted to the process exploiting this knowledge to integrate quality objective into the modelling process.

3.2 Capturing IS domain knowledge

The term knowledge refers to the part of the world investigated by a specific discipline and that includes a specific taxonomy, vocabulary, concepts, theories, research methods, and standards of justification (P. A. Alexander 1992). The first family of approaches concerns those focusing on methodological guides to ensure quality of models. In (Becker, Rosemann, et Uthmann 2000b) where the authors propose a set of guides to improve various characteristics of a process model such as clarity, comprehensibility, or accuracy (correctness). Other authors focus on improving the comprehensibility of models by providing naming rules, documentation, and use of icons or symbols graphs (Jan Mendling, Recker, et Reijers 2010). Other approaches, such as (Van Der Aalst et al. 2003) propose a set of best practices encapsulated in reusable and applicable patterns depending on the context. A second category of approaches concentrate on the formalization of rules related to methods and notations. In (Kent, Gaito, et Ross 1999) OCL rules are defined upon a UML meta-model to ensure the validity of structures. Other techniques, based on transformations, are also used to add semantics to meta-models (Chen et al. 2005). The last category explores the idea of combining ontologies, modelling and meta-modelling techniques to support conceptual modelling activity. In (Kaiya et Saeki 2006) the authors use a domain ontology to help in requirements elicitations. The same authors present in (Kaiya et Saeki 2005) a general framework to conduct a modelling activity by the help of meta-models and ontologies.

Our approach fits both in the second and third categories as it uses domain knowledge to improve quality of business process models and OCL rules to express notation's semantics.

Business process modelling is mainly used to document, analyze, and optimize workflows. Hence it contains a lot of domain knowledge. In this chapter, we explain how we exploited the IS domain knowledge from the business process modelling notation and the application domain knowledge. Thus we can apply on them to evaluate and improve the quality of BP models.

There are several elaborate graphical notations that can be used for business process modelling, ranging from the most popular BPMN (Omg 2004), EPC (Kühne et al. 2010), to less known Eriksson-Penker's notation (Penker et Eriksson 2000). Each notation has its own concepts and its own semantics. A severe problem is that several languages for business process modelling do not have formal execution semantics which is a pre-request to check correctness criteria. We aim to define constraints on a meta-model that converges all modelling notations.

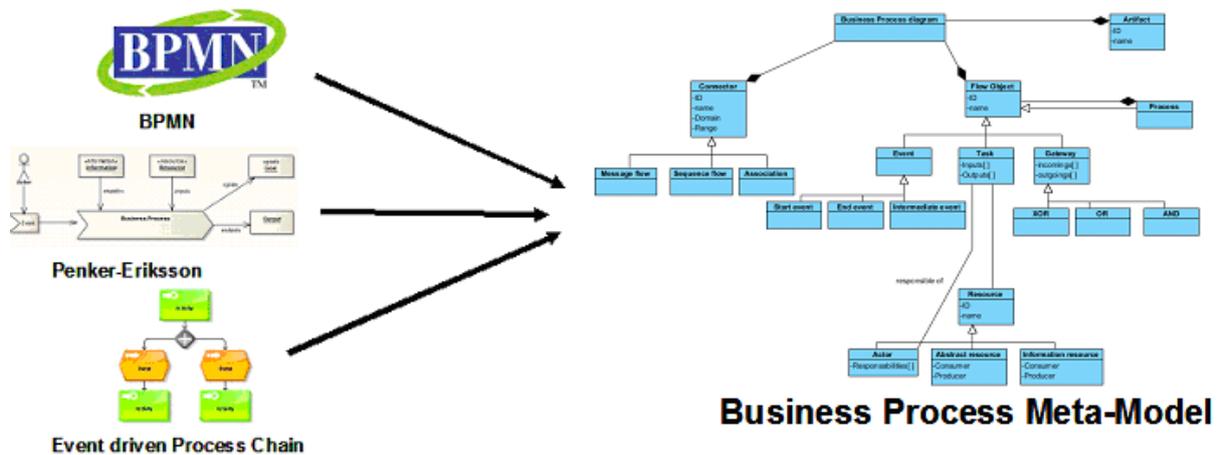


Figure 12: Independence of a given BP modelling notation

We aim to exploit this knowledge independently from any notation. Therefore we defined a meta-model for BP modelling that integrates these different notations by implementing an interoperability module that maps each notation concept to its corresponding one in our meta-model.

3.2.1 Business process meta-model

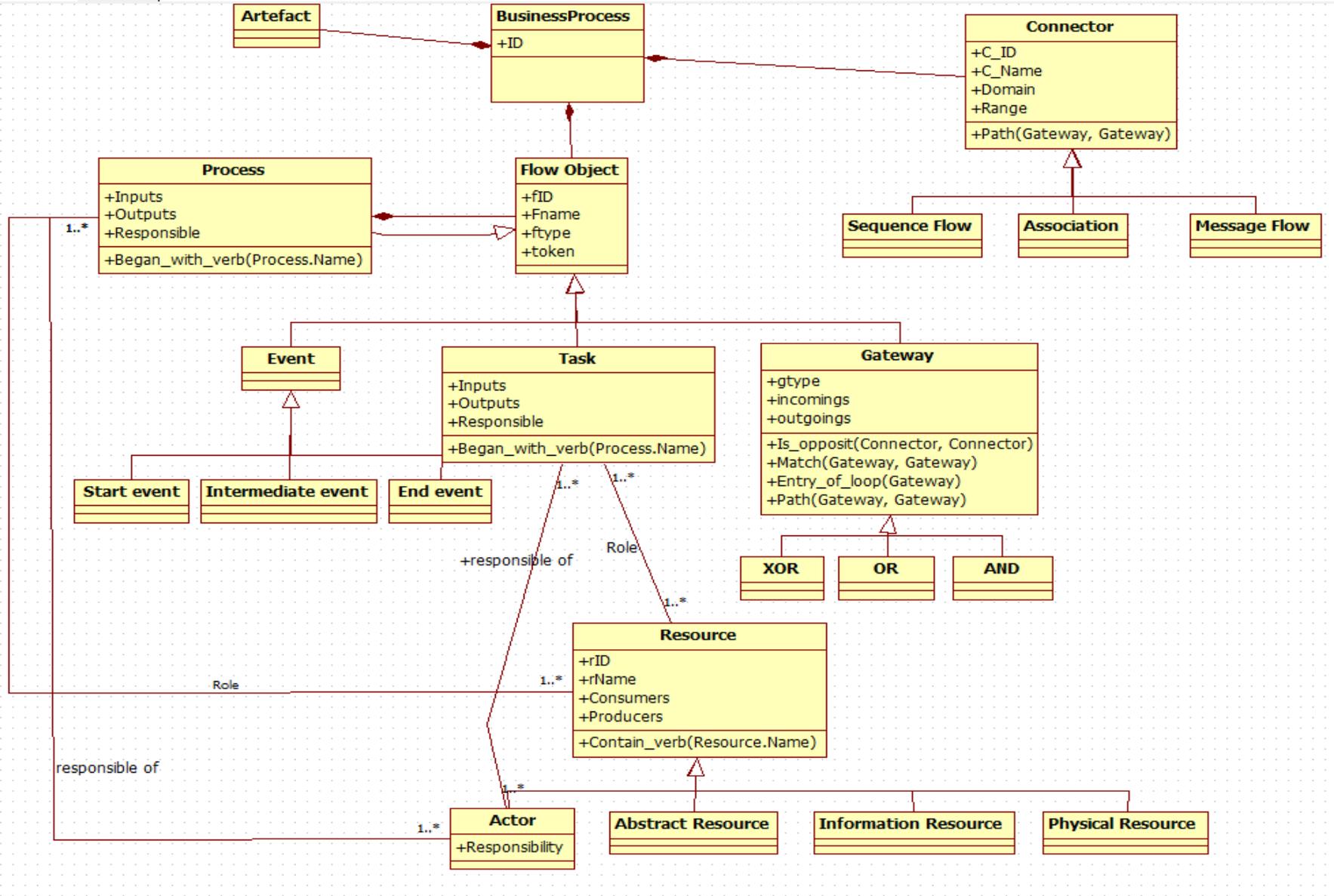
The meta-model defined in this section was constructed as a synthesis of a selection of concepts proposed by several authors, according to several notations and more specifically the work presented in (Aguilar-Savén 2004a).

The meta-model provides a synthetic vision of concepts used independently of specific notations helping in the understandability of models.

According to prominent BPM researcher van der Aalst (Wil M. P. van der Aalst, Hofstede, et Weske 2003b), BPM is defined as “a field of knowledge at the intersection between management and information technology, encompassing methods, techniques, and tools, to design, enact, control and analyse operational business processes involving humans, organizations, applications, documents, and other sources of information”. We are concerned with the design of operational processes.

We defined a BP modeling meta-model not to capture the key languages concepts as it is never used for modeling. We do not need completeness of the meta-model. It is considered as a pivot model used in order to annotate the different notations concepts so we can apply OCL constraints defined.

Figure 13: Business Process model Meta-Model



A process model is a tuple $BPM = \{A, E, G, C, IR, AR, PR\}$ where A represents the set of activities, E the set of events, G the set of gateways, C the set of connectors, IR, AR and PR represent respectively the information resources, abstract resources, physical resources and actors. Each of these sets are detailed in the sections below.

3.2.1.1 Flow objects

The meta-model expresses a business process model as composed of flows of objects and connectors. A flow object can be an event, an activity or a gateway. It is characterized by an identity (ID), a name and a token. A token is a boolean attribute that indicate if the flow object is executable or not.

- An event that occurs and impacts the progress of a process. The events could be of three types: start event that initiates the business process, end event that terminates the business process and intermediate events. Intermediate events also have subtypes that differ with each notation.

Let E be the set of Events, $E = \langle e_1..e_n \rangle$ where $e_i = \langle id, name, type, token \rangle$.

- An Activity can be a atomic and non decomposable presented by a task or a process if it is complex and has a visible structure. Each activity has an actor responsible of its execution.

Given a set A of activities, $A = \langle a_1..a_n \rangle$ where $a_i = \langle id, name, type, token, inputs[], outputs[], responsible[] \rangle$.

- A gateway is a mechanism able to manage the convergence and divergence of activities flows.
 - A XOR-gateway is used to model an "either/or" situation. Depending on conditions, exactly one outgoing flow will be chosen, the control flow is split. It can also be used as a 'Join' to show that no matter which exclusive path was taken from an XOR Decision, processing will continue with the single output from the 'Join' XOR Gateway.
 - OR-gateway is used when one or more of the outgoing Sequence Flows from the Decision may be taken. It can also be used as a 'Join' to show that no matter which paths were taken from an OR Decision, processing will continue with the single output from the 'merge' OR Gateway.
 - An And-gateway is used when Sequence Flows can run in parallel. All outgoing Sequence Flows are followed. It can also be used as a 'Join' when two or more parallel sequence flows need to join back into a single output flow.

Let G be the set of gateways, $G = \langle g_1..g_n \rangle$ where each gateway is described by its identity (ID), name, type, token, the number of incomings $\langle i_1..i_n \rangle$ and the number of outgoings

$\langle o_1 \dots o_n \rangle$. As a result $g_i = \langle id, name, type, token, incomings \langle i_1, \dots, i_n \rangle, outgoings \langle o_1 \dots o_n \rangle \rangle$.

3.2.1.2 Connectors

A connecting element can be an association, a sequence, or a message flow. A connector is a connection path from a concept to an other and described by an identity (ID), a name and a type, a domain and a range. Thus a domain represents the input extreme of a connector and the range represents the output extreme of a connector.

Let C be the set of connectors, $C = \langle c_1 \dots c_n \rangle$

- An association is a simple link between two concepts. It has a domain and a range.

Let As be the set of associations, $As = \langle as_1 \dots as_n \rangle$ where $as_i = \langle id, name, domain, range \rangle$.

- The Sequence Flow is used to show the order of activities that will be performed in a Process. It is used between flow objects. It is always direct from a domain to a range. The domain is the predecessor and the range is the successor.

Let Sf be the set of sequence flow, $Sf = \langle sf_1 \dots sf_n \rangle$ where $sf_i = \langle id, name, domain, range \rangle$.

- A message flow is used to represent exchange of information between two participants in the process. It is always directed from a sender (the domain) to a receiver (the range).

Let Mf be the set of message flow, $Mf = \langle mf_1 \dots mf_n \rangle$ where $mf_i = \langle id, name, domain, range \rangle$.

To conclude with the connectors, the sets Mf, Sf and As are included into the set C.

3.2.1.3 Resources

Resources are the objects that act or are used in the business. They are the concepts consumed, produced, transformed or used by the business process.

Activities refer to resources. A resource encompasses abstract concepts such as the human agent responsible for execution of the activity and also information produced or consumed by it.

- Information Resources represent information or knowledge as a bank account number, an address ... etc. An information resource is produced or consumed by an activity. So each one has a set of activity consumers $\langle co_1 \dots co_n \rangle$ and/or a set of activity producers $\langle po_1 \dots po_n \rangle$.

Let IR be the set of information resources, $IR = \langle ir_1 \dots ir_n \rangle$ with $ir_i = \langle id, name, consumers \langle co_1 \dots co_n \rangle, producers \langle po_1 \dots po_n \rangle \rangle$.

We have to note that the set of consumers can be an empty set, similarly to the set of producers but not both.

- Abstract Resource represents an idea or a concept. Contracts, accounts are examples of abstract resources. An abstract resource is produced or consumed by an activity. So each one has a set of activity consumers and/or a set of activity producers.

Let AR be the set of abstract resources, $AR = \langle ar_1..ar_n \rangle$ with $ar_i = \langle id, name, consumers \langle co_1..co_n \rangle, producers \langle po_1..po_n \rangle \rangle$.

Similarly, the set of consumers can be an empty set, similarly to the set of producers but not both.

- Physical Resource represents physical goods such as production processes. They are not administrative, management or service process, but a physical resource required in the process. Each one has a set of activity consumers and/or a set of activity producers.

Let PR be the set of physical resources, $PR = \langle pri, prn \rangle$ with $pri = \langle id, name, consumers \langle co_1..co_n \rangle, producers \langle po_1..po_n \rangle \rangle$.

- Actors represent the community responsible of the execution of certain activity in the process. Community may represent a company, an association, a human being, etc.

Let PR be the set of actors, $PR = \langle pr_1..pr_n \rangle$ with $pr_i = \langle id, name, responsibility \rangle$

The relation "responsible-of" that relates the activity to an actor, represents the responsible allocation to each activity.

Each activity (task or process) requires resources. They have inputs which are a set of resources consumed. And outputs, a set of resources produced.

The relation "require" that relates the activity to resources represents the incomings and outgoings of resources.

3.2.2 Business process model - Business process metamodel alignments

Each concept in a notation has a *name* and a *role* or, in other words, a *semantic*. Mappings from each notation concepts to the meta-model concepts are defined based on two functions types:

- Concepts semantics: maps two concepts of different notations that have the same roles.
- Name matching: maps two concepts from different notations that have the same names.

Relying on (Zha et al. 2008) and (Tscheschner 2010) we completed their mappings to encompass more notations.

Rule 1: Process

Notation	Concept
----------	---------

	Name	Semantic
BPMN	Process	An activity is a generic term for work that a company performs.
EPC	Function	A function is an activity which supports the completion of a business objective.
Eriksson-Penker	Process	A process is the execution of an action.

These different concepts are represented by a process in our meta-model, which represents the execution of an act in order to support the completion of a business objective.

As a result, the BPMN process, EPC function and Eriksson-Penker process are related to our meta-model Process.

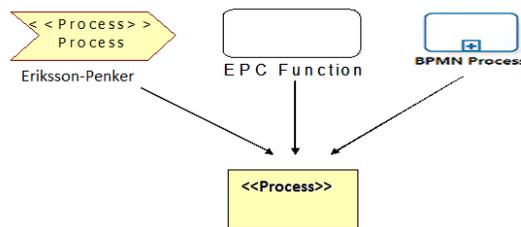


Figure 14: Mappings between different concepts having the same role: Process

Rule 2: Actor

Notation	Concept	
	Name	Semantic
BPMN	Pool/ Lane/ Performer	Used to organise and categorise activities according to a role.
EPC	Organization Unit	Represents a specific role within the process
Eriksson-Penker	People Resource	Is an actor acting in the process

All these concepts are represented within an Actor in our meta-model. An actor represents different type of participants that may be responsible of an execution of a Process.

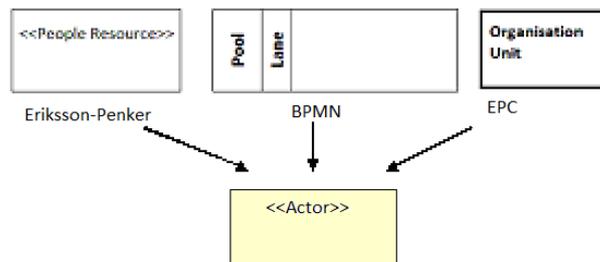


Figure 15: Mappings between concepts having the same role: Actor

Rule 3: Event

Notation	Concept	
	Name	Semantic
BPMN	Event	It initiates, terminates and presents something that happens during the execution of a business process
EPC	Event	Events define an occurred business situation, they also can trigger functions.
Eriksson-Penker	Event	Events initiate, terminate and denote something that happens.

By the name matching function, they are related to the event concept in our meta-model which represents a fact that occurs and changes, starts or even ends the process flow.

Rule 4: Gateway

There are different types of gateways: Split XOR, Split OR, Split AND, Join XOR, Join AND and Join OR.

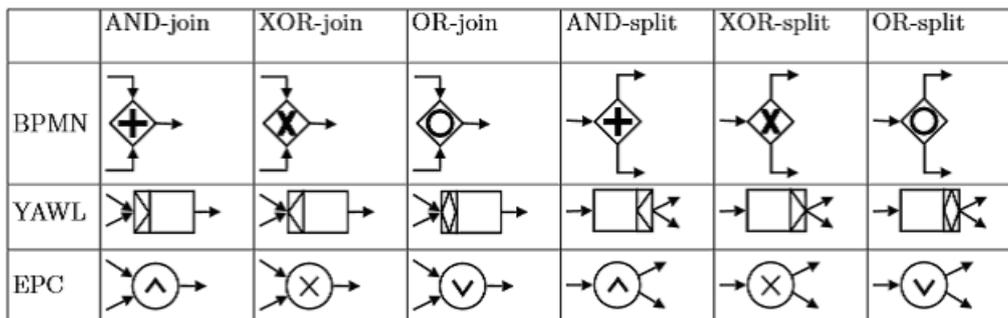


Figure 16: Connectors in different modeling languages (Gruhn et Laue 2010)

Notation	Concept	
	Name	Semantic
BPMN	Gateway	They are defined for controlling the divergence and convergence of the sequence flow.
EPC	Connector/ operator	They are defined as concatenation points in the process for events and functions.
Eriksson-Penker	Process decision	It represents the synchronization of the processes (junction, choice).

By a concept semantics function, it is related to the gateway in our meta-model which controls the convergence and the divergence of the flow.

Rule 5: Resource

Notation	Concept	
	Name	Semantic
BPMN	Artefact	Artefacts allow developers to bring some more information into the diagram. (data objects, group, annotations)
EPC	Resource	They are consumed/produced by functions. A resource can be an application system or a document.
Eriksson-Penker	Resource	Resources are the objects that act or are used in the business. They are concepts consumed, produced or used by the process.

These different concepts are presented by the concept Resource in our meta-model. Resources represent documents, information, abstract concepts that may be consumed, produced, transformed or even used by business processes.

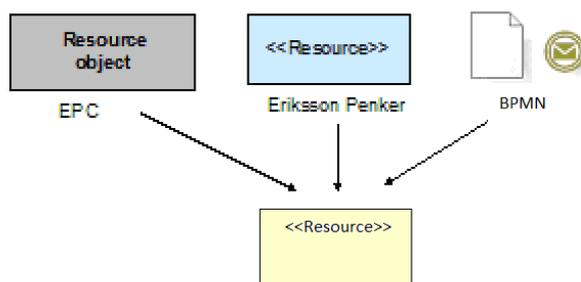


Figure 17: Resources in different notations

Rule 6: Sequence Flow

Notation	Concept	
	Name	Semantic
BPMN	Sequence flow	A sequence flow shows the order in which activities are performed in a process.
EPC	Control flow	A control flow creating a chronological sequence and logical interdependencies between events, functions and logical connectors.
Eriksson-Penker	Process flow	Shows which activity will follow another.

Accordingly, all these flows represent the order of execution of activities or processes. We present them by a sequence flow in our meta-model.

Rule 7: Message Flow

Notation	Concept	
	Name	Semantic
BPMN	Message flow	It tells which messages flow across processes.
EPC	Resource relation	Each resource has its own relation to a function.
Eriksson-Penker	Resource flow	It represents the consumption, production of resources (consume, product, require relations).

These flows are represented by a message flow in our meta-model representing the exchange of resources (information or abstract resources).

3.2.3 BPM notations' semantics formalization

The objective of defining this BP modeling meta-model is not centered on describing all the notations concepts that exists, but to define constraints for the notations standards. Each instance of the meta model will be restricted by the use of the constraints defined in the Object Constraint language (OCL). In this section we aim to demonstrate how IS domain knowledge can be used to improve the syntactic, semantic and pragmatic quality of business process models, and how it assists the designer in completing and validating the syntactic correctness of the model.

The main contribution is to define and homogenize the know-how of BP modellers through OCL constraints. Thanks to the object paradigm, we can define abstract constraints applicable to all BP model notations and concrete constraints dedicated to specific notations.

OCL constraints are defined on the meta-model level. They are used to detect and correct quality deficiencies. In fact, each constraint will check the validation of some conditions that the model should respect. Thus, checking the validation of these conditions will lead to the detection of quality deficiencies.

With respect to the categorization of quality in (Lindland, Sindre, et Solvberg 1994a), we separated the constraints into these three categories: syntactic, notations semantics and pragmatic constraints.

The notation errors represent the deviation from the syntax of the language rules. As a consequence, these errors affect the syntactic quality of the model. The syntax is mainly described by the meta-model presented in Section 3.2.1.

There are different aspects of a process model. Axenath et al. (Axenath, Kindler, et Rubin 2005) define three basic aspects of business process models:

- The behavioural aspect: defines the order in which the tasks of a business process are instantiated and in which the corresponding activities are executed. There are many different ways, formalisms, and notations for defining the order in which tasks, respectively the corresponding activities must be executed i.e. in a given state, it must be clear which tasks are enabled. This aspect can be detected by the process model workflow.
- The informational aspect: defines the information involved in a business process as well as the propagation of information among different activities. All information involved in a business process can be considered to be resources. Moreover, the information aspect defines how documents are propagated among activities.
- The organisational aspect of a business process model defines the structure of the organisation in which the business process is executed, its organisational units and the relations among them; and it defines the resources and actors within these organisational units.

Many verification approaches in the literature aim to detect as much defects as possible.

The approaches that focus on control flow are generally applied on EPC and Petri nets notations since they are dedicated to lay out business process workflows. Control flow defects are detected by the soundness, relaxed soundness, EPC soundness, decomposition and reduction criteria.

Soundness: Soundness is an important and prominent correctness criterion for business process models. It was first introduced in (Wil M. P. van der Aalst 1997). The original soundness property is defined for a Workflow net, a Petri net with one source and one sink, and requires that:

- i. For every state reachable from the source, there exists a firing sequence to the sink.
- ii. The state with a token in the sink is the only state reachable from the initial state with at least one token in it.
- iii. There are no dead transitions (Wil M. P. van der Aalst 1997).

It has been shown that soundness of a Workflow net is equivalent to liveness and boundedness of the corresponding short-circuited Petri net (Wil M. P. van der Aalst 1997). Therefore, several liveness and boundedness analysis techniques (Murata 1989) are directly applicable to the verification of soundness. Soundness identifies all deadlocks and lack of synchronization for process models with one start and one end node.

Relaxed Soundness: The soundness property of workflow models has stimulated the specification of several soundness derivatives, mainly because some soundness aspects proved to be too restrictive in certain application domains. In (Dehnert et Rittgen 2001) the authors define a process to be relaxed sound if every transition in a Petri net representation of the process model is included in at least one proper execution sequence (Dehnert et Rittgen 2001). The relaxed soundness property is used in a study on the verification of the SAP Reference Model (J. Mendling et al. 2006) (J. Mendling et al. 2007). Relaxed soundness does not necessary identify all deadlocks and lack of synchronization.

EPC Soundness: The property of EPC soundness proposed in (Jan Mendling et Aalst 2007), builds on the identification of a set of initial markings that covers all start events. The EPC soundness definition demands that:

- (i) There exists such a non-empty set of initial markings.
- (ii) That, for each initial marking in this set, proper completion is guaranteed.
- (iii) There must be a set of final markings reachable from some of these initial markings such that there exists at least one final marking in which a particular end arc holds a token.

If that is fulfilled, proper completion is guaranteed for a set of initial markings that cover all start arcs. The EPC soundness property is stricter than the relaxed soundness criterion since it requires a guarantee of proper completion. EPC soundness is used in another verification study of the SAP Reference Model (Jan Mendling, Neumann, et Van Der Aalst 2007b). The authors use a two-step approach using first reduction rules and then, if necessary, state space calculation. EPC soundness can identify all those deadlocks and lack of synchronization that occur for all start combinations in which a particular start event participates.

Decomposition: The approach reported in (Vanhatalo, Völzer, et Leymann 2013) builds on the decomposition of a workflow graph into single-entry-single-exit (SESE) components. The authors identify some heuristics for sound and for unsound components. In essence, these heuristics match EPC reduction rules as described in (J. Mendling et al. 2007). The decomposition approach identifies all those deadlocks and lack of synchronization that are defined in the heuristics.

Reduction: Another heuristic for checking the correctness of process models is used in (Gruhn et Laue 2007). The authors identify reduction rules for structured EPC models and heuristics to correct simple connector mismatch errors. This approach identifies all deadlocks and lack of synchronization that can be traced back to variants of unstructuredness.

These different approaches have in common that they detect the same type of defects: workflow defects in different ways. Thus workflow defects are related to logical relationships i.e. gateways in BPMN.

Our solution aim to gather the IS rules from several sources such as the above approaches and research papers on the most common errors in workflow as Gruhn et al. (Gruhn et Laue 2010) , (Jan Mendling 2009), Mendling and Aalst (Jan Mendling et Aalst 2007), Van der aalst et al. (W. M. P. van der Aalst et al. 2011), Fahland et al. (Fahland et al. 2009) and notations handbooks.

3.2.3.1 Syntactic constraints

1. Each activity has to reach the end event.

In every state that is reachable from a start event, there must be the possibility to reach an end event.

<i>context Process inv reach_end_event:</i>

```
Connector.allInstances() -> forAll(c| c.Domain.fID = self.fID implies Connector.allInstances()->
exists (c1| c1.Domain.fID= c.Range.fID and c1.Range.ftype = #end_event))
```

2. Each activity has to be reached from the start event

There should be a path from the start event to any activity in the model.

```
context Process inv reached_from_start_event:
Connector.allInstances() -> forAll(c| c.Range.fID = self.fID implies Connector.allInstances()->
exists (c1| c1.Range.fID= c.Domain.fID and c1.Domain.ftype = #start_event))
```

3. Each activity should have two relationships.

The activity should be a domain of a connector and a range of an other one.

```
context Process inv proces_sucessors_nb:
Connector.allInstances() -> forAll(c1, c2| c1.Domain.fID = self.fID implies c2.Domain.fID <>
self.fID)
```

4. Connectors should reach the extremes

When the modeller uses a connector, he/she has to check that the connector reaches the extreme concepts.

```
context Connector inv proces_sucessors_nb:
Connector.allInstances() -> forAll(c| c.Domain.fID <> " and c.Range.fID <> ")
```

5. An activity can not have two successors

It will create a confusion on which flow to trigger (Van Der Aalst 1998b).

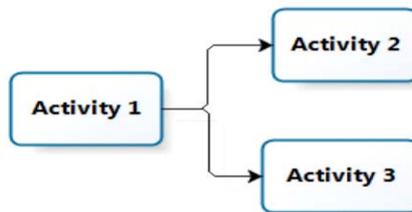


Figure 18: Activity with two outgoing edges

```
context Connector inv successors:
Connector.allInstances()->forAll(c1,c2| c1.Range.ftype =#process implies c2.Range.ftype <>
#process )
```

6. No connector should exist before the start event

There is no connector where range is a start event (Jan Mendling, Reijers, et van der Aalst 2008).

```
context Connector inv connector_range:
Connector.allInstances() -> forAll(c| c.Range.ftype <> #start_event)
```

7. No connector should exist after the end event

There is no a connector where domain is an end event (Jan Mendling, Reijers, et van der Aalst 2008).

```
context Connector inv connector_domain:
Connector.allInstances() -> forAll(c| c.Domain.ftype <> #end_event)
```

3.2.3.2 Notations semantics constraints

According to Esswein and al. (Esswein, Gehlert, et Seiffert 2004) , two instances of semantics of a model must be separated: first, semantics includes the rules for automated interpretation of a model, that is to say the concepts semantics and role in the model. Second, each modelling construct represents some real-world entities, which is another aspect of model semantics and is expressed by the labels of functions and events.

These errors have been found using the validation techniques mentioned above and some studies.

Other modelling languages like BPMN, Activity diagram, YAWL, Petri net or EPC use the same basic constructs for splits and joins. Thus these errors are common for all the business process modelling languages. We classified the workflow defects into five types: Deadlocks, Preventive deadlock, Lack of synchronization, Preventive lack of synchronization and Undesired situations.

- A deadlock results from a combination of logical relationships or gateways where one of these gateways is blocked waiting to be completed.
- Preventive deadlock results from situations where deadlock may or may not occur
- A process model suffers from a lack of synchronization if multiple branches are activated, but not synchronized later. In other words, we have a path with a handle.
- Preventive lack of synchronization results from situations where lack of synchronization may or may not occur.
- Undesired situations results from situations where there is paths not traversed or others traversed multiple times.

() summarize the different defects by explaining their causes and illustrating them.

Defect type	Defect Causes/Subtypes	
Deadlock	1. XOR-split with an AND-join: while the split only activates one control path, the AND is waiting for both to be completed	

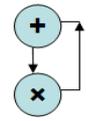
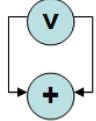
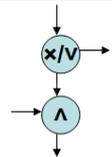
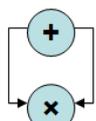
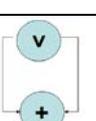
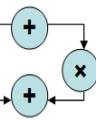
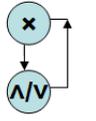
	2. An AND-Join is an entry into a loop: the AND-join will always receive both control paths, that is will not exit the loop.	
Preventive Deadlock	3. OR-split with an AND-join: the split may activate only one control path, the AND is waiting for both to be completed.	
	4. A (X) OR-split can lead the flow of control away from an AND-join which for this reason fails to synchronize its incoming flows.	
Lack of synchronization	5. AND-split with a XOR-join: multiple branches are activated due to the AND-split, but not synchronized later at XOR-join.	
Preventive Lack of synchronization	6. OR-split with an AND-join: the split may activate two control path, the AND is waiting for both to be completed.	
Undesired situations	7. An AND-split and an AND-join with one of the paths between them contain a XOR-join: The flow of control can reach the AND-join without having to pass the AND-split before.	
	8. An AND- or OR-join is an exit from a loop: which permits multiple terminations of the process. It is undesired.	

Table 9: Workflow defects

For specifying the constraints, we need some definitions. In order to define the situation that a join connector joins two (or more) control flow paths that have been started by a split before, we use two functions:

- Path-relation: $path(s,j)$ holds iff there is a path from s to j .
- Match-relation: $match(s,j)$ holds iff s is a split, j is a join and there are two paths from s to j whose only common elements are s and j .

1.

context Gateway inv DI:

Gateway.allInstances()->forall(g1, g2| g1.gtype =#XOR_split and g2.gtype =#AND_join implies Match(g1,g2)= false)

Chapter 3: IS and Application domain Knowledge

For all the instances of gateways, if there exist an instance g1 and an instance g2 where g1 is of type XOR_Split and g2 of type AND_Join, then a path from g1 to g2 will create a deadlock.

2.

context Gateway inv D2:

Gateway.allInstances()->forall(g1| g1.gtype =#AND_join implies Entry_of_a_loop(g1)= false)

Entry_of_a_loop is a boolean function that detects if the gateway is an entry of a loop. Hence for all the instances of gateways, an AND_Join gateway can not be the entry of a loop since it will create a deadlock.

3.

context Gateway inv PD3:

Gateway.allInstances()->forall(g1, g2| g1.gtype =#OR_split and g2.gtype =#AND_join implies Match (g1,g2)= false)

For all the instances of gateways, if there exist an instance g1 and an instance g2 where g1 is of type OR_Split and g2 of type AND_Join, then a path from g1 to g2 may create a deadlock.

4. In order to prevent such errors , the following conditions must hold:

- There is a path from the (X)OR-split s to an AND-join j
- There is an arc (s, s*) with s* = j and another arc (j*,j) with j* = s.
- There is no path from s to j* and no path from s* to j.

context Connector inv PD4:

Connector.allInstances() -> forall(c1,c2| Gateway.allInstances() -> forall(s, s1, j, j1| s.gtype =#XOR_split and j.gtype =#AND_join and c1.Domain.fID = s.fID and c1.Range.fID =s1.fID and s1.fID=j.fID and c2.Domain.fID =j1.fID and j1.fID =s.fID and c2.Range.fID=j.fID implies Path (s,j) and Path(s,j1) and Path(s1,j)=false))

5.

context Gateway inv L5:

Gateway.allInstances()->forall(g1, g2| g1.gtype =#AND_split and g2.gtype =#XOR_join implies Match (g1,g2)= false)

For all the instances of gateways, if there exist an instance g1 and an instance g2 where g1 is of type AND_Split and g2 of type XOR_Join, then a lack of synchronization is created.

6.

context Gateway inv PL6:

Gateway.allInstances()->forall(g1, g2| g1.gtype =#OR_split and g2.gtype =#XOR_join implies Match (g1,g2)= false)

For all the instances of gateways, if there exist an instance g1 and an instance g2 where g1 is of type OR_Split and g2 of type XOR_Join, then a lack of synchronization is created.

7.

context Gateway inv UND7:

```
Gateway.allInstances()->forall(s,j,e| Connector.allInstances()-> forall(c1,c2,c3| s.gtype
=#AND_split and j.gtype =#AND_join and e.gtype=#XOR_split and c1.Domain.fID =s.fID and
c1.Range.fID =j.fID and c2.Domain.fID =s.fID and c2.Range=e and c3.Domain =e implies
c3.Range.fID <> j.fID ))
```

For all the instances of gateways, if there exist an instance s of type AND_Split, e of type XOR_Split and j of type AND_Join, there shouldn't exist paths connecting s to e and e to j.

8.

context Gateway inv UND8:

```
Gateway.allInstances()->forall(g1, g2| g1.gtype =#AND_join implies Entry_of_a_loop (g1)= false)
```

For all the instances of gateways, if there exist an instance g1 of type AND_Join, it shouldn't be the entry of a loop since it will trigger the loop unlimitedly.

Some units and relations are presented differently in notations .Thus it will create potential defects specific to a notation rather than the others. Some of these defects can be detected by modelling tools either by not permitting the modeller to use a specific concept or by showing the defect.

9. Each activity should have a name.

An activity without label is a useless concept.

context Process inv lack_Activity_label:

```
Process.allInstances()->forall(p| p.fname<> ")
```

10. A Sequence flow can not cross the process boundary if it contains a start event

If the process does not contain a start event, the sequence flow should be connected to the trigger task in the process. This defect may occur in BPMN Lane/ Pool or in activity diagrams Swimlane module. Both notations contain concepts represented with boundaries.

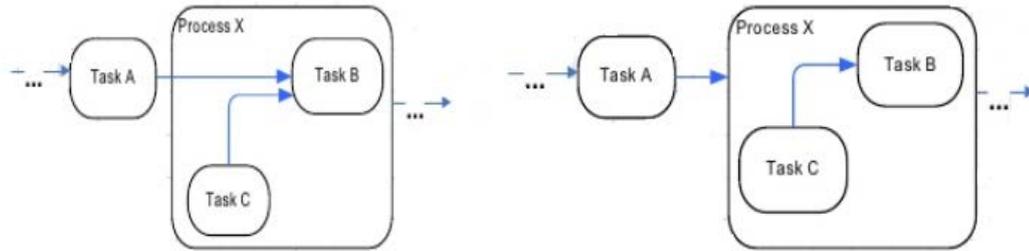


Figure 19: Sequence flow crosses the process boundary

11. Each X(OR) Split gateway should be labelled

Each Split XOR/OR gateway should have a name presenting the split event (Figure 20).

```
context Gateway inv lack_ORSplit_label:
Gateway.allInstances()->forAll(g| g.gtype= #OR_split implies g.fname<> ")
```

```
context Gateway inv lack_XORSplit_label:
Gateway.allInstances()->forAll(g| g.gtype= #XOR_split implies g.fname<> ")
```

12. Each X(OR) Split gateway outgoing edges should be labeled

Each Split XOR/OR outgoing edge should have a name presenting a condition (Figure 20).

```
context Gateway inv outgoing_edges_label:
self.gtype= #XOR_split or self.gtype= #OR_split implies self.outgoings->forAll( oc1, oc2| oc1 <> oc2
implies (oc1.C_Name <>"))
```

13. Each Split gateway have one incoming edge and at least two outgoing edges

Each type of gateway has constraints on the number of the edges coming in and going out from it (Dehnert et Rittgen 2001). Modelling "either" situations is performed via XOR- Split Gateways. Depending on conditions, exactly one outgoing flow will be chosen, the control flow is split. Thus a Split gateway should have one incoming edge and at least 2 outgoing edges (Figure 20) (« BPMN Tutorial Gateways - BPMN-Community » 2013).

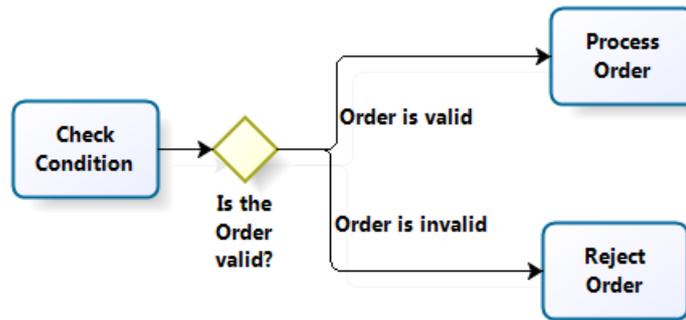


Figure 20: XOR Split Gateway

context Gateway inv Split_outgoing_edges:

self.gtype= #XOR_split or self.gtype= #OR_split or self.gtype= #AND_split implies self.outgoings->size()>=2

context Gateway inv Split_incoming_edges:

self.gtype= #XOR_split or self.gtype= #OR_split or self.gtype= #AND_split implies self.incomings->size()=1

14. Each Join gateway has one outgoing edge and at least two incoming edges

A Join gateway joins at least two separated paths into a common one. Thus it should have one outgoing edge and at least 2 incoming edges. (« BPMN Tutorial Gateways - BPMN-Community » 2013).

context Gateway inv Join_outgoing_edges:

self.gtype= #XOR_join or self.gtype= #OR_join or self.gtype= #AND_join implies self.outgoings->size()=1

context Gateway inv Join_incoming_edges:

self.gtype= #XOR_join or self.gtype= #OR_join or self.gtype= #AND_join implies self.incomings->size()>=2

15. An activity can not have two successors

It will create a confusion on which flow to trigger (Van Der Aalst 1998b).

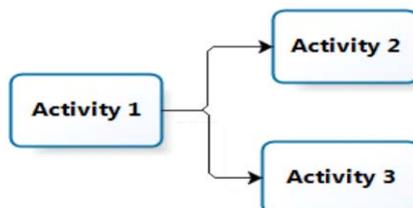


Figure 21: Activity with two outgoing edges

context Connector inv successors:

```
Connector.allInstances()->forAll(c1,c2| c1.Range.ftype =#process implies c2.Range.ftype <> #process )
```

16. Flow Objects can not be applied on the lane's, pool's or the swimlane's boundary

This defect may occur in BPMN or Activity diagrams module. Both notations contain concepts represented with boundaries.

17. A Pool or a Lane with no activities associated to it.

This defect may occur in BPMN or Activity diagrams module. A pool or a lane with no activities related to them or associated to them are useless.

18. A sequence flow relates activities and events

A sequence flow represents the order of activities. A resource can be neither the domain nor the range of a sequence flow. "sqdomain" constraint states that the domain of a sequence flow is a #start_event, an #intermediate_event, a #gateway or an #activity.

```
context Sequence_flow inv :
```

```
Sequence_flow.allInstances()->forAll(sf| sf.Domain.ftype = #flowObject and sf.Range.ftype = #flowObject)
```

19. Associations are used to connect artefacts to activities or intermediate events from the model.

An association is used to connect artefacts and flow objects.

```
context Association inv :
```

```
Association.allInstances()->forAll(as| as.Domain.ftype = #Artefact or sf.Range.ftype = #Artefact)
```

20. A Split XOR gateway is used only for an "either" situation.

If XOR split has an arc both to an event E1 as to another event E2, the semantics of the split means that both events can not take place at the same time thus E1.token = true implies E2.token = false. A bad application of XOR Split gateway may occur if E1 and E2 do not contradict each other and may take place at the same time, we can assume that there is an error in the model. The modeller can replace a XOR Split with an OR Split. In fact when applying the XOR split, it blocks one of the successors but semantically both successors can be processed.

```
context Gateway inv XOR_split:
```

```
Gateway.allInstances()->forAll(g| g.gtype=#XOR_split implies g.outgoings->forAll( oc1, oc2| oc1.Domain.fID=g.fID and oc2.Domain.fID=g.fID and oc1.Range.token= true implies oc2.Range.token = false))
```

21. A Split XOR gateway outgoing edges presents opposite conditions.

For all the instances of gateways, if there is an instance of type XOR_Split, the outgoing edges labels should be contradictory.

```
context Gateway inv XOR_split_edges_cdt:
Gateway.allInstances()->forAll(g| g.gtype=#XOR_split implies g.outgoings->forAll( oc1, oc2|
Is_opposit(oc1, oc2)))
```

22. Timer intermediate event semantics

Timer event can be used as a delay mechanism or as a task duration (Rozman et al. 2013).

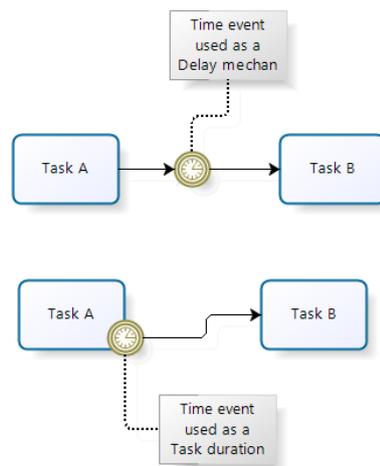


Figure 22: Timer intermediate event

Informational aspect defines the information involved in a business process. The information are presented by resources from different types. Thus informational constraints are the constraints associated to resources.

23. Resources and gateways can not be connected

Gateways should not receive or send any resources.

```
context Connector inv resource_connection:
Resource.allInstances()->forAll(r1,r2| self.Domain.fID =r1.rID implies self.Range.fID <> r2.rID)
```

For all the instances of resources, there is no connector that connect two resources, that is if there exist a resource r1 and an other r2, there exist no connector where domain is r1 and range r2.

```
context Connector inv resource_gateway_connection:
Resource.allInstances()->forAll(r| self.Domain.fID =r.rID implies Gateway.allInstances()->forAll(g|self.Range.fID <> g.fID))
```

```
context Connector inv resource_gateway_connection2:  
Resource.allInstances()->forAll(r| self.Range.fID =r.rID implies Gateway.allInstances()-  
>forAll(g|self.Domain.fID <> g.fID))
```

For all the instances of resources, there is no connector that connect a resource to a gateway, that is if there exist a resource r1 and a gateway g, there exist no connector where domain is r1 and range g.

24. A message flow relate activities to resources (abstract or information resource) and vice versa.

A message flow represents the exchange of messages/information between BPM concepts. For example, activities create outputs (information/abstract resources) and consume inputs (information/abstract resources).

```
context Message_flow inv Mf_domain:  
Message_flow.allInstances()->forAll(mf| mf.Domain.ftype = #abstractresource or mf.Domain.ftype =  
#informationresource implies mf.Range.ftype = #process)
```

```
context Message_flow inv Mf_range:  
Message_flow.allInstances()->forAll(mf| mf.Range.ftype = #abstractresource or mf.Domain.ftype =  
#informationresource implies mf.Domain.ftype = #process)
```

25. Each activity has an input and an output resource

A business process is focused upon the production of particular products. These may be physical products or less tangible ones like a service (W. V. D. Aalst et al. 2002). On the other hand, Davenport (Davenport 1998) sharpened his words and added that a business process is a specific ordering of work activities across time and place, with a beginning, and an end, and with clearly identified inputs and outputs - a structure for action.

```
context Process inv Process_Inputs_Allocation:  
Process.allInstances()->forAll(p| p.Inputs ->notEmpty())
```

```
context Process inv Process_Outputs_Allocation:  
Process.allInstances()->forAll(p| p.Outputs ->notEmpty())
```

26. Each activity has an actor responsible of its execution.

```
context Process inv Actor_Allocation:  
Process.allInstances()->forAll(p| p.Responsible ->notEmpty())
```

3.2.3.3 Pragmatic constraints

Pragmatic constraints do not detect notation defects but incomprehensible representations. It may propose a better way to represent an implicate idea.

1. **Each business process model has to contain one and only one start event and at least an end event.**

An occurrence of the Start event initiates the process and an End event terminates it (Jan Mendling, Reijers, et van der Aalst 2008). In contrast EPC allows multiple start events and multiple end events (Jan Mendling, Neumann, et Nüttgens 2005).

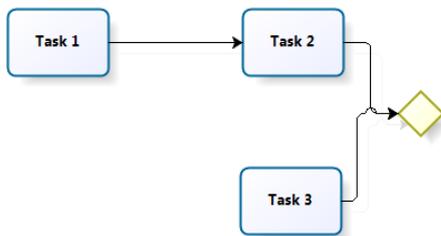


Figure 23: Process with no start event

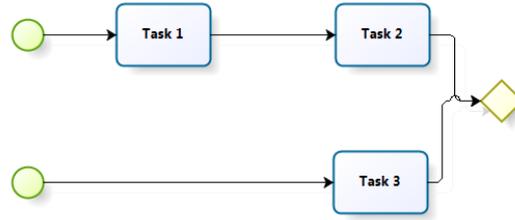


Figure 24: Process with multiple start events

```

context BusinessProcess inv NumberOfStartEvent:
BusinessProcess.allInstances()->forAll(bp|Starteventnb -> size()=1)
  
```

```

context BusinessProcess inv NumberOfEndEvent:
BusinessProcess.allInstances()->forAll(bp|Endeventnb -> size()>1)
  
```

If, in a process there is a parallel occurrence of start events, it may be presented by one start event succeeded by a parallel gateway that diverge the flow into two parallel paths.

2. **The presence of derived or redundant concepts makes the model less understandable.**

```

context FlowObject inv Redundancy:
FlowObject.allInstances()->forAll(f1, f2| f1.fname <> f2.fname)
  
```

An inappropriate concept name is considered as a pragmatic error.

3. **An activity name should begin with a verb** (Erreur ! Source du renvoi introuvable.).

```

context Process inv Process_label:
Process.allInstances()->forAll(p| Began_with_verb(p.fname) )
  
```

The function `begin_with_a_verb` is a Boolean function that return true if the label begin with a verb.

4. **A resource name cannot contain a verb** (Erreur ! Source du renvoi introuvable.).

```

context Resource inv Resource_label:
  
```

```
Resource.allInstances()->forAll(r| Contain_verb(r.rName)= false)
```

The function *contain-a-verb* is a Boolean function that return true if the label contain a verb.

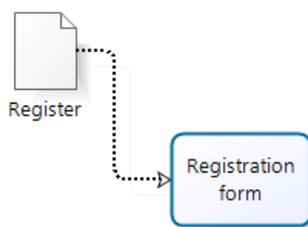


Figure 25: Erroneous model

3.3 Capturing application domain knowledge

Application domain knowledge refers to real-world problems. It can be extracted from several sources such as user's requirements statements, domain expertise or existing models related to the same problem or to similar problems.

Application domain knowledge covers 1) recurring and shared knowledge enabling the reuse of expertise and 2) application-specific knowledge required for development from scratch. We propose to capture the first category through domain ontology. In computer science, an ontology formally represents knowledge as a set of concepts within a domain, and of relationships between pairs of concepts. This knowledge can be the result of expertise capitalization or standardization effort within a given domain such as the e-tourism ontology for data exchange in tourism business domain. Such ontology can be built using requirements specification documents or requirements models. The definition of these ontologies is not in our scope. Domain ontologies are available in the literature, examples of foundational ontologies are: the Wonder Web Ontology Library (Masolo et al. 2003), in (Patron et al. 2008), the authors presented a mission plan ontology, Suggested Upper Merged Ontology (SUMO) (« Suggested Upper Merged Ontology » 2013), YAMATO (Cuiliqing 2013), UFO (Guizzardi 2007), etc.

On the other hand, in the paper of Guarino (Guarino 1998) present numerous domain ontologies are presented and classified into knowledge engineering, database design and integration, information retrieval and extraction.

In the first step, the approach consists in discovering the mappings between business process model elements and domain ontology elements. Similarly to make these alignment rules generic and independent of both the BP modelling notation and the ontology implementation language, we have defined an ontology meta-model presented in detail in Section 3.3.1.

Writing the rules at the notation level implies specifying the same rules several times to take into account the variety of vocabulary used to point out the same concept. The alignment rules aim to identify similarities between the process model elements and the domain ontology concepts. Once these similarities are identified, they serve as input for both semantic quality evaluation and improvement activities.

(Figure 26) illustrates an example of mappings between BP model concepts and domain ontology concepts. Based on these mappings, we exploit the domain knowledge by studying the different types of ontology relations.

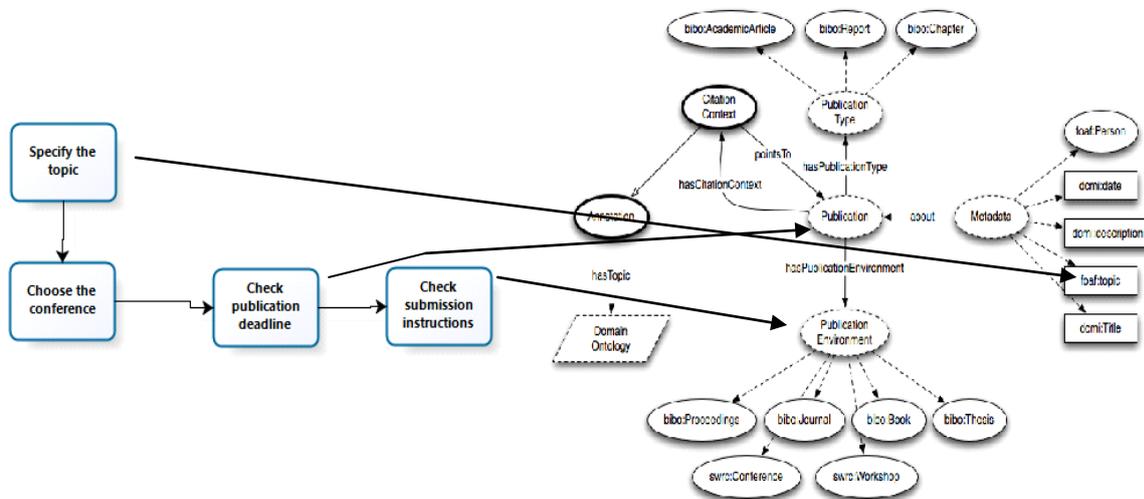


Figure 26: Mappings between Domain Ontology and Business Process Model

3.3.1 Ontology meta-model

The ontology meta-model allows representing domain ontologies using the same concepts independently of the language for their implementation. There are several contributions in literature concerning ontology meta-modelling. The authors in (Tziviskou et Keet 2007) introduced simple concepts and constructors (negation, conjunction, disjunction) to define complex concepts. They also defined several relationships including inheritance links, instantiation, and constraints. In (Kaiya et Saeki 2005) five types of concepts have been proposed to represent the functional requirements (function, object, and environment) and non-functional requirements (constraint, quality). Additionally several research projects have attempted to use some form of ontologies for conceptual design.

(Bergholtz et Johannesson 2001) propose an ontology to classify relationships based upon data abstractions and speech acts. (Storey, Chiang, et Chen 2005) presents an ontology for classifying the verb phrases of relationships based upon research in linguistics and semantic data models. (Storey et al. 1998) classify entities into various categories. These research efforts, however, concentrate on the classification schemes.

In our approach, we consider an ontology as a set of classes and relationships. This vision is largely adopted and more adequate for our approach as it gathers concepts and relations details that help us to align them with the BP model concepts.

We distinguish between three types of concepts: actor, action and artefact (Figure 27).

- An actor is an independent entity, able to perform actions.
- An action represents the execution of an action.
- An artefact is an inanimate object incapable of performing an action. An artefact may represent information or an abstract concept.

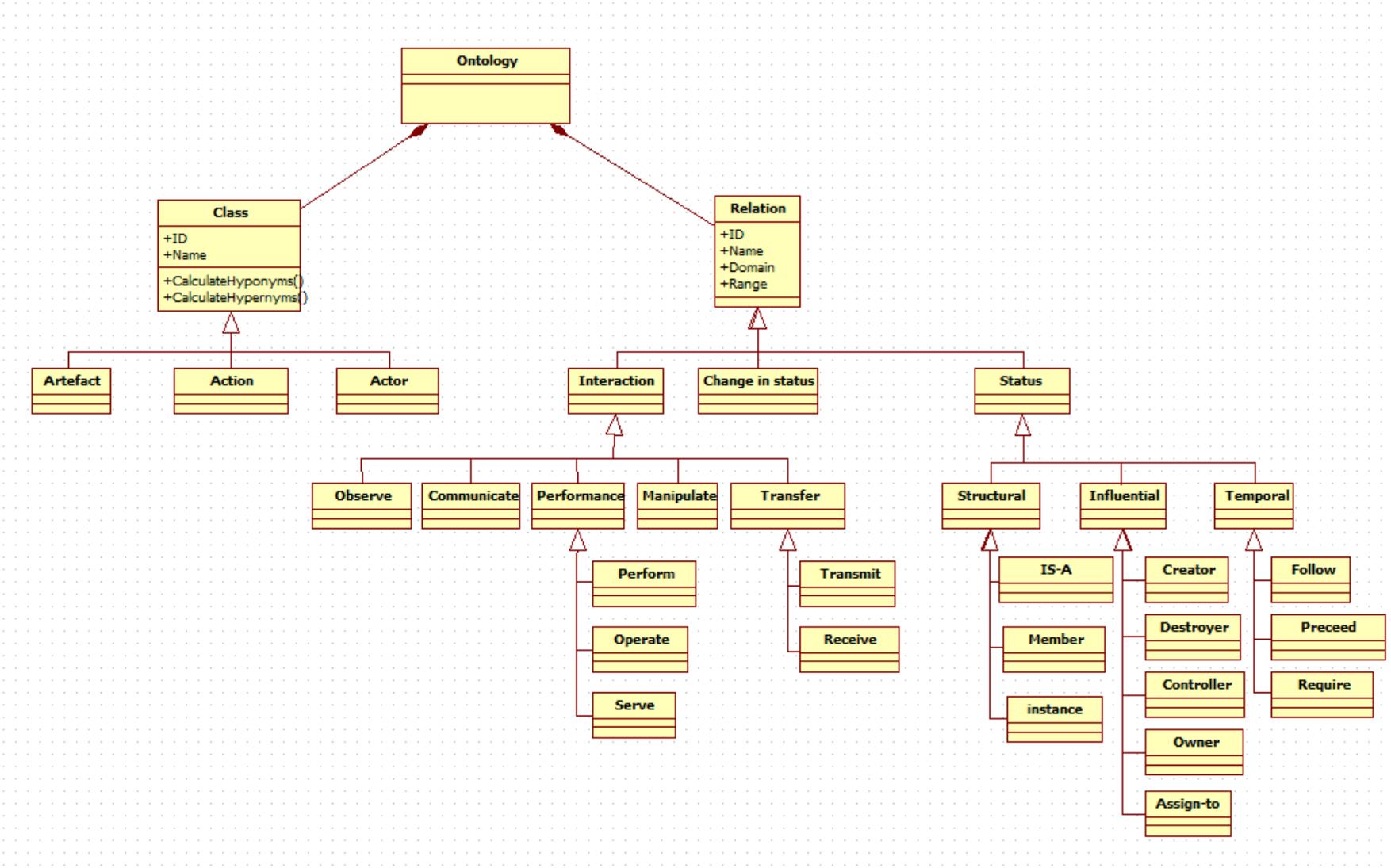
However, most meta-models take into account two kinds of relationships, namely inheritance and structural relationships.

For the needs of our approach we adapted the classification of relationships proposed in (Purao et Storey 2005) which has been initially defined to analyze semantics of relationships within a relational database and classify relationships by separating domain-dependent and domain-independent aspects of the relationship constructs (Sugumaran et Storey 2006). This classification offers several types of relationships allowing us to characterize precisely the nature of links between concepts. Each of these categories describes a connector role or semantics.

Relations are first decomposed into three categories:

- **Status:** represents relationships that may be structural (inheritance, composition, instantiation, etc.), influence (own, control, create, destroy, etc.), or temporal (follow, require, etc.).
- **Change of status:** reveals the occurrence of remarkable events. This type of relationship is primarily used to express the interdependence of status in the life cycle of an entity.
- **Interaction:** represents short-term relationships between entities. Several semantic relations are defined for interactions such as communication, observation, execution, etc.

Figure 27: Ontology Meta-model



3.3.2 Business Process metamodel - Ontology metamodel mappings

Thanks to the precise categorization of concepts in both ontology and process model meta-models we are able to predefine some concepts correspondences enabling the mapping of the domain ontology concepts with the business process model concepts.

We have defined two kinds of mapping, namely type-based mapping and instance-based mapping. Each concept in both business process and ontology meta-models have a type and a role. As a result, mappings are identified not only based on their types but also on the context of each instance.

3.3.2.1 Type-based mapping rules

This mapping involves the types of concepts in order to establish correspondences between the concepts at the meta-level. These correspondences allow reconciliation based on the types of concepts independently of their meaning. These rules are essential to avoid typing errors.

Let OD be the set of ontology concepts. $OD = \langle oc_1, oc_2, \dots, oc_n \rangle$ where oc_i is an ontology concept that can be an actor, an action or even a relation.

Let BP be the set of the business process model concepts. $BP = \langle bpc_1, bpc_2, \dots, bpc_m \rangle$ where bpc_i is a business process concept that can be a flow object, an activity or a connector.

Each business process model concept bpc is mapped to a set of ontology concepts oc [].

3.3.2.2 Concept based mappings

Based on the definition of each concept, we can define these mappings:

- An activity, which is the execution of an act in the BP metamodel is represented by an action in the ontology.
- An Actor, which represents the resource responsible of the execution of an activity, is represented by an actor in the ontology meta-model as it is responsible of the execution of an action. .
- An abstract resource in the BP meta-model represents an idea or a thought, it is mapped to an abstract concept in the ontology since it represents something formed in the mind, a thought or a notion.
- An Information resource in the BP meta-model is the Knowledge communicated or received. It is mapped to a knowledge concept in the ontology since it represents an information.

These mappings are summarized into (Table 10).

BP model meta-model concept	Domain Ontology meta-model concept
Actor	Actor
Abstract Resource	Abstract
Information Resource	Knowledge

Process/Activity	→	Action
------------------	---	--------

Table 10: Concept alignment

BP model meta-model connectors	Domain Ontology meta-model relations
Sequence Flow	Temporal
Message Flow	Communication Transfer
Role	Execution Manipulation Observation Influence

Table 11: Relationships alignment

3.3.2.3 Relation type based mappings

Similarly, we have established mappings between meta-model relations of BP model and of the ontology.

- A sequence flow in the BP meta-model presents the transition from an activity to an other or from a step to the following. It is represented by a temporal relation which shows the transition from a step to an other, from a lifecycle status to an other.
- A message flow in the BP meta-model represents the exchange of messages between two participants. It is represented by a communication and a transfer relation in the ontology since they have the same role.
- The Role relation in the BP model describes the role of a resource into an activity. It is presented with an execution, manipulation, observation or influence relation which shows the participation role of the different concepts.

3.3.2.4 Instance based mapping rules

Based on meta-models presented above, we developed a set of matching rules, allowing the mapping of the ontology field with the concepts of process models.

These rules are written in OCL (Brucker et al. 2011). The similarity computation uses the names of concepts, the synonyms, and keywords associated to ontology concepts. It is based on Wordnet and distance computation algorithms from literature such as Resnik information content (Resnik 1995), Jiang and Conrath (Jiang et Conrath 1997), Lin (Lin 1998), Wu and Palmer path length (Wu et Palmer 1994), Pathwardhan & Pedersen context vectors (Purandare et Pedersen).

There are six classes of matching rules. The rules are all defined as functions having, as input, one or several BP model concepts and returning one or several concepts from the domain ontology.

- 1) Name based similarity: It takes a BP model element and returns a set of ontology concepts which are syntactically equivalent (have the same names). It is based on Levenshtein distance (« Levenshtein Distance » 2013) which measures the difference between two sequences of strings. If it returns 0, it means that the two sequences are syntactically equivalent.

```

Name-Based-Similarity ( bp: Business process model element ) : Ontology-Concept[]
{
  /*We calculate the Levenshteindistance between the business process concept and all the
  ontology concepts thus OntologyConcepts.length is the number of concepts in the ontology */
  for (int i =0; i< OntologyConcepts.length; i++)
  if(Levenshteindistance (bp, o[i]) =0)
  return o[i];
}

```

```

LevenshteinDistance ( String a, String b ) : int
{
  /* len_a and len_b are the number of characters in string a and b respectively */
  int len-a = length (a);
  int len-b = length (b);
  /* if last characters of the strings match then cost = 0 else cost = 1*/
  if ( last character of a == last character of b ) cost = 0;
  else cost = 1;
  return minimum (
    LevenshteinDistance ( a without the last character, b ) +1;
    LevenshteinDistance ( a, b without the last character ) +1;
    LevenshteinDistance ( a without the last character, b
    without the last character ) + cost);
}

```

- 2) Partially name based similarity: It is used for composed names. For example let's consider a school teacher and teacher or school teacher and college teacher. These words are similar but are not detected with the name based similarity function. Partially name similarity function takes a BP model element and returns a set of ontology concepts which are partially equivalent. This function clean the concepts label from conjunctions as "and", " for", "the" and from the undesired characters as "_", "-", ..etc.

```

Partially-Name-Similarity( BPM concept bpm ) : Ontology-Concept[ ]
{
  int bpm = number of words that compose bpm;
  For all the ontology concepts do

```

```

    int lo = number of words that compose o;
    for (int i=0; i< lbpm; i++)
    for (int j=0; i< lo; j++)
    Calculate LevenshteinDistance ( bpm[i], o[j]);
    if(LevenshteinDistance=0) return o[i]
    endif
    enddo
}

```

3) Synonymy: returns a set of ontology concepts that are synonyms of the BP model concept in input.

The synonymy concepts are calculated via two ways. The first method relies on two steps:

- Let o be the ontology concept mapped to the BPM concept bp by the name based similarity function or partially name similarity function.
- Return all the synonyms of o

The second method aims to calculate the ontology concepts that are synonyms of bp.

The synonymy value is calculated by comparing the existence of common names or synonyms based on Wordnet (Fellbaum 1998) and several algorithms extracted from the literature: Resnik information content (Resnik 1995), Wu and palmer path length (Wu et Palmer 1994), Purandare and Pedersen context vectors (Purandare et Pedersen 2004).

a) Synonymy calculated by Resnik information content (Resnik 1995):

- It is established on the following hypothesis: the more information two concepts share in common, the more similar they are.

The information content of a concept (term or word) is the probability of finding the concept in a given corpus: $p(c)$ is the probability of encountering such concept c . The probability p of a concept c is ranged between 0 and 1. If $C1$ is related to $C2$ by an "IS-A" relation i.e. if $C1$ IS-A $C2$ then $p(c1)$ is less than $p(c2)$.

- By using the standard argumentation of information theory by (Ross 1976), information content of a concept is defined as the negative log likelihood, $-\log p(c)$.
- As probability increases, informativeness decreases, so the more abstract a concept is, the lower its information content.
- This quantitative characterization of information provides a new way to measure semantic similarity. The more information two concepts share in common, the more similar they are, and the information shared by two concepts is indicated by the information content of the concepts that subsume them in the taxonomy.

$$IC(c) = sim(C1, C2) = \max [-\log p(c)]$$

$0 \leq IC(c) \leq 18.75$ where larger values indicate greater relatedness

$Min = -\log(1) = 0 = IC(\text{root})$

$Max = -\log(1/\text{freq}(\text{root})) = 18.75 = IC(\text{least frequent i.e. most informative})$

Based on this algorithm, we can compute the ontology concepts that are synonyms of the BPM concept bp.

Ontology-Resnik-Synonyms(BPM bpm)

{ For all the ontology concepts

Calculate $IC = sim(bp, o)$;

if($IC > 15$) return $o[i]$

}

b) Synonymy calculated by Wu and Palmer path length (Wu et Palmer 1994):

(Wu et Palmer 1994) propose a measure that computes the similarity between two concepts by looking at the path length between them in the Wordnet taxonomy.

Intuitively, it seems reasonable to take the path length into account when determining semantic similarity: the closer two nodes are together in a taxonomy, the more semantically similar they are.

Wu and Palmer propose the following measure to measure the path length:

$$sim_{wp}(c_1, c_2) = \frac{2C}{A + B + 2C}$$

where $A = \text{len}(c_1, \text{lcs}(c_1, c_2))$, $B = \text{len}(c_2, \text{lcs}(c_1, c_2))$, and $C = \text{depth}(\text{lcs}(c_1, c_2))$.

Note that the length of the shortest path from synset c_i to c_j (measured by counting the edges between nodes) is denoted by $\text{len}(c_i, c_j)$. The least common subsumer $\text{lcs}(c_1, c_2)$ is the common subsumer for which the path length $\text{len}(c_1, c_2)$ is minimal, i.e. the most specific common superclass.

c) Synonymy calculated by Purandare and Pedersen context vectors (Purandare et Pedersen 2004):

Purandare developed a measure of semantic relatedness that represents a concept as a Context Vector. This is intended to be a more general representation than similarity measurements, since the source of the information for the context vectors is a raw corpus of text, not the paths found between concepts in an ontology. We start by creating Word Vectors, which are first order context vectors, for every content word w in our corpus of text. The dimensions of these vectors are content words from the same corpus of text (each dimension corresponding to one content word). The vector for a word w is created as follows:

1. Initialize the first order context vector to a zero vector $\sim w$.
2. Find every occurrence of word w in the given corpus.

3. For each occurrence of w , increment by 1 those dimensions of $\sim w$ which correspond to the words present in a specified window of context around w .

$$rel_{vector}(c_1, c_2) = \frac{v_1 \cdot v_2}{|v_1| |v_2|}$$

where v_1 and v_2 are the context vectors corresponding to c_1 and c_2 , respectively.

4) More general level: This function returns the list of the most general concepts of the ontology i.e. there is a relationship of superiority (also called hypernyms or IS-A relationship) between the business process concept in parameter and the ontology concepts in return. In other words, it returns the range of the IS-A relation.

The function More-General takes a flow object b as input and returns a set of ontology concepts $MG[]$ more general than b .

- It retrieves the Name-Based-Similarity(b), the Partially-Name-Based-Similarity(b) and the Synonymy(b) results
- Collect all their hypernyms. To collect the hypernyms we have to exploit the "IS-A" relations from the ontology, finally returning the IS-A range.

```
More-General (FlowObject b) : MG[ ]
```

```
{ /* The name based similarity ontology concepts are saved into a set O[ ] */
```

```
    O[ ] = Name-Based-Similarity(b) ;
```

```
/* The partially name based similarity ontology concepts are saved into a set O'[ ] , O'.length is the cardinality of the set O'*/
```

```
    O'[ ] = Partially-Name-Based-Similarity(b) ;
```

```
/* The synonyms ontology concepts of the BP concept are saved into a set O''[ ] , O''.length is the cardinality of the set O'' */
```

```
    O''[ ] = Synonyms(b) ;
```

```
Connector C;
```

```
    for( int i=0; i<O.length ; i++)
```

```
        { if( C.type=="IS-A" and C.domain = O[i])
```

```
            return C.range
```

```
        }
```

```
    for( int i=0; i<O'.length ; i++)
```

```
        { if( C.type=="IS-A" and C.domain = O'[i])
```

```
            return C.range
```

```
        }
```

```
    for( int i=0; i<O''.length ; i++)
```

```

        { if( C.type=="IS-A" and C.domain = O"[i])
          return C.range
        }
      }
    }
  }
}

```

5) More specific level: returns the ontology concepts having an inferiority relationship with a concept from the ontology detected as synonyms or syntactically equivalent to the BP model element.

The function More-Specific takes a flow object b as input and returns a set of ontology concepts $MG[J]$ more specific than b .

- It retrieves the Name-Based-Similarity(b), the Partially-Name-Based Similarity(b) and the Synonymy(b) results
- Collect all their hyponyms. To collect the hyponyms we have to exploit the "IS-A" relations from the ontology, finally returning the IS-A domain.

```

More-Specific (FlowObject b) : MG[ ]
{ /* The name based similarity ontology concepts are saved into a set O[ ] */
  O[ ] = Name-Based-Similarity(b) ;
/* The partially name based similarity ontology concepts are saved into a set O'[ ] */
  O'[ ] = Partially-Name-Based-Similarity(b) ;
/* The synonyms ontology concepts of the BP concept are saved into a set O''[ ] */
  O''[ ] = Synonyms(b) ;
Connector C;
  for( int i=0; i<O.length ; i++)
    { if( C.type=="IS-A" and C.range = O[i])
      return C. domain
    }
  for( int i=0; i<O'.length ; i++)
    { if( C.type=="IS-A" and C.range = O'[i])
      return C. domain
    }
  for( int i=0; i<O''.length ; i++)
    { if( C.type=="IS-A" and C.range = O''[i])
      return C. domain
    }
}
}

```

```
}

```

- 6) Related concepts: returns a set of ontology concepts linked to the results obtained by name based and synonym based similarity by a "temporal" or an "influence" relation to an other ontology concept.

The function Context takes a flow object b as input and returns a set of ontology concepts $MG[]$.

- It retrieves the Name-Based-Similarity(b), the Partially-Name-Based Similarity(b) and the Synonymy(b) results
- Collect all the concepts that form the context of b . To do so we have to exploit the connectors of type: "require", "follow", "precede", "assigned-to", "control", "destroy", "create" from the ontology, finally returning the domain of the connector.

```
Context (FlowObject b) : MG[]
{ /* The name based similarity ontology concepts are saved into a set O[] */
    O[] = Name-Based-Similarity(b) ;
    /* The partially name based similarity ontology concepts are saved into a set O'[] */
    O'[] = Partially-Name-Based-Similarity(b) ;
    /* The synonyms ontology concepts of the BP concept are saved into a set O''[] */
    O''[] = Synonyms(b) ;
Connector C;
    for( int i=0; i<O.length ; i++)
        { if( ( C.type=="influential" || C.type = "temporal" )and C.domain = O[i])
            return C. range
        }
    for( int i=0; i<O'.length ; i++)
        { if( ( C.type=="influential" || C.type = "temporal" )and C. domain = O'[i])
            return C. range
        }
    for( int i=0; i<O''.length ; i++)
        { if( (C.type=="influential" || C.type = "temporal" )and C.range = O''[i])
            return C. range
        }
}
}
```

These classes of rules are instantiated for each of the concepts of the BP meta-model. For each class, an example of instantiated rule for the Actor BP concept is given below.

The second type of mapping, presented in the following chapter, is richer, being based on the semantics of concepts.

3.4 Conclusion

To support business process, an enterprise needs to be aware of the process model quality. In fact, the structure of the model does not only affect the development of the system but also the comprehension of the model and leads to misunderstanding between the different participants. Hence it is important to ensure the models quality.

The first step of our approach is the domain knowledge exploitation. And based on Khatri and al. (Khatri et al. 2006), in information system, domain knowledge comprise two parts: IS domain knowledge and application domain knowledge. We explained in this chapter how we exploit each type of these knowledge.

Therefore, in order to exploit IS domain knowledge, we defined notation constraints that guarantees the absence of live-locks, deadlocks and other anomalies that can be detected without domain knowledge in order to improve the expressiveness of the model and the soundness of its structure. We gathered notation rules from different sources and we formalized them on a BP modeling metamodel level in order to ensure notation independancy. Thus to do so, we defined a BP modeling metamodel, BP model - BP metamodel alignments are defined and notation constraints are defined using OCL.

In order to exploit application domain knowledge, our approach rely on the existence of a domain ontology. We presented the alignment process that can be applied to each domain for which such ontology is available. We described the different types of alignments. BP metamodel - Ontology metamodel alignments: In order to evaluate the quality of BP models using domain knowledge.

The strength of our approach is that it is not domain-specific or notation specific. It may highly facilitate the task of BP modellers and lead to a significant improvement of BP models.

In the next chapter, we describe the evaluation and improvement process. Thus quality evaluation is done via quality metrics (syntactic, semantic and pragmatic) and quality improvement via functions and algorithms.

Chapter 4

Evaluation and Improvement of Business Process models Quality

Various parameters, frameworks and methodologies have been proposed in the literature to define and evaluate quality of conceptual models as (Lindland, Sindre, et Solvberg 1994c), (D. L. Moody, Shanks, et Darke 1998) and (D. L. Moody 1998). However, we are interested in evaluating BP models quality.

Modelling activity in general and BP modelling in particular are creative activities conducted by modellers using a given notation or modelling language. The result is of course highly dependent on the modeller experience in the notation practice, on his/her interpretation of the reality, and on the decisions he/she makes regarding the choice of concepts and details to be modelled. This explains the fact that several correct but different models can usually be generated from the same reality. However, these models are supposed to be faithful representations of the reality. Thus the definition of quality requirements for these models is, in fact, a mean to evaluate this modelling activity and ensure a better result. Many factors may be defined to characterize this quality.

To improve the quality of the models produced, several approaches are possible: assistance in the development process phase by generic methodological guides from experience, measurement and improvement of the specifications quality, reusing approved specifications fragments etc. In this thesis, we propose to exploit domain knowledge, which is supposed to reflect the knowledge shared by a community of actors, in order to improve the quality of process models.

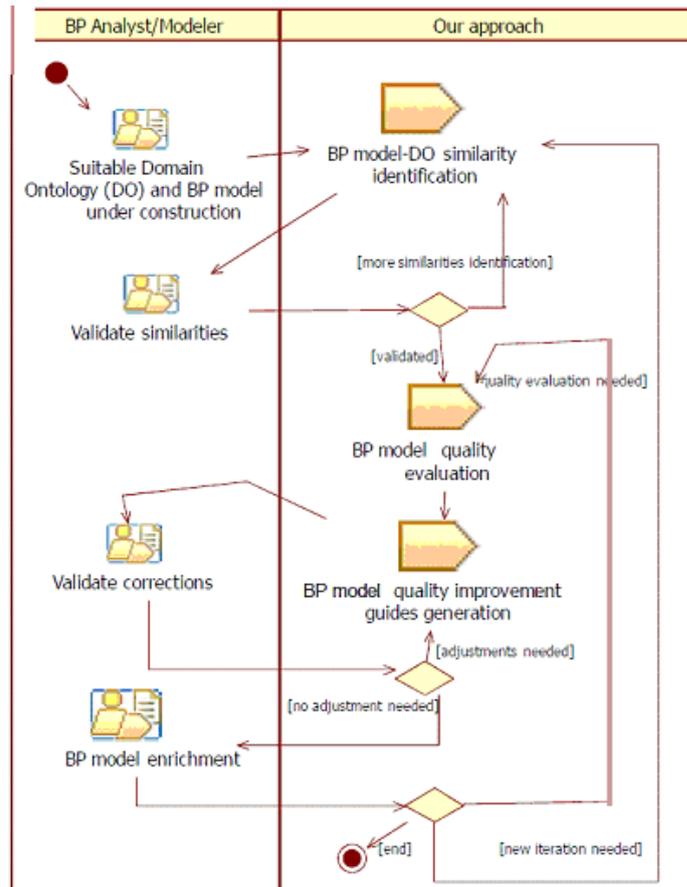


Figure 28: The overall approach for quality evaluation and improvement

The approach is based on the process described at (Figure 28). The entry point is composed of a business process model under construction and the associated domain knowledge provided by the process modeller. The process involves three steps:

- Identifying Model-Ontology similarities: In the first step, the approach consists in discovering the mappings between business process model elements and domain ontology elements. The alignment rules aim to identify similarities between the process model elements and the domain ontology concepts. Once these similarities are identified, they serve as input for both quality evaluation and improvement activities.
- Evaluating quality: The second step consists in evaluating the syntactic, semantic and pragmatic quality of business process models. Thus in order to evaluate syntactic quality, we have identified a set of syntactic metrics, related to the graphical symbols used. Many graphical notations are not consistent with principles of BP standards. In fact details of graphical syntax can have dramatic impact on understanding and problem solving performance. It can be measured by syntactic metrics such as quantitative or Boolean constraints. To evaluate the semantic quality, we have identified a set of what we call quality deficiencies such as incompleteness and ambiguity. These deficiencies result from modeling choices producing models that do not cover the intended requirements or with low

expressiveness. Such models lead to inadequate systems due to incompleteness or to misunderstanding during their implementation. Once a similarity has been identified between a BP model element, let it be bpm_i and an element from the domain ontology doj, our approach exploits the knowledge from the domain ontology related to doj to detect and measure semantic quality deficiencies according to quality metrics we have defined. Finally, to evaluate the pragmatic quality, we also identified a set of pragmatic metrics concentrated on the graphical communication effectiveness, it can be measured by the ease and accuracy in which information can be understood.

- **Quality improvement:** The quality improvement activity consists in suggesting to the analyst a set of improvement guidelines to improve the quality of their models. Our approach proposes different type of improvements: Textual guidelines, Preventive actions and corrective actions. Again, this step uses the domain knowledge to generate improvement actions. This means that the completeness and even the relevance of these guides rely partly on the quality of the domain ontology but this aspect is out of the scope of our approach. For example, if the approach identified a similarity between bpm_i – a BP model element and doj - an element from the domain ontology- and the domain ontology describes a relationship between doj and dok (an other element from the ontology), then our approach will propose an enrichment action on the BP model based on the relationship between doj and dok. The analyst, with the help of the suggestions made by the approach, adds/removes/rename/replace elements from the initial model that evolves.

However, quality improvement is never done in one shot and this is why the approach is iterative and incremental. The further sections detail the different steps of the proposed approach.

4.1 Syntactic quality evaluation

The syntactic quality checks the correctness of the model and evaluate the quality related to the concepts used.

In order to evaluate the syntactic quality we have identified a set of pragmatic quality metric. Each metric has a return type, a name and explanation (how is it calculated, what is the defect detected and a result analyse). These metrics are presented in (Table 12).

Metric return type	Metric Name	Explanation	Analyze of results
Boolean	Deadlock	This metric detects deadlocks. If a XOR split is succeeded by an AND join, Deadlock = true.	<ul style="list-style-type: none"> • Deadlock = true i.e. deadlock detected. • Deadlock = false i.e. no deadlock detected.
Boolean	Lack_of_synchronization	This metric detects Lack of synchronization. If an AND split is succeeded by a	<ul style="list-style-type: none"> • Lack_of_synchronization = true i.e. lack of synchronization detected. • Lack_of_synchronization = false i.e.

		XOR join, Lack of synchronization = true.	no lack of synchronization detected.
Integer	Start_event_nb	This metric detects start event defects. If number of start events in the model is less than 1 or higher than 1, a defect is detected.	<ul style="list-style-type: none"> • If $1 < \text{nb_start_event} < 1$, a defect is detected. • If $\text{nb_start_event} = 1$, no defect detected.
Integer	End_event_nb	This metric detects end event defects. If number of end events in the model is less than 1, defect is detected.	<ul style="list-style-type: none"> • If $\text{nb_end_event} < 1$, a defect is detected. • If $\text{nb_start_event} > 1$, no defect detected.
Boolean	Activity_Label	This metric detects unlabeled activities. If an activity has no name, thus $\text{name} = ""$, a defect is detected.	<ul style="list-style-type: none"> • If $\text{Activity.name} = ""$, an unlabeled activity is detected. • Else no defect detected.
Boolean	X(OR)_split_label	This metric detects unlabeled X(OR) gateways. If a X(OR) split gateway has no name, thus $\text{name} = ""$, a defect is detected.	<ul style="list-style-type: none"> • If $\text{X(OR)gateway.name} = ""$, an unlabeled X(OR) split gateway is detected. • Else no defect detected.
Integer	Split_gateway_outgoing_edges_nb	This metric detects bad Split gateway emplacement. If the number of outgoing edges is less than two, a bad emplacement gateway defect is detected.	<ul style="list-style-type: none"> • If $\text{Split_gateway_outgoing_edges_nb} < 2$ or $\text{Split_gateway_outgoing_edges_nb} = 0$, a bad emplacement gateway defect is detected. • If $\text{Split_gateway_outgoing_edges_nb} \geq 2$, no defect detected.
Integer	Split_gateway_incoming_edges_nb	This metric detects bad Split gateway emplacement. If the number of incoming edges is higher than one, a bad emplacement gateway defect is detected.	<ul style="list-style-type: none"> • If $\text{Split_gateway_incoming_edges_nb} > 1$ or $\text{Split_gateway_incoming_edges_nb} = 0$, a bad emplacement gateway defect is detected. • If $\text{Split_gateway_incoming_nb} = 1$, no defect detected.
Integer	Join_gateway_outgoing_edges_nb	This metric detects bad Join gateway emplacement. If the number of outgoing edges is higher than one, a bad emplacement gateway defect is detected.	<ul style="list-style-type: none"> • If $\text{Join_gateway_outgoing_edges_nb} > 1$ or $\text{Join_gateway_outgoing_edges_nb} = 0$, a bad emplacement gateway defect is detected. • If $\text{Split_gateway_outgoing_edges_nb} = 1$, no defect detected.
Integer	Join_gateway_incoming_edges_nb	This metric detects bad Join gateway emplacement. If the number of incoming edges is less than two, a bad emplacement gateway defect is detected.	<ul style="list-style-type: none"> • If $\text{Join_gateway_incoming_edges_nb} < 2$ or $\text{Join_gateway_incoming_edges_nb} = 0$, a bad emplacement gateway defect is detected. • If $\text{Split_gateway_incoming_edges_nb} \geq 2$, no defect detected.
Boolean	X(OR)_outgoing_edges_label	This metric detects Unlabeled outgoing edges. If a X(OR) outgoing edge has no name, thus $\text{name} = ""$, an unlabeled edge is detected.	<ul style="list-style-type: none"> • If $\text{X(OR)_outgoing_edges_label} = ""$, an unlabeled edge is detected. • Else no defect detected.

Boolean	Usefull_lane	This metric detects Useless lanes. If the lane does not contain any activity, thus Useful_lane= false and an unuseful lane is detected.	<ul style="list-style-type: none"> • If Useful_lane = true, an unuseful lane is detected. • If Useful_lane = false, no defect detected.
Integer	Actor_allocation	This metric detects lack of actor allocation. An activity with no actor is a defect.	<ul style="list-style-type: none"> • If Actor_allocation=0 a defect is detected. • If Actor_allocation\geq1, no defect is detected
Integer	Activity_consumed_resources	This metric detects lack of inputs. If an activity has no input resources it is an incomplete activity.	<ul style="list-style-type: none"> • If Activity_consumed_resources<1, an incompleteness defect is detected. • If Activity_consumed_resources>1, no defect is detected
Integer	Activity_produced_resources	This metric detects lack of outputs. If an activity has no output resources it is an incomplete activity.	<ul style="list-style-type: none"> • If Activity_produced_resources<1, an incompleteness defect is detected. • If Activity_produced_resources>1, no defect is detected

Table 12: Syntactic quality metrics

Each of these metrics detects a syntactic defect, by returning a Boolean or an integer value. They help the modeller to evaluate and improve the syntactic quality of their models.

4.2 Semantic quality evaluation

Semantic quality expresses the degree of correspondence between the information expressed within a model and the domain that is modelled (Krogstie, Lindland, et Sindre 1995b). It helps evaluating the ambiguous representations, incomplete representations and the meaningless states of the model (Wand et Wang 1996).

- The validity evaluation checks that all the statements of the model are correct. If the model do represent reality as close as possible.
- The completeness evaluation, however, verifies that all the pertinent statements from the domain are captured by the model.
- Meaningless states assessment verifies that the statements are related to the domain.

These deficiencies result from modelling choices producing models that do not cover the intended requirements or with low expressiveness and may be caused by an incorrect use of concepts. Such models lead to inadequate systems due to incompleteness or to misunderstanding during their implementation.

Different semantic measures have been proposed in the literature to evaluate the strength of the semantic link between two concepts or two groups of concepts from either two different ontologies (ontology alignment) or the same ontology. But in this thesis we are interested in the semantic quality of the business process models based on the knowledge provided by different sources (IS domain knowledge and application domain knowledge). Thus we need to study the strength of the semantic link between a business process model and ontology.

4.2.1 Quality defects detection

First, similarities have to be identified between BP model elements and domain ontology elements. Let bp_m be a BPM element and o a domain ontology element. Our approach exploits the knowledge from the domain ontology related to o to detect and measure semantic quality deficiencies.

- We have identified a set of quality deficiencies which result from modeling choices producing models that do not cover the intended requirements or with low expressiveness.
- We identified for each quality deficiency a set of quality metrics in order to measure the semantic quality in quantitative terms. These metrics reduce subjectivity in the evaluation process.

We group the quality defects into four categories: Meaningless states defects, ambiguous representation defects and incomplete representation defects (Wand et Wang 1996) .

4.2.1.1 Meaningless state defects

Meaningless states correspond to states and constructs from the models for which no correspondence is found in the corresponding ontology. This decreases the relevance of models and has an impact on its intelligibility. In fact the analyst may add an activity which does not represent a domain related role. In other words it represents meaningless data.

4.2.1.2 Ambiguous representation defects:

Ambiguous defects can be caused by:

- Ambiguity: two or more interpretations of a concept i.e. which states of the real-world should be mapped into the same state of the information system (Wand et Wang 1996).
- Abstraction concept level: This is usually viewed as a precision problem. In other words, there is no sufficient information to represent the concept of the real-world or vice versa.

Ambiguity

Ambiguity is the ability to express more than one interpretation for the same concept. It results from:

- Using the same name to express different realities. For example consider a model providing a property "telephone number". If the reality corresponding to this model allows several categories of telephone numbers such as personal, office, home and cellular telephone numbers having each its own usage then the property "telephone number" from the model is ambiguous since it represents several realities.

This makes models unclear and creates confusion when trying to understand them. To have a quantified evaluation of the ambiguity of concepts we defined quality metrics.

Clarity Metric

For quality evaluation we speak about clarity (as desired characteristics) instead of ambiguity which is a non quality property. This metric calculates the ratio of 1 to the total number of synonyms detected in the ontology.

$$\begin{aligned} & \text{if } |C_synonyms(o)| \neq 0 \\ & C_clarity(bp_m) = 1/|C_synonyms(o)| \\ & \text{if } |C_synonyms(o)| = 0 \\ & \text{we have a meaningless state.} \end{aligned}$$

The more the concept has synonyms, the more ambiguous it is. When the value of clarity equals 1 for bp_m , a given BP model element, this means that there is only one corresponding concept from the domain ontology and thus bp_m is not ambiguous. The value of clarity decreases when the number of synonyms in the ontology increases. When the value of clarity is meaningless, this means that there are no corresponding elements found for bp_m . In this case, bp_m can be a meaningless state. A validation from the modeller has to be performed since the domain ontology is not the exclusive source of knowledge.

Abstraction level

Abstraction level is related to the use of the suitable level of generality. Indeed, in some cases, using general concepts instead of specific and precise ones can decrease the efficiency of the processes, for example "analyze the request" activity is a general activity that may hide a lot of important details. On the contrary, using very specialized terms may decrease the understandability of the models, for example excessive splitting of steps as "check the hotel reservation invoice", "check the plan ticket", "check the registration invoice" activities can be replaced by "check the documents". The relevant choice of an abstraction level depends on several factors among which we can mention the nature of audience (developers or users), the objective of the model (explanation or implementation), etc.

- **Generality quality attribute:** The more the concept has hyponyms, the more general it is.

$$\text{Generality}(c) = \frac{1}{|hyponyms(c)|} \text{ if } |hyponyms(c)| \neq 0 \text{ and unknown otherwise}$$

Where c is a modelling element from the BP model and $|hyponyms(c)|$ is the number of hyponyms of c .

The value of generality metric decreases when the number of hyponyms increases. When generality metric returns unknown, this means that there are no hyponyms found for c .

- **Specification quality attribute:** The more the concept has hypernyms, the more specific it is.

$$\text{Specification}(c) = \frac{1}{|\text{hypernyms}(c)|} \text{ if } |\text{hypernyms}(c)| \neq 0 \text{ and unknown otherwise}$$

Where c is a modelling element from the BP model and $|\text{hypernyms}(c)|$ is the number of hypernyms of c .

The value of specification metric decreases when the number of hypernyms increases. When specification metric returns unknown, this means that there are no hypernyms found for c .

4.2.1.3 Incomplete representations defects

Incompleteness is related to an incomplete representation of the real world. It occurs when the process model misses some requirements. This incompleteness can result from the complexity of concepts for which only a sub-set of the description is captured within the process model or to a lack of domain knowledge of the analysts or both of them. For example, if a BP model considers only missions with train tickets costs reimbursement whereas the reality includes also missions with hotel costs then the BP model is incomplete. This is generally captured by completeness quality attributes.

We rely on the knowledge provided by the ontology to capture different types of incompleteness.

Incomplete defects can be caused by:

- Input incompleteness
- Resource allocation incompleteness
- Model richness

Input completeness

Each activity in the BPM needs to consume/produce resources. So incompleteness is detected if the inputs and/or outputs are missing for an activity.

- Let "oa" be the ontology action mapped to the BP model activity "a" by the Instance based mapping rule Name-based-similarity or by one of the synonymy rules described in Section 3.3.2.4.
- Let inputs [] be a set of ontology concepts related to "oa" by a "require" relation. As the "require" relation relates ontology actions to abstract and knowledge concepts. These concepts represent information and abstract resources. In other words, if "oa" requires i , where $i \in \text{inputs []}$ represent a knowledge or an abstract concept then the activity "a" should have as input.
- Let X be is equal to the number of inputs of the activity "a".
- Let Y be is equal to the number of concepts required for the action "oa".
- Calculate the ratio of X/Y.

The result will be a percentage indicating how complete the inputs of the activity "a" are.

Resource allocation completeness

As each activity (Process or task) is allocated to an actor responsible of its execution, we can rely on the knowledge provided by the ontology based on the "Influential" and "performance" relations to detect the completeness of the resource allocation. "Performance" relations as "Perform", "Operate", "Serve" and "Influential" relations as "Creator", "Destroyer", "Controller", "Owner", "Assigned to" represent the role of an actor over an action. As a result the resource allocation completeness metric detects the completeness of actor allocation of an activity a in the BP model.

- Let "oa" be the ontology action mapped to the BP model activity "a" by the Instance based mapping rule Name-based-similarity or by one of the synonymy rules described in section 3.3.2.4
- Let Actors [] be a set of ontology concepts related to "oa" by an "influential" or "Performance" relation.
- Let X be equal to the number of actors related to the action oa.
- Let Y be equal to the number of the number of actors related to the activity a.
- Calculate the ratio of X/Y .

The result will be a percentage indicating how complete the resource allocations of the activity a are.

4.2.1.4 Model- richness

Richness of the model is related to the total coverage of the domain. Does the model cover the domain? Does it represent all the steps required? Does it mention all the documentations that can be used in the process? To detect the richness of the model we defined two metrics: the Model Activity Richness and the Model Resource Richness.

- Let X be is equal to the number of activities presented in the model.
- Let Y be is equal to the number of actions in the ontology.
- Let U be is equal to the number of resources (abstract and information resources) in the business process model.
- Calculate V which is equal to the number of knowledge and abstract concepts in the ontology.

$$\text{Model Activity Richness} = X/Y$$

$$\text{Model Resource Richness} = U/V$$

4.3 Pragmatic quality evaluation

Pragmatic quality describes how well the schema is understood by its users (Lindland, Sindre, et Solvberg 1994a). A conceptual schema with good pragmatic quality has been described as a schema that can be interpreted with little cognitive effort (K. Siau et Tan 2005).

A complete and accurate understanding of conceptual schemas by future system users is essential in the process of validating the user requirements captured in the schemas, as they are the primary stakeholders in the conceptual modelling process concerned with understandability (D. Moody 2003).

Studies on the readability of Conceptual models can also be found in the literature. (Krogstie, Lindland, et Sindre 1995a), for instance, list graph aesthetics and diagram layout among the means to achieve pragmatic quality. Others, (Poels et al. 2011), reported upon an empirical study that evaluated Resources-Events-Agents (REA) model modelling. They present a laboratory experiment that measured the user understanding of diagrammatic conceptual schemas developed using the REA model. Their approach is based on pattern recognition phenomena and the resulting reduction in cognitive effort for understanding conceptual schema.

In order to evaluate the pragmatic quality we have identified a set of pragmatic quality metrics. Each metric, based on its calculation result, detect a defect. These metrics are presented in (Table 13).

Metric return type	Metric Name	Explanation	Analyze of results
Integer	Redundancy	This metric detects redundancies. If there are two or more concepts with the same name, a redundancy defect is detected.	<ul style="list-style-type: none"> • If Redundancy > 2, redundancy defect is detected. • If Redundancy < 2, no defect is detected.
Boolean	Activity_label	This metric detects bad activity label structure. If the activity label does not begin with a verb, Activity_label = false and a structure defect is detected	<ul style="list-style-type: none"> • If Activity_label = false, structure defect is detected. • If Activity_label = true, no defect is detected.
Boolean	Ressource_label	This metric detects bad resource label structure. If the resource label begins with a verb, Ressource_label = false and a structure defect is detected	<ul style="list-style-type: none"> • If Ressource_label = false, structure defect is detected. • If Ressource_label = true, no defect is detected.

Table 13: Pragmatic quality metrics

4.4 BPM quality improvement

Business processes are human-centered activities (V Keith, 2008). Thus they need to be continuously improved in order to fulfill the requirements of the customers. "Improvement" means the organized creation of beneficial change, the attainment of unprecedented levels of performance (Juran et Godfrey 1999). Performance is the achievement of a goal which depends on the context of study. In our study the aim of software process model improvement is to increase the business process models quality thus enhancing the understandability and the maintenance costs.

Unterkalmsteiner et al. (Unterkalmsteiner et al. 2012) seize from the literature different types of software process improvement (SPI) initiatives: 1) frameworks like CMM (Paulk 1995), SPICE (ISO / IEC 15504) ; 2) practices which can be applied in one or more phases of the software development life-cycle and 3) tools i.e. software applications that support software engineering practices.

SPI framework initiatives aim to define a basis for judging a process quality or for solving process problems (Paulk 1995). The objective of any improvement framework is to bring out a more effective guidance enabling process improvement program. In the literature we can find a lot of frameworks such as CMM (Paulk 1995) which is the most reported one, Six-Sigma (Thomas et Fehlmann), CMMI (Ahern, Clouse, et Turner 2008), etc. Practices are about identifying improvement opportunities and eliminating the differences constitutes the actual process improvement. As in these practice approaches, measurements to control process change and to confirm goal achievement are a central part. We can cite top-down, bottom up approaches (H.A. Reijers et Liman Mansar 2005), the Quality Improvement Paradigm (QIP)/ Experience Factory (Basili 1993), etc.

Based on the distribution of the SPI initiatives shown in (Figure 29), combinations of SPI initiatives are recorded explicitly. We can conclude that the majority of the contributions are in the frameworks. However, the frameworks provide guides of insufficient granularity for a real assistance to the improvement. In contrast, initiatives that provide practices and tools are not more than 4% of the publications. Within the context of this thesis we are concerned with providing improvement techniques and a prototype which implements our improvement approach.

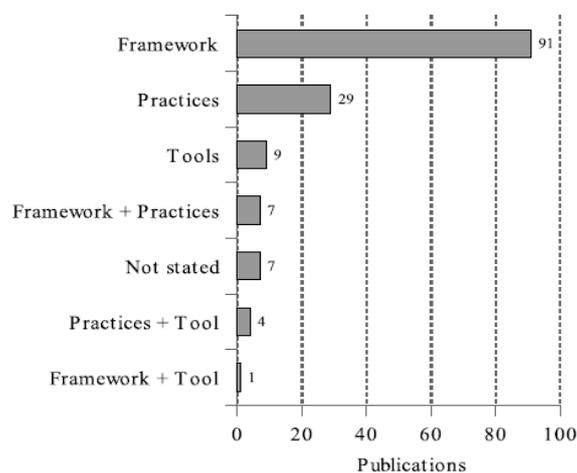


Figure 29: SPI Initiatives distribution of the publications (Unterkalmsteiner et al. 2012)

In this section we will present our proposed methods and techniques applied in the design phase of the business process in order to improve the business process models quality.

4.4.1 Business process improvement initiative

The mantra of many software process improvement frameworks and models originates in the Shwehart-Deming cycle (Deming 2000) plan–do–check–act:

- Plan: establish an improvement plan
- Do: implement the new process
- Check: measure the changed process
- Adjust: evaluate the effect of the adjust changes

The most dominant evaluation strategy is the "Pre-Post Comparison" (Unterkalmsteiner et al. 2012). It consists in comparing the success indicator's values before and after the improvements take place (Rozum 1993). Hence, it is necessary to define the success indicators i.e. the quality metric which can be used to evaluate the improvement, and to step-up a baseline from which the improvement can be approved.

Establishing that the quality of the process model has improved means that the number of quality defects decreased after the improvement process. As a result, our improvement strategy will contain two steps:

- a) Describe the improvement process: Explain what the possible ways to correct the defect are.
- b) Pre-Post comparison: Compare the success indicator after and before the improvement process.

4.4.1.1 Improvement process

The improvement process comprises three steps.

- The first step performs the improvements in order to correct the defects captured by the application of the IS constraints.
- The consists in activity label refactoring
- The third step implements the improvement that corrects the defects captured by the ontology.

We will detail each of these steps (Figure 30) using BPMN.

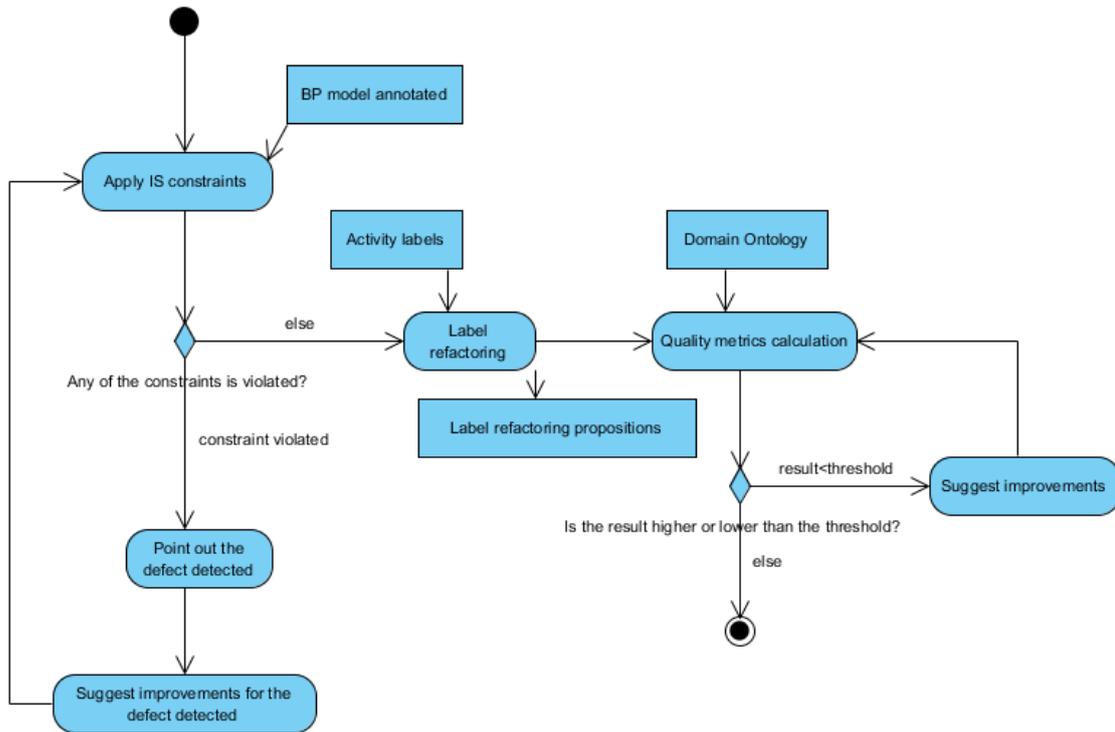


Figure 30: Our improvement process

- The IS constraints are violated when a quality defect is captured. The IS constraints solve the quality defects related to how the concepts are used in the model i.e. is the concept correctly used and the way this concept is elaborated in, does it yield the necessary information? Thus these constraints point out the problem but do not solve them. We aim to propose improvements to each of these defects.
- Activity label refactoring consists in categorizing the labels into classes and extracting information from them. The categorization function and information extraction algorithm will be detailed in Section 4.4.2.
- Based on the quality metrics value, quality defects are detected. In fact, semantic quality metrics depend on the knowledge provided by the domain ontology i.e. on the mappings between the process model concepts and the ontology concepts. But these mappings depend on the type and the semantic of the business process model concepts which we assume that they are improved by the IS constraints improvement propositions. Similarly, these metrics point out the problems but do not solve of. We aim to propose improvements to each of these defects.

4.4.2 Label refactoring

Process model quality has been approached from different angles, including verification, error probability, and comprehension. Also the small pieces of text that capture the names of activities (activity labels) have been investigated from a usability perspective. Usually, information about data is

hidden in activity labels. The latter represent actions, which take place during the execution of a business process.

Activity labels can be a complete sentence or a sentence fragment. They may contain a verb that can present an activity, a noun that may present a business object, etc

In this type of improvement, we aim to extract the information from the activity labels in order to complete the model. Based on the work of Leopold et al. (Leopold, Smirnov, and Mendling 2010), who developed an algorithm for the label style recognition and categorize the sentence fragments into noun phrase labels, verb phrase labels and irregular category. We completed their algorithm to classify the complete sentences into Object-verb-by-Subject, time-porter labels or subject-verb-object labels.

(Table 14) describe the label classes by explaining the structure and providing an example for each class.

Label Class	Structure	Example
Noun phrase	Label that began with a noun.	Invoice[O] creation
Verb phrase	Composed of at least one verb and the dependents of that verb: objects, complements. It does not include the subject.	Creating [V] invoice.
Irregular	Label contains characters ":", "_ " or "-"	Creation-Level.
Subject-verb-object	The subject comes first, the verb second, and the object third (« SVO (Subject-Verb-Object) » 2013).	The children [S] eat [V] buns [O].
Object-verb-by-Subject	Used to indicate the subject that does the verb in a passive sentence.	The flowers [O] were delivered [V] by a postman [S].
Time-porter	Contains two clauses related by a time subordinating conjunction. Time subordinating conjunctions are words that introduce dependent clauses in a sentence with respect to the time.	While the politician was speaking to a large crowd [C1], he fainted from heat exhaustion [C2].

Table 14: Label classes

The first algorithm aims to categorize the labels.

i. Name: Label_categorization

Input: Activity label

Output: Label Style

Description:

The input of the algorithm is a label, the output is the label style.

- First the algorithm examines if the label contains characters such as ":", "_ " or "- " that allow classifying it as irregular.
- The algorithm precedes with the sentence fragments by checking if the label contains a subject, a verb and a noun, it is classified as a Subject-verb-object. If the label

contains in order an object, a verb, the preposition "by" and a subject respectively then it is classified as Object-verb-by-Subject.

- Otherwise, we are in a complete sentence case. If it begins with a verb, in this case the label is categorized as verb phrase. If it begins with a noun, it is categorized as noun phrase label.
- The algorithm continues seeking for time conjunctions. If time conjunctions are found, the label is classified as time-porter.

Leopold et al. (Leopold, Smirnov, and Mendling 2010), also implemented an algorithm that aims to extract resources from the activity labels. We will adopt their algorithm in order to complete the BP model by proposing, for each activity, an input and an output.

ii. Name: Extract_Resource

Input: Activity label categorized as noun phrase or verb phrase.

Output: Resource

Description:

If an activity label is a list of words with a special structure, based on these words, we aim to extract resources (information or abstract) so that we can complete our model.

The extraction algorithm is decomposed into three steps:

- Label analysis: In order to determine the action (presented by the verb), the resource (presented by the noun) and the state (presented by the adverb). It is done by tagging the words of the label as verb, noun or adverb.
- Specify the output of the activities: We have several cases, each case has its own analysis. In the first case which is the basic one, an activity a1 precedes an activity a2. In this case, the resource created is using the noun of the activity label in hand to present the object and the adverb to present the state. The second case occurs when the activity is preceded by a XOR split. In this case, the object and its state are extracted from the XOR condition. The condition is the edge label.
- Specify the input of the activities. The third step aims to create inputs. In the contrary of the outputs, to specify the inputs, we have to analyze the successor.

iii. Name: Extract_BP_concepts

Input: Activity label categorized as time-porter.

Output: BP_concept

Description:

Activity labels categorized as time-porter may contain information that can be presented by a business process concept.

- Time subordinating conjunctions: They present the time transitional words in the sentence. The latter have the function of limiting, restricting, and defining time. The definitions of the time conjunctions were taken from the English-French Dictionary (« English-French Dictionary WordReference.com » 2013). Time transitional words are the subordinating conjunctions time clauses. Conjunctions relate two clauses in the time: Clause1 **Time_Conjunction** Clause2. The most common are: while, after, before and as soon as. Their functions are presented in (Table 15).

	Function	Example	Present in the BP model
While	It indicates that two separate clauses occur at the same time. Clause1 happens in the same time as Clause2.	<i>The phone rang while I was watching TV.</i>	<ul style="list-style-type: none"> • Create a parallel gateway "pg". • Add activity (v1) as a successor of "pg" where v1.name = verb extracted from Clause1. • Add activity (v2) as a successor of "pg" where v2.name = verb extracted from Clause2.
After	After is used to express that one event follows another. Clause1 happens after Clause2.	<i>The workers ended the strike [C1] after management agreed to give them a pay raise [C2].</i>	<ul style="list-style-type: none"> • Create a sequence flow. • Add activity (v1) as the domain of the sequence flow where v1.name = C1.verb + C1.O • Add activity (v2) as a range of the sequence flow where v2.name = C2.name + C2.O
Before	Before is used to express that one event precedes (comes before) another. Clause1 happens before Clause2.	<i>The workers ended the strike [C1] before management agreed to give them a pay raise [C2].</i>	<ul style="list-style-type: none"> • Create a sequence flow. • Add activity (v1) as the range of the sequence flow where v1.name = C1.verb + C1.O. • Add activity (v2) as a domain of the sequence flow where v2.name = C2.name + C2.O.
As soon as	As soon as is used to express that one event happens first and another happens immediately after . Clause1 happens after Clause2.	<i>The workers ended the strike [C1] as soon as management agreed to give them a pay raise[C2].</i>	<ul style="list-style-type: none"> • Create a sequence flow. • Add activity (v1) as the domain of the sequence flow where v1.name =C1.verb +C1.O. • Add activity (v2) as a range of the sequence flow where v2.name = C2.verb +C2.O.

Table 15: Time conjunction

iv. Name: Extract_Actor

Input: Activity label categorized as Subject–verb–object or Object-verb-by-Subject

Output: Actor

Description:

Activity labels categorized as Subject–verb–object or Object-verb-by-Subject may contain information about actor.

- If the label is categorized as Subject-verb-Object, the algorithm will return the first word.
- If the label is categorized as Object-verb-by-Subject, the algorithm will return the word that succeeds the preposition "by". In fact "by" is an English preposition that is used to indicate the person that does something in a passive voice sentence.

4.4.3 Quality improvement categorization

We have identified several types of improvement techniques. These improvements can be in the form of any of the following types: textual guidelines, corrective actions and preventive actions (Figure 31).

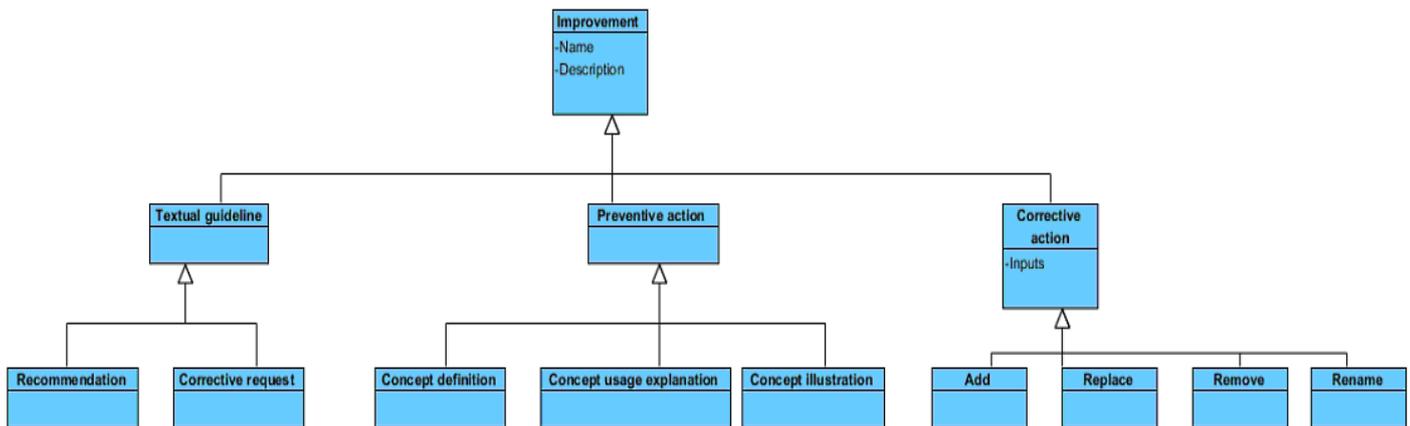


Figure 31: Improvements type

4.4.3.1 Textual guidelines

Textual guidelines are corrective improvements in the form of text for improving the quality of the business process models. They can point out the problem by describing the quality defect and/or suggest a solution. For some defects, it is necessary to correct them while for others it is recommended to rectify them. Thus the analysts can employ the suggested guideline for improving the quality of his model. Each textual guideline has a name and a description. Moreover, a textual guideline can be a recommendation or a corrective action:

- A recommendation is a corrective advice, a suggested improvement action: it consists in suggesting to the analyst a solution to improve the quality of his/her models.
- A corrective request is an obligation. It consists in obliging the analyst to perform the solution.

When an IS constraint is violated, a textual guideline will indicate to the analyst the quality defect and/or how to correct it. Thus, for each constraint, a proper textual guideline is attached to it. We classify below the textual guidelines in the same way as the IS constraints.

4.4.3.2 Textual improvements for syntactic constraints

Each textual guideline have a name and a description. The description provide an improvement guideline for the modeller.

<i><u>Name:</u></i> <i>Start_event constraint corrective guideline.</i>
<i><u>Description:</u></i> <i>A start event is missing. You should add a start event to your model in order to initiate your process.</i>

<i><u>Name:</u></i> <i>Multi_Start events constraint corrective guideline.</i>
<i><u>Description:</u></i> <i>There are several start events in the model. It is recommended that the model contains one start event.</i>

<i><u>Name:</u></i> <i>End_event constraint corrective guideline.</i>
<i><u>Description:</u></i> <i>An end event is missing. You should add an end event to your model in order to end your process.</i>

<i><u>Name:</u></i> <i>Start_event_emplacement constraint corrective guideline.</i>
<i><u>Description:</u></i> <i>A start event is preceded by an other concept. You should remove any concept that precedes the start event.</i>

<i><u>Name:</u></i> <i>End_event_emplacement constraint corrective guideline.</i>
<i><u>Description:</u></i> <i>An end event is succeeded by an other concept. You should remove any concept that succeeds the end event.</i>

<i><u>Name:</u></i> <i>Useless_activity constraint corrective guideline.</i>
<i><u>Description:</u></i> <i>The activity "Activity_ID" is without a name. It is recommended to rename the activity.</i>

<i><u>Name:</u></i> <i>Activity_Bad_Synchronization constraint corrective guideline.</i>
<i><u>Description:</u></i> <i>The Activity "Activity_name" can not reach the end event. You should create a path from the "Activity_name" to the end event respectively.</i>

<i><u>Name:</u></i> <i>Lost_Activity constraint corrective guideline.</i>
<i><u>Description:</u></i> <i>The Activity "Activity_name" is not reachable from the start event. You should create a path from the start_event to "Activity_name".</i>

<i><u>Name:</u></i> <i>Connector_Bad_Emplacement constraint corrective guideline.</i>
<i><u>Description:</u></i> <i>The connector "Connector_ID" is not connected.</i>

<i><u>Name:</u></i> <i>SequenceFlow_Bad_Emplacement constraint corrective guideline.</i>
<i><u>Description:</u></i> <i>The sequence flow "SequenceFlow_ID" crosses the boundary of "Process_name". The process "Process_name" will be initiated with its start event.</i>

<i><u>Name:</u></i> <i>Join_AND_gateway constraint corrective guideline.</i>
<i><u>Description:</u></i> <i>The input edges number is less than two. Or the output edges number is higher than one. In both cases, the gateway doesn't serve its purpose.</i>

Name: *Split_AND_gateway_constraint_corrective_guideline.*
Description: *The output edges number is less than two. Or the input edges number is higher than one. In both cases, the gateway doesn't serve a purpose.*

Name: *Unlabeled_XOR_Split_gateway_constraint_corrective_guideline.*
Description: *The XOR_Split gateway has no label. It is not clear which path will be triggered. It is recommended to rename the XOR split gateway*

Name: *Unlabeled_Split_OR_gateway_constraint_corrective_guideline.*
Description: *The gateway has no name. It is not clear which path the process triggers.*

Name: *Join_OR_gateway_edges_nb_constraint_corrective_guideline.*
Description: *The input edges number is less than two. Or the output edges number is higher than one. In both cases, the gateway doesn't serve its purpose.*

Name: *Split_OR_gateway_edges_nb_constraint_corrective_guideline.*
Description: *The output edges number is less than two. Or the input edges number is higher than one. In both cases, the gateway doesn't serve its purpose.*

Name: *Join_XOR_gateway_edges_nb_constraint_corrective_guideline.*
Description: *The input edges number is less than two. Or the output edges number is higher than one. In both cases, the gateway doesn't serve its purpose.*

Name: *Split_XOR_gateway_edges_nb_constraint_corrective_guideline.*
Description: *The input edges number is less than two. Or the output edges number is higher than one. In both cases, the gateway doesn't serve its purpose.*

Name: *Unlabeled_outgoing_Edge_constraint_corrective_guideline.*
Description: *The edge "Edge_ID" has no name. The edge name after a XOR/OR Split gateway represents the condition that should be approved to initiate the next activity. It is recommended to rename the edge in order to define the condition that initiates each path.*

Name: *Activity_synchronization_constraint_corrective_guideline.*
Description: *The Activity "Activity_name" has two outgoing connectors. It is recommended to create an AND Split gateway.*

Name: *Activity_Bad_Emplacement_constraint_corrective_guideline.*
Description: *The Activity "Activity_name" is detected on the lane's boundary. It should be assigned to one lane.*

Name: *Resource_constraint_corrective_guideline.*
Description: *The Resource "Resource_name" is connected to "Gateway_name". Nevertheless, a gateway does not receive nor send messages.*

We should note that all the behavioural constraints are assigned with textual guidelines that point out the error but all of them will recommend the modeller to replace one of the gateways.

4.4.3.3 Textual improvements for semantic constraints

Semantic constraints detect semantic defects. Hence textual guidelines can recommend ways to correct the defect but there are always many ways to correct a semantic defect. The modeller can change the way of thinking and thus change the concepts in the model.

<i><u>Name:</u></i> Resource_connection constraint corrective guideline.
<i><u>Description:</u></i> The Resource "Resource_name" and the Resource "Resource_name" are connected. There is no communication between two resources.

<i><u>Name:</u></i> Sequence_flow constraint corrective guideline.
<i><u>Description:</u></i> The sequence flow "Sf_ID" should not relate a resource to an activity. The sequence flow role is to represent the order of the activities in a process. It is recommended that you replace it with a message flow.

<i><u>Name:</u></i> Message_flow constraint corrective guideline.
<i><u>Description:</u></i> The message flow "Mf_ID" should not relate two flow objects. The message flow role is to represent the exchange of messages. It is recommended that you replace it with a sequence flow.

<i><u>Name:</u></i> Association constraint corrective guideline.
<i><u>Description:</u></i> The association "Af_ID" should not relate two flow objects. The association role is to relate artefacts with flow objects. It is recommended that you replace it with a sequence flow.

<i><u>Name:</u></i> XOR_Split_Bad_emplacement corrective guideline.
<i><u>Description:</u></i> The activities that succeed the XOR_split "XOR_name" are not contradictory. It is recommended to replace the XOR_split with an OR_split.

<i><u>Name:</u></i> OR_Split_Bad_emplacement corrective guideline.
<i><u>Description:</u></i> The activities that succeed the OR_split "OR_name" are contradictory. It is recommended to replace the OR_split with an XOR_split.

<i><u>Name:</u></i> XOR_Split_edges_labels corrective guideline.
<i><u>Description:</u></i> The outgoing edges of the XOR_Split "XOR_name" do not present opposite conditions. It is recommended to rename the outgoing edges.

<i><u>Name:</u></i> Actor_allocation constraint corrective guideline.
<i><u>Description:</u></i> The activity "A_name" is not associated to a actor. Each activity should have a resource responsible of its execution.

<i><u>Name:</u></i> Input_Output_Resource_allocation constraint corrective guideline.
<i><u>Description:</u></i> The activity "A_name" do not have any input nor output resources. Each activity should have inputs and outputs. It is recommended to add resources to the activity.

4.4.3.4 Textual improvements for pragmatic constraints

Recommendations are also suggested in order to improve the pragmatic quality of the model.

<i><u>Name:</u></i> Redundancy constraint corrective guideline.

Description: *The concept "concept_name" and the concept "concept_name" have the same names. It is recommended to rename one of them.*

Name: *Activity_label corrective guideline.*

Description: *The Activity "Activity_name" does not respect an activity label structure. It is recommended to rename the activity.*

Name: *Resource_label corrective guideline.*

Description: *The Resource "Resource_name" does not respect a resource label structure. It is recommended to rename the resource.*

4.4.3.5 Corrective actions

Corrective actions are the actions that allow the user to add, rename, replace and remove concepts from the model. Each concept has its own method that respects the concept definition and constraints. A corrective action has a name, inputs and a description.

Add actions

We can add different types of BP model concepts: an activity, an abstract/ information/actor, a connector, a start/end/intermediate event, a gateway or an edge for a gateway.

i. Name: Add_Activity

Inputs: Activity, Predecessor, Successor.

Description:

To add an activity "a" in the BP model, we should mention its successor and predecessor. Depending on the type of each of them, a different type of connector is used.

- If the predecessor is a flow object then a sequence flow will connect the flow object to the activity.

- If the predecessor is a of type resource then a message flow will connect the resource to the activity.

- If the successor is a flow object then a sequence flow will connect the activity to the flow object.

- If the successor is a resource then a message flow will connect the activity to the resource.

As a result, when we want to add a new activity we should mention its successor and predecessor.

ii. Name: Add_Resource

Inputs: Resource, Predecessor, Successor.

Description:

Similarly to add a resource, we should mention its successor and its predecessor. However, based on the IS domain knowledge constraints, a resource can not be related to an other resource. Thus

to meet the IS constraint, a resource can be related only to flow objects. When relating a resource to a flow object the only type of relation that can be used is a message flow. In contrary to AddActivity, Add Resource can have none successor or none predecessor.

iii. Name: Add_Actor

Inputs: Actor, Successor.

Description:

An actor is always related to an activity i.e. its successor is always an activity. It usually doesn't have a predecessor. And it is always linked by an association.

iv. Name: Add_StartEvent

Inputs: Successor.

Description:

A start event is related to its successor by a sequence flow.

v. Name: Add_EndEvent

Inputs: Predecessor.

Description:

An end event is related to its predecessor by a sequence flow.

vi. Name: Add_Gateway

Inputs: Type, Predecessor [], Successor [].

Description:

Each gateway type differs by the number of incomings and outgoings.

- If the gateway type is a Join gateway, then the number of predecessors should be greater than one and the number of successors is equal to one. For each predecessor, a sequence flow is created to relate it to the gateway.
- If the gateway type is Split gateway then the number of successors should be greater than one and the number of predecessors is equal to one. For each successor, a sequence flow is created to relate it to the gateway.

vii. Name: Add_SequenceFlow

Inputs: Predecessor, Successor.

Description:

A Sequence flow relates two flow objects. The successor and the predecessor should be of type flow object in order not to violate the Sequence flow constraint.

viii. Name: Add_MessageFlow

Inputs: Predecessor, Successor.

Description:

A Message flow relates a resource to a flow object. Thus to meet the Message Flow constraint, the successor or the predecessor should be of type flow object, the other one should be of type resource.

ix. Name: Add_Association

Inputs: Predecessor, Successor.

Description:

An association relates an artefact to a flow object. Thus to meet the Association constraint, the successor or the predecessor should be of type flow object, the other one should be of type artefact.

Rename action

It consists in renaming the concept i.e. it still has the same type, it does just changes the name.

i. Name: Rename

Inputs: Old_Concept , New_Concept.

Description:

Rename (a, b): This function aims to rename the BP model Old_concept with the name of the New_concept. No constraints are attached to this function since neither the successor's type nor the predecessor's type is changed.

Replace action

It consists in replacing the concept by an other with a different type.

ii. Name: Replace

Inputs: Old_Concept, New_Concept.

Description:

This function replaces the concept “a” by another named “b”. As the type of the concept changes, the type of the relations associating it to its successor and predecessor may change. This function examines the type of the replacement to check if it has to change the type of the connector.

Remove actions

Name: Remove

Inputs: Concept.

Description:

Remove actions can not be applied on all cases. Each of the remove actions should make sure not to violate any of the IS constraints. We can remove a concept "a" in two cases:

If a.Predecessor is a flow object and a.Successor is a flow Object or a resource.

If a.Predecessor is a resource and a.Successor is a flow Object.

If a.Predecessor is a flow Object and a.Successor is a resource.

One of the cases where the action Remove can not be applied if the predecessor and the successor are of type resources. In this case two resources will be related to each other.

And we should always adapt the relations based on the concept types.

4.4.3.6 Preventive actions

The ability to provide improvement advices that aim to prevent potential problems is essential to reduce the error probability. These improvement advices are presented under the form of preventive actions.

If the usage of a business process model concept indicates that a possible problem is, we develop a preventive action to warn the potential situation.

Three types of preventive actions are: Concept definition, Concept usage patterns and Concept illustration.

Concept definition

A concept definition displays the role of each concept. It will help the modeller to know what is the adequate concept to use in order to model a specific requirement.

- A sequence flow is used to show the order that activities will be performed in a process. Each sequence flow has only one domain and only one range.
- A message flow is used to show the flow of messages between two participants.
- An association is used to link information and artefacts.
- An activity is an action performed in the process.
- Split_OR gateways are locations within the business process where the sequence flow can take one, two or more alternative paths.
- Split_XOR gateways are locations within the business process where the sequence flow can take only one path from several based on a validate condition.
- Split_AND gateways are locations within the business process where the sequence flow take two or more paths in the same time.
- Join_OR gateways are locations within the business process where at least one path can trigger the activity that succeeds.
- Join_AND gateways are locations within the business process where two or more sequence flows merge in a common path.

- The start event indicates when a business process will start.
- The end event indicates when a business process will end.
- The intermediate event indicates when an event happens between the start and the end event.
- The message intermediate event presents information or an abstract message.

Concept usage explanations

Usage explanations describe the behaviour of a business process model concept. A concept behaviour meets to specific conditions. Thus each concept has to be associated with documentation on how it should be used. The conditions should be respected:

- The usage explanation of an "Activity":
 - Each activity should have an actor responsible of its execution.
 - The name of an activity should begin with a verb.
 - A path should relate a start event to the activity
 - The end event should be reached from the activity.
- The usage explanation of a "Split_XOR":
 - A split_XOR should have a name.
 - A split_XOR should have more than one successor.
 - A split_XOR should have only one predecessor.
 - The output edges should have a name.
- The usage explanation of a "Join_XOR":
 - A Join_XOR should have more than one predecessor.
 - A Join_XOR should have only one successor.
- The usage explanation of an "Actor":
 - An Actor should be associated to at least one activity.
- The usage explanation of a "Resource":
 - A Resource should be associated to at least one activity.
 - A Resource label should not contain a verb.
- The usage explanation of a "Message flow":
 - It connects a flow object to a resource or vice versa.
- The usage explanation of a " Sequence flow":
 - It connects two flow objects.
- The usage explanation of an "Association":
 - It connects an artefact to a flow object.

Illustration Concept

Illustrations are business process templates containing the best practices that show different usages of a specific concept. They contain the knowledge needed to explain how a concept can be applied in a process.

4.4.3.7 Quality improvement patterns

A Business Process Improvement Pattern is an abstract form of a recurring instance of a process modification step used in a Business Process Improvement activity (C. Alexander, Ishikawa, et Silverstein 1977b). Quality patterns use IS constraints and Quality metrics in order to detect the problem and suggest improvements. They capitalize the experience and provide an establish solution to a recurring problem. Thus a quality pattern can use multiple metrics and multiple constraints. Similarly metrics and constraints can be used by multiple patterns.

Based on the results of different metrics and constraints, numerous textual guidelines and corrective actions are proposed for improvements.

Although a pattern can generally be described in any form, we will adopt the problem-solution-context described by Alexander and al. (C. Alexander, Ishikawa, et Silverstein 1977b). This description was used and tested and proved in other fields of application (Forster 2013b):

Name: The pattern name summarizes the pattern objective. It should categorize each pattern.

Context: The pattern context defines the situation and the conditions where the pattern can be applied.

Problem: The problem to be solved is described

Solution: The recommendations or the corrective actions to solve the problem are defined. An implementation guideline may be given.

We defined five improvement patterns: Concept ambiguity, Model completeness, Organizational aspect correctness, Model behavioural correctness and Model semantic correctness patterns.

A. Pattern Name: concept ambiguity

Context:

To check the concept ambiguity with respect to the number of synonyms, hyponyms, and hypernyms present in the domain ontology, as well as to detect unclearness in any concept label.

This pattern is applicable on each concept in the model.

Problem:

- i) If a concept has a big number of synonyms in the domain, that is to say it is not clear then it will create an ambiguity.
- ii) If a concept has a big number of hyponyms in the domain then it may hide a lot of details.

- iii) If a concept has a big number of hypernyms in the domain then it may conceal a lot of details and create an ambiguity.
- iv) If an activity label does not contain a verb, then it does not present any action. The role of the activity is ambiguous.
- v) If a resource label contains a verb, it does not present information, nor an abstract idea nor an actor. The role of the resource is ambiguous.
- vi) If an activity has no name.
- vii) If there is a redundancy of concepts

To check if this pattern is relevant for the current situation:

- a) *Calculate the clarity metric of the concept.*
- b) *Calculate the generality metric of the concept.*
- c) *Calculate the specification metric of the concept.*
- d) *Check if the Activity_label constraint is valid.*
- e) *Check if the Resource_label constraint is valid.*
- f) *Check if the lack_Activity_label constraint is valid.*

Solution:

The following corrective actions can be employed:

- i) Rename the concept with one of the synonyms provided by the ontology.
- ii) Rename the concept with one of the hyponyms provided by the ontology.
- iii) Rename the concept with one of the hypernyms provided by the ontology.
- iv) Refactor the concept label.
- v) Remove a redundant concept.

The model ambiguity is related to the concepts labels, specially the activities labels. The concept ambiguity pattern enumerates many cases that create ambiguity such as redundancy, an incorrect label structure, a label that represents different concepts in the domain, etc.

Ambiguity is detected via quality metrics such as clarity, generality and specification of a concept and through IS constraints such as Activity label, resource label constraints.

Each of the metrics and the constraints provide textual guidelines that ask the modeller to rename a concept and remove redundant concepts.

B. Pattern Name: model completeness

Context: To check the completeness of the model. This pattern is suitable for unfinished and incomplete models.

Problem:

- i. Sometimes models present only a subset of the domain
 - a. Very vague activities
 - b. Activities with no inputs
 - c. Activities with no outputs
 - d. Activities with no actor allocated to it.
- ii. Some concepts attributes may complete the requirement presented in the model.
 - a. Missing X(OR) Split gateway name
 - b. Missing X(OR) Split outgoing edge name
- iii. The lack of basic concepts in the model
 - a. A model with no start event
 - b. A model with no end event.
- iv. To check if this pattern is relevant for the current situation.
 - a. Calculate the Model Richness metric
 - b. Calculate the Resource Richness metric
 - c. Calculate the Input Richness metric
 - d. Calculate the Actor_Allocation metric
 - e. Check the validity of the Start_event constraint
 - f. Check the validity of the End_event constraint
 - g. Check the validity of the X(OR)_Split_name constraint
 - h. Check the validity of the X(OR)_Split_outgoing_edges_name constraint
 - i. Check the validity of the Actor_Allocation constraint
 - j. Check the validity of Activity_Input constraint
 - k. Check the validity of Activity_Output constraint.

Solution:

The following corrective actions can be employed:

- i. Add Activity from the hyponyms, hypernyms or keywords provided by the ontology
- ii. Add Start event
- iii. Add End event
- iv. Add Resource
- v. Add Actor
- vi. Rename a gateway
- vii. Rename an edge.

The model completeness is related to the complete representation of requirements and the presence of the functional concepts in the model. Such as very general activities that mask a lot of details, activities with no resources consumed or produced, activities with no actor, missing gateways name, or missing gateways outgoing edges names, missing start event, missing end event.

Thus to correct these defects, quality metrics can be calculated and IS constraints are checked. Each of these metrics and constraints suggest to the modeller via textual guidelines different corrective guidelines.

C. Pattern Name: model organizational aspect correctness

Context: The model correctness is related to the correctness of the concepts emplacement in the model.

Problem:

- i. Bad emplacement of concepts
 - i. A predecessor of a start event or a successor of an end event.
 - ii. A connector does not reach the extremes or a sequence flow that crosses the boundary of a process
 - iii. A split gateway with less than two outgoing edges.
 - iv. A join gateway with less than two incoming edges.
 - v. An activity without successor or that can not be reached from a start event i.e. lost activity

To check if this pattern is relevant for the current situation:

- vi. Check Start_event constraint
- vii. Check End_event constraint
- viii. Check Reach_extremes constraint
- ix. Check Process_Boundary constraint
- x. Check Split_Outgoing_edges constraint and Join_Outgoing_edges constraint
- xi. Check Activity and gateways constraints

Solution:

- i. Each constraint provides to the modeller corrective guidelines that ask the modeller to use the corrective actions Add, Rename, Replace or Remove.

The organizational aspect correctness of the model is related to the incorrect emplacement of concepts in the models. Such as Start event predecessor, an end event successor, a bad emplacement of connectors, an incorrect use of gateways (number of incoming and outgoing edges) and a lost activity or a bad activity synchronization.

All these defects are detected by IS constraints.

- Start event constraint detects if the start event exists and if it has a predecessor.
- End event constraint detects if the end event exists and if it has a successor.
- Reach extremes constraint detects if there is any connector that does not reach the extremes (domain and range).
- Process_Boundary constraint detects if a connector crosses a process boundary and creates confusion regarding the starting point of the process.
- Split_Outgoing_edges and Join_Outgoing_edges constraints detect if the number of incoming and outgoing edges are incorrect.
- Activity constraint detects if the activity is not reachable from the start event or can not reach the end event.
- Gateway constraint detect if the gateway does not have a name.

Non validated constraints detect a defect and propose to the modeller improvements.

The solutions provided by the pattern through the constraints are corrective guidelines that ask the modeller to:

- Remove any predecessor of a start event or add a start event if it is missing.
- Remove any successor of an end event or add an end event if it is missing.
- Replace a connector if it does not reach the extremes.
- Rename a gateway.
- Add or Remove outgoing/ incoming edges.
- Add new activities that succeed a bad synchronized activity or add a connector from this activity to the end event.

D. Pattern Name: model behavioural correctness

Context: The model behavioural correctness is related to the workflow correctness. Workflow is related to the gateways emplacement in the model.

Problem:

- i. Deadlocks
- ii. Lack of synchronization
- iii. To check if this pattern is relevant for the current situation
- iv. Check the behavioural aspect constraints

Solution:

- i. Deadlock or Lack of synchronization defects are related to the gateways in the model. Thus the solutions provided by the constraints are: Replace the gateway or remove it.

The behavioural aspect correctness of the model is related to the workflow defects in the model, such as Deadlocks and lack of synchronization.

These defects are detected by the behavioural aspect constraints such as D1, D2, PD3, PD4, L5, PL6, UND7 and UND8 defined in Section.3.2.3.2.

The solutions, provided by the pattern through the constraints, are corrective guidelines that ask the modeller to:

- Add a gateway.
- Remove a gateway.
- Rename a gateway.
- Replace a gateway.

E. Pattern Name: model semantic correctness

Context: The model correctness is related to the semantic correctness of the concepts. Semantic defects may be present due to bad concepts emplacement in the model or inappropriate mappings.

Problem:

- i. Resource connected to an other resource or to a gateway
- ii. Incorrect use of Sequence flow
- iii. Incorrect use of Message flow
- iv. Incorrect use of Association
- v. Incorrect use of X(OR) gateway

To check if this pattern is relevant for the current situation:

- a. Check resource constraint
- b. Check gateway constraint
- c. Check sequence flow constraint
- d. Check sequence flow constraint
- e. Check association flow constraint
- f. Check X(OR) constraint

Solution:

- i. The connector's constraints suggest replacing the connector by others more adequate.
- ii. The XOR constraints ask the modeller if the activities splited by the gateway are contradictory. If it is not, the modeller is recommended to replace a gateway by an other.

The model semantic correctness is related to the incorrect semantic emplacement of concepts in the models. Such as communication between resources, communication between a resource and a gateway, incorrect use of a connector and incorrect use of a gateway.

All these defects are detected by IS constraints.

- Resource constraint checks the resource communication.
- Gateway constraints check the gateways communication.

- Sequence Flow, Message flow and association constraints check the type of domain and range.
- X(OR) gateways check the labels semantics of the split gateways successors and the Join gateways predecessors.

The solutions provided by the pattern through the constraints, are corrective guidelines that ask the modeller to:

- Replace a connector.
- Replace a gateway.
- Remove a flow that connects a resource to another or a resource to a gateway.

4.5 Conclusion

In this chapter, we presented our proposed quality evaluation process encompassing quality defects and quality metrics to evaluate the business process models with respect to a domain ontology.

In the syntactic and pragmatic evaluation part, we defined quality metrics that detect syntactic and pragmatic defects respectively.

In the semantic evaluation part, we define quality defects such as ambiguous representation, incomplete representation and meaningless states. Whereas in the second part, we identified new quality metrics for each deficiency.

The strength of our guided process lies in the fact that the quality measurement metrics provide the BP modellers with a quantified evaluation of the quality of their process models regarding a given domain ontology.

In addition, we described different processes involved in the identification/creation of the quality improvement such as textual guidelines, corrective actions and preventive actions. We proposed different types of improvements: textual guidelines, preventive actions and corrective actions. Quality improvement is the process of reducing the number of quality deficiencies. For each quality defect, different quality improvement initiatives are proposed and the impact of these improvements is measured and evaluated.

In the next chapter, we present our software prototype “*BPM-Quality*” that implements our proposed solution and processes. We discuss the architecture of the prototype along with different interfaces available in “*BPM-Quality*”..

Chapter 5

BPM-Quality: Software Prototype Implementing the Proposed Approach

We designed and developed a prototype, “*Business Process Quality*”, which implements our quality approach. This implementation has two core objectives. It first helps in demonstrating the feasibility of the approach. The second is related to the validation of the approach. However, the tool's purpose is to enable the evaluation and the semi-automatic improvement of the Business Process models.

"Business Process Quality" exploits the BP model concepts and the domain ontology concepts. It has an import functionality based on XML allowing the evaluation of the semantic quality of BP models.

5.1 General architecture

(Figure 32) illustrates the general architecture of the solution. “Business Process Model Quality” *BPM-Quality* is able to accept conceptual models designed using any modelling tool that export models into XMI, XML or even XPD. It is important to mention here that XML file contains all the model elements and connectors information.

“*BPM-Quality*” is conceived for two types of user: Quality Experts and modellers. Quality expert is a user having in depth knowledge about quality concepts. He/She is responsible for defining the quality concepts such as quality metrics, BP meta-model _ontology mappings, quality improvements, etc. In contrast the modeller is a normal user who is familiar with modelling notations and is responsible for designing different business process models. Analysts can only evaluate the models employing the quality concepts defined by quality experts.

“*BPM-Quality*” is divided into two core modules: the annotation module and the Domain Knowledge Based Quality Management Module.

- The annotation module where the quality expert is responsible for defining the BP Model – Meta-model mappings.
- The Domain Knowledge Based Quality Management Module where the quality expert is responsible of the quality management by defining the quality metrics, quality improvement guides, etc. Similarly, he/she is also responsible of the domain knowledge management by

defining the meta-models (BP meta-model and Ontology meta-model), defining the mappings rules, etc. And where the modeller is responsible of executing the constraints, quality metrics and improvements.

- The two modules access a common knowledge base. In fact “*BPM-Quality*” contains a knowledge base storing different files as BP notations rules, BP and ontology meta-models, etc. defined by the modeller/analyst. Moreover, the knowledge base also stores the BP models domain quality sessions.

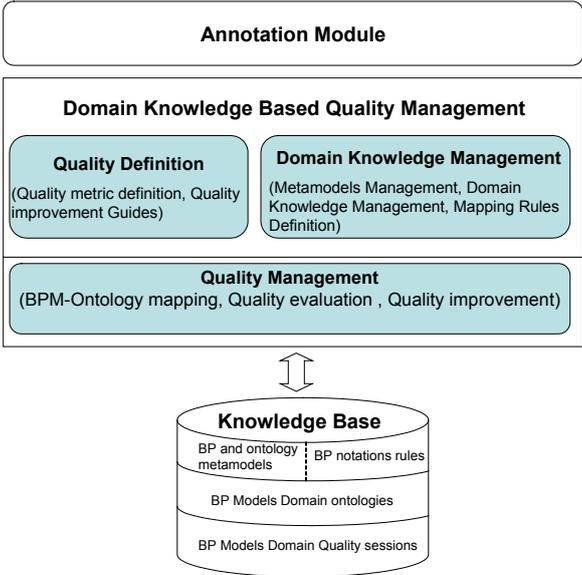


Figure 32: Prototype Architecture

5.2 Functional view for quality expert

(Figure 33) illustrates the systems behaviour with respect to quality expert. It can be noticed that quality experts interact with “*BPM-Quality*” with an aim to define the quality concepts in the following way:

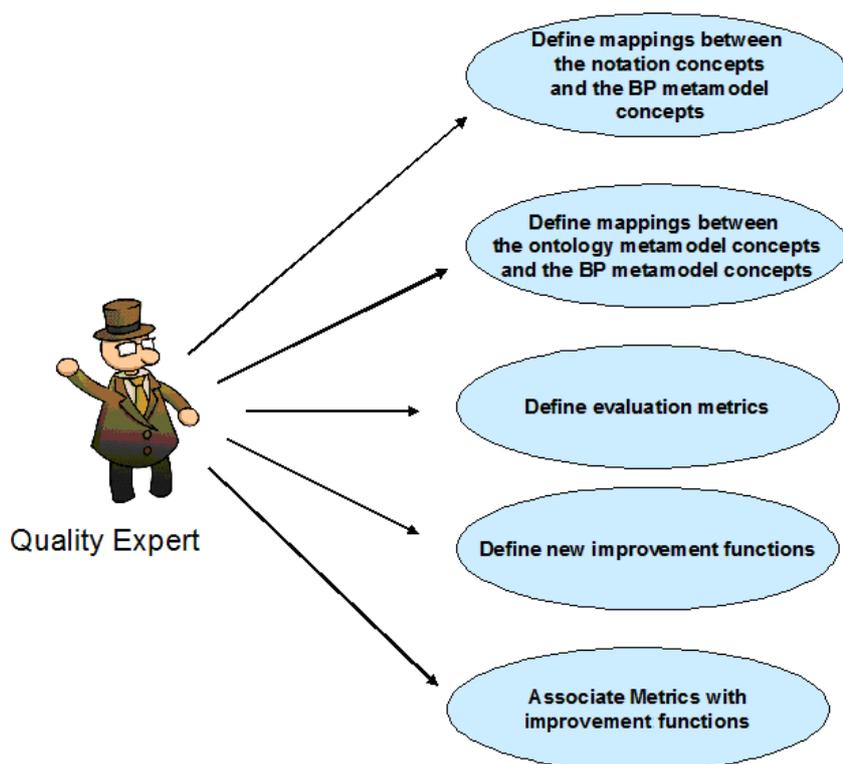


Figure 33: Use Case Diagram: Define the Quality Concepts

- i. To define the mappings between the notation concepts and the BP meta-model concepts. These mappings are used in the annotation module in order to converge all the notations to match our meta-model. The annotation module is explained in details in Section 5.4.
- ii. To define mappings between the ontology meta-model concepts and the BP meta-model concepts. This requires that the notation concepts are already mapped to the BP meta-model concepts. These mappings are used in the Domain knowledge management submodule in order to detect the similarities between the BPM concepts and the ontology concepts. The domain knowledge management submodule is explained in details in Section 5.5.
- iii. To define evaluation metrics. The metrics are used in the Quality definition submodule. The Quality definition submodule is explained in details in Section 5.5.2.1.
- iv. To define new improvements functions. The improvement functions are used in the Quality management submodule. Also, the quality management submodule is detailed in Section 5.5.2.2.
- v. To associate the metrics with improvement functions. As the improvements are provided to the modeller based on the metrics values, mappings between them have to be identified. This requires that the metrics and the improvements are already identified.

5.3 Functional view for modeller

(Figure 34) illustrates the Modeller use case diagram to evaluate quality. This use case describes the systems behaviour with respect to a modeller. It can be noticed that modellers interact with “*BPM-Quality*” to evaluate and improve their models. Their interactions include the following:

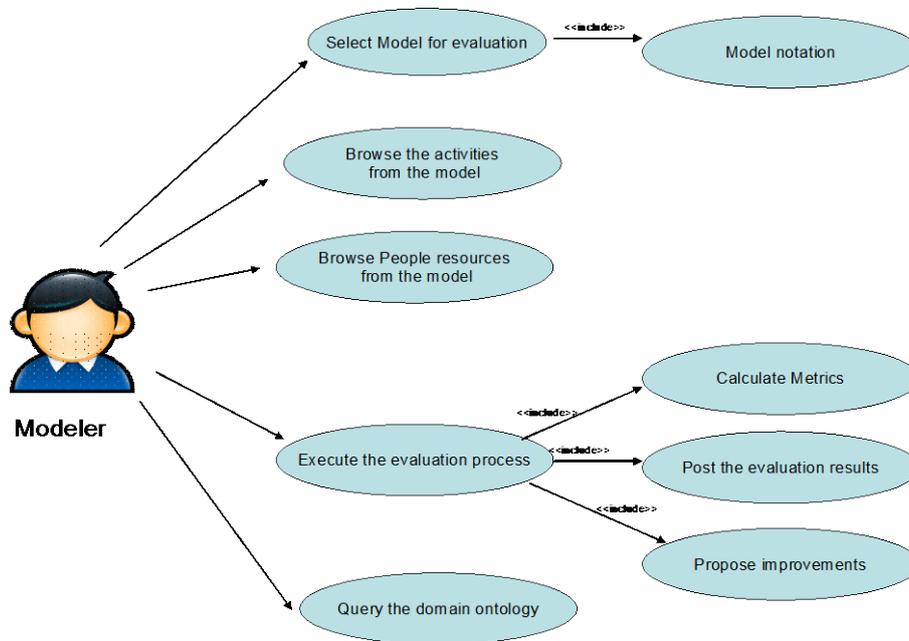


Figure 34: Use Case Diagram: Evaluate the Quality

- i. The modeller selects the target model for evaluation. In order to evaluate it, the model should be annotated by the BPM_Quality annotation module.
- ii. The modeller can browse his/her model's activities
- iii. The modeller can browse his/her model's actor concepts.
- iv. The modeller can execute the evaluation process with respect to each concept selected. In other words, after browsing the activities/actor, the tool will post the evaluation results with respect to the activity/actor selected by the user. The evaluation process includes calculating metrics and proposing improvements. Each metric is associated to an improvement by the quality expert.
- v. The modeller can request the ontology for synonyms, hyponyms, hypernyms or keywords in order to replace the original concept, add new concepts or remove old concepts based on the improvements proposed.

5.4 Annotation module

As our tool aims to support quality evaluation of BP models modelled by different notations and produced by different tools, we aim to reach notation independency. The interoperability model allows supporting quality evaluation of BP models produced by commercial or free BP modelling tools.

Indeed, the current version of the prototype is interoperable with several BP modelling tools such as Bizagi, ARIS or Star UML. These tools use different modelling notations: BPMN for Bizagi, EPC for ARIS, Eriksson and Penker's notation for StarUML etc. And they provide export utilities based on XML or XPD. Moreover, the several BP modelling tools use different notations. To solve this problem, we have developed an annotation module able to deal with several export languages by annotating the exported models to make the BP models compatible with our meta-model.

In fact each notation has its own concepts and its own semantics. We want to converge the diagrams from different notations to match our meta-model. To do so, mappings from each notation concepts to the meta-model concepts are defined based on the concepts semantics and name matching rules. And these mappings are presented by adding tags to the exported diagrams.

In BPMN, a Process describes the function or the work that must be done. A Process in Eriksson-Penker notation is a set of activities designed to produce a specified output (Eriksson et Penker 2012). And in EPC, functions are activities which need to be executed (Jan Mendling, Neumann, et Nüttgens 2005). In our meta-model we represent these different concepts that have the same meaning with a Process. As a result in a XML/XPDL file exported from a BPMN diagram, each Activity is annotated by a "Process". In a XPDL file exported from an EPC diagram, each function is annotated by a "Process". Similarly, an Actor in Eriksson-Penker notation is an actor acting in the process. In BPMN the Pool, Lane and Performer represent the participant in the process. And in EPC, an organization unit is a specific role within the process. All these concepts are represented within an Actor in our meta-model. As a result, in a XPDL file exported from a BPMN diagram, each Pool/Lane/Performer is annotated by an "Actor". The annotation process provides a business process model annotated and compatible with our meta-model.

To summarize, the annotation module, it takes as input the BP model exported in XML, XPDL or XMI and BP notation concepts-metamodel mapping rules the annotation rules to generate a BP model annotated and congruent to the BP meta-model Figure 35 .

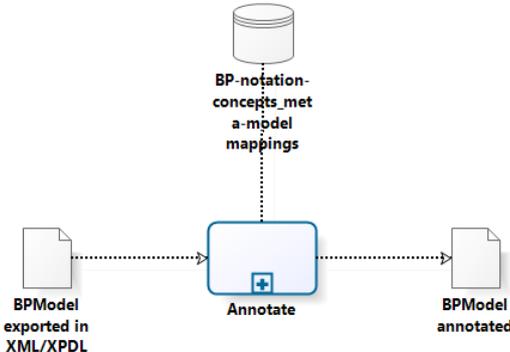


Figure 35: Annotation Process

The mapping rules are defined as Java functions which take one parameter: the tag's type and create a XML comment as shown in Fig. 3.

```
<Activity Id="d6032406-4c4d-4032-b8b3-97ad391607b9" Name="">
  <Description/>
  <Route GatewayType="Parallel" Instantiate="true" ParallelEventBased="true"/>
  <Documentation/>
  <NodeGraphicsInfos>
    <NodeGraphicsInfo BorderColor="-5855715" BorderVisible="true" FillColor="-52" Height="40" ToolId="BizAgi_Process_Modeler" Width="40">
      <Coordinates XCoordinate="1889" YCoordinate="1226"/>
    </NodeGraphicsInfo>
  </NodeGraphicsInfos>
  <ExtendedAttributes/>
  <!--BCType: Gateway Parallel--></Activity>
  <Activity Id="bc056027-54cd-4763-811e-85826136ad3d" Name="Billing">
    <Description/>
    <Implementation>
      <Task/>
    </Implementation>
    <Performers/>
    <Documentation/>
    <InputSets/>
    <OutputSets/>
    <Loop LoopType="None"/>
    <NodeGraphicsInfos>
      <NodeGraphicsInfo BorderColor="-16553830" BorderVisible="true" FillColor="-1249281" Height="60" ToolId="BizAgi_Process_Modeler" Width="90">
        <Coordinates XCoordinate="1221" YCoordinate="708"/>
      </NodeGraphicsInfo>
    </NodeGraphicsInfos>
    <ExtendedAttributes/>
  <!--BCType: activity--></Activity>
```

Figure 36: BPM annotated

5.5 Domain knowledge based quality management module

This module represents the core of the tool. It contains two submodules.

- Knowledge Management Module
 - Application domain knowledge management: relies on the domain ontology.
 - IS knowledge management: relies on the knowledge provided by the notations and the methods.
- Quality Management Module
 - Quality Metrics
 - Quality Improvement guidelines

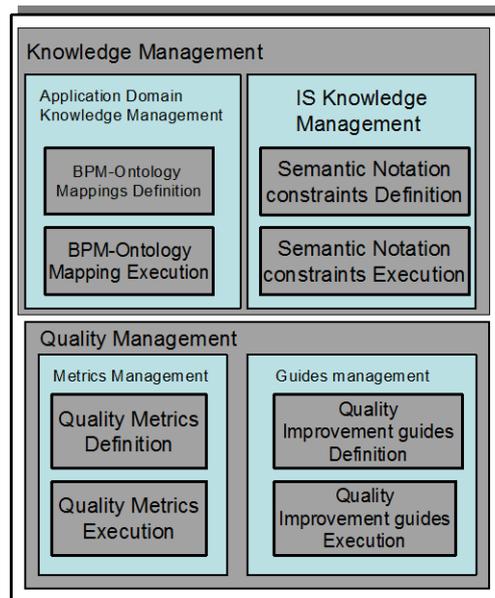


Figure 37: Quality Management Module architecture

5.5.1 Knowledge management submodule

As our approach exploit IS domain knowledge and application domain knowledge, our tool has two subsubmodels: 1) Application domain knowledge management and 2) IS domain knowledge management.

5.5.1.1 Application domain knowledge management

The application domain knowledge management subsubmodule consist of three steps:

- The quality expert defines the mappings between the BP meta-model concepts and the ontology meta-model concepts. Moreover, he/she can use the mappings defined by default. These mappings are stored as Java functions in the Knowledge base.
- After that the mappings are executed, synonyms, hyponyms, hypernyms, keywords and responsibilities are calculated.
- The results are posted to the modeller via an interface. The activity's interface show the synonyms, hyponyms, hypernyms and actor related to the action mapped to the activity (Figure 38). The actor's interface shows the synonyms, hyponyms, hypernyms and responsibilities (Figure 39).

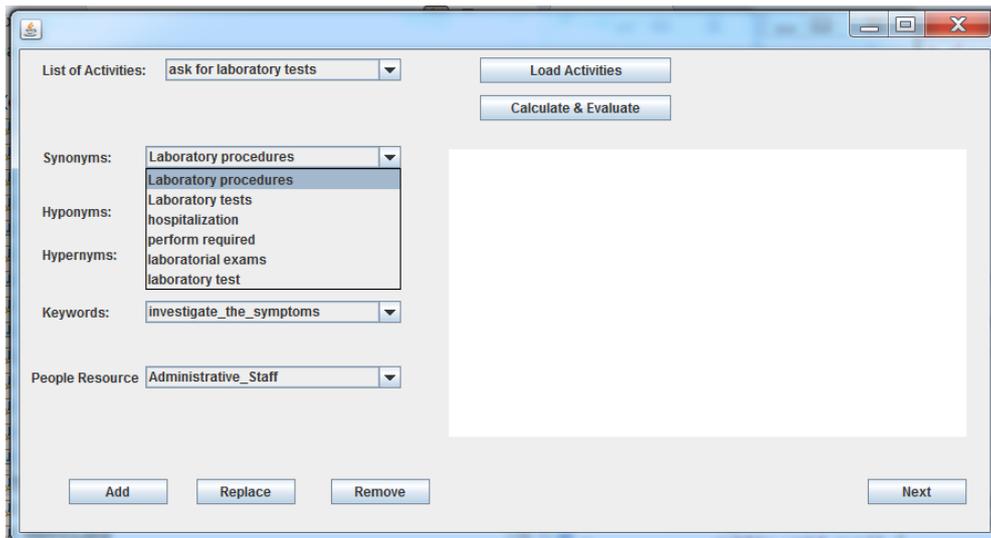


Figure 38: Activity synonyms interface

When the modeller chooses an activity from his/her model (List of activities), the “*BPM-Quality*” tool calculate the mappings between the current activity and the ontology actions. Synonyms, hypernyms, hyponyms, keywords and Actor are displayed. Moreover, quality metrics are calculated and posted in the interface's text Area. The modeller can Add a new activity, he/she can also replace the old activity with a new one and he/she can remove the old activity.

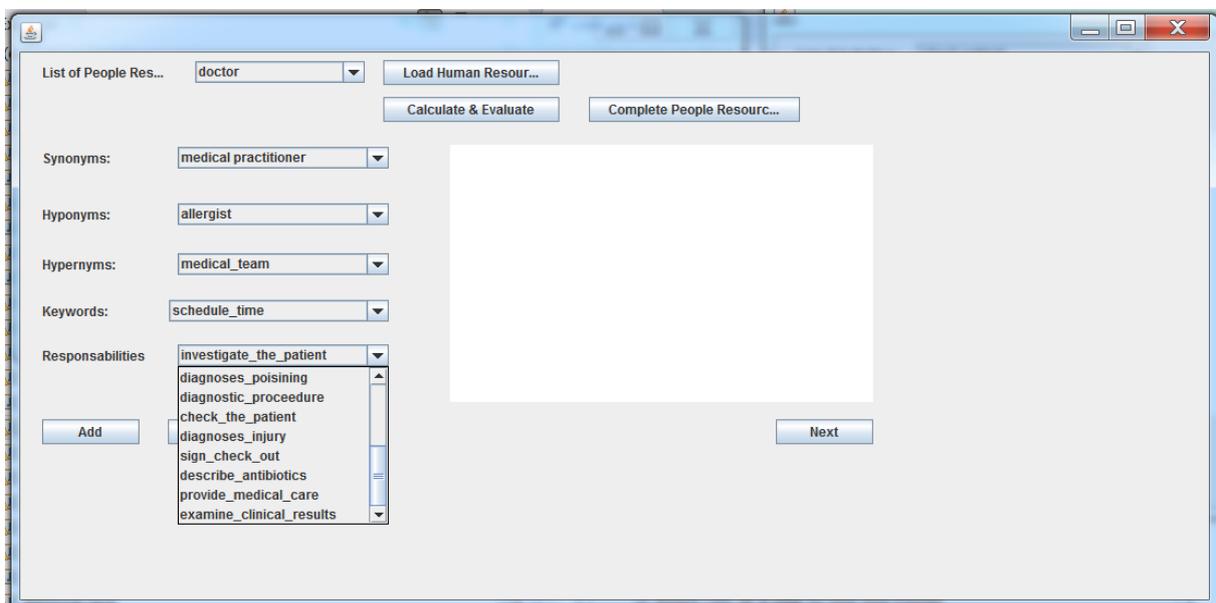


Figure 39: Actor hyponyms interface

Similarly to the activities, when the modeller chooses an actor from his/her model (List of actors), the *BPM-Quality* tool calculate the mappings between the current actor and the ontology actors. Synonyms, hypernyms, hyponyms, keywords and responsibilities are displayed. Moreover, quality metrics are calculated and posted in the interface's text Area. The modeller can add a new actor. He/She can add new responsibilities to the actor that is to add new activities where the latter is

responsible of its execution. Also he/ she can replace the old actor with a new one and he/she can remove the old actor.

5.5.1.2 IS domain knowledge management

The IS domain knowledge management subsubmodule consists of two steps:

- The quality expert defines the IS constraints. He/ She can use the constraints defined by default. The constraints are stored as Java functions in the knowledge base.

```
if(bpm.StartEventList.size() ==0)
{ textArea.append("\n"+ "No start event");
  textArea.append("\n"+" The model should contain at Least one Start
  event");
}
```

- Applying the IS constraints on the model.

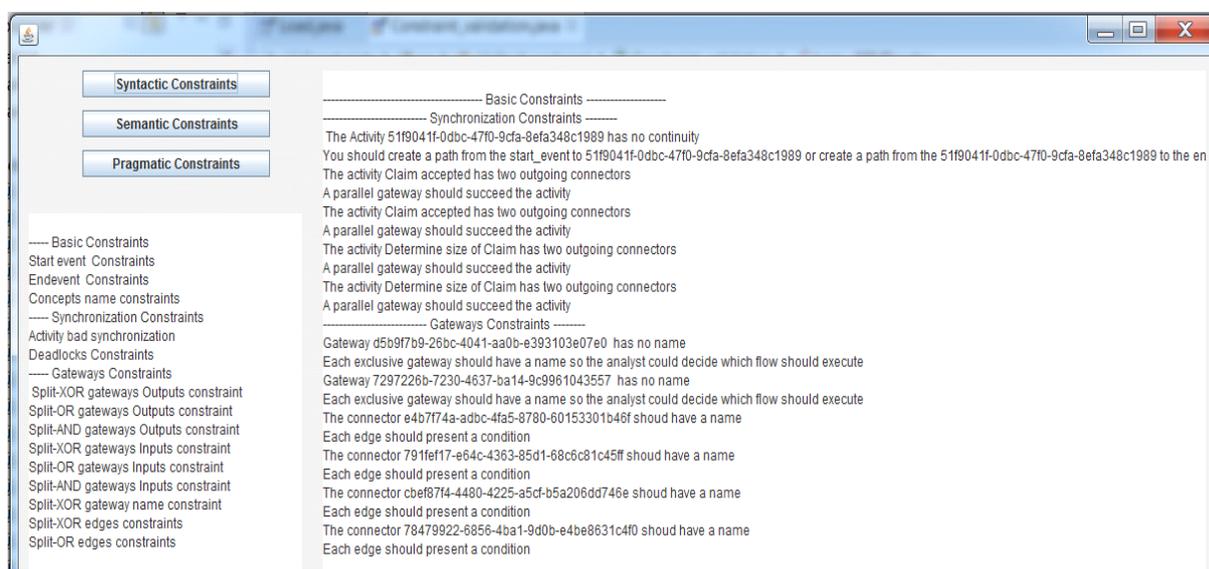


Figure 40: IS knowledge application interface

As explained in Chapter 3, three types of constraints are defined: syntactic, semantic and pragmatic. Thus “*BPM-Quality*” allows the modeller to check each type of these constraints and displays the constraint violated.

5.5.2 Quality management subModule

The quality management submodule in “*BPM-Quality*” is responsible for defining quality metrics and improvements. It is also responsible for executing the improvements chosen and calculating the quality metrics defined. The relationships between the metrics and the improvements are defined in this submodel.

5.5.2.1 Metric management

The Metric management subsubmodule in “*BPM-Quality*” is responsible for quality metrics definition and the execution of these metrics.

- The quality metric definition utility includes a rich set of predefined quality metrics with their definitions. It also offers to the quality expert the possibility to add new metrics as Java functions and document them with definitions.

```
Double Clarity (Concept a)
{
return 1./ a.Synonyms.lenght();
}
```

- The metric execution utility is responsible of identifying the metric functions inputs and posts the calculation results to the modeller.

5.5.2.2 Guides management

Based on the obtained quality metrics values, improvements are proposed to the modeller. The Guide management subsubmodule is responsible for the quality improvement definition such as textual guidelines, corrective actions and preventive actions, the execution of these improvements and defining the relations between metrics, IS constraints and improvements.

- The quality improvement definition utility allows the quality expert to define new improvements and associate them with metrics and constraints. Improvements are proposed according to a given quality threshold of metrics or a constraint violation.
- The quality improvement execution utility is responsible of answering the "If" states of the Java function where it evaluates the results of the calculated metrics. If the metric value is lower/ higher than a specific value, then it proposes the associated improvement function.

5.5.3 Knowledge base

(Figure 41) illustrates the levels of *BPM-Quality* 's knowledge base. It is composed of three different abstraction levels. The highest level contains the Business Process and the ontology meta-models, BP-ontology mapping rules and annotation mappings rules. The intermediate level is dedicated to quality metrics, semantic notation constraints and improvement guidelines. Finally, the lowest level stores the results of the improvement sessions.

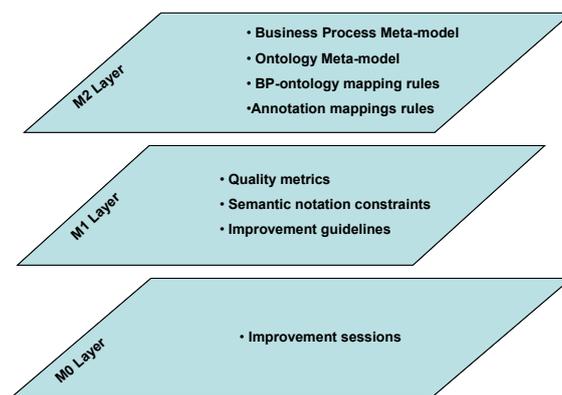


Figure 41: Knowledge base Structure

5.6 Quality evaluation in BPM-Quality

The scenario presented in this section illustrates how the “*BPM-Quality*” can be used to evaluate and improve the models. The quality definition module is conceived for quality experts only since it requires in depth knowledge about quality concepts. However as mentioned before, we provide two types of mechanisms to accommodate both basic analysts and quality experts for quality evaluation module. Quality experts can skip the metrics and the improvements definition and directly start the evaluation process by selecting the quality patterns or quality metrics from the knowledge base. However, it requires that they are aware of all the existing contents of the knowledge base and what each concept proposes.

The quality evaluation process starts by selecting the model to be evaluated. The model must first be exported to the XML file format. XML (XPDL) has widely been used in the industry for enhancing the portability of business process models among different modelling tools. Our prototype is based on XML. Thus it can be used to evaluate models designed using any modelling tool capable of exporting its models into XML. Thus, the user selects the model to be evaluated. Once the model is selected, the *BPM-Quality*'s module annotation, annotates the exported model and produces an XPDL file annotated. Once the XPDL file is annotated, the user is proposed the following six options:

- i. Formulate new quality metrics: create new Java functions (for quality experts only).
- ii. Use existing quality metrics i.e. the quality metrics existing in the Knowledge base and which were previously used for evaluation.
- iii. Formulate new quality constraints: create new Java functions (for quality experts only).
- iv. Use existing quality constraints i.e. the quality constraints existing in the Knowledge base and which were previously used for evaluation.
- v. Formulate new quality improvement procedures: create new Java functions (for quality experts only).

- vi. Use existing quality improvement procedures i.e. the quality improvement procedures existing in the Knowledge base and which were previously used for evaluation.

This scenario doesn't include the housekeeping of the knowledge base i.e. it doesn't discuss about the insertion, modification or deletion of quality patterns, quality metrics and constraints to the knowledge base. It uses the existing contents of the knowledge base for evaluation and propositions.

Step-1: Validation of IS notation constraints

After that the model is annotated, IS notation constraints are checked. The modeller can choose to check the syntactic constraints, the semantic constraints and the pragmatic constraints respectively. Each constraint violated is posted to the modeller with improvement textual guidelines.

In the second step, the modeller applies manually the guidelines on his/her model. He/She can recheck the constraints in order to verify the non-violation of previous constraints.

Step-2: Model evaluation and improvement

(Figure 38) displays the screen of the application domain knowledge exploitation. The first combo box on the screen contains all the activities seized from the model. Upon the selection of the activity to evaluate:

- The second combo box will list all the chosen activity synonyms provided by the ontology.
- The third combo box will list all the chosen activity hyponyms provided by the ontology.
- The fourth combo box will list all the chosen activity hypernyms provided by the ontology.
- The fifth combo box will list all the chosen activity keywords provided by the ontology.
- The sixth combo box will list all the chosen activity actor provided by the ontology.

A text area will post all the quality metrics results and the textual guidelines associated to the metrics. Thus based on the textual guidelines, the modeller applies the corrective actions recommended.

- The modeller can replace the current activity with one of its synonyms/ hyponyms/ hypernyms by choosing the synonym from the synonyms/ hyponyms/ hypernyms list respectively and clicking on the "Replace" Button.
- The modeller can add a new activity by choosing an activity from any list and clicking on the "Add" Button. By clicking on the Add button, a new interface will be created to identify the successor and the predecessor of the new activity.
- The modeller can remove the current activity by clicking on the "Remove" Button.

- The modeller can add resources. He/ She chooses a resource from the keywords lists and by clicking the "Add" button a new interface will be created to choose the successor and the predecessor of the resource.
- Finally the modeller can add a new actor by choosing one from the actors list. It will create a new actor but it's up to the modeller to replace manually the activity. And this disadvantage is due to the fact that the XML file does not provide the capability to attach an activity to a pool or a lane in BPMN. Thus if it is in EPC or Eriksson-Penker notation, the new actor will be added as any other resource by identifying the successor and the predecessor.

The same step is done for the actor evaluation. As the "*BPM-Quality*" tool provides the same interface but for the actors seized from the model. The modeller can always re-evaluate the transformed model. And by re-evaluating the modeller new synonyms, hypernyms, hyponyms and keywords will be provided as there are new concepts added to the model.

5.7 Comparison of BPM-quality with other existing software

In Chapter 2, we used (Table 16) to compare existing evaluation approaches. Now we will use the same criteria and some others to compare our prototype with respect to existing evaluation tools.

Product Criteria	Bizagi	Star-UML	Visual Paradigm (Logizian)	Aris	Méga	Woflan	BPM-Quality
For use by business users or engineers?	Business	Engineers	Engineers	Business and engineers	Business and engineers	Business and engineers	Business and engineers
Does it give preventive improvement guidelines?	yes	no	no	no	no	no	yes
Does it provide BPM evaluation?	yes (Syntactic evaluation)	no	yes (Syntactic evaluation)	yes (Syntactic evaluation)	yes (Syntactic evaluation)	yes (Syntactic evaluation)	yes (Syntactic, Semantic and Pragmatic evaluation)
Granularity of evaluation criteria	Metrics	Metrics	Metrics	Metrics	Metrics	Metrics	Metrics, Quality patterns and IS constraints.
Possibility to Add/edit evaluating criteria	no	no	no	no	no	no	yes
Does it provide improvement guidelines?	no	no	yes	no	yes	no	yes
Does it give the possibility to define improvements?	no	no	no	no	no	no	yes
Adopt which	BPMN	Eriksson-Penker And AD	BPMN, EPC, DEF	EPC	BPMN	Petri_nets	All Business process

notation?							modelling notations.
-----------	--	--	--	--	--	--	----------------------

Table 16: Comparing BPM-Quality with existing evaluation approaches

Following findings can be deduced from (Table 16):

- i. Only “*BPM-Quality*” (our prototype) provides syntactic, semantic and pragmatic evaluation metrics. Woflan provides a workflow evaluation to detect deadlocks and lack of synchronization but it does not evaluate the semantics nor the pragmatic quality of the EPC model. Méga evaluates the BPMN model structure and checks the BPMN standard violation, also it do not evaluate the semantic nor the pragmatic quality of the model. Similarly to Mega, Logizian checks the structure of the BPMN model. It does not evaluate the semantic nor the pragmatic quality of the model.
- ii. Only Visual paradigm and Méga provide improvement guidelines. But their guidelines are restricted to the BPMN standards and they do not cover all the notation defects. In contrast to “*BPM-Quality*” which covers a big number of quality defects and provide different types of improvements (textual guidelines, preventive actions and corrective actions).
- iii. Only “*BPM-Quality*” provides the possibility to define quality metrics using the Quality Metrics definition subsubmodel.
- iv. Only “*BPM-Quality*” provide the possibility to define quality improvements using the Quality improvement guides definition subsubmodule. It also allows the quality expert to map the metrics to the improvements provided.
- v. Only Bizagi and “*BPM-Quality*” provides preventive improvement guidelines. When the modeller chooses a concept, a definition of the current concept is posted. But “*BPM-Quality*” do post an addition to the definition, the concept usage explanation to prevent from emplacement errors.
- vi. Only “*BPM-Quality*” supports the evaluation and the improvements of models from a big number of notations.

In view of the above findings, we can say that “*BPM-Quality*” avoids several shortcomings present in other existing evaluation applications. Moreover, “*BPM-Quality*” can be used for several notations.

5.8 Conclusion

In this chapter, we presented our software prototype “*BPM-Quality*” that implements our proposed solution and process. We started by illustrating the application architecture and we detailed each of the

modules. We took examples to demonstrate the flow of “*BPM-Quality*”. Following are the limitation of “*BPM-Quality*”:

- i. “*BPM-Quality*” evaluates the complete model and proposes recommendations for the entire model thus if user is interested in evaluating only certain parts of the model then he can only do that by turning those parts into an independent model.
- ii. Some of the improvements suggested by the tool as the textual guidelines can not be applied automatically, the modeller should apply them on his model and re-evaluate it by the tool.

In the next chapter, we present the two validation experiments conducted for our approach. In the first experiment, we validated the selection of the IS notation constraints whereas in the second experiment, we validated the efficacy of our proposed quality approach including quality metrics, quality improvement patterns.

Chapter 6

Validation

The main objective of our validation is to evaluate the approach and show its value. We conducted two experiments to validate our proposed approach at different stages.

Thus, a first experiment was performed to ensure that the quality constraints defined in Section 3.2.3 in order to detect a big number of the important quality defects in the evaluation of Business process models. This interim validation exercise was performed on a collection of Business process models from BPM Academic initiative (« BPM Academic Initiative | Signavio » 2013). The main objectives behind this first validation were the following:

- i. To ensure that these constraints detect a big number of the important quality defects.
- ii. To make sure that the concepts are well used in the model.
- iii. To study the percentage of the more common defects.

The results of the above experiment helped us in validating the selection of our quality constraints and in correcting the usage of the BPM concepts.

We performed the second experiment to validate the efficacy of our complete approach including domain ontology. We have carefully selected the respondents for the validation experiment such that all the respondents have prior knowledge about conceptual model and some modelling experience. This experiment consisted that every respondent model the distributed scenario.

The main objectives behind this second validation were the following:

- i. To make sure that the quality metrics identified are useful in evaluating the quality of the BPM
- ii. To ensure that the knowledge provided by the domain ontology can provide improvements to the model.
- iii. To study the mostly used improvements.

The second experiment helped us in validating the strengths and benefits of employing our proposed approach in evaluating and improving the BP model. In both experiments, we have used multiple charts and tables to analyze the most spreading defects. We were mostly interested in how to detect and correct these defects. In the next sections, we describe the two experiments and analyze the results.

6.1 Validating the selected quality constraints

In the domain of BPM quality, researchers have proposed different constraints for evaluation but the main problem lies in their disparity and non-converging solutions. Moreover there is not any framework that proves the completeness of the constraints identified i.e. if they cover all possible defects or not. We aim to apply the identified constraints on a big number of models so that we can try to cover as many quality defects as possible. In order to do that, we applied them on a collection of models provided by BPM Academic initiative (« BPM Academic Initiative | Signavio » 2013).

- i. We categorized the defects into syntactic, notation semantics and pragmatic constraints.
- ii. We studied the percentage distribution of these defects.
- iii. We analyzed the causes of the most common defects.
- iv. We compared the defects detected by our approach and those detected by two other open source tools.

In total 100 Business process models extracted from the collection provided by BPM Academic initiative were contacted to complete our validation. The advantage of this collection is that we have a big number of models from different domains but the disadvantage is that we don't have any feedback on the modeller's profile. These models contain connector base, notation semantic and pragmatic defects detected by our constraints. The classification of the defects is based on the classification of the IS constraints that detect them. In fact, they were classified in the same way as the IS constraints: into connector base, notation semantic and pragmatic

We have been able to perform an analysis of all models and detect the number of the discussed defects. Thus we quantified the defects errors.

(Table 17) provides the classification of the different types of defects.

	Defect type		The number of defected models
Syntactic	Lost Activity	In every state that is reachable from a start event, there must be the possibility to reach an end event.	60
	Misplaced Activity	Each activity should reach the end event.	21
	Waste Activity	Each activity should have at least two relationships.	2
	Improper use of flow elements	A connector that does not reach the extremes has no role.	0
	Start event bad emplacement	There shouldn't exist any predecessor for a start event.	0
	End event bad emplacement	There shouldn't exist any successor for a start event.	0

	Bad Activity Synchronization	An activity cannot have two activities as successors	7
Notation Semantic Quality	Bad use of gateway	<ul style="list-style-type: none"> • A XOR-Split gateway should have at least 2 outgoing edges. • An OR-Split gateway should have at least 2 outgoing edges. • An AND-Split should have at least 2 outgoing edges. • An AND-Join Gateway must have at least two incoming edges. • An OR-Join Gateway must have at least two incoming edges. A XOR-Join Gateway must have at least two incoming edges.	13
	Unlabeled Split X(OR) gateways outgoing edges	Each Split XOR/OR outgoing edge should have a name presenting a condition.	23
	Useless Pool / Lane	A pool or a lane with nothing associated to it is useless.	5
	Ambiguous activities	Activities can not be placed on the Pool / Lane boundaries	0
	Gateway-Resource communication	Gateways should not receive or send any resources.	0
	Deadlock	Deadlock occurs when there is no continuity	9
	Lack of synchronization	It occurs when there are two active paths and only one should continue.	14
	Unlabeled activity	An activity with no name is a useless activity	17
	Process boundary crossed	If the process has a start event the sequence flow should not cross its boundaries.	1
	Unlabeled Split X(OR) gateway	Each Split XOR/OR gateway should have a name presenting the split event.	43
	Incorrect use of Sequence flow	A Sequence flow relates flow objects.	13
	Incorrect use of Message flow	A message flow relates activities to resources and vice versa.	47
	Incorrect use of Association	An association relates artefacts to flow objects or to resources and vice versa.	4
	Resources communication	A flow that relates two resources is wrong. There is no communication between two resources.	6
	Incorrect use of X (OR) gateway.	A XOR gateway presents an 'either' situation. An OR gateway represents an "AND/ OR"	2

		situation.	
	Incorrect outgoing edges labelling	The XOR outgoing edges should present opposite conditions.	4
	Incorrect use of timer events	The modeller should differentiate between the activity duration event and the delay event.	0
	Lack of actor	Each activity should have an actor responsible of its execution.	26
	Lack of Inputs / Outputs	Each activity should have an input and an output.	90
Pragmatic Quality	Process without Start event	Each business process model has to contain one start event.	9
	Process without End event	Each business process model has to contain at least one end event.	11
	Process with Multiple Start events.	Each business process model has to contain one start event.	23
	Bad structure of the activity label	An activity name should begin with a verb.	28
	Bad structure of the Actor label	An actor name cannot contain a verb.	2
	Redundancy defect	The presence of derived or redundant concepts makes the model less understandable	19

Table 17: Categorization of the defects

(Table 17) sum up the defects with respect to the number of models where the defect has been detected. Based on these results, we aim to analyze the causes of the most common process modelling defects.

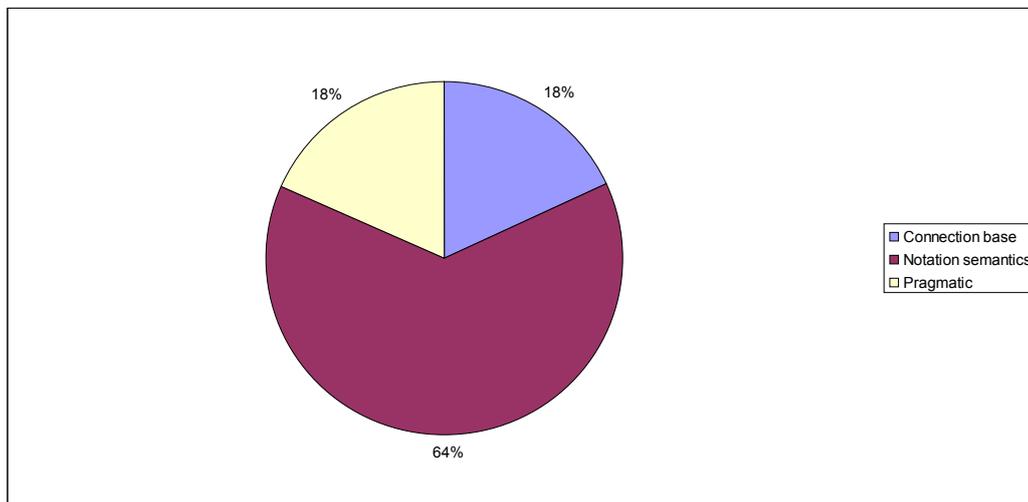


Figure 42: Defect types percentage

(Figure 42) gives a good indication that in this model collection, the notation semantic defects present the higher percentage (64%) followed by the syntactic and pragmatic defects (18%). These results confirm the studies in the literature. This percentage distribution is mainly due to the classification of

defects as a high number of defects are classified as notation semantic. Besides, the literature studies confirm our analysis. First Mendling and al. (Jan Mendling et al. 2012) used a collection of 429 EPC models of an Australian financial institution for validation. Based on their validation results, they tabulated the probabilities of finding different types of errors. The types of errors they studied, based on our classification, are notation semantic errors. In fact, they studied the size, connection, modularity, connector interplay and complex behaviour and they concluded that the complex and the big size models represent the highest percentage.

On the other hand, Rozman et al. (Tomislav Rozman 2009) analyzed the most common process modelling mistakes in BPMN process models and they found that :

- 1) Bad activity synchronization represented by not connected activities and hanging concepts i.e. not reachable from the start event are 34% of the errors.
- 2) Processes without start event are 4% of the errors.
- 3) Processes with multiple start events are 6% of the errors.
- 4) Processes without end event are 13% of the errors.
- 5) Incorrect use of sequence flows are 16% of the errors.
- 6) Incorrect use of associations is 1% of the errors.
- 7) Incorrect use of message flow is 3% of the errors.
- 8) Misplaced concepts are 5% of the errors.
- 9) Improper use of flow elements are 18% of the errors.

Gruhn et al. (Gruhn et Laue 2010) present a new heuristic approach for finding errors. They define patterns related to a violation of soundness property or bad modelling style. To validate their approach, they analyzed almost 1000 models. For this purpose, they used three different model checkers tools. The results of these tools have been compared with the results given by their approach. 534 of the models were detected for soundness errors i.e. almost 50% of the models. The most remarkable errors are related to bad modelling style. That is from the 443 OR-joins in the models, no less than 106 should be replaced by an AND- or XOR-join which can model the given situation more precisely. They also realized that OR-connectors are used too often.

Back to our model collection validation, (Figure 43) depicts the partition of notation semantic defects in our study. It can be seen that the majority of the notation semantic defects belongs to the lack of inputs/outputs (28%), Incorrect use of message flow (15%), followed by unlabeled split X(OR) gateway (14%).

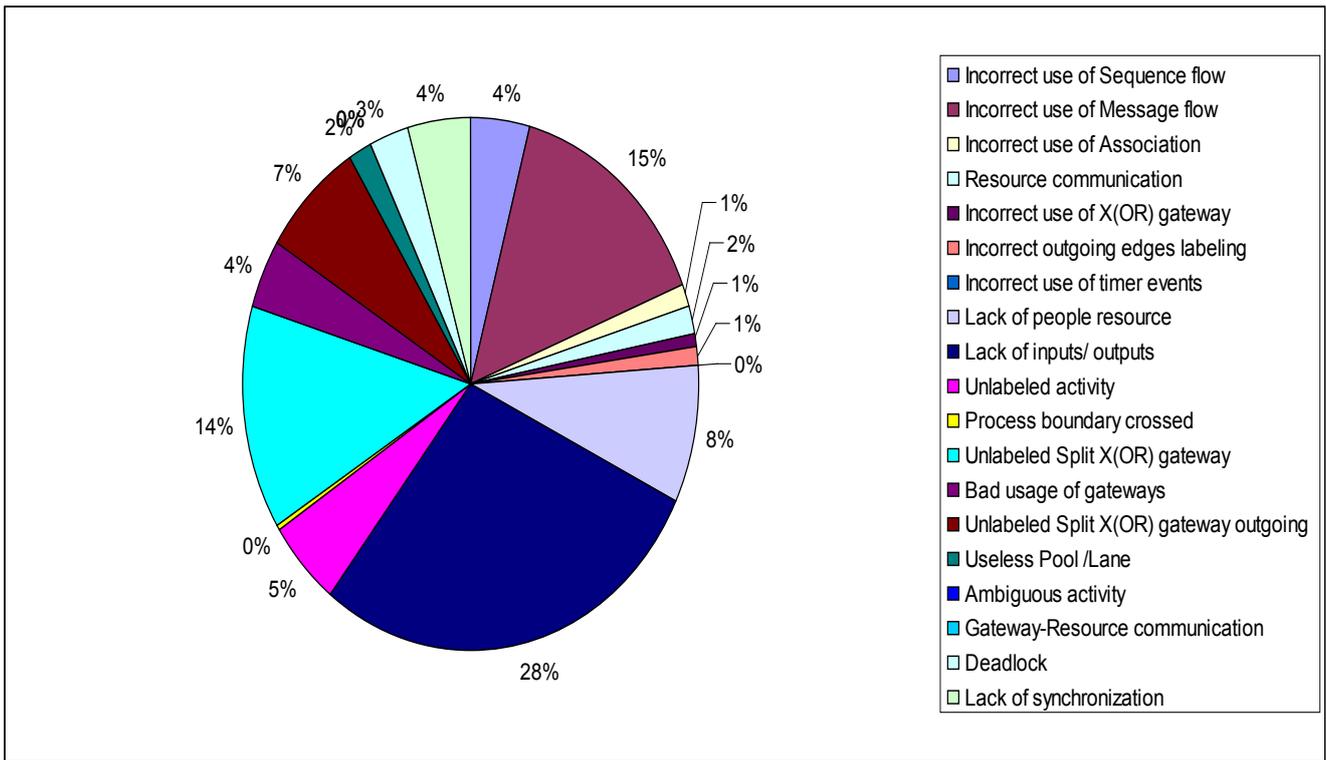


Figure 43: Notation semantic defects percentage distribution

Lack of inputs and outputs in the model may be the result of various causes:

- i. None of the models prevent the modeler to add resources (objects, message events..) into their models.
- ii. For the modeler, the objective of a business process model is to describe the order of execution of activities. They do not concentrate on the presence of resources.

The common causes of the message flow constraint are:

- iii. Some of the modelling tools detect this defect but none of them explains the cause.
- iv. None of the notations gives a detailed explanation of the message flow's semantics. Most of them define it as the messages exchange representation without specifying how it should be applied in a model.

The unlabeled Split XOR/OR gateway defect may be the result of various causes:

- i. None of the modelling tools prevents the modeller that a Split-XOR/OR gateway should be labelled.
- ii. For the modeller, the type of gateway used is sufficient to present the idea. The Split-XOR gateway should represent the question that should be answered in order to know what is the valid condition that triggers the next activity.

The lack of actor allocation is due to the fact that:

- i. None of the modelling tools oblige the modeller to define the actors, they allow the modellers to begin their models without identifying the actors (lanes, actors...).
- ii. An interesting number of modellers define the actor in the activity label.

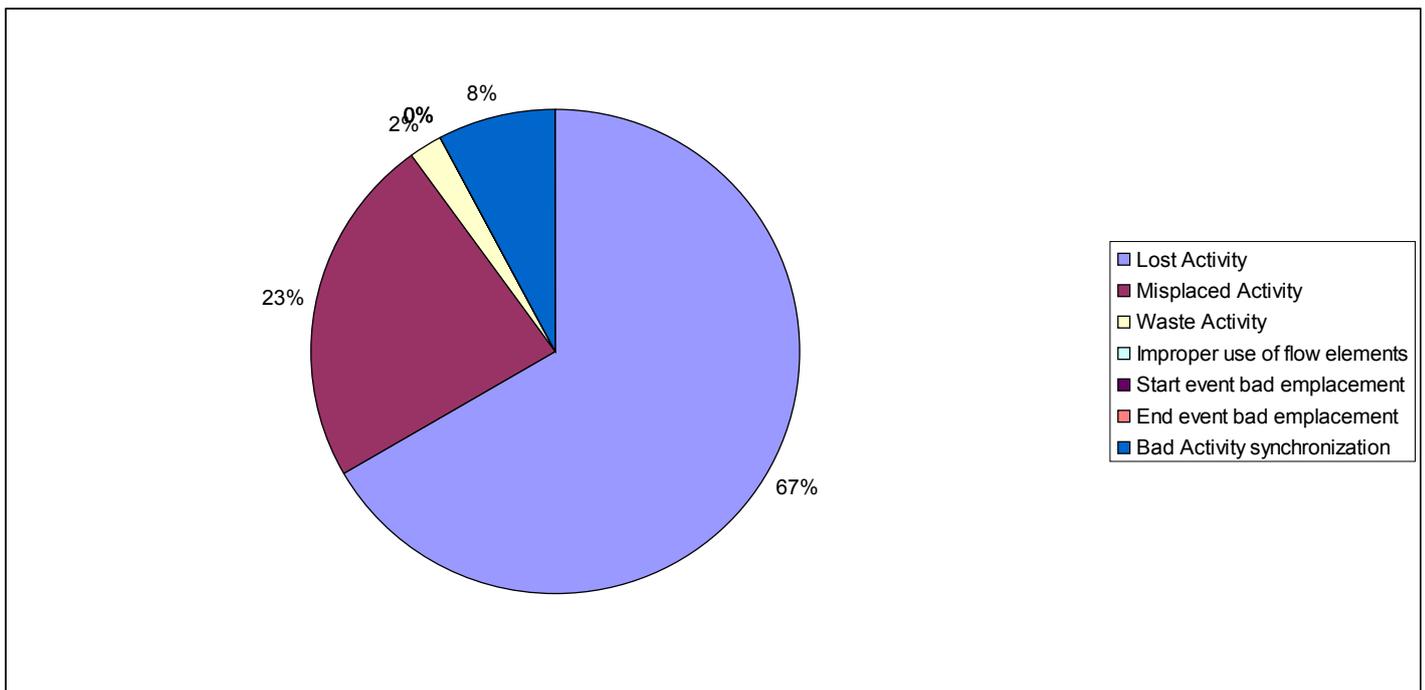


Figure 44: Syntactic defects percentage distribution

(Figure 44) depicts the partition of syntactic defects. It can be seen that the majority of the syntactic defects belongs to the lost activity (67%) followed by misplaced activities (23%).

The lost activity defect may also be the result of various causes:

- i. The modeller does not double check on his/her model after finishing it.
- ii. The modeller concentrates on the sequence of activities in a specific requirement rather than in the scenario as a whole.

On the other hand, based on the same causes, a misplaced activity is followed.

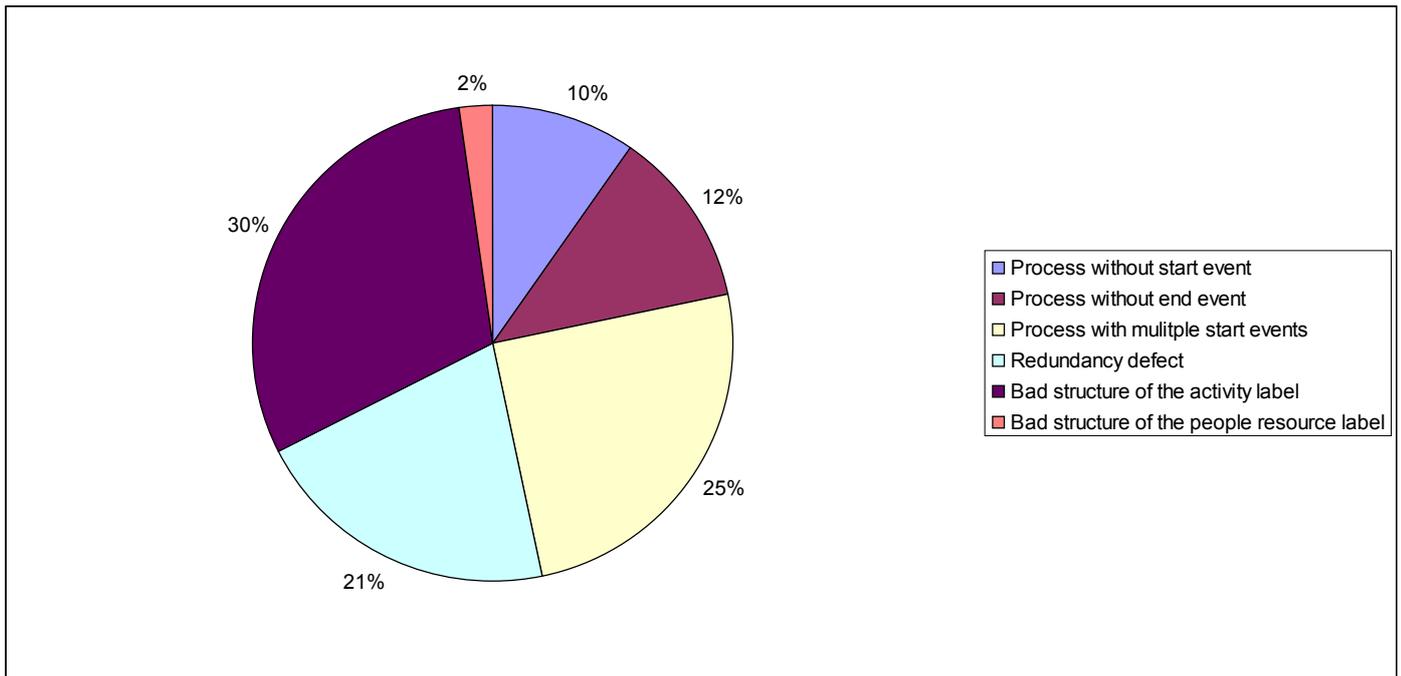


Figure 45: Pragmatic errors percentage

Finally, (Figure 45) depicts the partition of pragmatic defects. It can be seen that the majority of the pragmatic defects belongs to the bad structure of activity labels (30%), Processes with multiple start events (25%), followed by the redundancy defect (21%).

The activity label defect may be due to:

- i. None of the modelling notations defines what should be the activity label structure.
- ii. The modeller can name the activity anything he/she wants in all the modelling tools.

Process with multiple start events may be due to:

- i. For some notations, processes can be triggered by multiple events.
- ii. The modeller want to describe the real-world case, but only in BPMN, there exists multi types of start events that describe different ways for triggering a process.

The redundancy defect may be due to:

- When an activity occurs twice in a scenario, it is much easier for the modeller to recreate the activity rather than search for the old one and reconnect it.
- When the same flow object is reused many times, the model became unreadable as the connectors overlap.

As mentioned before, these errors have been found using business process models modelled in BPMN. When we applied the IS constraints, an interesting observation from these first results is that the highest percentage of errors is dedicated to the notation semantic errors, which may be due to the big number of defects categorized as notation semantic.

However, the percentage distribution of the defects is related to the modelling language, because the latter does have an influence in the occurrence of error types. For example, in BPMN it is common to have defects related to the connector's semantics, or in EPC there is no such type of error because there is only one type of connectors. Additionally, deadlock, livelock and lack of synchronization are more common in EPC than in BPMN.

6.2 Quality evaluation

In order to compare our heuristic results to the results of the existing evaluations in the literature, we selected four well-known tools that check business process models for the modelling defects: Logizian, Mega, Blueworks and Bizagi. All these four tools support business process modelling with BPMN. After doing the analysis with the different tools, we compared the results.

- Visual paradigm, Logizian simulacion 10.2: It is BPMN 2.0 Business workflow design software. Logizian identify and Fix Design issues with model quality checker. When running the model quality checker, a series of checking are performed on every model element in the project. When it finds a design issue, marks are deducted from that model element (« Logizian 10.2 » 2013).
- IBM Blueworks Live: Blueworks Live combines features of WebSphere Lombardi Blueprint and IBM BPM Blueworks into one online cloud-based BPM tool that enables collaboration and process automation. Blueworks implement modifications adapted on the process model. It does not allow the modeller to misplace the concepts which prevent any syntax error (« IBM Blueworks Live » 2013).
- Mega Process: It provides capabilities for modelling, documenting, and improving business processes and organizational structures. It analyse the information flows and detects structure errors. Mega can enforce rules to make sure that the main modelling standards are respected when the modeller describe his/her model. BPMN standards are activated so that it can show which BPMN standard is broken (« MEGA Process » 2013).
- The Ford tool, Bizagi: It is business process management software which provides leading process automation and workflow automation solutions. After the modeller finishes describing his/her model, Bizagi checks the model structure and posts the defects detected (« Bizagi » 2013).

Defect subtype	Blueworks	Logizian	Mega	Bizagi	BPM_Quality
----------------	-----------	----------	------	--------	-------------

Process without Start event	Y	N	Y	N	Y
Process without End event	Y	N	Y	N	Y
Process with Multiple Start events.	Y	N	N	N	Y
Start event bad emplacement	Y	Y	Y	Y	Y
End event bad emplacement	Y	Y	Y	Y	Y
Unlabeled activity	Y	N	Y	N	Y
Lost Activity	Y	Y	N	N	Y
Misplaced Activity	Y	Y	N	N	Y
Waste Activity	Y	Y	N	N	Y
Improper use of flow elements	Y	Y	Y	Y	Y
Process boundary crossed	Y	N	N	N	Y
Unlabeled Split X(OR) gateway	Y	N	Y	N	Y
Bad Activity Synchronization	Y	N	N	N	Y
Bad use of gateway	Y	N	Y	N	Y
Unlabeled Split X(OR) gateways outgoing edges	N	N	Y	N	Y
Useless Pool / Lane	N	Y	N	N	Y
Ambiguous activities	Y	N	N	N	Y
Activity with two successors	Y	Y	N	N	Y
Gateway-Resource communication	N	N	N	N	Y
Deadlock	Y	N	N	N	Y
Lack of synchronization	Y	N	N	N	Y
Incorrect use of Sequence flow	N	N	N	N	Y
Incorrect use of Message flow	N	N	N	N	Y
Incorrect use of Association	N	N	N	N	Y
Resources communication	Y		N	N	Y
Incorrect use of X (OR) gateway.	N	N	N	N	Y
Incorrect outgoing edges labelling	N	N	N	N	Y
Incorrect use of timer	N	N	N	N	Y

events					
Lack of Actor	Y	Y	N	N	Y
Lack of Inputs / Outputs	N	N	N	N	Y
Bad structure of the activity label	N	N	N	N	Y
Bad structure of the Actor label	N	N	N	N	Y
Redundancy defect	N	N	N	N	Y

Table 18: Tools comparison

After doing the analysis with the different tools, the comparison between our tool results and each of the four tools results showed that:

- The syntactic and some of the semantic notation defects were detected by Blueworks, almost as well as our tool. Nevertheless, it does not detect the unlabeled X(OR) gateways outgoing edges, the useless pools and lanes nor the gateway-resource communication. In fact Blueworks is a Pre-evaluation tool, it enforces the modeller to add a limited number of concepts in a specific position which will prevent a lot of syntactic and semantic notation defects. On the other hand, it does not allow the modeller to create a path between two gateways thus it prevent from any deadlocks and lack of synchronization. Moreover, Méga detect the standard BPMN defects. It is a Post-evaluation tool, any defect that broke a BPMN standard is detected. It also provides recommendation on how to correct the defect. But the BPMN standards are not valid to the other notations. Also some defects detected by our tool are not included in the BPMN standards, such as multiple start events, misplaced activity, waste activity, etc... Finally, Bizagi detects the basic BPMN standards such as the start event and the end event bad emplacement by not allowing the modeller to create them in incorrect places.

To conclude, Blueworks and Bizagi do not provide any recommendations in order to explain why the modeller is enforced in a specific list of concepts but Méga provides some of the BPMN recommendations. “*BPM-Quality*” tool detects all the syntactic and semantic notation defects defined above with improvements (corrective actions, preventive actions and recommendations) to each of the defects detected. Only the actor allocation defect was detected by Blueworks and Logizian. None of the tools analyzes the emplacement of connectors based on their semantic. It may be due to the differences among the tools when it comes to defining the connectors’ semantics (message flow, sequence flow and associations). All the modelling and evaluation tools of business processes concentrate on correcting the structure of the model with no concern on what is missing nor on the semantic of the concepts. “*BPM-Quality*” tool detects the notation semantic defects relying on the semantic constraints that should be validated.

- Finally, none of the pragmatic defects were detected by the tools. “*BPM-Quality*” detects the redundant concepts and checks the structure of the concepts labels relying on Wordnet's tags.

6.3 Validating the proposed quality approach

In Chapter 3 we proposed quality approach based on domain ontology. However, the viability of our proposed quality approach can only be validated through an experiment. Thus, we used medical domain ontology and an hospitalization process scenario in order to apply the experience. This experiment consists of the following steps:

- i. In the first step, respondents who have a minimum base knowledge about conceptual models were given the scenario that was followed to create the business process model. Respondents were required to analyze the scenario, to identify the tasks and to create a business process model in any language they prefer.
- ii. In the second step, each of these models was evaluated by the IS constraints. Quality improvement suggestions were provided to each defect detected.
- iii. In the third step, using our quality approach tool, type based mappings and semantic based mappings were calculated between the domain ontology and each of these models in order to provide the knowledge needed for not only evaluating the semantic quality of the models but also providing quality improvement.
- iv. In the fourth step, each of the transformed models is re-evaluated by the quality metrics in order to compare it to the original model.

In short, this experiment consists in evaluating and improving each of the models to finally compare them and prove the efficiency of our approach.

6.3.1 Samples

This experiment was given to different students in Conservatoire national des arts et métiers (CNAM) who have prior knowledge and experience about CM. Eight respondents submitted the business process model. We studied the feedback of each of the respondents with five questions.

1. Occupation: Respondents were required to select their occupation from a list of three pre-defined occupations: PhD student, Master student, Professional.
2. How many Conceptual model courses have they taken? We classify them into four classes:
 - If they took zero courses: None domain knowledge.
 - If they took one course: Average domain knowledge.
 - If they took two courses: Good domain knowledge.
 - If they took more than two courses: Very good domain knowledge.

3. Do they have any modelling experience? Respondents were required to select their modelling experience from three classes:
 - No experience: If they didn't have any practical course on conceptual models.
 - Average experience: If they took practice courses on conceptual models.
 - Maximum experience: If they are experts in modelling.

4. Do they have any business process modelling experience? Respondents were required to select their modelling experience from three classes:
 - No experience: If they didn't have any practical course on business process modelling.
 - Average experience: If they took practice courses of BP modelling.
 - Maximum experience: If they are domain experts.

5. What notation are they more comfortable with? Respondents were required to select from a list of five pre-defined notations.
 - BPMN
 - Eriksson-Penker
 - Activity diagram
 - EPC
 - Petri-net

6. Are they familiar with the scenario' domain?
 - If they have never been hospitalized: No familiarity with the scenario domain.
 - If they have been hospitalized once: Average familiarity with the scenario domain.
 - If they have been hospitalized more than once: Good familiarity with the scenario domain.

Respondent	1	2	3	4	5	6	7	8
Occupation	PhD student	PhD student	PhD student	PhD student	PhD student	PhD student	PhD student	Master student
Number of CM courses taken	Average knowledge	Average knowledge	Average knowledge	Average knowledge	Average knowledge	Average knowledge	Very good knowledge	Good knowledge
Modelling experience	Average experience	Average experience	Good experience	Good experience	Average experience	Good experience	Good experience	Good experience
BP modelling experience	No experience	No experience	No experience	No experience	No experience	No experience	Average experience	Average experience
Familiar notation	Activity diagram	Activity diagram	Activity diagram	Activity diagram	Activity diagram	Activity diagram	Activity diagram	BPMN
Familiarity with the scenario's	Good familiarity	Good familiarity	Average familiarity	Good familiarity	Average familiarity	Good familiarity	Good familiarity	Average familiarity

domain.								
---------	--	--	--	--	--	--	--	--

Table 19: Respondents information with respect to their occupation

(Table 19) provides the respondent's profile with basic information. It can be seen that our respondents have a prior knowledge and experience about conceptual models. All the respondents have taken at least one course in conceptual models. Only one respondent has good experience in business processes, which explains the familiar notation choice: BPMN. In spite of this, the respondents are very familiar with the scenario's domain and they have good experience in activity diagram's modelling.

In Section 4.1 we proposed quality metrics to evaluate BP models and, in Section 4.4, we defined improvement actions so that they can be associated with the metrics values. However, the viability of our proposed quality metrics and improvements can only be validated through an experiment. This experiment consists of the following five steps:

- i. In the first step, respondents were given a scenario to create a Business process model. As the respondents are not familiar with a specific notation, they were asked to use any notation they prefer.
- ii. In the second step, the models were annotated with the “*BPM-Quality*” annotation module. This module annotates the models so they can be mapped to the domain ontology concepts.
- iii. In the third step, IS constraints were applied on each of the models. The defects detected by these constraints are posted to the modeller with improvements that includes textual guidelines. Thus the IS constraints help in identifying the problem but the modeller must rely on the improvements provided for the solution.
- iv. In the forth step, the quality metrics are calculated with respect to each of the concepts selected by the modeller.
- v. In the fifth step, respondents were provided the list of the improvements including textual guidelines and improvement actions in order to improve their models.

6.3.2 Evaluation process

After insuring that the type based mappings are correctly done, the semantic based mappings are executed. And the models are evaluated using the quality metrics identified in Section 4.2.

Based on the metrics calculation results, improvements are proposed to the modeller. In the example shown in Module-4 (Appendix B), we will analyze each of the metrics results and the improvements proposed.

The list of activities in the model are seized by the tool and evaluated by the quality metrics. We will present an extract of the results.

Quality Attribute	Metric		Original Model
Ambiguity	Clarity	Patient	0.33
		Secretary	0.16
		Interview the patient	0.5
Abstraction level	Generality	Doctor	0.5
		Guide the patient to his room	0.5
		Enter the patient information	0.16
	Specification	Check Patient	0.1
		Doctor	0.125
		Symptoms	0.05
		Patient	0.5
Incompleteness	Input incompleteness	Obtaining information regarding the patient	0.25
		Complete the registration form	0.1
		Fill the admission form	0.5
	Actor allocation completeness	Model-3	0.002
	Model richness	Model-3	0.15

Table 20: Extract from the computed Metrics for Model-3

1. The clarity metric of the actor "Secretary" is equal to 0.33 i.e. it has a lot of synonyms in the domain, that is the actor label is not clear and creates an ambiguity since it may present several meanings in the domain.
2. The specification metric of the activity "Check patient" present a very low value and this is due to the provided hyponyms of the concept extracted from the ontology. The more the concept has a big number of hyponyms, the more it is general. The user is recommended to replace the activity by one or more specific activities.
3. The specification metric of the actor "Doctor" presents a very low value that is it has a big number of hyponyms in the ontology. The user is recommended to replace the actor by one or more specific ones.
4. The Input completeness of the activity "Complete the registration form" is equal to zero since it does not have any input resources. The modeller is recommended to complete his/her model by adding inputs suggested by the ontology.
5. The Model-Richness metric's low value confirms that the model does not cover a sufficient part of the domain. The modeller is asked to enrich his model. The model can be enriching by either adding new activities suggested by the ontology or replacing an activity by several more specific. It will increase the number of activities i.e. it will increase the model richness' metric value.

6.3.3 Improvements

As demonstrated in Section 4.4, improvements are proposed for the modeller to improve the quality of his/her model. Once the metrics are calculated, corresponding corrective actions or textual guidelines can be proposed to improve and correct the model. Thus, in view of the above metrics results and their interpretations, the following improvements can be applied to improve the model's quality:

- For improving the ambiguity of the actor "Secretary", the ontology provides synonyms extracted from Wordnet and presented by the data property "has synonyms". The user may use the action Replace (Secretary, front-desk staff).
- For improving the abstraction level of the actor "doctor", similarly the ontology provides hyponyms. (Figure 47) is an extract of the domain ontology that represents these hyponyms. Based on the type based mapping and the semantic based mapping the actor "doctor" is related to the actor "Doctor" as the function Name_Based_Similarity detects that both concepts have the same names. All the relations in (Figure 47) are "is-a" relations that present some of the concepts hyponyms. The modeller may use the action Replace (Doctor, Surgeon), or/and AddActor (Generalist, fill the form of evaluation).

The ontology is an OWL file. The information is extracted from OWL tags. (Figure 46) is an extract presenting the 'is-a' relations between the actor "Doctor" and its hyponyms.

```

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#is_a"/>
  <NamedIndividual IRI="#allergist"/>
  <NamedIndividual IRI="#doctor"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#is_a"/>
  <NamedIndividual IRI="#surgeon"/>
  <NamedIndividual IRI="#doctor"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#is_a"/>
  <NamedIndividual IRI="#cardiologist"/>
  <NamedIndividual IRI="#doctor"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#is_a"/>
  <NamedIndividual IRI="#pediatrist"/>
  <NamedIndividual IRI="#doctor"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#is_a"/>
  <NamedIndividual IRI="#gynecologist"/>
  <NamedIndividual IRI="#doctor"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#is_a"/>
  <NamedIndividual IRI="#generalist"/>
  <NamedIndividual IRI="#doctor"/>
</ObjectPropertyAssertion>

```

Figure 46: OWL extract

Reading a OWL tags isn't practical Thus we will present the ontology information in a graphical way (Figure 47).

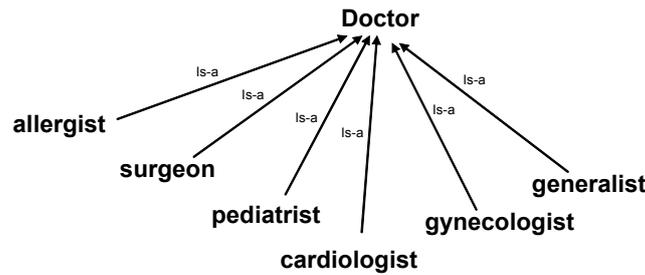


Figure 47: Doctor hyponyms

Similarly to the others, for improving the abstraction level of the activity "Check Patient", the ontology provides hyponyms. To improve the activity "Complete the registration form" input completeness, the user can rely on the context provided by the ontology to add input resources, etc.

(Table 21) sums up the most common improvements used in this experience. However, it is interesting to note that the improvements are related to the type of defects detected.

- Resources were added to the eight models due to the lack of resources detected by the Input completeness metric and due to the knowledge provided by the ontology.
- Activities were renamed due to the undesirable clarity, specification and generality metrics values and due to the different abstraction levels of the same activity provided by the ontology.
- Message flow replaced by an other type of connector. It means that there are twenty seven incorrect uses of the message flow detected by the Message flow constraint.
- Moreover, eighteen unlabeled edges were detected. Thus they renamed outgoing edges.

Improvement type	M1	M2	M3	M4	M5	M6	M7	M8	Sum
Add Resource	6	9	7	15	19	19	3	10	88
Rename Activity	3	7	6	4	7	9	4	10	50
Replace the Message flow	5	4	7	7	1	4	4	2	34
Rename outgoing edges	0	6	0	4	6	4	0	2	22
Add Activity	1	9	8	5	3	8	2	7	43
Add Actor	0	4	3	0	3	3	0	3	16
Replace the Sequence flow	4	0	0	0	1	1	0	1	7
Remove /Add outgoing edges	0	2	0	2	0	0	0	0	4
Add Gateway	0	3	2	0	0	0	0	0	5
Add End event	0	0	0	2	0	0	0	0	2
Add Start event	0	0	0	1	0	0	0	0	1
Add Sequence flow	0	0	0	4	0	0	0	1	5
Add Message flow	0	0	0	0	0	0	0	0	0
Add Association	0	0	0	0	0	0	0	0	0

Table 21: Improvements occurrence

6.4 Quality evaluation

Once all the improvements are applied to the original model, the same set of metrics is recalculated to demonstrate the quality improvement due to our proposed quality evaluation and improvement process. The results are placed in (Table 22).

Quality Attribute	Metric	Original Model	Post transformation models
Ambiguity	Clarity	0.32	0.66
Abstraction level	Specification	0.6	0.64
	Generality	0.31	0.71
Incompleteness	Input completeness	0.01	0.07
	Actor allocation completeness	0.05	0.1
	Model richness	0.17	0.215

Table 22: Metrics average calculation

From (Table 22), we can see that the *clarity* metric average was “0.32” in the original model whereas it is “0.66” in the transformed model meaning that the concepts are clearer. In fact, we have to point that not if the clarity metric is equal to 1 means that we are in the best situation. When the clarity metric is equal to 1, it means that there are no more synonyms provided by the domain ontology which have nothing to do with the best choice of concepts. The best choice of concept in the model is subjective and related to the modeller. When the clarity metric has increased, it means that the modeller replaced the concept with another provided by the ontology that has less synonyms in the domain i.e. less ambiguous.

Similarly, the specification metric average was “0.6” in the original models whereas it is “0.64” in the transformed models. The metric value has increased with a tiny difference which means that the hypernym choices do not answer the modellers’ requirements.

In contrast to the generality metric average, it was “0.31” in the original model whereas it is “0.71” in the transformed models, signifying that most of the concepts are replaced by the adequate level of concepts. The metric value variation for each model is shown in (Figure 48). The growth of the histogram bars means that the generality metric increases i.e. the number of hyponyms decrease. Similarly to the clarity metric, if the generality metric is equal to 1, it does not mean that we are in the best situation. The most adequate level does not mean the more specific level. (Figure 48) shows that the generality value of the models M2, M3 and M4 increase significantly that is to say that the model's activities were very vague and the ontology provides the modeller with a big number of hyponyms. If a specification metric is not increased, it may result from two causes: 1) the modeller did not replace the concept or 2) an incomplete ontology.

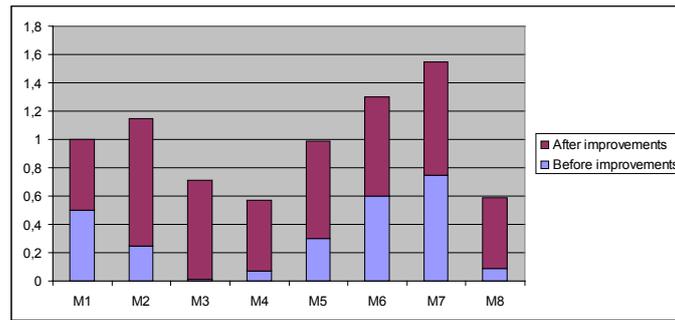


Figure 48: Generality metric variation

The specification and the generality metrics change at every ontology pass since when the modeller replaces an old concept with a new concept from the ontology, the new concept has different number of hyponyms and hypernyms, thus different value of specification and generality metrics.

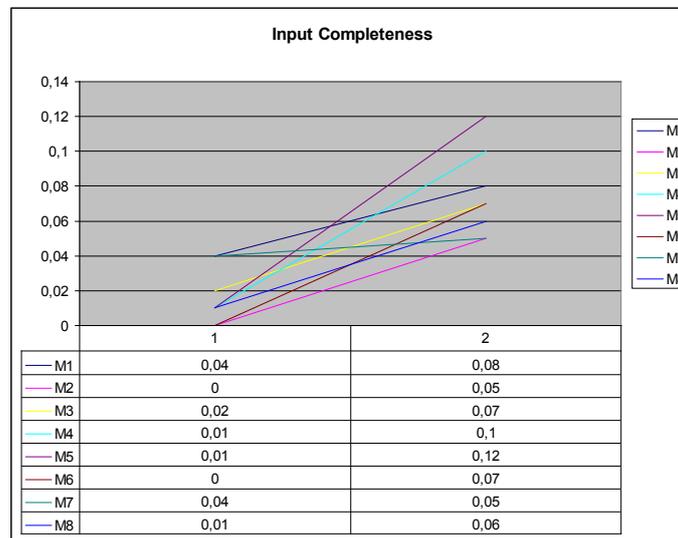


Figure 49: Input completeness metric variation

The Input completeness metric average value in the original models were 0.01 whereas in the transformed models is 0.07. The metric value is due to the big number of resources in the ontology. We can conclude that the majority of the models don't include resources for the activities. The simplicity of the inserting resources process to the activities has increased remarkably the value of the Resource allocation completeness metric. (Figure 49) shows the significant growth of the M5 curve, that is, the ontology provides the modeller with a big number of input resources. In contrast of the M7 curve where the ontology did not provide a big number of input resources, this may be due to three reasons: 1) there are no sufficient mappings between the BPM activities and the ontology actions (the BPM can be irrelevant or badly described activities), 2) the modeller didn't add the resources proposed by the ontology, 3) the ontology is incomplete.

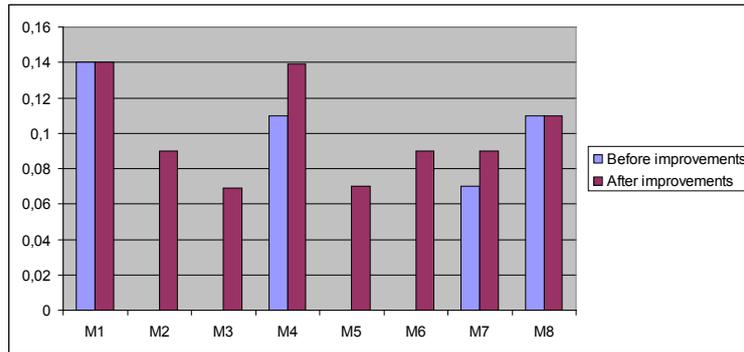


Figure 50: Actor allocation metric variation

The Actor allocation completeness metric average in the original models was 0.05 whereas in the transformed models it is 0.1, that is to say that an average of two actors was added to each model even if the ontology provide much more actors. It is due to the complexity of adding actors. The modeller must repartition the activities with respect the actors added.

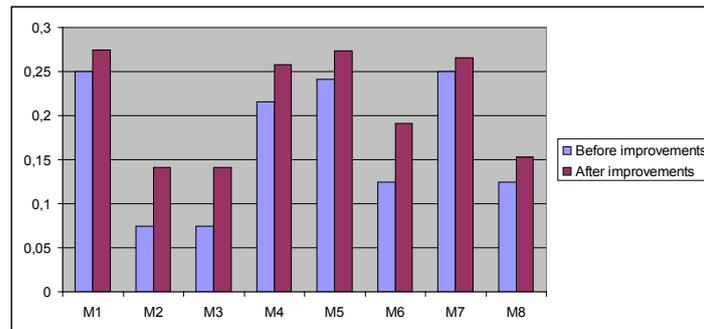


Figure 51: Model richness metric variation

At last, the average value of the model richness metric was "0.19" before the improvement process and it increases to "0.26" i.e. an average of 5 activities were added in each model. The activities added to the model are provided by the ontology through: 1) hyponyms of an other activity, as hyponyms describe details that the modeller may have skipped, 2) context of an other activity, the ontology exploits the temporal relations (succeed, precede) of each action in the ontology.

We can conclude that all the metrics associated to the completeness such as input completeness, resource allocation completeness and model richness metrics have significantly higher values compared to the original model implying that the transformed models are comparatively more complete compared to the original model.

An interesting observation from these results is that the respondent's feedback seems to have an influence on the occurrence of defects.

- The respondents that are familiar with the domain create models with sufficient details. The specification metric's value and the input completeness value were very acceptable.

- The respondents that have a good experience in Business process models have less annotation defects than those who have an experience in conceptual models in general.

6.5 Conclusion

In this chapter, we applied our proposed solution and processes on a set of models as case studies to evaluate their efficacy. We identified the domain ontology and the set of respondents. All the metrics were calculated. Improvements were generated and applied to the original model. The resulting transformed models were re-evaluated to check if the proposed improvements actually improved the model or not. In the end, the initial results were compared with the post-transformation results. We identified that the transformed models have significantly better metrics results than the original models suggesting that the proposed approach has helped in improving the original model.

In the next chapter, we conclude the thesis and provide some perspectives and future extensions of the work.

Chapter 7

Conclusion

It has now been widely agreed that the effectiveness, consistency and transparency of the companies' business operations depends on the quality of the Business Process models. Therefore, a good analysis and evaluation method should ensure that BPM quality. Hence if the Business process models are scanned for defects and then corrected, it is likely to reduce the number of change requests for the end system. Moreover, these errors and deficiencies in the BPMs will not be propagated along the development process.

Research on BPM quality evaluation is rather young and lacks any known standards. However, approaches in the literature concentrate on providing advices and best practices to ensure mainly the syntactic quality of models i.e. the structure of the model. Therefore, an effort on semantic and pragmatic quality definitions needs to be provided. Often the modellers are left with a proposed set of evaluation metrics that can be used for evaluation without helping in improving the models.

In spite of the wide acceptance of the benefits of process awareness, the improvement of Business Processes is still vague and incomplete. Literature review shows that little or no theoretical guidance is provided to modellers and analysts on how to improve the models.

Another major hurdle in evaluating the quality of Business process models is the lack of tools automating the evaluation process and proposing corrective actions for improvements.

7.1 Contributions

The research presented in this thesis is a step forward in semantics based quality evaluation and improvement using domain knowledge. In order to formulate a complete solution, we extracted knowledge from the notations semantics and from domain ontology. The approach takes into account the variety of business process model notations by using a meta-model. The domain ontology is represented by the means of ontologies where semantics is enriched by using several kinds of relationships among the concepts. One advantage of our approach is that it is not domain-specific or notation specific. The alignment process can be applied to each domain for which such ontology is available. It may highly facilitate the task of BP modellers and lead to a significant improvement of BP models. The other advantage is that it encourages capitalization of expertise. Indeed, in many fields, there is an effort of definition of structured and shared knowledge in several areas: medical practices, HR processes, e-learning etc. The proposed approach is an actual usage of such knowledge.

Most of the existing quality frameworks on BPM propose quality metrics relying on the structure of the model. In reply, we proposed:

- fifteen syntactic quality metrics: each metric has a return value that can be a boolean or an integer. We explained each of these metrics by defining their role, the defect they detect and analyzing their return value.
- six semantic quality metrics: all of these metrics have an integer return value and rely on the domain ontology to be calculated.
- three pragmatic quality metrics: each metric has a return value, two of them rely on wordnet in order to be calculated.

The proposed quality metrics evaluate in a quantitative way the syntactic, semantic and pragmatic quality of the model, relying not only on the structure of the model but also on the concepts semantics.

Relying on the proposed metrics, our approach detects syntactic structure defects (X (OR) Split gateways defects, Deadlocks, Unlabeled Activities, etc), semantic structure defects (Incorrect use of sequence flow, lack of actor, etc) and pragmatic structure defects (eliminate the redundant concepts...etc). In addition to semantic quality metrics based on the knowledge of the domain ontology such as clarity, generality, specification, model richness, etc. Based on the constraints violation and the metrics values, improvements are proposed. Improvements are classified into: textual guidelines, corrective actions and preventive actions. Moreover, quality improvement patterns are identified; each pattern has a name, a context, specific to a problem and provides solutions. Moreover, we were able to correct syntactic defects such as the lack of actors, bad emplacement of gateways, unlabeled activities, etc. Additionally to semantic defects such as ambiguity, input completeness, etc and pragmatic defects such as eliminating the redundant concepts, refactoring the activity labels, etc.

After the formulation of the IS constraints, quality metrics and improvements, we conducted two validation experiments. The first was performed on a set of one hundred business process models from BPM Academic initiative to ensure that the quality constraints defined detected a big number of the important quality defects in the evaluation of Business process models. The second was performed on models described by respondents in order to validate the strengths and benefits of employing our proposed approach in evaluating and improving the BP models. In fact we evaluated the semantic quality of the BP models by calculating the semantic metrics and analyzing the results.

Additionally, we compared the original models's and the transformed models's metrics results in order to demonstrate the effectiveness of the improvements applied. This activity enabled us in identifying the most common used improvement procedures.

We developed a software prototype “*BPM-Quality*” implementing our proposed quality approach and guidance process for evaluating and improving BPMs. Our prototype automates all the above mentioned steps of our approach and implements a workflow enabling its users to evaluate and

improve BPMs efficiently and effectively. “*BPM-Quality*” maintains a knowledge base storing different quality metrics, IS constraints and improvement procedures. After applying the IS constraints and calculating the quality metrics “*BPM-Quality*” proposes improvements procedures.

The approach suffers from some limits. The main one is the fact that the domain knowledge in hand is not necessarily complete. This is the reason why we propose to enrich the sources of domain knowledge with IS constraints. Hence the quality of the domain ontology affects the reliability of the metrics results and the effectiveness of the improvements proposed. We can seize from the literature research papers on the quality evaluation of domain ontologies such as (Tartir et al. 2005), (Stvilia 2007), etc.

An other limit is the completeness of the quality constraints and metrics. We can not assure that we can detect all the defects that a modeller can create, yet we aggregated as much defects from the literature as possible and defined new defects based on experiments done.

Finally, our approach requires further validation models to prove the improvement process robustness. As a matter of fact, we applied our approach on eight models allowing us to calculate and illustrate the semantic metrics on each concept in the BP models and to convey the different improvements proposed for each defect.

7.2 Future Work – Perspectives

Our thesis opens multiple directions for future research. This future work could be extensions to the actual work or new perspectives that are new work.

The first research direction aiming at extending our work includes:

- The improvement of distances computing the mappings between the BP models and the ontologies.
- The enrichment of quality metrics, thus defining new semantic quality criteria.
- Define thresholds for the quality metrics. For example, we know that when clarity metric value is decreasing the concept is more ambiguous but there is no specified threshold value.
- The quality metrics results are ranged into the interval $[0, 1]$. However, the clarity metric value "0.1" has a different meaning than the generality metric value "0.1". For that reason, and as a continuity to this thesis, metrics values should be compared to each others and new boundaries should be analyzed for each metric.
- Validation was done on a small number of models, that is, to analyze and apply the improvements proposed by the approach. We need to broaden the validation on a bigger number of models in order to cover as much defects as possible. The integrity of defects

detection can not be assured, but the bigger the number of models evaluated, the higher the number of defects are analyzed and detected.

- The synonyms in our approach are calculated via Wordnet. Thus some of the synonyms are not related to the domain. Integrating synonymy in the domain ontology will provide us with more adequate synonyms.

The second research direction, for improving the current work, includes the extension of our prototype in the following ways:

- i. To integrate a behavioural domain ontology (if exists) which present domain actions behaviour in order to help the identification of the sequence of activities. Thus a behavioural ontology may detect semantic defects related to the activities flow (two activities cannot happen in parallel, thus relating them with an AND split gateway is incorrect, etc).
- ii. To include the automatic application of proposed transformations. In our approach transformations are a category of recommendations. The current version of our prototype provides details about the relevant transformations whereas the application of these transformations is a manual process.

The third research direction could be to formulate a pivot model that integrates all the business process modelling notation concepts. And based on this pivot model we can transform models from a notation to an other.

The fourth research direction can be to create a SharePoint Online that provides an integrated location where modellers can share their models. As a result, we will have an online database for business process models, this database will enable the modellers to share their knowledge and find organizational resources and information. It will also help to extend the evaluation experiences, large-scale experiences, hence detecting the largest number of defects that can be made and hopefully attend a level where no new defects are detected. Based on the defects detected, new constraints and new metrics will be created and stored into the knowledge base and efficiently help the modellers to improve the quality of their models.

Appendix A

Mappings between the ontology concepts and the BP model concepts

Mappings between the ontology concepts and the BP model concepts:

These mappings use:

- Wordnet data base:

```
Protected static final String WORDNET_DATABASE_DIR_PROPERTY "wordnet.database.dir";
System.setProperty(WORDNET_DATABASE_DIR_PROPERTY,(new File("resources/dict/")).getAbsolutePath());
```

- Wordnet functions:

```
JWS jw= new JWS("C:/Users/sarah/Desktop/Wordnet plugin","wordnet similarity");
sResnik r = jw.getResnik();
sTreeMap<String, Double> p = jw.getPath().path("doctor", "entity", "n");
sLin l = jw.getLin();
WuAndPalmer w = jw.getWuAndPalmer();
```

Name based similarity: Take a business process model concept "a" and return all the ontology concepts that are syntactically equivalent to "a". It exploits all the ontology concepts and calculate the LevenshteinDistance distance.

```
public ArrayList<String> Name_Based_Similarity(String a)
{
    ArrayList<String> Name_Based_Similarity = new ArrayList<String> ();
    Name_Based_Similarity = null;
    for(int i=0; i<OntoReader.listofterms.size(); i++)
    {
        OntoConcept o = (OntoConcept) OntoReader.listofterms.get(i);
        if(bp.LevenshteinDistance(a, o.name)==0)
            Name_Based_Similarity.add( o.name.substring(1));
    }
    return Name_Based_Similarity;
}
```

Partially Based Similarity: Take a business process model concept "a" and return all the ontology concepts that are partially equivalent to "a", that is at least a word in common. It exploits all the ontology concepts and calculate the LevenshteinDistance distance between each word in both concepts.

```
public ArrayList<String> Partially_Based_Similarity(String a)
{
    ArrayList<String> Partially_Name_Similarity = new ArrayList<String> ();
    Partially_Name_Similarity = null;

    for(int i=0; i<OntoReader.listofterms.size(); i++)
    {
        OntoConcept o = (OntoConcept) OntoReader.listofterms.get(i);
        if(bp.Partialyequivalence(o.name, a))
```

```

        Partially_Name_Similarity.add(o.name.substring(1));
    }
    return Partially_Name_Similarity;
}

```

The partially equivalent used is described below: It decomposes each of the BP model concept and the ontology concept. Thus it compare each of the split results.

```

public Boolean Partialequivalence(String ab, String a)
{
    for(int i=0; i<OntoReader.getListeofterms().size(); i++)
    {
        OntoConcept o = (OntoConcept) OntoReader.getListeofterms().get(i);
        if((a.contains("_") || a.contains(" ") || a.contains("for") || a.contains("to") || a.contains("and") ||
        a.contains("or")) && (o.name.contains("_") || o.name.contains(" ")))
        {String T[];
        String Q[] = null;
        if (a.contains("_"))
            T = a.split("_");
        else
            T = a.split(" ");

        if (o.name.contains("_"))
            Q = o.name.substring(1).split("_");
        else if (o.name.contains(" "))
            Q = o.name.substring(1).split(" ");
        for(int c = 0; c < T.length ; c++)
            for(int b = 0; b < Q.length ; b++)
            {
                if(T[c].compareToIgnoreCase(Q[b]) == 0 )
                    return true;
            }
        }
    }
    return false;
}

```

Synonyms calculated based on Wordnet based:

- **Resnik similarity** takes a BP model concept "a" , it calculates the Resnik distance between "a" and each of the ontology concepts.

```

public ArrayList<String> Resnik_Similarity( String a)
{
    String T[] = a.split(" ");

    ArrayList<String> resnik_result = new ArrayList<String>();

    for(int i=0; i<OntoReader.listeofterms.size(); i++)
    {
        OntoConcept o = (OntoConcept) OntoReader.listeofterms.get(i);
        String Q[] = o.name.split("_");

        for (int l=0; l<T.length; l++)
            for (int j=0; j<Q.length; j++)
            {
                TreeMap t = r.res(Q[j], T[l], "n");
            }
    }
}

```

```

    Collection<Double> tab = r.res(Q[j], T[l], "n").values();
    java.util.Iterator<Double> it=tab.iterator();
    while (it.hasNext())
    {
        Double on = it.next();
        if(on > 5)
            resnik_result.add(o.name.substring(1));
    }
    }
    }
    return resnik_result;
}

```

- **lin similarity** takes a BP model concept "a" , it calculates the Lin distance between "a" and each of the ontology concepts.

```

public ArrayList<String> Lin_Similarity(String a)
{ArrayList<String> lin = new ArrayList<String> ();
String T[] = a.split(" ");

    for(int i=0; i<OntoReader.listeofterms.size(); i++)
    {
        OntoConcept o = (OntoConcept) OntoReader.listeofterms.get(i);
        String Q[] = o.name.split("_");

        for (int m=0; m<T.length; m++)
            for (int j=0; j<Q.length; j++)
            {
                TreeMap t = l.lin(Q[j], T[m], "n") ;

                Collection<Double> tab = l.lin(Q[j], T[m], "n").values();

                java.util.Iterator<Double> it=tab.iterator();
                while (it.hasNext())
                {
                    Double on = it.next();
                    if(on > 0.7)
                        lin.add(o.name.substring(1));
                }
            }
    }
    return lin;
}

```

- **Wun And Palmer** similarity takes a BP model concept "a" , it calculates the Win and Palmer distance between "a" and each of the ontology concepts.

```

public ArrayList<String> Wun_Palmer_Similarity(String a)
{String T[] = a.split(" ");
ArrayList<String> Wun_Palmer = new ArrayList<String> ();

```

```

for(int i=0; i<OntoReader.listeofterms.size(); i++)
{
    OntoConcept o = (OntoConcept) OntoReader.listeofterms.get(i);
    String Q[] = o.name.split("_");
    for (int m=0; m<T.length; m++)
        for (int j=0; j<Q.length; j++)
            {

TreeMap t = r.res(Q[j], T[m], "n") ;

        Collection<Double> tab = w.wup(Q[j], T[m], "n").values();

        java.util.Iterator<Double> it=tab.iterator();
        while (it.hasNext())
            {
                Double on = it.next();
                if(on > 0.9)
                    Wun_Palmer.add(o.name.substring(1));
            }

            }
        }
    return Wun_Palmer;
}

```

More general : It returns the hypernyms of the equivalent and the partially equivalent functions results.

```

public ArrayList<String> More_general (String a)
{OntoConcept x = null;
    ArrayList<String> more_general = new ArrayList<String>();
    String T[] = a.split(" ");
    for (int j=0; j<T.length; j++)
for(int i=0; i<OntoReader.listeofterms.size(); i++)
    {
        OntoConcept o = (OntoConcept) OntoReader.listeofterms.get(i);
        // System.out.println(o.name.substring(1));

        if(o.name.substring(1).contains(T[j]))
        { //more_general.add(o.name);
            for (int k =0; k<o.Hypernyms.size(); k++)
                more_general.add(o.Hypernyms.get(k).substring(1));
        }

    }

    for(int i=0; i<OntoReader.listeofterms.size(); i++)
        {
            OntoConcept o = (OntoConcept) OntoReader.listeofterms.get(i);
            for (int ab=0; ab<o.Synonyms.size(); ab++)
                {if (o.Synonyms.get(ab).compareTo(a)==0)
                    x=o;
                }
        }

    if(x!=null)
        { for (int n=0; n<x.Hypernyms.size(); n++)

```

```

        more_general.add(x.Hyponyms.get(n).substring(1));
    }
    System.out.println(" more_general size than " +a +"" + more_general.size() );

    for (int m=0; m<more_general.size(); m++)
    System.out.println(" more_general than " +a +"" + more_general.get(m) );
    return more_general;
}

```

More specific : It returns the hyponyms of the equivalent and the partially equivalent functions results.

```

public ArrayList<String> More_specific (String a)
{OntoConcept x = null;
    ArrayList<String> more_specific = new ArrayList<String>();
    String T[] = a.split(" ");
    for (int j=0; j<T.length; j++)
    for(int i=0; i<OntoReader.listeofterms.size(); i++)
    {
        OntoConcept o = (OntoConcept) OntoReader.listeofterms.get(i);

        if(o.name.substring(1).contains(T[j]))
        {
            for (int k =0; k<o.Hyponyms.size(); k++)
            more_specific.add(o.Hyponyms.get(k).substring(1));
        }
    }
    for(int i=0; i<OntoReader.listeofterms.size(); i++)
    {
        OntoConcept o = (OntoConcept) OntoReader.listeofterms.get(i);
        for (int ab=0; ab<o.Synonyms.size(); ab++)
        {if (o.Synonyms.get(ab).compareTo(a)==0)
            x=o;
        }
    }
    if(x!=null)
    {for (int n=0; n<x.Hyponyms.size(); n++)
        more_specific.add(x.Hyponyms.get(n).substring(1));
    }
    for (int m=0; m<more_specific.size(); m++)
    System.out.println(" more_specific than " +a +"" + more_specific.get(m) );
    return more_specific;
}
}

```

Quality Metrics:

- **Generality:**

```

public Double Generality( String a)
{Double gen;
    int size =ak.Hyponymresults.size();
    if(size ==0)
        gen =0.0;
    else gen = 1./size;
    return gen;
}

```

```
}

```

- **Specification:**

```
public Double Specification (String a)
{Double spec;
int size = ak.Hyponymresults.size();
    if(size ==0)
        spec =0.0;
    else spec = 1./size;
    return spec;
}
```

- **Clarity:**

```
public Double Clarity( String a)
{Double clar;
    int size = ak.Synonymresults.size();
    if(size ==0)
        clar =0.0;
    else clar = 1./size;
    return clar;
}
```

- **Input Richness:**

```
public int Input_richness( String a)
{
    int ratio;
    String rrange ="";
    // X is the number of inputs of the activity a.
    int x=0;
    for (int kk=0; kk< BPMReader.ConceptList.size(); kk++)
        for (int i=0; i<BPMReader.Resource_input_List.size(); i++)
            {
                if(BPMReader.ConceptList.get(kk).ID.compareTo(BPMReader.Resource_input_List.get(i).range)==0)
                    {
                        rrange = BPMReader.ConceptList.get(kk).name;
                        x++;
                    }
            }
    //y is nb of inputs to oa
    int y= ak.Contextresults.size();
    if( y==0) ratio = 0;
    else ratio = x/y;

    return ratio;
}
```

- **Activity Richness:**

```
public int Activity_Richness()
{
    int ar;
    //Calculating Activity Richness
    int nb_BPM_activities ;
    nb_BPM_activities = BPMReader.ActivityList.size();
}
```

```

int nb_onto_actions=0;
for (int i=0; i< OntoReader.listeofterms.size(); i++)
    if(OntoReader.listeofterms.get(i).type.compareTo("#action")==0)
        nb_onto_actions++;
if(nb_onto_actions==0) ar =0;
else
ar = nb_BPM_activities/nb_onto_actions;
return ar;
}

```

- **Ressource Richness:**

```

public Double Ressource_Richness()
{ Double rr;
double nb_BPM_resources ;
Double nb_BPM_artifact;
Double nb_BPM_info;
nb_BPM_artifact = (double) BPMReader.ArtifactList.size();
nb_BPM_info = (double) BPMReader.InformationRessourceList.size();
nb_BPM_resources = nb_BPM_artifact + nb_BPM_info;
int nb_onto_artefacts=0;
for (int i=0; i< OntoReader.listeofterms.size(); i++)
if((OntoReader.listeofterms.get(i).type.compareTo("#abstract")==0) ||
(OntoReader.listeofterms.get(i).type.compareTo("#artefact")==0)||
(OntoReader.listeofterms.get(i).type.compareTo("#knowledge")==0))
nb_onto_artefacts++;
if(nb_onto_artefacts==0) rr =0.0;
else
rr = nb_BPM_resources/nb_onto_artefacts;
return rr;
}

```

BPM-Quality interfaces:

The first “BPM-Quality” interface is to load the BP model adress and the domain ontology adress.

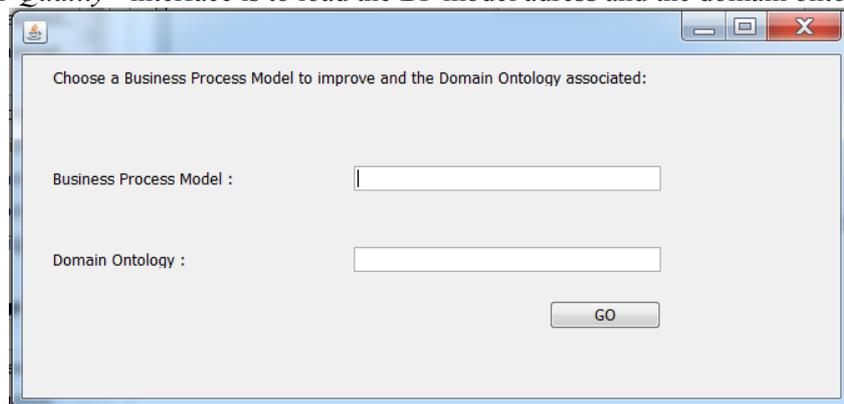


Figure 52: Load interface

First interface is the IS constraints application interface. It allows the modeler to check each of the syntactic, semantic and pragmatic constraints.

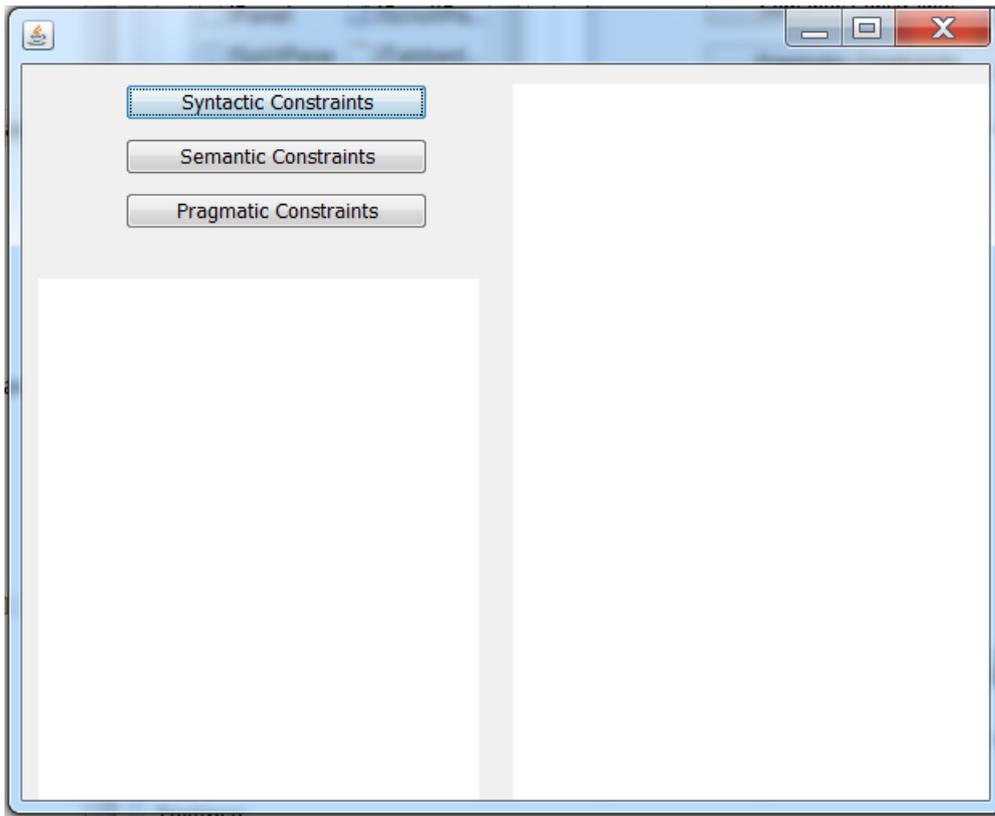


Figure 53: constraints interface

The first textArea will post the name of each constraint checked. The second textArea will post the results of the constraints checked.

The third interface is the activity evaluation and improvement interface. It loads the BP model activities, evaluate them and suggest improvements.

- List of activities comboBox will load all the BP model activities.
- Synonyms comboBox post all the chosen activity synonyms.
- Hyponyms comboBox post all the chosen activity hyponyms.
- Hypernyms comboBox post all the chosen activity hypernyms.
- Keywords comboBox post all the concepts (except people resources) related to the actions mapped to the chosen activity.
- People Resource comboBox post the people resources related to the actors mapped to the chosen activity.

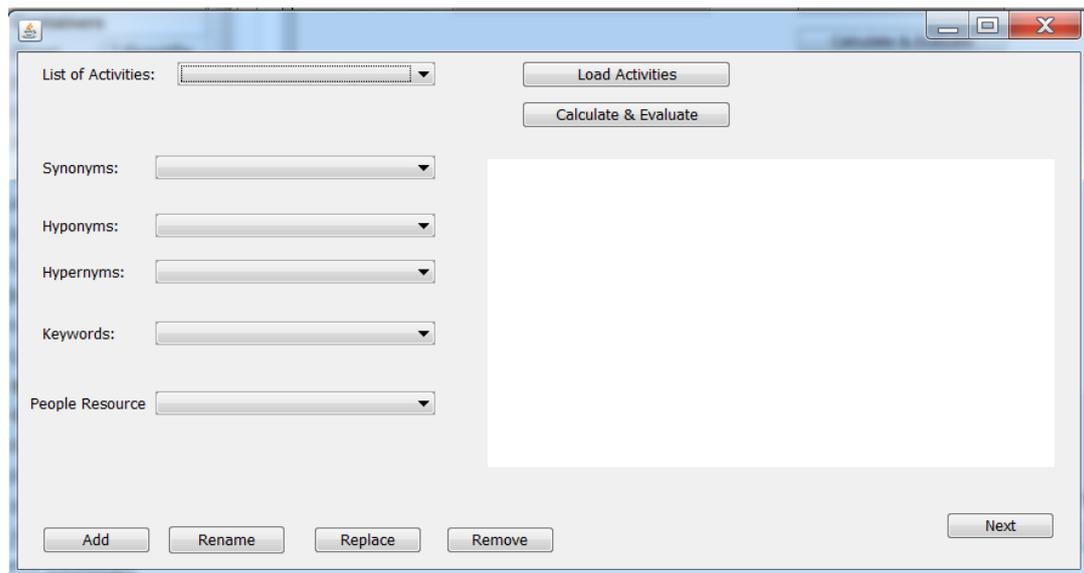


Figure 54; Activity evaluation and improvement interface

The improvements proposed by the interface are automatically applied by the tool.

- Add a new activity which will create a new interface to seize the successor and the predecessor of the new activity.
- Rename an activity which will apply the Rename function and take the selected activity from the list of activities and the new one selected from any of the comboBox lists.
- Replace an activity which will apply the Replace function and take the selected activity from the list of activities and the new one selected from any of the comboBox lists.
- Remove an activity.

The fourth interface is the people resource evaluation and improvement interface. It loads the BP model people resources, evaluates them and suggests improvements.

- List of people resources comboBox will load all the BP model people resources.
- Synonyms comboBox post all the chosen people resource synonyms.
- Hyponyms comboBox post all the chosen people resource hyponyms.
- Hypernyms comboBox post all the chosen people resource hypernyms.
- Keywords comboBox post all the concepts (except people resources) related to the actors mapped to the chosen people resource.
- Responsible comboBox post the activities related to the actions mapped to the chosen people resource.
- Complete People Resources button will display a new interface that contains new people resources not posted in the comboBoxes. It will allow the modeler to complete his model with new actors.

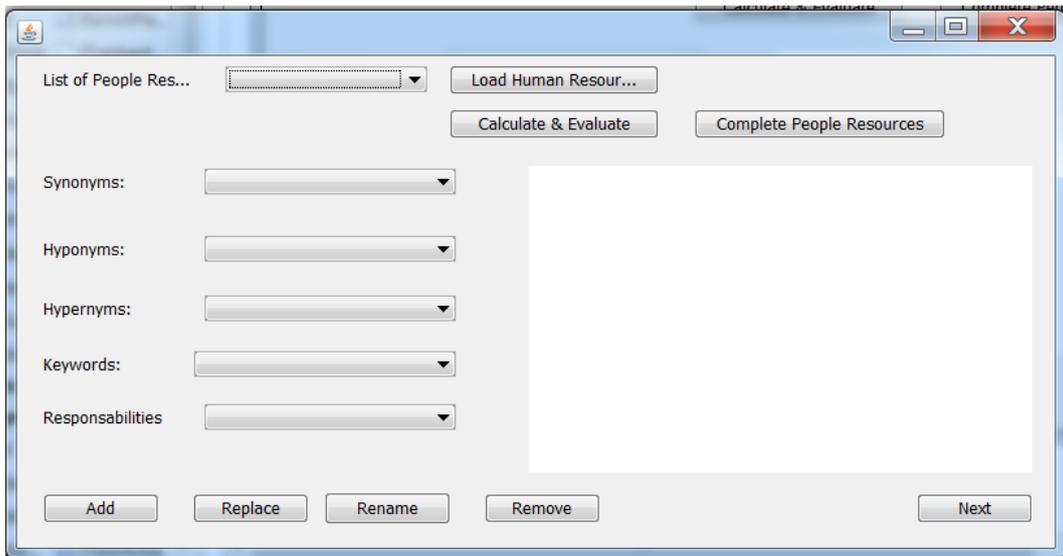


Figure 55: People Resource evaluation and improvement interface

The improvements proposed by the interface are automatically applied by the tool.

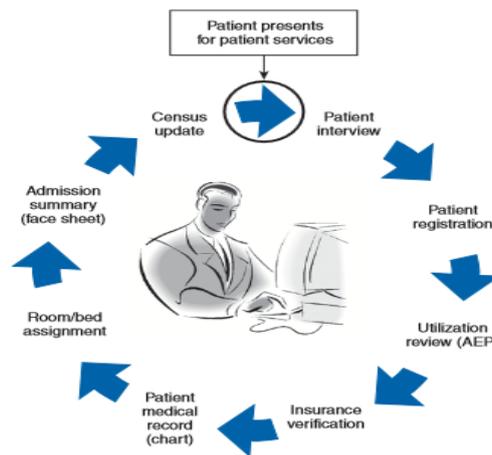
- Add a new people resource.
- Rename a people resource which will apply the Rename function and take the selected people resource from the list of people resources and the new one selected from any of the comboBox lists.
- Replace a people resource which will apply the Replace function and take the selected people resource from the list of people resources and the new one selected from any of the comboBox lists.
- Remove a people resource.

Appendix B

The validation Scenario:

The purpose of this document is to describe the different steps covered in a hospitalization process.

Pre-admission Process



A *patient* arrives to the hospital, he reports to the reception of the hospital. The *secretary*, enquires about her/his problem by interviewing her/him. The interview is conducted for the purpose of obtaining information regarding the patient situation and complete her/his medical record.

If it's a *new patient*, he must complete a registration form at the first visit ,which becomes part of the patient's permanent medical record. Meanwhile the *secretary* create his medical record.

Patient information is obtained by the hospital through the use of a patient registration form. The form generally contains fields for information regarding the *patient (Demographic information)*, **insurance carrier (information regarding the insurance carrier or government program underwhich the patient has coverage)**, **guarantor, diagnosis, and physician (Information regarding the patient's condition and thereferring or admitting physician)**. The patient registration form is completed by the patient and reviewed by the *admission representative*. The registration process consists of creating a patient account on the hospital's computer system and entering patient information obtained during the patient interview. The patient's account is the computerized record by which the patient information is recorded and maintained. An account and medical record number are assigned to each patient by the system. In fact, the patient's medical record is a folder where the patient's information is stored, including **demographic, insurance, financial, and medical information**.

Hospitalization Process

The *secretary* refers the patient to the concerned department/*doctor*. He is assigned a room and/or a bed as semi private rooms have two beds in each room, therefore the *patient* would be assigned a room and a bed. Room assignment is also performed in outpatient areas such as Ambulatory Surgery, the Emergency Department, or Observation.

The concerned *doctor* investigates the patient's case history. If requires, patient is advised for admission in the hospital. In case of admission, the patient is given the **admission date and admission form** for further formalities except for emergency cases. This will be recommended by the specialist during the patient's outpatient consultation. Meanwhile, the patient is counseled by the secretary regarding the treatment package which includes: **estimated bill size, average length of stay, various modes of payment accepted, documents to bring on day of admission**.

Further, when the patient arrives at the ward, the hospital *ward staff* will orientate the patient to the ward and to his room. Patient will be clerked by the *doctor*, this involves:

- asking for symptoms
- taking a detailed medical history
- ordering of tests if necessary

Based on the **symptoms** reported by the *Patient*, **laboratory tests** are demanded, such as **blood tests, X-ray procedures...** Results are collected by the *nurse* and notified to the *doctor*. Progress notes outline **the patient's status, results of diagnostic studies, and response to treatments**. The *doctor* completes progress notes each time the patient is seen.

The nurse **records notes** during each shift. She collect the **monitoring results**. Her progress notes indicates the patient status, response regarding his **condition and vital signs such as blood pressure, pulse, temperature**. And finally, notify the results to the *ward staff* and observe the symptoms during all the patient stay.

The *doctor* analyse the **results of laboratory exams** and classify the problem. According to the patient's presentation a course of **antibiotics** might be indicated and in some cases a **surgical consult** is necessary. Medications and other pharmaceuticals required during the patient stay are provided by the *Pharmacy Department* in accordance with the physician's orders.

Check out Process

Finally, before leaving the ward, patient is handed over with detailed discharge summary, which includes **doctor's advice on their further follow-up treatment, daily routine diet, and medical prescription**. The doctors may give the patient an appointment for follow-up. If the patient needs to reschedule the outpatient appointment after discharge, they can feel free to contact the concerned doctor.

Create a Business process model or an activity diagramme:

- The actors are presented into Lanes in the business process models, or into Swimlanes if it's an activity diagramme.
- You can describe a specific case i.e. declare specific symptoms as high temperature or dizziness. Based on these symptoms a specific laboratory tests are demanded as blood test.

Models used in the 2nd experience:

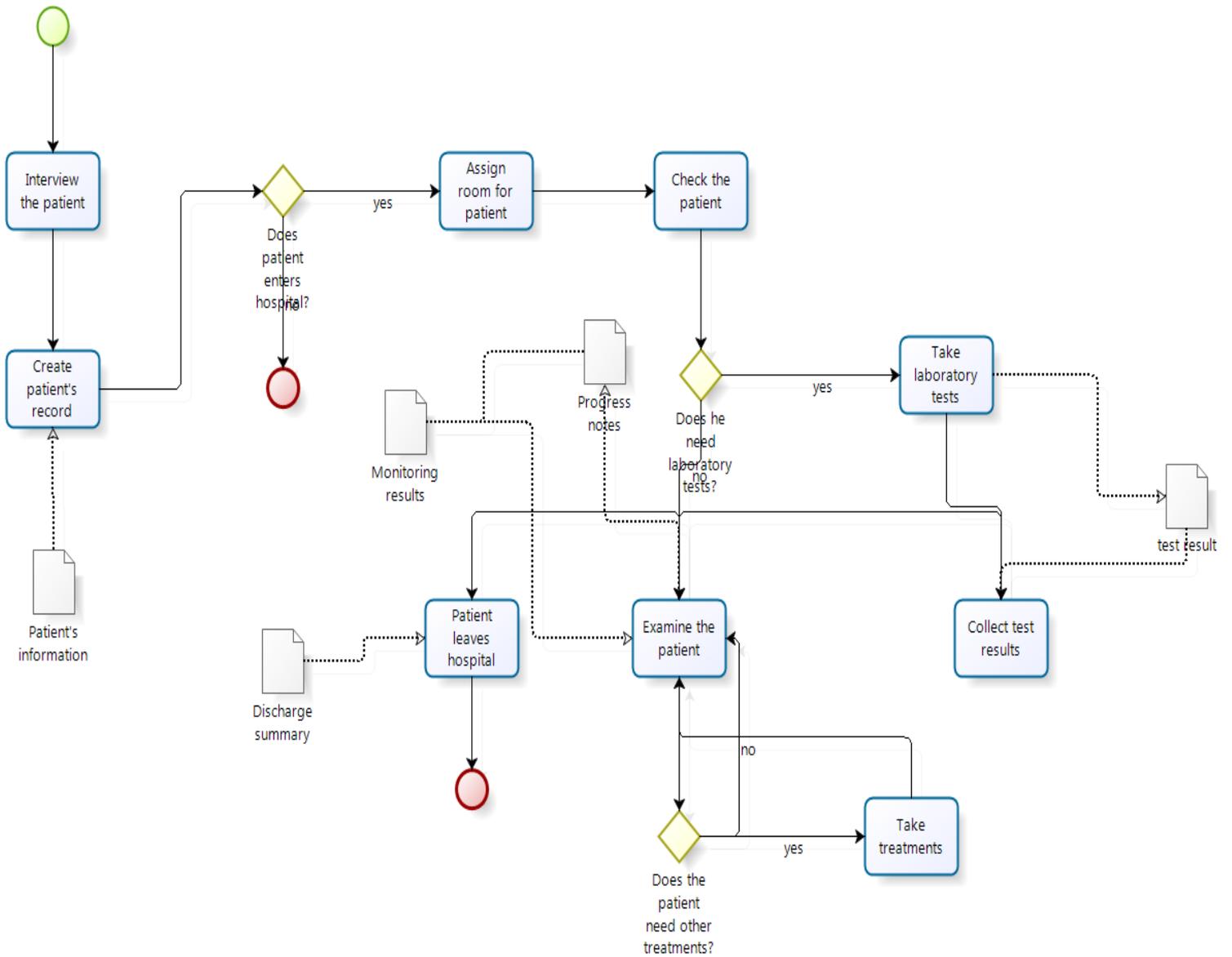


Figure 56: Model-1

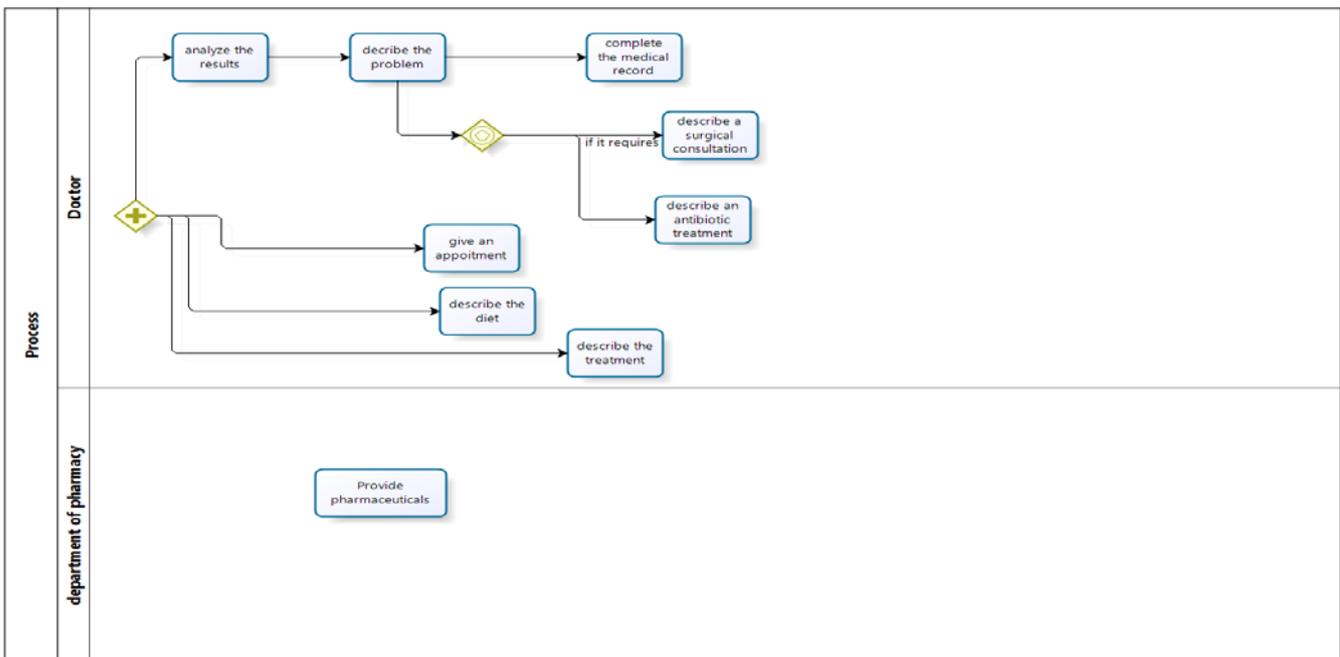
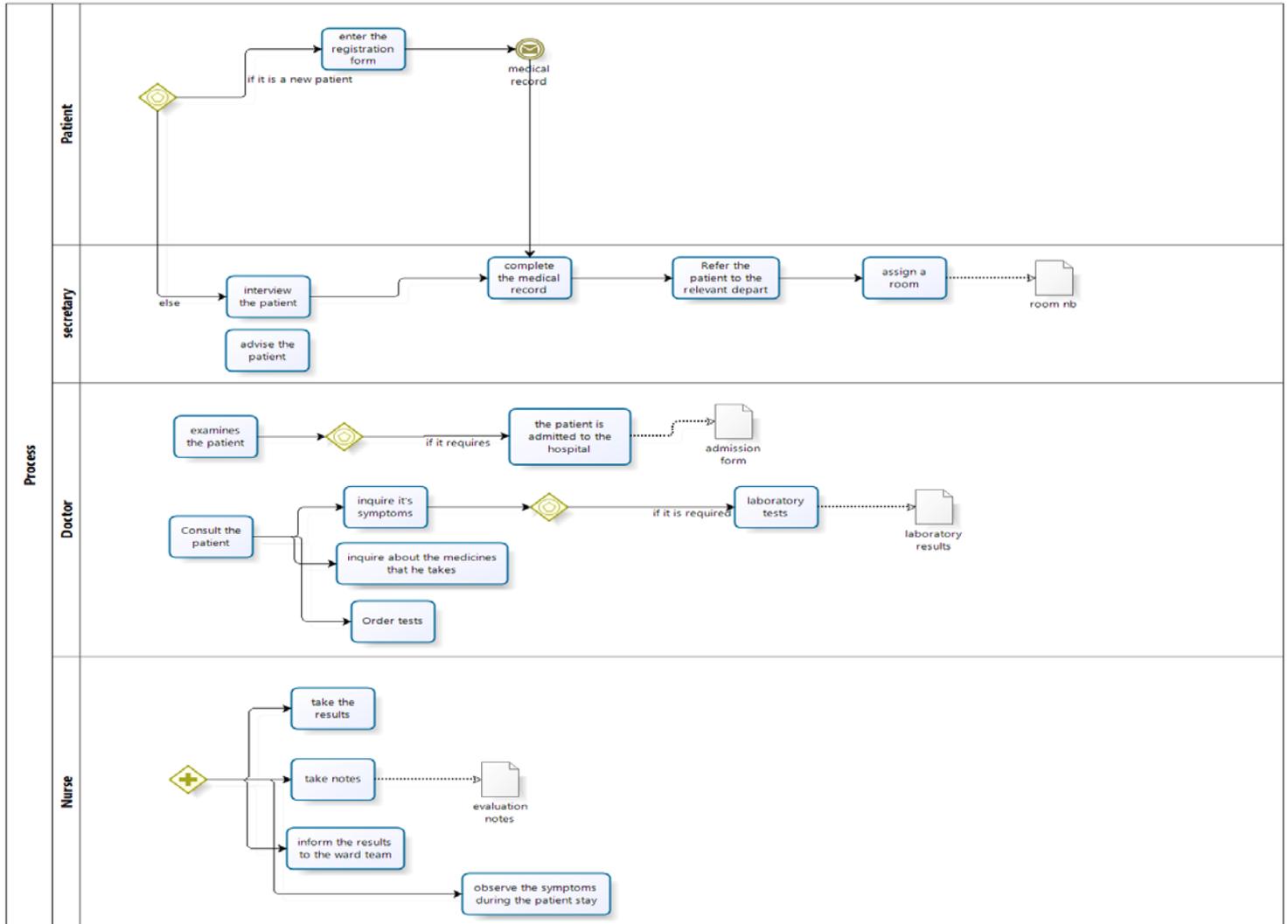


Figure 57: Model-2

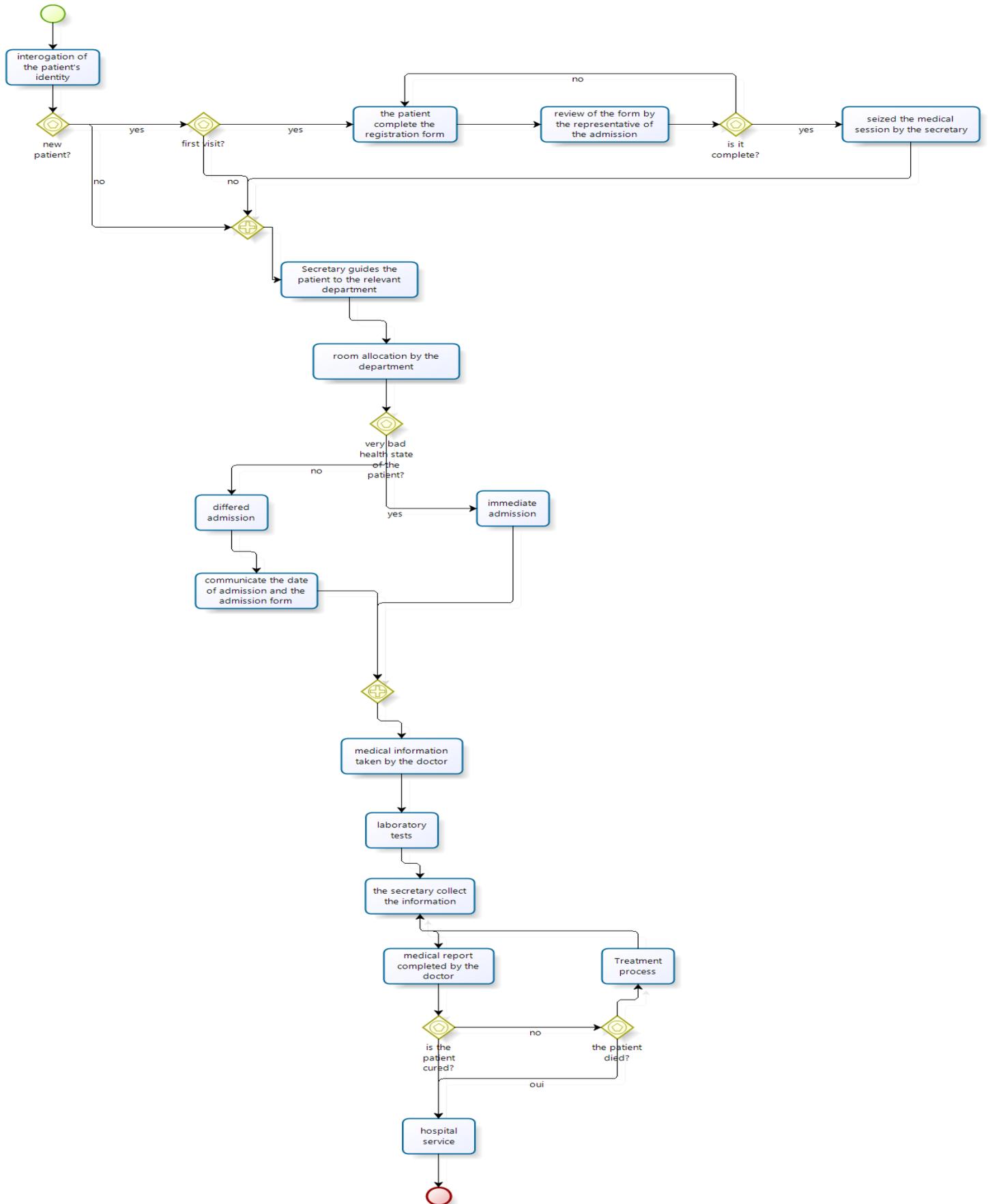


Figure 58: Model-3

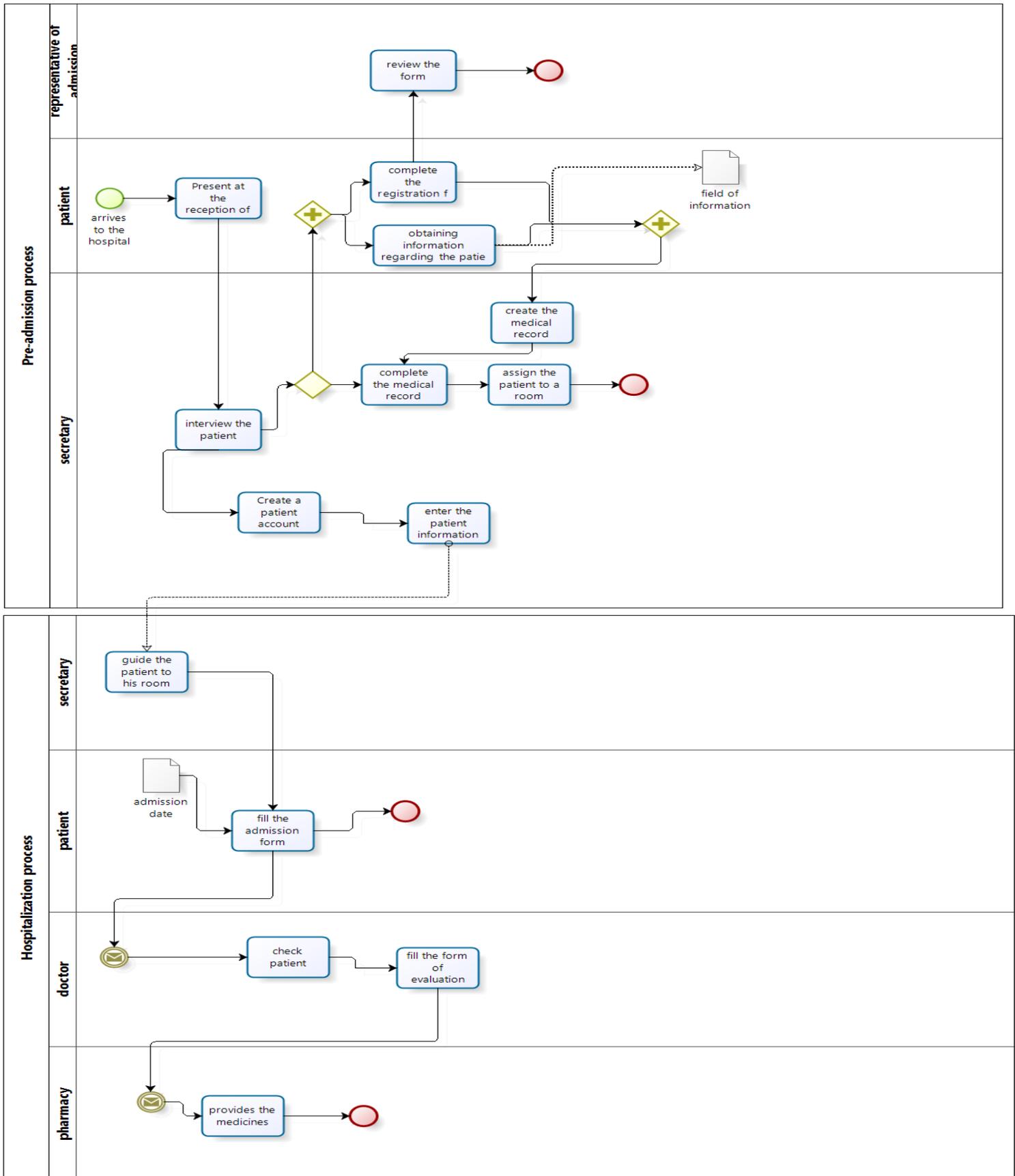


Figure 59: Model-4

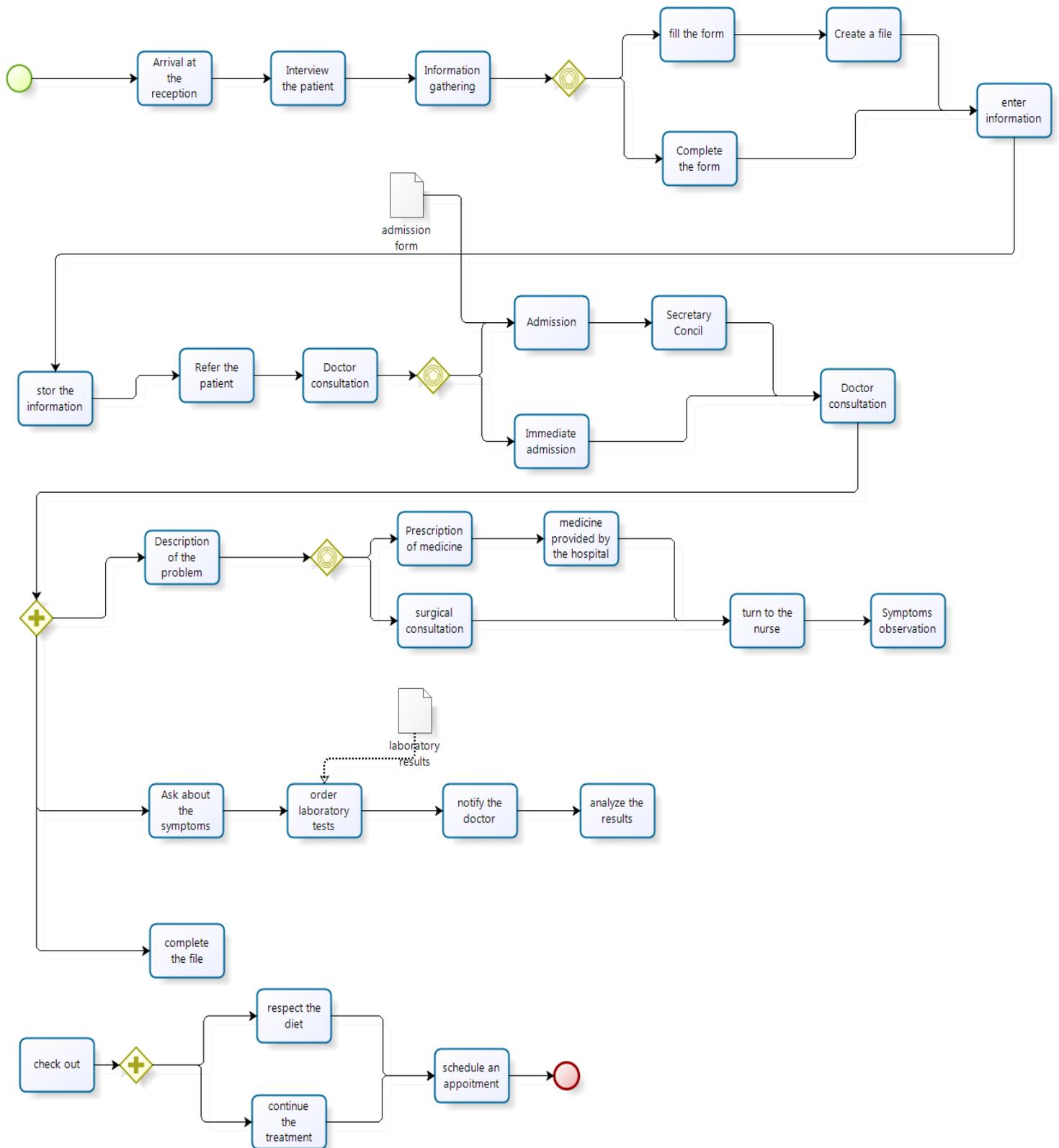


Figure 60: Model-5

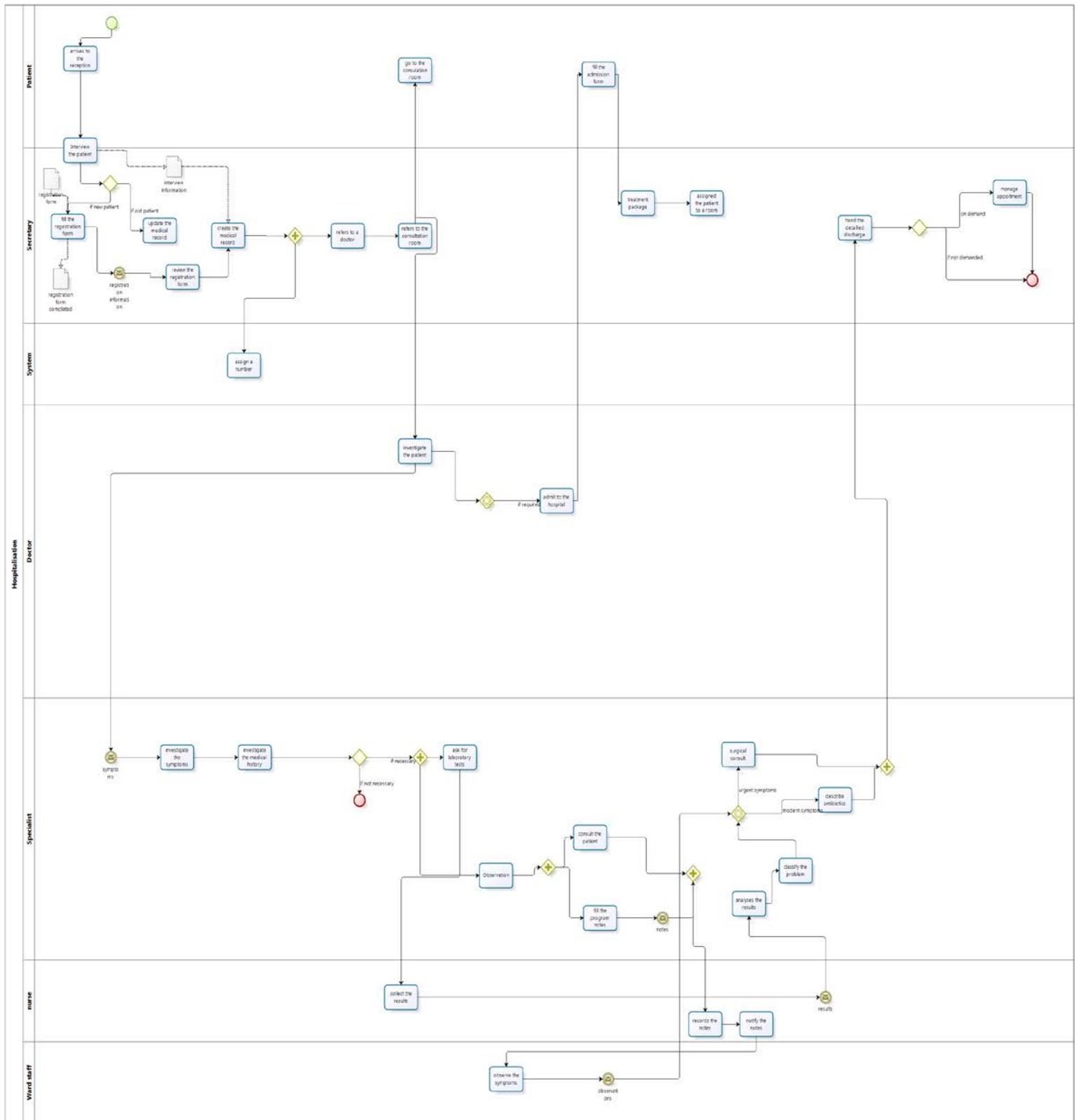


Figure 61: Model-6

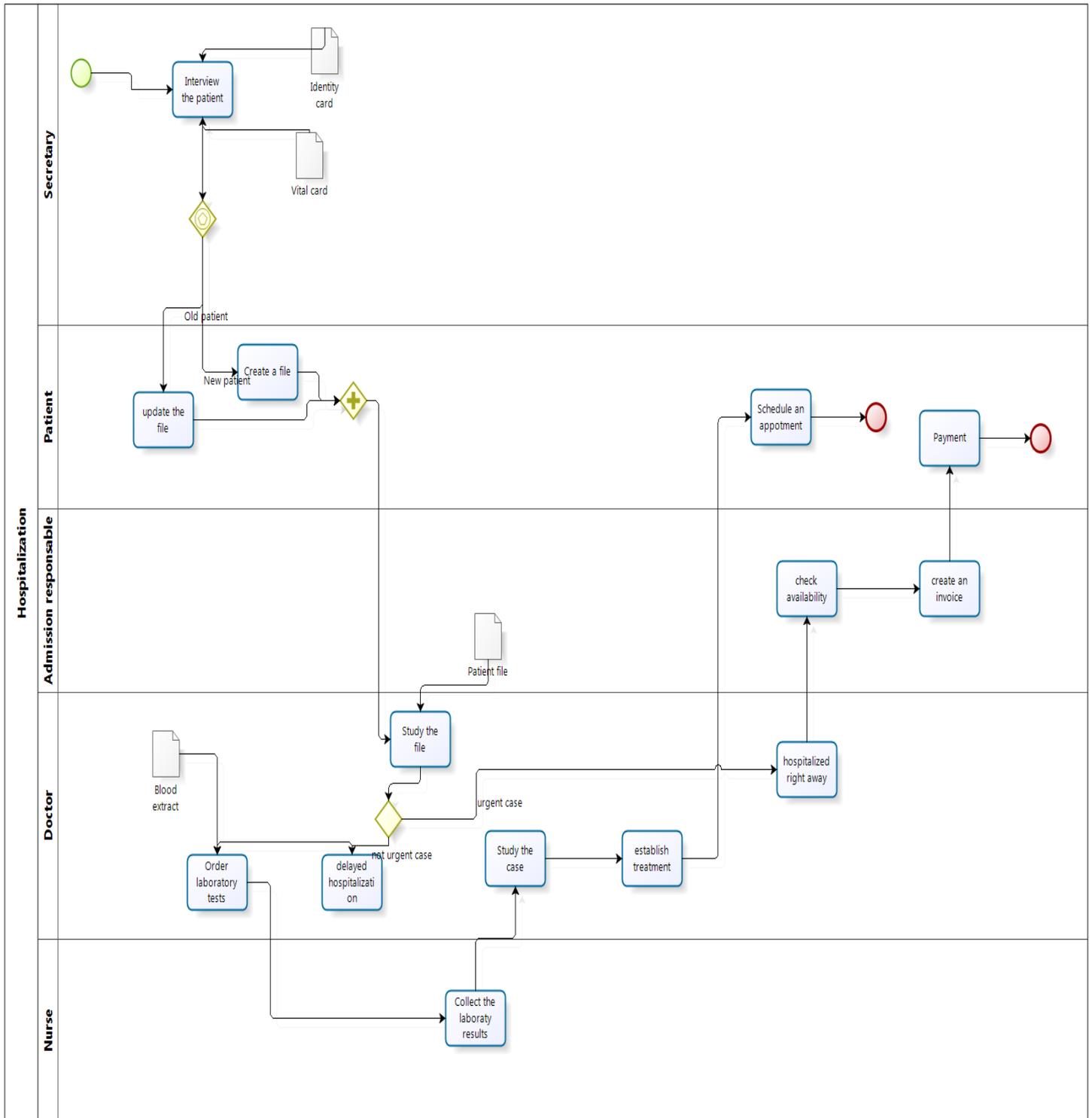


Figure 62: Model-7

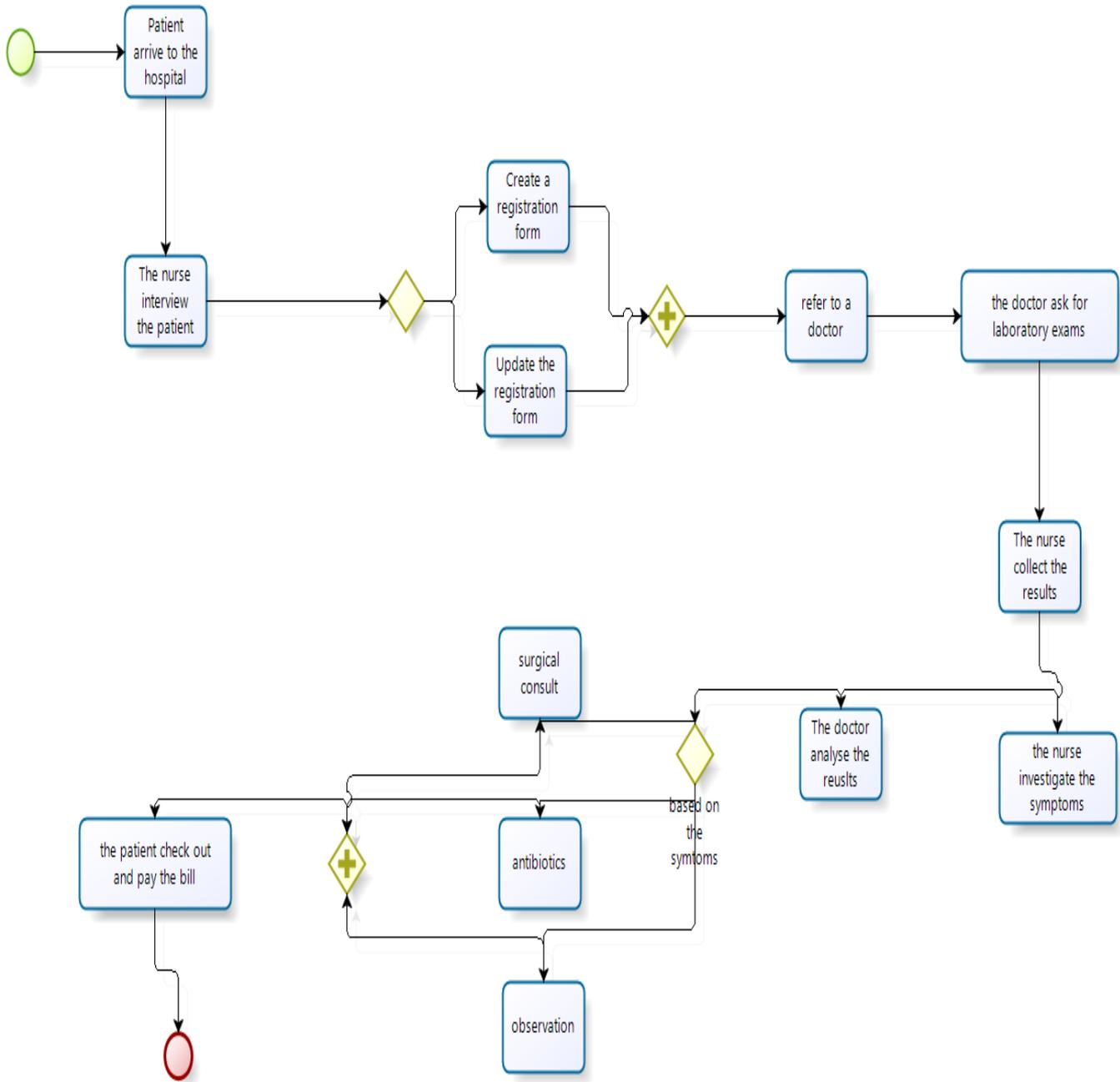


Figure 63: Model-8

Appendix C

The domain Ontology

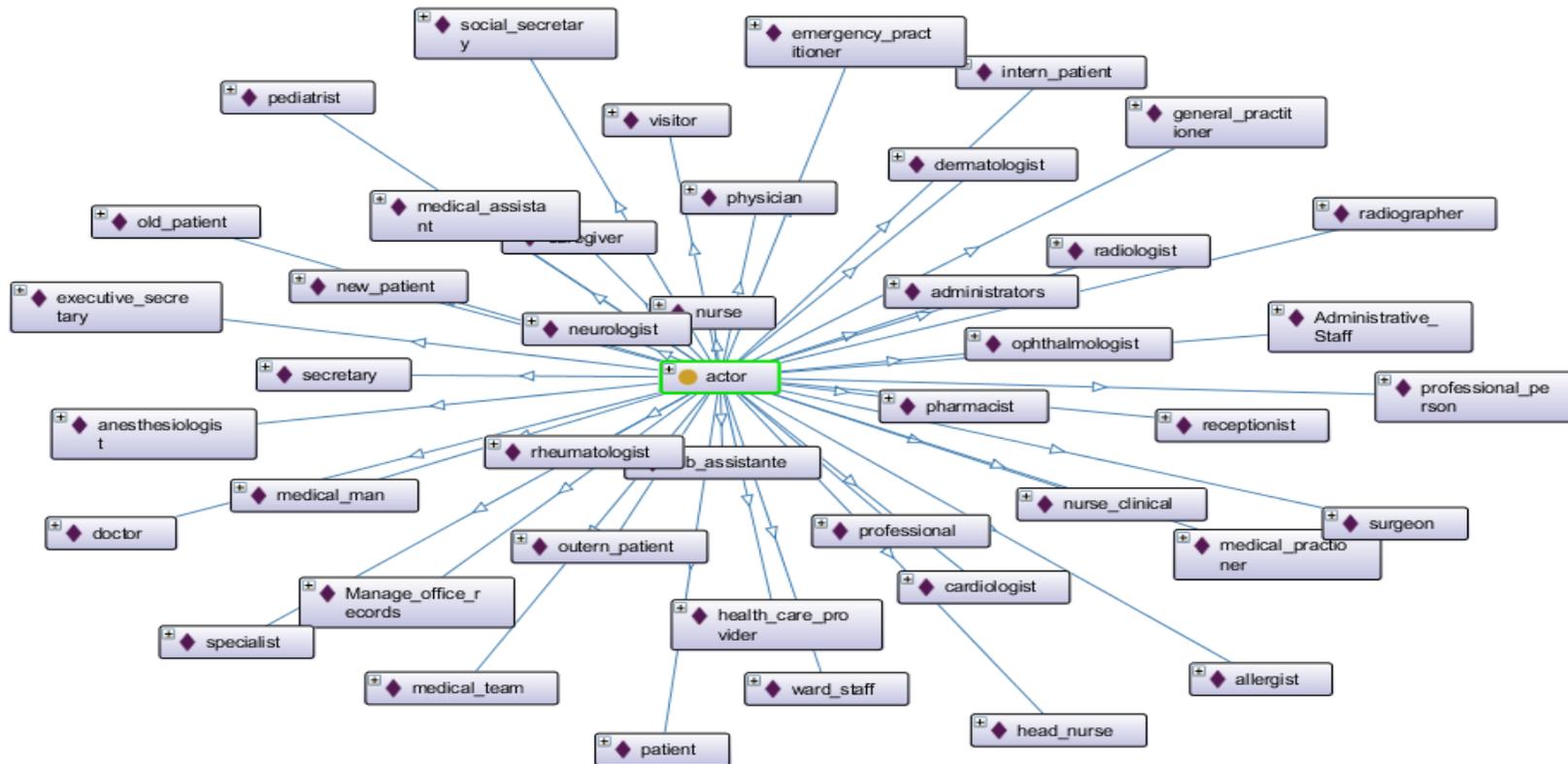


Figure 64: Actor instances

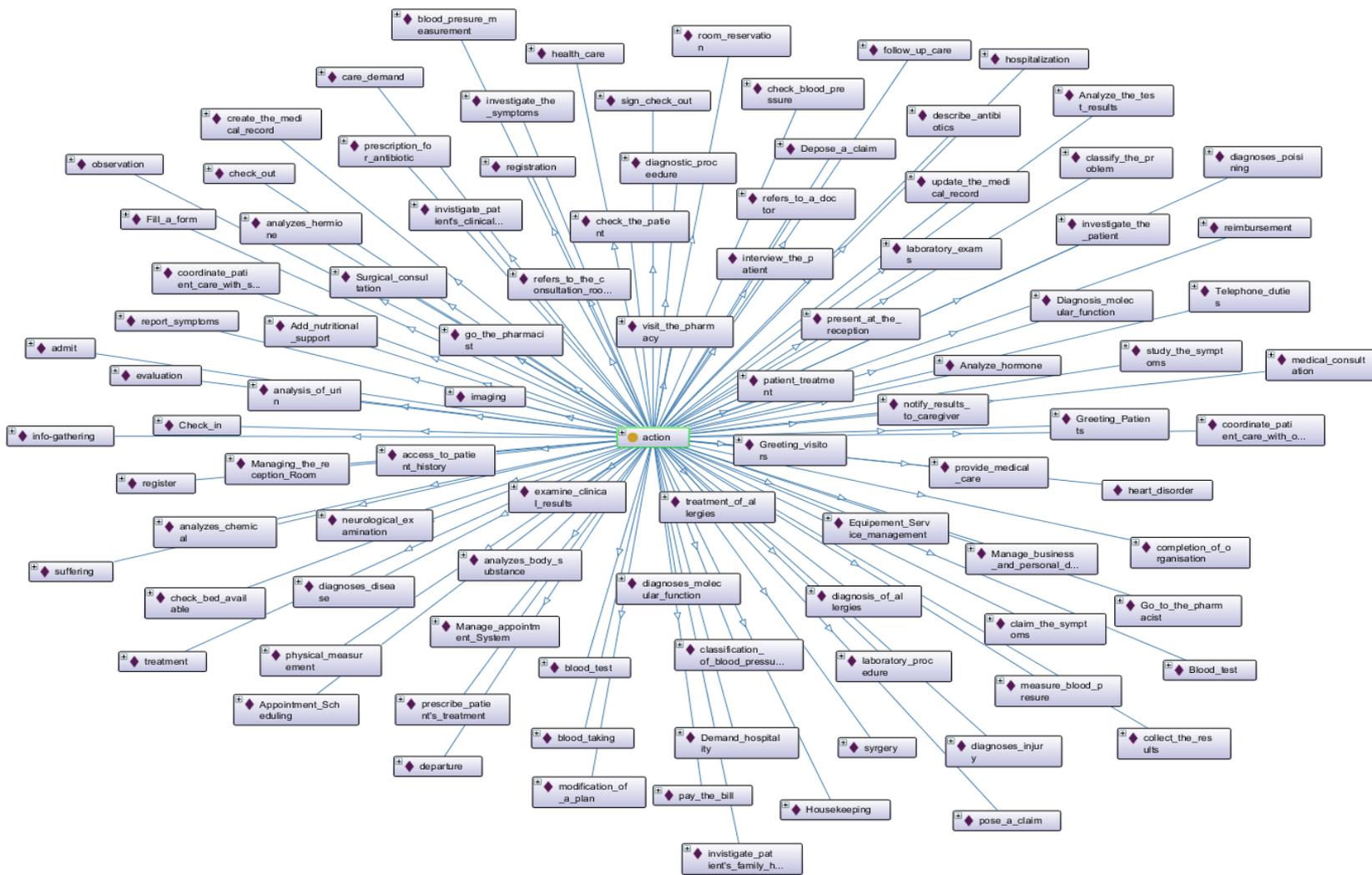
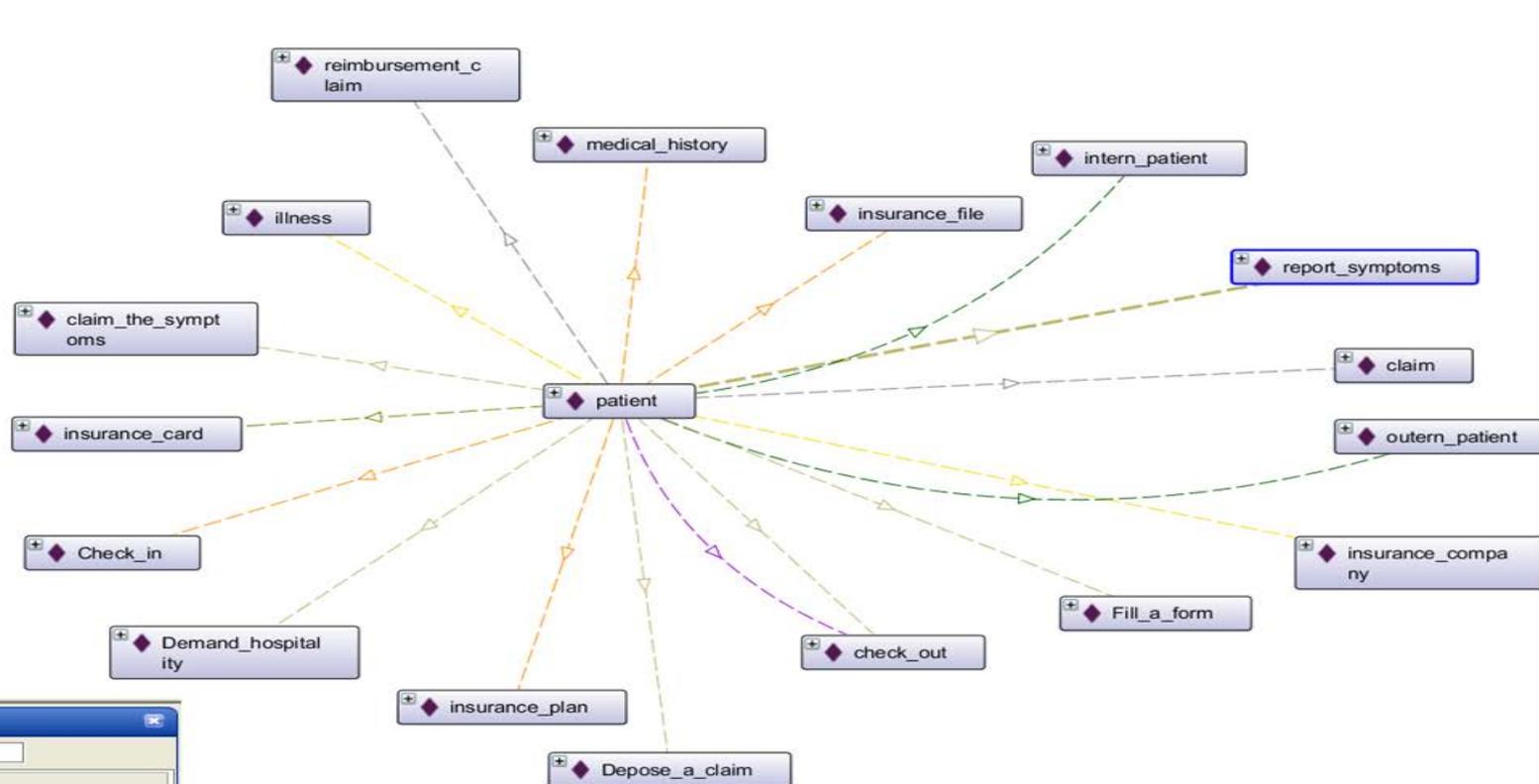


Figure 65: Action instances



Arc Types

type filter text

- assigned_to
- create
- has individual
- has subclass
- is_a
- operate
- owner
- perform
- require
- view

Figure 69: Patient relations

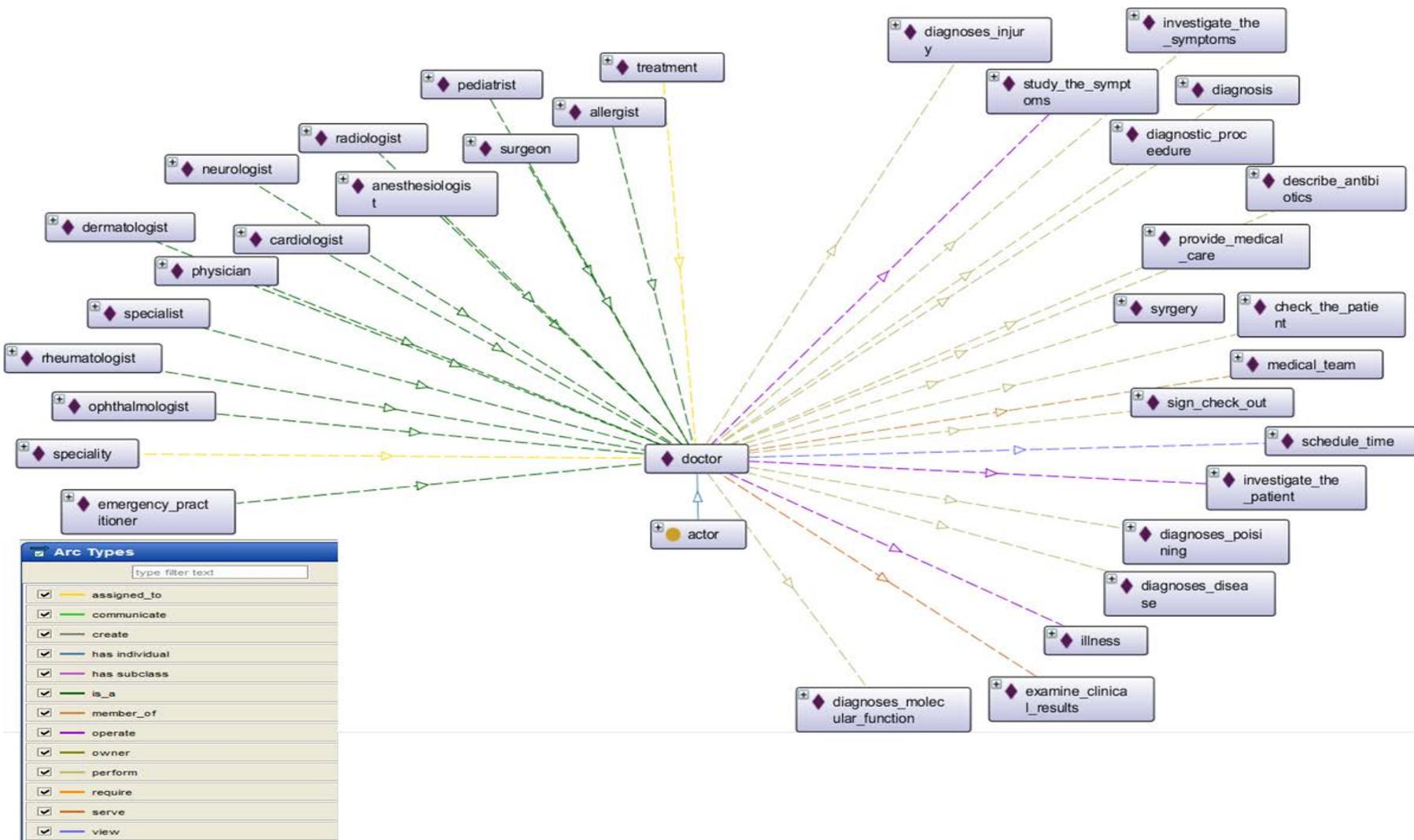


Figure 70: Doctor relations

Bibliography

- « A Weighted Coupling Metric for Business Process Models ». 2013.
- Aalst, Van Der. 1998a. *The Application of Petri Nets to Workflow Management*.
- Aalst, W. M. P. van der. 1999. *Formalization and Verification of Event-driven Process Chains*.
- Aalst, W. M. P. van der, K. M. van Hee, A. H. M. ter Hofstede, N. Sidorova, H. M. W. Verbeek, M. Voorhoeve, et M. T. Wynn. 2011. « Soundness of Workflow Nets: Classification, Decidability, and Analysis ». *Formal Aspects of Computing* 23 (3) (mai 1): 333-363. doi:10.1007/s00165-010-0161-4.
- Aalst, Wil M. P. van der. 1997. « Verification of Workflow Nets ». In *Proceedings of the 18th International Conference on Application and Theory of Petri Nets*, 407-426. ICATPN '97. London, UK, UK: Springer-Verlag.
- Aalst, Wil M. P. van der, Arthur H. M. ter Hofstede, et Mathias Weske. 2003a. « Business Process Management: A Survey ». In *Business Process Management*, édité par Wil M. P. van der Aalst et Mathias Weske, 1-12. Lecture Notes in Computer Science 2678. Springer Berlin Heidelberg.
- Aalst, W. M. P. Van Der. 1998. *The Application of Petri Nets to Workflow Management*.
- Aalst, Wil Van Der, Kees Van Hee, Prof Dr Kees, Max Hee, Remmert Remmerts De Vries, Jaap Rigter, Eric Verbeek, et Marc Voorhoeve. 2002. *Workflow Management: Models, Methods, and Systems*.
- Abd Ghani, Abdul Azim, Tieng Wei Koh, Geoffrey Muchiri Muketha, et Pei Wen Wong. 2008. « Complexity metrics for measuring the understandability and maintainability of Business Process Models using Goal-Question-Metric (GQM) ». Article. *International Journal of Computer Science and Network Security*. ACKOFF, PROFESSOR RUSSELL L. 1968. « Management misinformation systems ». *Management Decision* 2 (1) (décembre 31): 4-8. doi:10.1108/eb000823.
- Agarwal, Ritu, Atish P. Sinha, et Mohan Tanniru. 1996. « The role of prior experience and task characteristics in object-oriented modeling: an empirical study ». *International Journal of Human-Computer Studies* 45 (6) (décembre): 639-667. doi:10.1006/ijhc.1996.0072.
- Aguilar, Elvira, Francisco Ruiz, Félix García, et Mario Piattini. 2006. « Applying Software Metrics to evaluate Business Process Models ». *CLEI Electron. J.* 9 (1).
- Aguilar-Savén, Ruth Sara. 2004a. « Business process modelling: Review and framework ». *International Journal of Production Economics* 90 (2) (juillet 28): 129-149.
- Ahern, Dennis M, Aaron Clouse, et Richard Turner. 2008. *CMMI Distilled: A Practical Introduction to Integrated Process Improvement*. Upper Saddle River, NJ: Addison-Wesley.
- Aldin, Laden, et Sergio de Cesare. 2009. « A Comparative Analysis Of Business Process Modelling Techniques ». *UK Academy for Information Systems Conference Proceedings 2009* (mars 31).
- Alexander, Christopher, Sara Ishikawa, et Murray Silverstein. 1977a. *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press.
- Alexander, Patricia A. 1992. « Domain Knowledge: Evolving Themes and Emerging Concerns ». *Educational Psychologist* 27 (1): 33-51. doi:10.1207/s15326985ep2701_4.
- Alexandra Galatescu, Taisia Greceanu. 2004. « Ontologies for Analysis and Improvement of Business Process Quality in a Virtual Enterprise. » 117-128.

- « Analysis of most common Process Modelling Mistakes in BPMN Process Models ». 2013. *Business Process Incubator*.
- Antón, Annie I., et Colin Potts. 1998. « The Use of Goals to Surface Requirements for Evolving Systems ». In *Proceedings of the 20th International Conference on Software Engineering*, 157–166. ICSE '98. Washington, DC, USA: IEEE Computer Society.
- Anupindi, Ravi, Sunil Chopra, Sudhakar D. Deshmukh, Jan A. Van Mieghem, et Eitan Zemel. 2011. *Managing Business Process Flows*. 3^e éd. Prentice Hall.
- Arpinar, I. Budak, Ruoyan Zhang, Boanerges Aleman-Meza, et Angela Maduko. 2005. « Ontology-Driven Web Services Composition Platform ». *Information Systems and E-Business Management* 3 (2) (juillet 1): 175-199. doi:10.1007/s10257-005-0055-9.
- Assenova, Petia, et Paul Johannesson. 1996. « Improving quality in conceptual modelling by the use of schema transformations ». In *Proceedings 15 th Int. Conf. of Conceptual Modelling (ER'96), Cotbus*, 277–291. AddisonWesley.
- Axenath, Björn, Ekkart Kindler, et Vladimir Rubin. 2005. *The Aspects of Business Processes: An Open and Formalism Independent Ontology*.
- Ayora, Clara, Victoria Torres, Barbara Weber, Manfred Reichert, et Vicente Pelechano. 2013. « Enhancing Modeling and Change Support for Process Families through Change Patterns ». In *Enterprise, Business-Process and Information Systems Modeling*, édité par Selmin Nurcan, Henderik A. Proper, Pnina Soffer, John Krogstie, Rainer Schmidt, Terry Halpin, et Ilia Bider, 246-260. Lecture Notes in Business Information Processing 147. Springer Berlin Heidelberg.
- Badica, C., M. Teodorescu, C. Spahiu, A. Badica, et C. Fox. 2005. « Integrating role activity diagrams and hybrid IDEF for business process modeling using MDA ». In *Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2005. SYNASC 2005*, 4 pp. - . doi:10.1109/SYNASC.2005.40.
- Badica, costin, et chris Fox. 2003. « A formal IDEF-Based notation for business process modelling ». *CAISE*: 535-549.
- Basili, Victor R. 1993. « The Experience Factory and its relationship to other Improvement Paradigms ». In *Software Engineering — ESEC '93*, édité par Ian Sommerville et Manfred Paul, 68-83. Lecture Notes in Computer Science 717. Springer Berlin Heidelberg.
- Batra, D., et J. G. Davis. 2013. « Conceptual Data Modelling in Database Design : Similarities and Differences between Expert and Novice Designers ». *International Journal of Man-Machine Studies* 37 (1): 83-101.
- Batra, Dinesh, et Joseph G. Davis. 1992. « Conceptual data modelling in database design: similarities and differences between expert and novice designers ». *International Journal of Man-Machine Studies* 37 (1) (juillet): 83-101. doi:10.1016/0020-7373(92)90092-Y.
- Becker, Jörg, Michael Rosemann, et Christoph von Uthmann. 2000a. « Guidelines of Business Process Modeling ». In *IN: BUSINESS PROCESS MANAGEMENT: MODELS, TECHNIQUES AND EMPIRICAL STUDIES. EDS*, 30–49. Springer-Verlag.
- Belle, Jean-paul Van. *A Framework for the Evaluation of Business Models and its Empirical Validation*.
- Bergholtz, Maria, et Paul Johannesson. 2001. « Classifying the Semantics of Relationships in Conceptual Modeling by Categorization of Roles ». In , 199-203.
- Birkmeier, Dominik, Sebastian Klöckner, et Sven Overhage. 2010a. « AN EMPIRICAL COMPARISON OF THE USABILITY OF BPMN AND UML ACTIVITY DIAGRAMS FOR BUSINESS USERS ».

- Boehm, Barry W. 1981. *Software Engineering Economics*. 1^{re} éd. Prentice Hall.
- Bosilj-Vuksic, Vesna, George M. Giaglis, et Vlatka Hlupic. 2001. « IDEF diagrams and petri nets for business process modeling: suitability, efficacy, and complementary use ». In *Enterprise information systems II*, édité par Bernadette Sharp, Jose Cordeiro, et Joaquim Filipe. Boston, USA: Kluwer Academic Publishers.
- « BPM Academic Initiative | Signavio ». 2013.
- « BPMN Tutorial Gateways - BPMN-Community ». 2013.
- Brucker, Achim D., Matthias P. Krieger, Delphine Longuet, et Burkhart Wolff. 2011. « A specification-based test case generation method for UML/OCL ». In *Proceedings of the 2010 international conference on Models in software engineering*, 334–348. MODELS'10. Berlin, Heidelberg: Springer-Verlag.
- Buchman, J., et Christian Harsana Ekadharmawan. 2009. « Barriers to sharing domain knowledge in software development practice in SMEs ». In *Proceedings of the 3rd international workshop on knowledge collaboration in software development (KCSD2009)*, 2–16.
- Buijs, Joos, Marcello La Rosa, Hajo A. Reijers, Boudewijn van Dongen, et Wil M. P. van der Aalst. 2013. « Improving business process models using observed behavior ». In *SIMPDA 2012*, édité par P. Cudre-Mauroux, P. Ceravolo, et D. Gasevic, 162:44–59. Campione d'Italia, Italy: Springer.
- Burton-Jones, Andrew, et Ron Weber. 1999. « Understanding Relationships with Attributes in Entity-relationship Diagrams ». In *Proceedings of the 20th International Conference on Information Systems*, 214–228. ICIS '99. Atlanta, GA, USA: Association for Information Systems.
- « Business-Driven Development ». 2005. CT316. décembre 9.
- Caetano, Artur, Marielba Zacarias, Antonio Rito Silva, et Jose Tribolet. 2005. « A Role-Based Framework for Business Process Modeling ». In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 1 - Volume 01*, 13.3–. HICSS '05. Washington, DC, USA: IEEE Computer Society. doi:10.1109/HICSS.2005.49.
- Cardoso, Jorge. 2006. « Process control-flow complexity metric: An empirical validation ». In *IEEE International Conference on Services Computing, 2006. SCC '06*, 167–173. doi:10.1109/SCC.2006.82.
- Cardoso, Jorge. 2007. « Business process quality metrics: log-based complexity of workflow patterns ». In *Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems: CoopIS, DOA, ODBASE, GADA, and IS - Volume Part I*, 427–434. OTM'07. Berlin, Heidelberg: Springer-Verlag.
- Cardoso, Jorge. « Control-flow Complexity Measurement of Processes and Weyuker's Properties ». In *in: 6th International Enformatika Conference, Transactions on Enformatika, Systems Sciences and Engineering*, 213–218.
- Chan, Nguyen Ngoc, Walid Gaaloul, et Samir Tata. 2012. « Assisting Business Process Design by Activity Neighborhood Context Matching ». In *Service-Oriented Computing*, édité par Chengfei Liu, Heiko Ludwig, Farouk Toumani, et Qi Yu, 541–549. Lecture Notes in Computer Science 7636. Springer Berlin Heidelberg.
- Chen, Kai, Janos Sztipanovits, Sherif Abdelwalhed, et Ethan Jackson. 2005. « Semantic anchoring with model transformations ». In *In ECMDA-FA, volume 3748 of LNCS*, 115–129. Springer.
- Cherfi, Samira Si-said, Jacky Akoka, et Isabelle Comyn-Wattiau. 2006. « Use Case Modeling and Refinement: A Quality-Based Approach ». In *Conceptual Modeling - ER 2006*, édité par David W. Embley, Antoni Olivé, et Sudha Ram, 84–97. Lecture Notes in Computer Science 4215. Springer Berlin Heidelberg. Cherfi, Samira Si-Saïd, Isabelle

- Comyn-Wattiau, et Jacky Akoka. 2008. « Quality Patterns for Conceptual Modelling ». In *Proceedings of the 27th International Conference on Conceptual Modeling*, 142–153. ER '08. Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-540-87877-3_12.
- Chiesi, Harry L., George J. Spilich, et James F. Voss. 1979. « Acquisition of domain-related information in relation to high and low domain knowledge ». *Journal of Verbal Learning and Verbal Behavior* 18 (3) (juin): 257–273. doi:10.1016/S0022-5371(79)90146-4.
- Clark, Malcolm, Yunhyong Kim, Udo Kruschwitz, Dawei Song, Dyaa Albakour, Stephen Dignum, Ulises Cerviño Beresi, Maria Fasli, et Anne De Roeck. 2012. « Automatically structuring domain knowledge from text: An overview of current research ». *Information Processing & Management* 48 (3) (mai): 552–568. doi:10.1016/j.ipm.2011.07.002.
- cuiligqing. 2013. « YATO Yet Another Top-level Ontology ». *Docstoc.com*.
- Daniyal, Ali, Samina Raza Abidi, et Syed Sibte Raza Abidi. 2009. « Computerizing Clinical Pathways: Ontology-Based Modeling and Execution ». *Studies in Health Technology and Informatics* 150: 643–647.
- Davenport, Thomas H. 1993. *Process Innovation: Reengineering Work Through Information Technology*. Harvard Business Press.
- Davenport, Thomas H. 1998. « Putting the enterprise into the enterprise system ». *Harvard Bus. Rev.* 76 (4) (juillet): 121–131.
- Davies, Islay, Peter Green, Michael Rosemann, Marta Indulska, et Stan Gallo. 2006. « How do practitioners use conceptual modeling in practice? ». *Data Knowl. Eng.* 58 (3) (septembre): 358–380. doi:10.1016/j.datak.2005.07.007.
- Dehnert, Juliane, et Peter Rittgen. 2001. « Relaxed Soundness of Business Processes ». In *Proceedings of the 13th International Conference on Advanced Information Systems Engineering*, 157–170. CAiSE '01. London, UK, UK: Springer-Verlag.
- Deming, W. Edwards. 2000. *Out of the Crisis*. Cambridge, Mass.: MIT Press.
- Dijkman, Remco M., Marlon Dumas, et Chun Ouyang. 2008. « Semantics and analysis of business process models in BPMN ». *Information and Software Technology* 50 (12) (novembre): 1281–1294. doi:10.1016/j.infsof.2008.02.006.
- Dongen, B. F. Van, et J. Mendling. 2006. « Structural Patterns for Soundness of Business Process Models ». In *In EDOC '06: Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06)*, 116–128. IEEE Computer Society.
- Dumas, Marlon, et Arthur H. M. ter Hofstede. 2001. « UML Activity Diagrams as a Workflow Specification Language ». In , 76–90. Springer Verlag.
- Ehrig, Marc, Agnes Koschmider, et Andreas Oberweis. 2007. « Measuring similarity between semantic business process models ». In *Proceedings of the fourth Asia-Pacific conference on Conceptual modelling - Volume 67*, 71–80. APCCM '07. Darlinghurst, Australia, Australia: Australian Computer Society, Inc.
- Eick, Christoph F. 1991. « A Methodology for the Design and Transformation of Conceptual Schemas ». In *Proceedings of the 17th International Conference on Very Large Data Bases*, 25–34. VLDB '91. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Eid-Sabbagh, Rami-Habib, et Mathias Weske. 2013. « Analyzing Business Process Architectures ». In *Advanced Information Systems Engineering*, édité par Camille Salinesi, Moira C. Norrie, et Óscar Pastor, 208–223. Lecture Notes in Computer Science 7908. Springer Berlin Heidelberg.

- Endres, Dr Albert, et Prof Dieter Rombach. 2003. *A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories*. 1^{re} éd. Addison Wesley.
- Eriksson, Hans-Erik, et Magnus Penker. 2012. « Business Modeling With UML: Business Patterns at Work ».
- Eshuis, Rik, et Roel Wieringa. 2002. « Comparing Petri Net and Activity Diagram Variants for Workflow Modelling - A Quest for Reactive Petri Nets ». In *In Weber et al*, 321–351. Springer.
- Esswein, Werner, Andreas Gehlert, et Grit Seiffert. 2004. « Towards a Framework for Model Migration ». In *Advanced Information Systems Engineering*, édité par Anne Persson et Janis Stirna, 463-476. Lecture Notes in Computer Science 3084. Springer Berlin Heidelberg.
- Fahland, Dirk, Cédric Favre, Barbara Jobstmann, Jana Koehler, Niels Lohmann, Hagen Völzer, et Karsten Wolf. 2009. « Instantaneous Soundness Checking of Industrial Business Process Models ». In *Business Process Management*, édité par Umeshwar Dayal, Johann Eder, Jana Koehler, et Hajo A. Reijers, 278-293. Lecture Notes in Computer Science 5701. Springer Berlin Heidelberg.
- Fellbaum, Christine. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- Ferdian. 2001. « A Comparison of Event-Driven Process Chains and UML Activity Diagram for Denoting Business Processes ». Studienarbeit, Technische Universität Hamburg-Harburg.
- Gamma, Erich, Richard Helm, Ralph Johnson, et John Vlissides. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. 1st éd. Addison-Wesley Professional. Part of the Addison-Wesley Professional Computing Series series.
- García, Félix, Mario Piattini, Francisco Ruiz, Gerardo Canfora, et Corrado A. Visaggio. 2006. « FMESP: Framework for the modeling and evaluation of software processes ». *Journal of Systems Architecture* 52 (11) (novembre): 627-639. doi:10.1016/j.sysarc.2006.06.007.
- GEAMBASU, Cristina Venera. 2012. « BPMN vs. UML Activity Diagram for Business Process Modeling ». *Journal of Accounting and Management Information Systems* 11 (4): 637-651.
- Gemino, Andrew, et Yair Wand. 2005. « Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties ». *Data & Knowledge Engineering* 55 (3) (décembre): 301-326. doi:10.1016/j.datak.2004.12.009.
- Gruhn, Volker, et Ralf Laue. 2007. « What business process modelers can learn from programmers ». *Science of Computer Programming* 65 (1) (mars 1): 4-13. doi:10.1016/j.scico.2006.08.003.
- Gruhn, Volker, et Ralf Laue. 2010. « A heuristic method for detecting problems in business process models ». *Business Process Management Journal* 16 (5) (septembre 14): 806-821. doi:10.1108/14637151011076485.
- Gruhn, Volker, et Ralf Laue. « Complexity metrics for business process models ». In *in: W. Abramowicz, H.C. Mayr (Eds.), 9th International Conference on Business Information Systems (BIS 2006), Lecture Notes in Informatics*, 1–12.
- Guarino, Nicola. 1998. « Formal Ontology and Information Systems ». In , 3–15. IOS Press.
- Guizzardi, Giancarlo. 2007. « On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models ». In *Proceedings of the 2007 conference on Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference DB&IS'2006*, 18–39. Amsterdam, The Netherlands, The Netherlands: IOS Press.

- H. James Nelson, Geert Poels. 2005. « Quality in conceptual modeling: five examples of the state of the art. » *Data Knowl. Eng.* 55: 237-242. doi:10.1016/j.datak.2004.12.004.
- Hallerbach, A., T. Bauer, et M. Reichert. 2009. « Guaranteeing Soundness of Configurable Process Variants in Provop ». In *IEEE Conference on Commerce and Enterprise Computing, 2009. CEC '09*, 98-105. doi:10.1109/CEC.2009.62.
- Hammer, Michael, et James Champy. 2003. *Reengineering the Corporation: A Manifesto for Business Revolution*. New York: HarperBusiness Essentials.
- Harmon, Paul. 2003. *Business Process Change: A Manager's Guide to Improving, Redesigning, and Automating Processes*. Morgan Kaufmann.
- Harrington, H. James. 1998. « Performance improvement: the rise and fall of reengineering ». *The TQM Magazine* 10 (2) (avril 1): 69-71. doi:10.1108/09544789810211353.
- Havey, Michael. 2005. *Essential Business Process Modeling*. 1^{re} éd. O'Reilly Media.
- Heinrich, R., et B. Paech. 2010. « Defining the Quality of Business Processes. » *Modellierung*: 133-148.
- Heravizadeh, Mitra, Jan Mendling, et Michael Rosemann. 2009. « Dimensions of Business Processes Quality (QoBP) ». In *Business Process Management Workshops*, édité par Danilo Ardagna, Massimo Mecella, et Jian Yang, 80-91. Lecture Notes in Business Information Processing 17. Springer Berlin Heidelberg.
- Hollingsworth, D. 1995. « Workflow Management Coalition - The Workflow Reference Model ». Workflow Management Coalition.
- Holschke, O., P. Gelpke, P. Offermann, et C. Schröpfer. 2008. « Business Process Improvement by Applying Reference Process Models in SOA-a Scenario-based Analysis ». *Multikonferenz Wirtschaftsinformatik. GITO-Verlag, Berlin, München, Germany*.
- Holschke, Oliver, et Christian Schröpfer. *Business Process Improvement by Applying Reference Process Models in SOA – a Scenario-based Analysis*.
- « IBM Blueworks Live ». 2013.
- « ISO 8601 ». 2013. *Wikipedia, the Free Encyclopedia*.
- Jiang, Jay J., et David W. Conrath. 1997. « Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy ». *arXiv:cmp-lg/9709008*
- Johansson, Lars-Olof, Magnus Wärja, Harald Kjellin, et Sven Carlsson. 2008. « Graphical Modeling Techniques and Usefulness in the Model Driven Arcitechture : Which Are the Criteria for a "Good" Computer Independent Model? » In Mittuniversitet.
- Jr, Jack L. Brock, John P. Finedore, et Deborah A. Davis. 1997. *Business Process Reengineering Assessment Guide*. DIANE Publishing.
- Juran, Joseph M., et A. Blanton Godfrey. 1999. *Juran's Quality Handbook*. McGraw Hill.
- Kaiya, H., et M. Saeki. 2005. « Ontology based requirements analysis: lightweight semantic processing approach ». In *Fifth International Conference on Quality Software, 2005. (QSIC 2005)*, 223-230. doi:10.1109/QSIC.2005.46.
- Kaiya, H., et M. Saeki. 2006. « Using Domain Ontology as Domain Knowledge for Requirements Elicitation ». In *Requirements Engineering, 14th IEEE International Conference*, 189 -198. doi:10.1109/RE.2006.72.
- Kent, S., S. Gaito, et N. Ross. 1999. « A Meta-model Semantics for Structural Constraints in UML ». Book section.
- Khatri, Vijay, et Iris Vessey. 2010a. « Information use in solving a well-structured IS problem: the roles of IS and application domain knowledge ». In *Proceedings of the 29th international conference on Conceptual modeling*, 46-58. ER'10. Berlin, Heidelberg: Springer-Verlag.

- Khatri, Vijay, Iris Vessey, V. Ramesh, Paul Clay, et Sung-Jin Park. 2006. « Understanding Conceptual Schemas: Exploring the Role of Application and IS Domain Knowledge ». Research-article.
- Khlif, Wiem, Lobna Makni, Nahla Zaaboub, et Hanene Ben-Abdallah. 2009. « Quality metrics for business process modeling ». In *Proceedings of the 9th WSEAS international conference on Applied computer science*, 195–200. ACS'09. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS).
- Khlif, Wiem, Nahla Zaaboub, et Hanene Ben-Abdallah. 2010. « Coupling metrics for business process modeling ». *W. Trans. on Comp.* 9 (1) (janvier): 31–41.
- Kindler, Ekkart. 2004. « On the Semantics of EPCs: A Framework for Resolving the Vicious Circle ». In *International Conference on Business Process Management (BPM 2004), volume 3080 of Lecture Notes in Computer Science*, 82–97. Springer-Verlag.
- Ko, Ryan, Stephen Lee, et Eng Lee. 2009. « Business process management (BPM) standards: a survey ». *Business Process Management Journal* 15 (5): 744–791. doi:10.1108/14637150910987937.
- Kobryn, Cris. 1999. « UML 2001: a standardization odyssey ». *Commun. ACM* 42 (10) (octobre): 29–37. doi:10.1145/317665.317673.
- Koehler, Jana, et Jussi Vanhatalo. 2007. « IBM Research | Technical Paper Search | Process Anti-Patterns: How to Avoid the Common Traps of Business Process Modeling(Search Reports) ». ZZ999.
- Kosslyn, S M. 1988. « Aspects of a Cognitive Neuroscience of Mental Imagery ». *Science (New York, N.Y.)* 240 (4859) (juin 17): 1621–1626.
- Kouneli, Aggeliki, Georgia Solomou, Christos Pierrakeas, et Achilles Kameas. 2012. « Modeling the Knowledge Domain of the Java Programming Language as an Ontology ». In *Advances in Web-Based Learning - ICWL 2012*, édité par Elvira Popescu, Qing Li, Ralf Klamma, Howard Leung, et Marcus Specht, 152–159. Lecture Notes in Computer Science 7558. Springer Berlin Heidelberg.
- Krogstie, John. 1998. « Integrating the understanding of quality in requirements specification and conceptual modeling ». *SIGSOFT Softw. Eng. Notes* 23 (1) (janvier): 86–91. doi:10.1145/272263.272285.
- Krogstie, John. 2002. « A Semiotic Approach to Quality in Requirements Specifications ». In *Proceedings of the IFIP TC8 / WG8.1 Working Conference on Organizational Semiotics: Evolving a Science of Information Systems*, 231–249. Deventer, The Netherlands, The Netherlands: Kluwer, B.V.
- Krogstie, John, Odd Ivar Lindland, et Guttorm Sindre. 1995a. « Defining quality aspects for conceptual models ». In *Proceedings of the IFIP international working conference on Information system concepts: Towards a consolidation of views*, 216–231. London, UK, UK: Chapman & Hall, Ltd.
- Krogstie, John, Guttorm Sindre, et Håvard Jørgensen. 2006. « Process Models Representing Knowledge for Action: A Revised Quality Framework ». *European Journal of Information Systems* 15 (1): 91–102. doi:10.1057/palgrave.ejis.3000598.
- Krogstie, John, Arne Sølvberg, 2000. « Information Systems Engineering: Conceptual Modeling in a quality perspective ».
- Kühne, Stefan, Heiko Kern, Volker Gruhn, et Ralf Laue. 2010. « Business Process Modeling with Continuous Validation ». *Journal of Software Maintenance and Evolution: Research and Practice* 22 (6-7): 547–566. doi:10.1002/smr.517.
- Kunze, Matthias, Alexander Luebbe, Matthias Weidlich, et Mathias Weske. 2011. « Towards Understanding Process Modeling – The Case of the BPM Academic Initiative ». In *Business Process Model and Notation*, édité par Remco Dijkman, Jörg Hofstetter, et

- Jana Koehler, 44-58. *Lecture Notes in Business Information Processing 95*. Springer Berlin Heidelberg.
- Lankhorst, Marc. 2013. *Enterprise Architecture at Work Modelling, Communication and Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg : Imprint: Springer.
- Lars-Olof, Johansson, Warja Magnus, et Carlsson Sven. 2009. « An evaluation of business process model techniques, using Moody's quality criterion for a good diagram ».
- Leondes, Cornelius T. 2010. *Intelligent Knowledge-Based Systems: Business and Technology in the New Millennium*. Springer.
- Leopold, Henrik, Sergey Smirnov, et Jan Mendling. 2010. « Refactoring of process model activity labels ». In *Proceedings of the Natural language processing and information systems, and 15th international conference on Applications of natural language to information systems*, 268–276. NLDB'10. Berlin, Heidelberg: Springer-Verlag.
- Leopold, Henrik, Sergey Smirnov, et Jan Mendling.. 2011. « Recognising Activity Labeling Styles in Business Process Models ». *Enterprise Modelling and Information Systems Architectures* 6 (1): 16-29.
- Leopold, Henrik, Sergey Smirnov, et Jan Mendling.. 2012. « On the refactoring of activity labels in business process models ». *Information Systems* 37 (5) (juillet): 443-459. doi:10.1016/j.is.2012.01.004.
- « Levenshtein Distance ». 2013. *Wikipedia, the Free Encyclopedia*.
- Lin, Dekang. 1998. « An Information-Theoretic Definition of Similarity ». In *In Proceedings of the 15th International Conference on Machine Learning*, 296–304. Morgan Kaufmann.
- Lindland, O.I., G. Sindre, et A. Solvberg. 1994a. « Understanding quality in conceptual modeling ». *IEEE Software* 11 (2): 42-49. doi:10.1109/52.268955.
- « Logizian 10.2 ». 2013.
- Luo, Wenhong, et Y. Alex Tung. 1999. « A framework for selecting business process modeling methods ». *Industrial Management & Data Systems* 99 (7) (novembre 1): 312-319. doi:10.1108/02635579910262535.
- Malone, Thomas W., et Kevin Crowston. 1994. « The interdisciplinary study of coordination ». *ACM Comput. Surv.* 26 (1) (mars): 87–119. doi:10.1145/174666.174668.
- Masolo, Claudio, Stefano Borgo, Aldo Gangemi, Nicola Guarino, et Alessandro Oltramari. 2003. « WonderWeb Deliverable D18 : Ontology Library (final) ». Trento, Italy: Laboratory For Applied Ontology - ISTC-CNR.
- McAllister, Chad A. 2006. *Requirements Determination of Information Systems: User and Developer Perceptions of Factors Contributing to Misunderstandings*. ProQuest.
- McCusker, James P., Joanne Luciano, et Deborah McGuinness. 2013. « Towards an Ontology for Conceptual Modelling ».
- McGuinness, Deborah L. 2011. « Towards an Ontology for Conceptual Modeling ». « MEGA Process ». 2013.
- Mendling, J., M. Moser, G. Neumann, H. M. W. Verbeek, et B. F. Van Dongen. 2006. « Faulty EPCs in the SAP Reference Model ». In eds, *International Conference on Business Process Management (BPM 2006)*, 451–457. Springer-Verlag.
- Mendling, J., H. M. W. Verbeek, B. F. van Dongen, W. M. P. Aalst, et G. Neumann. 2007. *Detection and Prediction of Errors in EPCs of the SAP Reference Model*.
- Mendling, Jan. 2009. « Empirical Studies in Process Model Verification ». In *Transactions on Petri Nets and Other Models of Concurrency II*, édité par Kurt Jensen et Wil M. P. van der Aalst, 208-224. *Lecture Notes in Computer Science 5460*. Springer Berlin Heidelberg.

- Mendling, Jan, et Wil Van Der Aalst. 2007. « Formalization and Verification of EPCs with OR-Joins Based on State and Context ». In *Proceedings of CAiSE 2007, LNCS 4495*, 439–453. Springer-Verlag.
- Mendling, Jan, Gustaf Neumann, et Markus Nüttgens. 2005. « Yet Another Event-driven Process Chain - Modeling Workflow Patterns with yEPCs ». *Enterprise Modelling and Information Systems Architectures, EMISA 1*: 3–13.
- Mendling, Jan, Gustaf Neumann, et Wil Van Der Aalst. 2007a. « Understanding the occurrence of errors in process models based on metrics ». In *Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems: CoopIS, DOA, ODBASE, GADA, and IS - Volume Part I*, 113–130. OTM'07. Berlin, Heidelberg: Springer-Verlag.
- Mendling, Jan, Jan Recker, et Hajo A. Reijers. 2010. « On the Usage of Labels and Icons in Business Process Modeling ». *International Journal of Information System Modeling and Design 1 (2)*: 40–58. doi:10.4018/jismd.2010040103.
- Mendling, Jan, Hajo A. Reijers, et Jorge Cardoso. 2007. « What Makes Process Models Understandable? ». In *Business Process Management*, édité par Gustavo Alonso, Peter Dadam, et Michael Rosemann, 48–63. Lecture Notes in Computer Science 4714. Springer Berlin Heidelberg.
- Mendling, Jan, Hajo Reijers, et Wil M. van der Aalst. 2008. « Seven Process Modeling Guidelines (7PMG) ». *Faculty of Science and Technology*.
- Mendling, Jan, Laura Sánchez-González, Félix García, et Marcello La Rosa. 2012. « Thresholds for error probability measures of business process models ». *Journal of Systems and Software 85 (5) (mai)*: 1188–1197. doi:10.1016/j.jss.2012.01.017.
- Mendling, Jan, et Mark Strembeck. 2008. « Influence Factors of Understanding Business Process Models ». In *Business Information Systems*, édité par Witold Abramowicz et Dieter Fensel, 142–153. Lecture Notes in Business Information Processing 7. Springer Berlin Heidelberg.
- « Michael Hammer and James Champy On Business process reengineering (BPR) ». 2013.
- Moody, Daniel. 2003. « Dealing with Complexity in Information Systems Modeling: Development and Empirical Validation of a Method for Representing Large Data Models ». *ICIS 2003 Proceedings* (décembre 31).
- Moody, Daniel.. 2007. « What Makes a Good Diagram? Improving the Cognitive Effectiveness of Diagrams in IS Development ». In *Advances in Information Systems Development*, édité par Wita Wojtkowski, W. Gregory Wojtkowski, Jože Zupancic, Gabor Magyar, et Gabor Knapp, 481–492. Springer US.
- Moody, Daniel L. 1998. « Metrics for Evaluating the Quality of Entity Relationship Models ». In *Conceptual Modeling – ER '98*, édité par Tok-Wang Ling, Sudha Ram, et Mong Li Lee, 211–225. Lecture Notes in Computer Science 1507. Springer Berlin Heidelberg.
- Moody, Daniel L. 2005. « Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions ». *Data & Knowledge Engineering 55 (3) (décembre)*: 243–276. doi:10.1016/j.datak.2004.12.005.
- Moody, Daniel L., et Graeme G. Shanks. 1994. « What makes a good data model? Evaluating the quality of entity relationship models ». In *Entity-Relationship Approach — ER '94 Business Modelling and Re-Engineering*, édité par Pericles Loucopoulos, 94–111. Lecture Notes in Computer Science 881. Springer Berlin Heidelberg.
- Moody, Daniel L., Graeme G. Shanks, et Peta Darke. 1998. « Improving the Quality of Entity Relationship Models—Experience in Research and Practice ». In *Conceptual Modeling – ER '98*, édité par Tok-Wang Ling, Sudha Ram, et Mong Li Lee, 255–276. Lecture Notes in Computer Science 1507. Springer Berlin Heidelberg.

- Moody, Daniel L., Guttorm Sindre, Terje Brasethvik, et Arne Sølvberg. 2002. « Evaluating the Quality of Process Models: Empirical Testing of a Quality Framework ». In *Proceedings of the 21st International Conference on Conceptual Modeling*, 380–396. ER '02. London, UK, UK: Springer-Verlag.
- Muehlen, Michael Zur, et Jan Recker. 2008. « How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation ». In *Proceedings of the 20th international conference on Advanced Information Systems Engineering*, 465–479. CAiSE '08. Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-540-69534-9_35.
- Murata, T. 1989. « Petri nets: Properties, analysis and applications ». *Proceedings of the IEEE* 77 (4): 541–580. doi:10.1109/5.24143.
- Nelson, H. James, Geert Poels, Marcela Genero, et Mario Piattini. 2012. « A conceptual modeling quality framework ». *Software Quality Control* 20 (1) (mars): 201–228. doi:10.1007/s11219-011-9136-9.
- Nurcan, Selmin, et Marc-Henri Edme. 2005. « Intention-Driven Modeling for Flexible Workflow Applications ». *Software Process: Improvement and Practice* 10 (4): 363–377. doi:10.1002/spip.240.
- Nysetvold, Anna Gunhild, et John Krogstie. 2005. « Assessing Business Process Modeling Languages Using a Generic Quality Framework ». In *In Proceedings of the CAiSE'05 Workshops*, 545–556. Idea Group.
- OMG Final Adopted Specificatio. 2006. « Business Process Modeling Notation Specification » (février).
- Omg, Object Management Group. 2004. « Business Process Modeling Notation (BPMN) Version 1.0 » 50 (October 11, 2010): 170. doi:10.1007/s11576-008-0096-z.
- Ouyang, Chun, Marlon Dumas, et Arthur H. M. Ter Hofstede. *From Business Process Models to Process-oriented Software Systems: The BPMN to BPEL Way* ★.
- Overhage, PD Dr Sven, Dominik Q. Birkmeier, et Sebastian Schlauderer. 2012. « Quality Marks, Metrics, and Measurement Procedures for Business Process Models ». *Business & Information Systems Engineering* 4 (5) (octobre 1): 229–246. doi:10.1007/s12599-012-0230-8.
- Panetto, H., M. Dassisti, et A. Tursi. 2012. « ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment ». *Advanced Engineering Informatics* 26 (2) (avril): 334–348. doi:10.1016/j.aei.2011.12.002.
- Pankratius, Victor, et Wolfried Stucky. 2005. « A Formal Foundation for Workflow Composition, Workflow View Definition, and Workflow Normalization Based on Petri Nets ». In *In Proc. Second Asia-Pacific Conference on Conceptual Modelling (APCCM2005)*, 79–88. Society, Inc.
- Patron, P., E. Miguelanez, Y.R. Petillot, D.M. Lane, et J. Salvi. 2008. « Adaptive mission plan diagnosis and repair for fault recovery in autonomous underwater vehicles ». In *OCEANS 2008*, 1–9. doi:10.1109/OCEANS.2008.5151975.
- Paulk, Mark C. 1995. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, Mass.: Addison-Wesley Pub. Co.
- Penker, Magnus, et Hans-Erik Eriksson. 2000. *Business Modeling With UML: Business Patterns at Work*. 1^{re} éd. Wiley.
- Pirró, Giuseppe. 2009. « A semantic similarity metric combining features and intrinsic information content ». *Data Knowl. Eng.* 68 (11) (novembre): 1289–1308. doi:10.1016/j.datak.2009.06.008.
- Poels, Geert, Ken Decreus, Ben Roelens, et Monique Snoeck. 2013. « Investigating Goal-Oriented Requirements Engineering for Business Processes ».

- Poels, Geert, Ann Maes, Frederik Gailly, et Roland Paemeleire. 2011. « The Pragmatic Quality of Resources- Events-Agents Diagrams: An Experimental Evaluation ». *Information Systems Journal* 21 (1): 63–89. doi:10.1111/j.1365-2575.2007.00253.x.
- Purandare, Amruta, et Ted Pedersen. *Word Sense Discrimination by Clustering Contexts in Vector and Similarity Spaces*.
- Purao, Sandeep, et Veda C. Storey. 2005. « A multi-layered ontology for comparing relationship semantics in conceptual models of databases ». *Appl. Ontol.* 1 (1) (janvier): 117–139.
- Ravichandran, T., et Arun Rai. 1994. « The dimensions and correlates of systems development quality ». In *Proceedings of the 1994 computer personnel research conference on Reinventing IS : managing information technology in changing organizations: managing information technology in changing organizations*, 272–282. SIGCPR '94. New York, NY, USA: ACM. doi:10.1145/186281.186456.
- Recker, J., M. zur Muehlen, K. Siau, J. Erickson, et M. Indulska. 2009. « Measuring Method Complexity: UML versus BPMN ». In , 1-10. Association for Information Systems.
- Reeves, Carol A., et David A. Bednar. 1994. « Defining Quality: Alternatives and Implications ». *The Academy of Management Review* 19 (3) (juillet): 419. doi:10.2307/258934.
- Reijers, H.A., et S. Liman Mansar. 2005. « Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics ». *Omega* 33 (4) (août): 283-306. doi:10.1016/j.omega.2004.04.012.
- Reijers, Hajo A. 2005. « Process Design and Redesign ». In *Process-Aware Information Systems*, édité par rlon Dumas, Wil M. P. van der Aalst, et Arthur H. M. ter Hofstede, 205–234. John Wiley & Sons, Inc.
- Reijers, Hajo A., et Irene T. P. Vanderfeesten. 2004. « Cohesion and Coupling Metrics for Workflow Process Design ». In *Business Process Management*, édité par Jörg Desel, Barbara Pernici, et Mathias Weske, 290-305. Lecture Notes in Computer Science 3080. Springer Berlin Heidelberg.
- Resnik, Philip. 1995. « Using information content to evaluate semantic similarity in a taxonomy ». In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, 448–453. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Riehle, Dirk, et Heinz Züllighoven. 1995. « A Pattern Language for Tool Construction and Integration Based on the Tools and Materials Metaphor ». In *Pattern Languages of Program Design*, 9–42. Addison-Wesley.
- Rittgen, Peter. 2010. « Quality and perceived usefulness of process models ». In *Proceedings of the 2010 ACM Symposium on Applied Computing*, 65–72. SAC '10. New York, NY, USA: ACM. doi:10.1145/1774088.1774105.
- Ross, Sheldon. 1976. « A First Course in Probability par Sheldon Ross: MacMillan Hardcover - AbeBooks.fr ».
- Rozinat, A., et W. M. P. Van Der Aalst. 2013. « Conformance testing : Measuring the fit and appropriateness of event logs and process models ». In *Lecture notes in computer science*, 163-176. Springer.
- Rozum, J. 1993. « Concepts on Measuring the Benefits of Software Process Improvement ». *Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-93-TR-009*.
- Russell, Nick, Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, et David Edmond. 2005. « Workflow Resource Patterns: Identification, Representation and Tool Support ». In *Advanced Information Systems Engineering*, édité par Oscar Pastor et João Falcão e

- Cunha, 216-232. *Lecture Notes in Computer Science 3520*. Springer Berlin Heidelberg.
- Russell, Nick, Arthur H. M. ter Hofstede, David Edmond, et Wil M. P. van der Aalst. 2005. « Workflow Data Patterns: Identification, Representation and Tool Support ». In *Conceptual Modeling – ER 2005*, édité par Lois Delcambre, Christian Kop, Heinrich C. Mayr, John Mylopoulos, et Oscar Pastor, 353-368. *Lecture Notes in Computer Science 3716*. Springer Berlin Heidelberg.
- Russell, Nick, Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, et Petia Wohed. 2006. « On the suitability of UML 2.0 activity diagrams for business process modelling ». In *Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling - Volume 53*, 95-104. APCCM '06. Darlinghurst, Australia, Australia: Australian Computer Society, Inc.
- Sánchez-González, Laura, Félix García, Francisco Ruiz, et Mario Piattini. 2013. « TOWARD A QUALITY FRAMEWORK FOR BUSINESS PROCESS MODELS ». *International Journal of Cooperative Information Systems* 22 (01) (mars): 1350003. doi:10.1142/S0218843013500032.
- Sánchez-González, Laura, Francisco Ruiz, Félix García, et Mario Piattini. 2013. « Improving Quality of Business Process Models ». In *Evaluation of Novel Approaches to Software Engineering*, édité par Leszek A. Maciaszek et Kang Zhang, 130-144. *Communications in Computer and Information Science 275*. Springer Berlin Heidelberg.
- Scheer, August-Wilhelm, et Markus Nüttgens. 2000. « ARIS Architecture and Reference Models for Business Process Management ». In *Business Process Management, Models, Techniques, and Empirical Studies*, 376-389. London, UK, UK: Springer-Verlag.
- Scheer, August-Wilhelm, et Kristof Schneider. 2006. « ARIS — Architecture of Integrated Information Systems ». In *Handbook on Architectures of Information Systems*, édité par Assoc Professor Dr Peter Bernus, Professor Dr Kai Mertins, et Professor Dr Günter Schmidt, 605-623. *International Handbooks on Information Systems*. Springer Berlin Heidelberg.
- Seel, Christian, et Dominik Vanderhaeghen. 2005. « Meta-model based extensions of the epc for inter-organisational process modelling ». In *IN: PROCEEDINGS 4TH GI-WORKSHOP EPK 2005 - GESCHÄFTSPROZESSMANAGEMENT*, 117-136.
- Shanks, Graeme. 2007. « Conceptual Data Modelling: An Empirical Study of Expert and Novice Data Modellers ». *Australasian Journal of Information Systems* 4 (2) (juillet 29). doi:10.3127/ajis.v4i2.360.
- Shanks, Graeme, Elizabeth Tansley, et Ron Weber. 2003. « Using ontology to validate conceptual models ». *Commun. ACM* 46 (10) (octobre): 85-89. doi:10.1145/944217.944244.
- Siau, K., et Xin Tan. 2005. « Technical communication in information systems development: the use of cognitive mapping ». *IEEE Transactions on Professional Communication* 48 (3): 269-284. doi:10.1109/TPC.2005.853933.
- Siau, Keng, et Qing Cao. 2001. « Unified Modeling Language: A Complexity Analysis ». *Journal of Database Management* 12 (1): 26-34. doi:10.4018/jdm.2001010103.
- Siau, Keng, John Erickson, et LihYunn Lee. 2005. « Theoretical vs. Practical Complexity: The Case of UML ». *Journal of Database Management* 16 (3): 40-57. doi:10.4018/jdm.2005070103.
- Simsion, Graeme C. 1994. *Data Modeling Essentials: Analysis, Design, and Innovation*. Van Nostrand Reinhold.

- Soffer, Pnina, Maya Kaner, et Yair Wand. 2012. « Towards Understanding the Process of Process Modeling: Theoretical and Empirical Considerations ». In *Business Process Management Workshops*, édité par Florian Daniel, Kamel Barkaoui, et Schahram Dustdar, 357-369. Lecture Notes in Business Information Processing 99. Springer Berlin Heidelberg.
- Soffer, Pnina, et Yair Wand. 2004. « Goal-driven analysis of process model validity ». In *IN: PROCEEDINGS OF CAISE'04*, 521-535. Springer-Verlag.
- Storey, Veda C., Roger Chiang, et G. Lily Chen. 2005. « Ontology Creation: Extraction of Domain Knowledge from Web Documents ». In *Conceptual Modeling – ER 2005*, édité par Lois Delcambre, Christian Kop, Heinrich C. Mayr, John Mylopoulos, et Oscar Pastor, 256-269. Lecture Notes in Computer Science 3716. Springer Berlin Heidelberg.
- Storey, Veda C., Debabrata Dey, Harald Ullrich, et Shankar Sundaresan. 1998. « An ontology-based expert system for database design ». *Data Knowl. Eng.* 28 (1) (octobre): 31-46. doi:10.1016/S0169-023X(98)00012-3.
- Stvilia, Besiki. 2007. « A Model for Ontology Quality Evaluation ». *First Monday* 12 (12) (novembre 21).
- « Suggested Upper Merged Ontology ». 2013. *Wikipedia, the Free Encyclopedia*.
- Sugumaran, Vijayan, et Veda C. Storey. 2006. « The role of domain ontologies in database design: An ontology management and conceptual modeling environment ». *ACM Transactions on Database Systems (TODS)* 31: 1064-1094.
- « SVO (Subject-Verb-Object) ». 2013. *About.com Grammar & Composition*.
- Talwar, Rohit. 1993. « Business re-engineering—a strategy-driven approach ». *Long Range Planning* 26 (6) (décembre): 22-40. doi:10.1016/0024-6301(93)90204-S.
- Tartir, Samir, I. Budak Arpinar, Michael Moore, Amit P. Sheth, et Boanerges Aleman-meza. 2005. « OntoQA: Metric-based ontology quality analysis ». In *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*.
- « The Idea behind Business Process Improvement : Toward a Business Process Improvement Pattern Framework [Florian Forster] ». 2013a. Text.
- « The Importance of the Conceptual Design Phase ». 2013. *Business & Entrepreneurship - azcentral.com*.
- Thomas, Dr, et M. Fehlmann. *Six Sigma for Software*.
- Tomislav Rozman. 2009. « Analysis of Most Common Process Modelling Mistakes in BPMN Process ... »
- Tscheschner, Willi. 2010. « Transformation from EPC to BPMN ». *Hasso-Plattner-Institute, Potsdam*.
- Tziviskou, Christina, et C. Maria Keet. 2007. « A meta-model for ontologies with ORM2 ». In *Proceedings of the 2007 OTM confederated international conference on On the move to meaningful internet systems - Volume Part I*, 624-633. OTM'07. Berlin, Heidelberg: Springer-Verlag.
- Unterkalmsteiner, M., T. Gorschek, A.K.M.M. Islam, Chow Kian Cheng, R.B. Permadi, et R. Feldt. 2012. « Evaluation and Measurement of Software Process Improvement #x02014;A Systematic Literature Review ». *IEEE Transactions on Software Engineering* 38 (2): 398-424. doi:10.1109/TSE.2011.26.
- V Keith, Swenson V. Jim Farris. « Human-Centered Business Process Management ».
- Van-Belle, Jean-Paul. 2004. « Empirical Evaluation and Comparison of Enterprise Models: A Framework and Its Application ». *PowerShow*.
MDJiN/Empirical_Evaluation_and_Comparison_of_Enterprise_Models_A_Framework_k_and_Its_Application_powerpoint_ppt_presentation.

- Van der Aalst, W. M. P. 1999. « Woflan: a Petri-net-based workflow analyzer ». *Syst. Anal. Model. Simul.* 35 (3) (mai): 345–357.
- Van Der Aalst, W. M. P., A. H. M. Ter Hofstede, B. Kiepuszewski, et A. P. Barros. 2003. « Workflow Patterns ». *Distrib. Parallel Databases* 14 (1) (juillet): 5–51. doi:10.1023/A:1022883727209.
- Van der Aalst, W.M.P., et A.H.M. ter Hofstede. 2005. « YAWL: yet another workflow language ». *Information Systems* 30 (4) (juin): 245–275. doi:10.1016/j.is.2004.02.002.
- Van der Aalst, Wil, Arya Adriansyah, et Boudewijn van Dongen. 2012. « Replaying history on process models for conformance checking and performance analysis ». *Wiley Int. Rev. Data Min. and Knowl. Disc.* 2 (2) (mars): 182–192. doi:10.1002/widm.1045.
- Van Herk, D. 2006. « Activity Monitoring Buzz or Business. Master thesis Information Science ».
- Vanhatalo, Jussi, Hagen Völzer, et Frank Leymann. 2013. « Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition ». In *Service-Oriented Computing – ICSOC 2007*, édité par Bernd J. Krämer, Kwei-Jay Lin, et Priya Narasimhan, 4749:43–55. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Verbeek, H. M. W., T. Basten, et W. M. P. van der Aalst. 1999. « Diagnosing Workflow Processes Using Woflan ». *THE COMPUTER JOURNAL* 44: 2001.
- Vergidis, K., A. Tiwari, et B. Majeed. 2008. « Business Process Analysis and Optimization: Beyond Reengineering ». *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 38 (1): 69–82. doi:10.1109/TSMCC.2007.905812.
- Vesna Bosilj-Vuksic, George M. Giaglis. 2000. « IDEF Diagrams and Petri Nets for Business Process Modelling: Suitability, Efficacy, and Complementary Use. » 242–247.
- Vessey, Iris, et Sue Conger. 1993. « Learning to Specify Information Requirements: The Relationship Between Application and Methodology ». *J. Manage. Inf. Syst.* 10 (2) (septembre): 177–201.
- Wahl, Terje, et Guttorm Sindre. 2005a. « An Analytical Evaluation of BPMN Using a Semiotic Quality Framework ». In *In CAiSE'05 Workshops. Volume*, 533–544.
- Walrad, C., et E. Moss. 1993. « Measurement: The key to application development quality ». *IBM Systems Journal* 32 (3): 445–460. doi:10.1147/sj.323.0445.
- Wand, Yair, et Richard Y. Wang. 1996. « Anchoring data quality dimensions in ontological foundations ». *Commun. ACM* 39 (11) (novembre): 86–95. doi:10.1145/240455.240479.
- Weber, Barbara, Manfred Reichert, Jan Mendling, et Hajo A. Reijers. 2011. « Refactoring large process model repositories ». *Computers in Industry* 62 (5) (juin): 467–486. doi:10.1016/j.compind.2010.12.012.
- Weber, Barbara, Manfred Reichert, et Stefanie Rinderle-Ma. 2008. « Change patterns and change support features – Enhancing flexibility in process-aware information systems ». *Data & Knowledge Engineering* 66 (3) (septembre): 438–466. doi:10.1016/j.datak.2008.05.001.
- Weske, Mathias. 2012. *Business Process Management: Concepts, Languages, Architectures*. Springer.
- Wetzstein, Branimir, Zhilei Ma, Agata Filipowska, Monika Kaczmarek, Sami Bhiri, Silvestre Losada, Jose-Manuel Lopez-Cob, et Laurent Cicurel. 2007. « Semantic Business Process Management: A Lifecycle Based Requirements Analysis. » In *SBPM*, édité par Martin Hepp, Knut Hinkelmann, Dimitris Karagiannis, Rüdiger Klein, et Nenad Stojanovic. Vol. 251. CEUR Workshop Proceedings. CEUR-WS.org.

- « What is business process quality management? definition and meaning ». 2013. *BusinessDictionary.com*.
- « while - English-French Dictionary WordReference.com ». 2013.
- White. 2009. « Defining Business Rules ~ What Are They Really? (Abstract and Table of Contents) ».
- White, Stephen. 2004. « Process Modeling Notations and Workflow Patterns ».
- White, Stephen A. 2004. « Introduction to BPMN ». *Elements* 15 (c): 1-11. doi:10.3727/000000006783982421.
- White, Stephen A. « Process Modeling Notations and Workflow Patterns ».
- Wiley, Jennifer. 1998. « Expertise as Mental Set: The Effects of Domain Knowledge in Creative Problem Solving ». *Memory & Cognition* 26 (4) (juillet 1): 716-730. doi:10.3758/BF03211392.
- Wohed, P., W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede, et N. Russell. 2006. « On the Suitability of BPMN for Business Process Modelling ». In *Business Process Management*, édité par Schahram Dustdar, José Luiz Fiadeiro, et Amit P. Sheth, 161-176. Lecture Notes in Computer Science 4102. Springer Berlin Heidelberg.
- Wolf, Karsten. 2007. « Generating Petri net state spaces ». In *Proceedings of the 28th international conference on Applications and theory of Petri nets and other models of concurrency*, 29-42. ICATPN'07. Berlin, Heidelberg: Springer-Verlag.
- Wu, Zhibiao, et Martha Palmer. 1994. « Verbs semantics and lexical selection ». In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, 133-138. ACL '94. Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/981732.981751.
- Zha, Haiping, Yun Yang, Jianmin Wang, et Lijie Wen. 2008. « Transforming XPDL to Petri Nets ». In *Business Process Management Workshops*, édité par Arthur ter Hofstede, Boualem Benatallah, et Hye-Young Paik, 197-207. Lecture Notes in Computer Science 4928. Springer Berlin Heidelberg.
- Zultner, Richard E. 1993. « TQM for technical teams ». *Commun. ACM* 36 (10) (octobre): 79-91. doi:10.1145/163430.163577.

Published Papers

2013

International Journal with reviewers

S. Si-Said Cherfi , S. Ayad, I. Comyn-Wattiau - Improving Business Process Model Quality Using Domain Ontologies, Springer JoDS - Journal on Data Semantics , vol. 2(special issue on Evolution and Versioning in Semantic Data Integration Systems), pp. 75-87, 2013, (doi:10.1007/s13740-013-0022-4)

2012

International Conference with reviewers

S. Ayad , S. Si-Said Cherfi , I. Comyn-Wattiau - Towards A Semantic Quality Based Approach for Business Process Models Improvement, 23rd Australasian Conference on Information Systems (ACIS'12), March 2012, pp.1-11, Geelong, Victoria, Australia,

S. Ayad , S. Si-Said Cherfi - Domain Knowledge Based Quality for Business Process Models, The 17th International Conference on Information Quality (ICIQ'12), November 2013, pp.70-84,

S. Si-Said Cherfi , S. Ayad , I. Comyn-Wattiau - Aligning Business Process Models and Domain Knowledge: A Meta-Modeling Approach, 16th East-European Conference on Advances in Databases and Information Systems (ADBIS'12), January 2012, Vol. 186, pp.45-56, Series Advances in Intelligent Systems and Computing series, (DOI: 10.1007/978-3-642-32741-4_5)

S. Ayad - A Quality Based Approach for the Analysis and Design of Business Process Models, IEEE Sixth International Conference on Research Challenges in Information Science (RCIS), November 2013, pp.1-5,

National Conference with reviewers

S. Ayad - Gestion de la qualité des Modèles de Processus Métier: Méthode et Outil, INFORSID, November 2013, pp.601-609,

AYAD Sarah

Business Process Models Quality: Evaluation and Improvement

Résumé

La problématique scientifique abordée correspond à la modélisation et à l'amélioration des processus métiers. Ce problème est d'un intérêt croissant pour les entreprises qui prennent conscience de l'impact indéniable que peuvent avoir une meilleure compréhension et une meilleure gestion des processus métiers (PM) sur l'efficacité, la cohérence et la transparence de leurs activités. Le travail envisagé dans le cadre de la thèse vise à proposer une méthode et un outil pour mesurer et améliorer la qualité des modèles de processus métier. L'originalité de l'approche est qu'elle vise non seulement la qualité syntaxique mais aussi la qualité sémantique et pragmatique en s'appuyant notamment sur les connaissances du domaine.

Mots clés : Modelisation des processus metier, qualite de model, qualite semantique, connaissance de domaine, ontologie de domaine, evaluation de la qualite, amelioration de la qualite.

Abstract :

Business process modelling has been receiving increasing attention from companies as well as the academic community, since it is likely to improve practice and working processes in organizations

in the future. Indeed, to keep up and remain competitive, companies must be able to assess the quality of their products and the efficiency of their services. This requires understanding the current system and analyzing the potential benefits of the underlying processes. Using models helps reaching such an objective as they provide an abstract and coherent vision of the business and facilitates understanding and reasoning about alternative solutions to reach an agreement (Nurcan et Edme 2005). Business process models (BPM) can be realized through methodologies, techniques, or software, in a way that helps organizations bring together processes and their context including people, documents, information sources, organizational structures, and applications (Van Herk 2006). As a methodology, BPM help organizations in gaining control over their business processes by modelling, validating, analyzing, and monitoring the processes.

Keywords : Business Process Modeling, model quality, semantic quality, domain knowledge, domain ontology, quality evaluation, quality improvement.