

## CHAPITRE 3

# Adaptation des processus d'entreprise

**ERIC ANDONOFF, CHIHAB HANACHI,  
SELMIN NURCAN**

## 1. CONTEXTE

Il est maintenant reconnu que le succès économique d'une entreprise dépend, entre autres, de la qualité de son Système d'Information (SI). De plus, depuis quelque temps maintenant, l'analyse et la conception des systèmes d'information d'entreprise s'articulent autour des processus : on parle de Systèmes d'Information orientés Processus (Process Aware Information System dans la littérature anglophone) [Dumas et al., 2002]. L'analyse et la conception d'un SI d'entreprise ne sont plus uniquement abordées à travers les données et les applications, mais intègrent les processus qui permettent de mieux mettre en cohérence, ou aligner, le SI avec l'usage qui en sera fait par les acteurs de l'entreprise.

D'autre part, le monde économique dynamique, concurrentiel et ouvert dans lequel les entreprises évoluent les amène à ajuster fréquemment et rapidement les processus pour les mettre en conformité avec leurs pratiques de travail (prise en compte des nouveaux besoins clients, prise en compte de nouvelles réglementations, etc.). Il en résulte que le succès économique d'une entreprise dépend de la capacité de son SI à intégrer les changements que l'environnement impose à l'entreprise. Cela vient justifier l'importance que revêt la problématique de l'adaptation des processus d'entreprise [Smith et Fingar, 2003], [Reijers, 2006], [Weske, 2007].

Les Systèmes d'Information orientés Processus (SIOp) constituent le cadre dans lequel s'inscrit ce chapitre. L'objet de l'adaptation sont ici les processus ou un réseau de processus. Nous considérons plus particulièrement des processus (i) explicites, c'est à dire pour lesquels il est possible de définir un schéma, et (ii) potentiellement exécutables à l'aide d'outils informatiques tels que des Systèmes de Gestion de Workflow, des ERP ou des orchestrateurs de services.

## 1.1 Procédé d'adaptation

Le procédé d'adaptation introduit en préface de cet ouvrage définit les invariants de l'adaptation et identifie quatre étapes qui s'instancient de la manière suivante dans le cadre de ce chapitre.

L'étape d'*observation du système* a pour objectif d'identifier les éléments à observer ; elle indique également les moyens d'observation. Dans le contexte des processus d'entreprise, ce sont les processus en fonctionnement dans une entreprise qui sont observés. Ces processus peuvent s'exécuter avec ou sans support informatique. Plus précisément, ce sont des instances de processus (ou cas) qui sont observées, autrement dit des traces d'exécution, manuelles ou automatiques, des processus. Ces traces décrivent le travail effectivement réalisé par les acteurs de l'entreprise : tâches réalisées, durée, informations utilisées et produites... Les moyens utilisés pour cette observation peuvent être des démarches d'analyse du travail réalisé par les acteurs de l'entreprise dans le cas de processus non informatisés, ou des outils d'administration et de supervision de l'exécution des processus informatisés produisant des traces numériques (log).

L'étape d'*interprétation des données prélevées* a pour objectif d'indiquer ce que représentent ces éléments et comment les qualifier. Nous identifions un ensemble d'indicateurs pour les processus en fonctionnement qui peuvent être organisés en trois aspects :

1) Aspects comportementaux (les tâches et la structure de contrôle du processus) :

- La conformité entre les cas observés et leur schéma. Il s'agit de déterminer l'écart entre le processus prescrit (modèle) et ce qui est effectivement réalisé (instance). Ces différences concernent les tâches et la manière dont elles s'enchaînent.
- La distribution des cas (instances) parmi les différents chemins possibles décrits dans leur schéma. Il peut en être déduit le chemin le plus fréquenté ou des chemins jamais parcourus.

2) Aspects organisationnels (les acteurs et leurs modes de collaboration) :

- Le nombre d'acteurs impliqués dans les cas.
- La structure des communications entre acteurs. Il s'agit d'identifier les réseaux que constituent les acteurs lorsqu'ils délèguent et/ou sous-traitent des activités.

- Les groupes de compétences qui sont des ensembles de personnes pouvant réaliser les mêmes activités.
- Les acteurs incontournables.

3) Aspect performance :

- Temps d'exécution moyen, minimum et maximum des cas.
- Temps d'exécution moyen, minimum et maximum pour chaque chemin.
- Temps moyen d'exécution pour chaque tâche.

Les valeurs prises par ces indicateurs permettent de diagnostiquer des dysfonctionnements.

La troisième étape du procédé d'adaptation est l'*identification des ajustements à opérer* pour résoudre les dysfonctionnements de coordination, d'organisation ou de performance diagnostiqués précédemment. Les ajustements peuvent amener à adapter les schémas de processus au niveau de la structure de contrôle, des tâches, de l'organisation (structure, effectif) impliquée dans le processus, des informations manipulées ou des politiques d'attribution de tâches. Il est à remarquer que certains ajustements sont d'ordre managérial tandis que d'autres relèvent de l'ingénierie des processus.

Enfin, la quatrième étape du procédé d'adaptation, appelée application *des ajustements*, indique comment opérer les ajustements à apporter aux processus. Les solutions proposées dans la littérature pour mener à bien ces ajustements seront décrites plus loin ; elles reposent sur une typologie de l'adaptation (par flexibilité, par déviation, par évolution) et sur l'étape du cycle de vie dans laquelle elles opèrent.

## **1.2 Définitions préliminaires et cadre d'analyse**

Avant de rentrer plus avant dans les détails de l'adaptation des processus, il est nécessaire de définir le vocabulaire du domaine, d'introduire le cycle de vie des processus et de présenter une typologie de l'adaptation des processus en rapport avec ce cycle de vie.

### **1.2.1 Concepts de base**

Nous introduisons dans un premier temps les notions de schéma de processus et d'instances de processus, appelées aussi cas.

### 1.2.1.1 *Processus et schéma de processus*

Un *processus* est un ensemble coordonné de tâches mis en œuvre par des acteurs (humains ou matériels) au sein d'une organisation pour aboutir à un résultat déterminé. Il est décrit par un *schéma de processus* qui représente, dans une forme supportant des manipulations automatiques, différentes perspectives inter connectées [Aalst et Hee, 2002].

Les *perspectives* généralement prises en compte sont les perspectives comportementale, organisationnelle, informationnelle et intentionnelle [Aalst et al., 2003-a].

La *perspective comportementale* explicite les tâches et leur coordination (règles d'enchaînements ou structure de contrôle). De nombreux patrons ont été proposés dans la littérature pour aider à définir cette structure de contrôle ; on peut par exemple citer les patrons d'aiguillage de type Ou-exclusif, de synchronisation de type Et, de synchronisation de type Ou, etc. Une étude exhaustive des patrons utiles pour la définition des structures de contrôle est menée dans [Aalst et al., 2003-b].

La *perspective organisationnelle* structure les acteurs impliqués dans la réalisation du processus en groupes d'acteurs pouvant jouer le même rôle ou en groupes d'acteurs appartenant à une même unité organisationnelle. Elle indique également les relations qu'entretiennent ces acteurs entre eux (hiérarchie, délégation, responsabilité...). La notion de rôle est primordiale, car elle permet de définir le mode d'attribution des tâches aux acteurs sans faire référence explicitement aux acteurs, mais plutôt à leurs capacités. Par conséquent, l'indisponibilité ou la suppression d'un acteur n'a aucune influence sur le schéma de processus.

La *perspective informationnelle* représente les structures des formulaires, des documents et des données qui sont utilisés et produits par le processus. Ces informations sont importantes, car leur existence et leurs valeurs peuvent déterminer les conditions d'exécution des tâches.





Enfin, la *perspective intentionnelle* spécifie le but, c'est-à-dire l'objectif du processus. Cette fonction est sous-jacente à la modélisation des processus, mais n'est que rarement exploitée lors de leur mise en œuvre.

De nombreuses notations, souvent graphiques, existent dans la littérature pour représenter des schémas de processus. On peut citer par exemple les notations UML [Engels et al., 2002], EPC [Scheer et al., 2002] ou BPMN [BPMN, 2011], et le formalisme des réseaux de Petri [Desel, 2002], particulièrement certains de ses dialectes (Wf-net [Aalst et Hofstede, 2005], réseaux de Petri à objets [Sibertin, 1985], réseaux de Petri colorés [Russel et

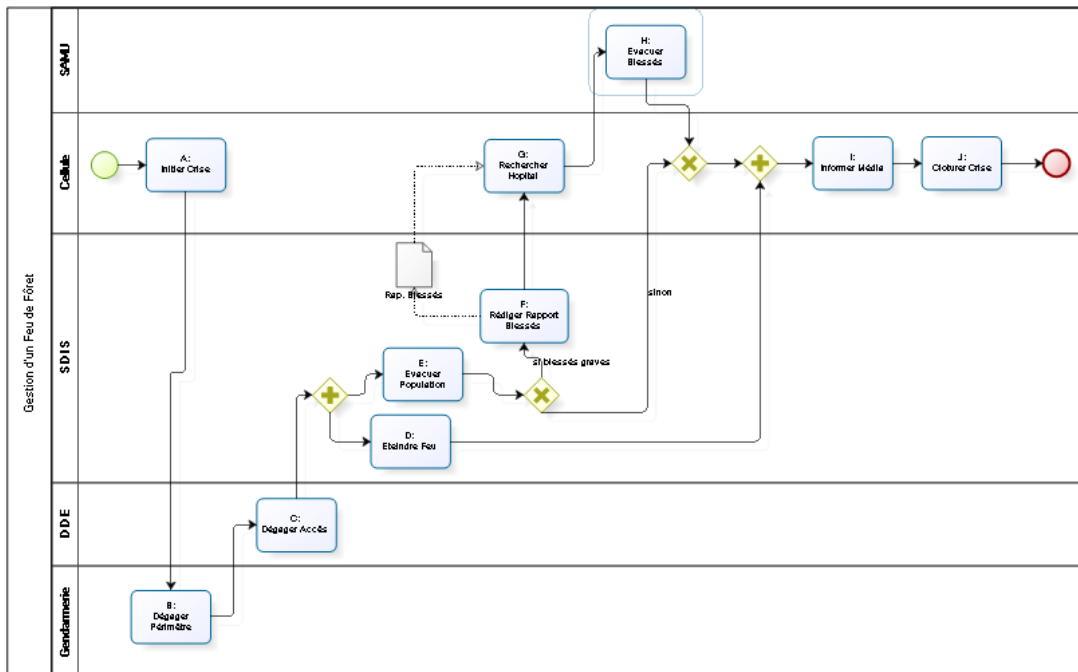
al., 2005]). La notation BPMN, promue par l'OMG, est devenue la notation standard pour la définition des schémas de processus d'entreprise. Elle a l'avantage d'être facile à appréhender par des utilisateurs non informaticiens mais elle se centre principalement sur la perspective comportementale des processus [Wohed et al. 2006]. Par exemple, en BPMN, la perspective organisationnelle se limite uniquement à la notion de rôle via le concept de *lane*.

La Figure 1 ci-après donne un exemple de schéma de processus utilisé dans le cadre de la résolution d'une crise, plus précisément un feu de forêt.

Dans le cadre d'une crise de type feu de forêt, le préfet met en place une cellule de crise où siègent des représentants du SDIS (Service Départemental d'Incendie et de Secours, c'est-à-dire les pompiers), du SAMU, de la gendarmerie et de la DDE (Direction Départementale de l'Équipement) et déroule le processus ad hoc suivant pour faire face à cette crise. Les gendarmes s'occupent dans un premier temps de dégager un périmètre de sécurité pour permettre aux autres acteurs d'intervenir. La DDE intervient ensuite pour déblayer l'accès à la forêt. Le SDIS prend ensuite la main et dépêche des pompiers pour éteindre le feu et évacuer la population des habitations menacées par l'incendie. Le SDIS est également en communication avec la cellule de crise pour gérer les blessés graves qu'il ne peut pas prendre en charge et qui ont besoin de soins d'urgence (par exemple des brûlés). Dans ce cas-là, le SDIS transmet à la cellule de crise un rapport détaillé sur la nature des blessures. Cette dernière, à la lumière de ce rapport, se charge de contacter les hôpitaux de la région les plus adaptés (proximité, présence d'un service spécialisé, disponibilité de lits et de médecins), et sollicite ensuite le SAMU pour évacuer ces blessés vers les hôpitaux sélectionnés. Lorsque l'ensemble des tâches énumérées ci-dessus est terminé (le feu est éteint, les blessés sont soignés...), la cellule de crise informe les médias et clôture ce processus.

Les acteurs de la résolution de la crise sont visualisés dans des lignes (*lanes*) et les tâches menées par ces acteurs sont représentées par des rectangles (A : Initier Crise, B : Dégager Périmètre...). Deux patrons sont utilisés pour synchroniser les tâches réalisées: le patron  pour traduire le parallélisme et le patron  pour traduire l'alternative. Enfin, les cercles  et  représentent respectivement les événements initiant et terminant le processus.

Ce processus symbolique est le plan de référence pour la gestion d'un feu de forêt. Lors d'une occurrence effective d'un feu, la cellule se réunit et peut être amené à l'adapter en fonction des particularités du cas vécu.



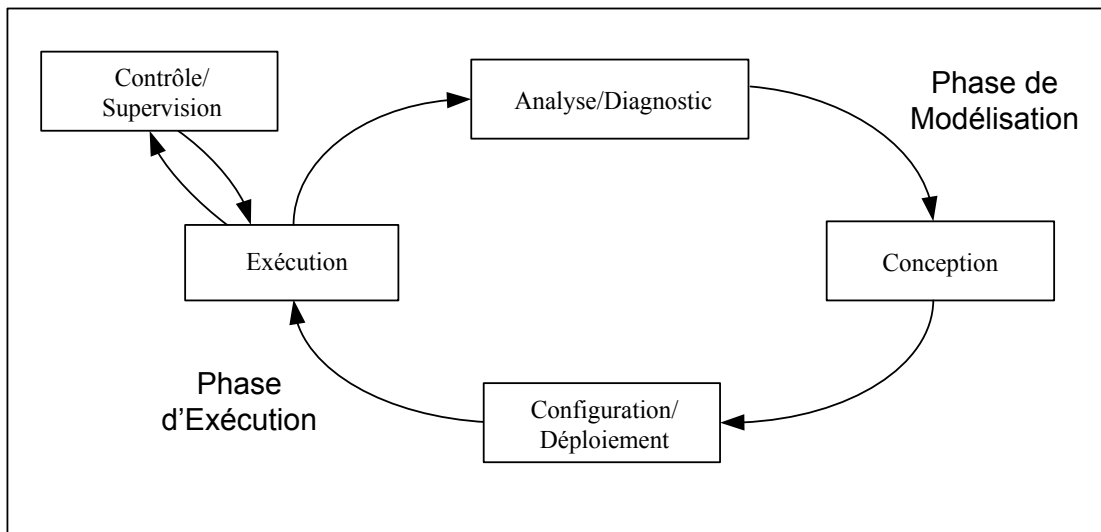
**Figure 1.** Schéma BPMN du processus Gestion d'un Feu de Forêt

### 1.2.1.2 Instance de processus

La réalisation effective d'un processus est appelée instance de processus (ou cas) et correspond à une exécution d'un schéma de processus. L'instance de processus s'exécute conformément à son schéma. Si un schéma de processus définit plusieurs chemins d'exécution possibles, l'instance n'adopte, quant à elle, que l'un de ces chemins. Plusieurs instances d'un même processus peuvent s'exécuter concurremment : il peut y avoir plusieurs feux de forêts simultanés gérés indépendamment. Dans l'exemple adopté, on peut distinguer deux chemins d'exécution possibles : le cas où il n'y a pas de blessés (le chemin A.B.C.(D//E).I.J) et le cas où il y a des blessés ( le chemin A.B.C.(D//(E.F.G.H)).I.J).

### 1.2.2. Cycle de vie des processus

Le cycle de vie des processus est composé de cinq étapes décrites dans la Figure 2. L'adaptation peut porter sur chacune de ces étapes.



**Figure 2.** Cycle de vie des processus

L'*étape de conception* a pour objectif la modélisation des processus. Elle inclut la définition des tâches menées dans le cadre des processus modélisés, la définition de l'enchaînement de ces tâches (ou structure de contrôle), la définition des acteurs et informations impliqués dans le processus et enfin la définition de l'objectif du processus. Des notations, souvent graphiques (BPMN ou ad-hoc), servent de support à cette étape. Il est également possible, dans cette étape, de s'assurer de la validité des schémas de processus produits. Des outils supportant la simulation et la vérification de propriétés sont utilisables pour mener à bien cette validation.

L'*étape de configuration/déploiement* a pour but de produire une forme exécutable du processus. Il s'agit d'informatiser les tâches restées à un niveau abstrait à l'issue de l'étape précédente, de connecter le processus au SI de l'organisation (rôles à relier aux acteurs réels, via LDAP par exemple ; accès effectifs aux données via les SGBD utilisés, etc), et de paramétrer le logiciel support (paramètres d'exécution, droits des utilisateurs). Cette étape peut également comprendre des tests à échelle réelle avant le déploiement final.

Pendant l'étape d'*exécution*, les instances de processus sont exécutées conformément à leur schéma. Cette étape est confiée à un moteur dont le rôle principal est de soumettre les tâches aux différents acteurs selon la politique d'allocation choisie.

L'étape de *contrôle/supervision* permet de surveiller l'exécution des instances de processus afin de s'assurer de leur bon déroulement. Des cas (instances) peuvent être en retard ou bloqués et cette étape permet de gérer de telles situations en désactivant des instances, en modifiant le schéma de

certaines instances, en relançant des acteurs pour qu'ils mènent à bien leur travail etc.

Enfin, l'étape d'*analyse/diagnostic*, qui succède à celle d'exécution, peut amener à revenir sur l'étape de conception. Elle consiste à analyser les traces d'exécution des instances de processus pour mesurer l'écart entre les processus réellement réalisés et leurs modèles. Cette analyse peut être menée manuellement ou être informatisée (exploration/fouille de processus).

### 1.2.3 Typologie de l'adaptation

Afin de fournir un cadre à la discussion que nous allons mener autour de l'adaptation des processus, nous proposons de définir une typologie de l'adaptation. Cette typologie s'inspire des principales typologies définies dans la littérature [Nurcan, 2008], [Schonenberg et al., 2008], [Weber et al., 2009], et identifie trois formes d'adaptation :

- L'*adaptation par flexibilité* porte sur la conception du processus et consiste à jouer sur la granularité du contrôle c'est-à-dire à retarder les choix de modélisation ou d'exécution, et à permettre la personnalisation de l'exécution en introduisant par exemple des points de concertation pour guider le processus, lever l'indéterminisme, ou délibérer sur les choix possibles.
- L'*adaptation par déviation* est réactive et fait face aux imprévus et aux exceptions qui apparaissent en cours d'exécution des instances de processus.
- L'*adaptation par évolution* est une amélioration a posteriori par réaction à l'analyse d'exécutions passées. Cette forme d'adaptation prend en compte l'évolution des processus au cours du temps sur la base d'expériences et de diagnostics du fonctionnement des instances de ce processus notamment en intégrant de la connaissance ou des bonnes pratiques.

Ces formes d'adaptation ne sont pas exclusives et peuvent être combinées dans une même démarche méthodologique ou sur une même plateforme technologique. Il est à noter que cette typologie est uniquement consacrée à l'adaptation des processus : adaptation des modèles (schémas) de processus ou adaptation des instances (cas). Mais il est clair que l'adaptation pourrait porter sur d'autres aspects relevant de l'ingénierie des processus à savoir le processus d'ingénierie, qui correspond à la démarche méthodologique adoptée pour la modélisation des processus, ou l'environnement d'exécution. Plus précisément, on pourrait être amené à :



- modifier le cycle de vie des processus et par exemple introduire d'autres étapes que celles présentées précédemment ;
- remettre en cause la notation utilisée pour améliorer son pouvoir d'expression ;
- changer l'infrastructure qui supporte l'exécution des processus considérés. Par exemple, on pourrait migrer d'une architecture conventionnelle vers une architecture orientée service.

Dans le contexte de ce chapitre, nous aborderons uniquement l'adaptation des processus et ce, à la lumière de la typologie précédente.

## **2. APPROCHES EXISTANTES**

Cette section se limite à présenter l'existant. Elle dresse tout d'abord un panorama des directions empruntées pour traiter du problème d'adaptation des processus. Elle présente ensuite les principales contributions de la littérature sur ce problème. C'est dans la section 3 qu'une synthèse de ces approches sera proposée en prenant en compte la typologie introduite précédemment, et notamment les différentes formes d'adaptation.

### **2.1 Panorama des travaux**

Ces dernières années, la communauté des processus d'entreprise a fait de nombreux travaux pour proposer des solutions au problème de l'adaptation des processus. Plusieurs directions différentes ont été empruntées par ces solutions. Nous les avons classées comme suit :

- L'approche par évolution,
- l'approche par cas,
- l'approche par conception,
- l'approche par contrainte,
- l'approche par intention et contexte,
- l'approche par intégration de protocoles.

Les premières contributions des chercheurs du domaine des processus d'entreprise ont considéré le problème de l'évolution de ces processus en abordant l'évolution de leurs schémas et la migration de leurs instances. On

parle d'**approche par évolution**. Ces contributions ont parfois également étudié la notion de version de processus. Plus précisément, les premiers travaux ayant abordé ce problème d'évolution des processus sont ceux de [Casati et al., 1996-a] et de [Kradolfer et Geppert, 1999], mais la principale contribution de cette direction est probablement celle ADEPT [Reichert et Dadam, 1998], [Reichert et al., 2003-a], [Reichert et al., 2003-b], [Reichert et al., 2005]. Ces travaux, lorsqu'ils ont étudié la notion de version de processus [Reichert et Dadam, 1998] [Kradolfer et Geppert, 1999], ont étudié cette notion pour les schémas de processus en ne considérant que la perspective comportementale des processus. D'autres travaux ne se sont pas limités à uniquement considérer la perspective comportementale lors de la définition de versions de processus, mais ont également pris en compte les autres perspectives des processus. Par exemple, [Zhao et Liu, 2007] a intégré la perspective informationnelle tandis que la proposition de [Châabane et al., 2010] considère l'ensemble des perspectives introduites pour les processus ainsi qu'une autre perspective, appelée perspective intentionnelle, qui définit le contexte d'utilisation des versions de processus.

Des travaux ont également été menés pour traiter ce problème d'adaptation des processus en occultant la notion de schéma de processus et en ne considérant que celle d'instance de processus. On parle d'**approche par cas**. Plusieurs contributions vont dans ce sens. Parmi elles, nous pouvons citer Bonita [Grigori et al., 2004], [Charoy et al., 2006] particulièrement orienté vers les processus coopératifs, Corepro [Muller et al., 2008] ou [Rinderle et Reichert, 2006] dans lesquels l'adaptation est dirigée par les données, ou Flower [Aalst et Berens, 2001], [Aalst et al., 2005] dans lequel le concept central est l'instance de processus, et non le schéma de ce processus c'est-à-dire les activités le composant et l'ordonnancement de ces activités.

D'autres travaux ont essayé de traiter ce problème d'adaptation des processus en évitant d'avoir à adapter les processus. Ces travaux adoptent une **approche par conception**. Parmi ces travaux, certains ont proposé de définir, lors de la spécification du schéma des processus, l'ensemble de tous les chemins possibles. Un tel choix est fortement lié à la puissance de modélisation du langage de spécification des schémas de processus, notamment en ce qui concerne les patrons de coordination [Aalst et al., 2003-b] ; ce choix a également, dans certains travaux, donné naissance à la notion de variante [Lu et Sadiq, 2006], [Lu et al., 2006], [Lu et al., 2009], ou d'alternative [Châabane et al., 2008], [Châabane et al., 2010]. D'autres travaux utilisent la notion d'héritage pour définir des schémas génériques de processus [Aalst, 2001]. D'autres encore proposent d'utiliser des techniques

retardées pour l'exécution ou la modélisation des activités composant un processus [Adams et al., 2006].

Un certain nombre de travaux ont décidé d'utiliser une **approche par contrainte** pour modéliser les schémas de processus. Dans ce type d'approche, le schéma des processus est défini par un ensemble de contraintes (ou de règles) que les instances de ce schéma doivent respecter au cours de leur exécution. Le problème de l'évolution des schémas est simplifié, car faire évoluer le schéma consiste à ajouter ou supprimer des contraintes. Les principaux travaux adoptant cette approche sont les travaux précurseurs menés autour de WIDE [Casati et al., 1996-b], [WIDE, 1999] et ceux menés plus récemment qui se sont concrétisés par les propositions suivantes : AgentWork [Muller et al., 2004], BPAL [Lezoche et al., 2008], Declare [Pesic et Aalst, 2006] [Aalst et al., 2009] et Ecape/BP-Fama [Boukhbouze et al., 2009].

D'autres travaux préconisent d'utiliser les notions d'intention et de contexte pour spécifier des processus facilement adaptables. On parle d'**approche par intention et contexte**. Le raisonnement sur les buts (intentions) est une clé pour comprendre si les activités métiers sont en concordance avec la stratégie de l'organisation [Nurcan et al., 2005], [Nurcan et Edme, 2005]. Les approches par intention supportent plus naturellement la co-production des représentations. Elles facilitent la compréhension des problèmes et permettent de communiquer sur les aspects essentiels (le pourquoi) avant de passer à la mise en œuvre des objectifs en utilisant des représentations organisationnelles et opérationnelles qui sont forcément plus détaillées et dans lesquelles les divergences locales ne vont pas tarder à faire surface. D'autres travaux préconisent d'utiliser la notion de contexte, couplée à celle d'intention pour spécifier des processus plus systématiquement adaptables (au lieu de l'être de manière ad hoc). On parle d'approche par contexte. Les travaux de Saidani et Nurcan illustrent ce type d'approche [Saidani et Nurcan, 2006-a], [Saidani et Nurcan, 2006-b] et [Saidani et Nurcan, 2007-a], [Saidani et Nurcan, 2007-b]. Dans [Saidani et Nurcan, 2006-a], la notion de responsabilité d'un rôle au sein d'un processus, appelée mission, est utilisée comme concept clé pour faciliter les évolutions organisationnelles et fonctionnelles des organisations dans lesquelles les processus sont déployés. Dans [Saidani et Nurcan, 2007-a], [Saidani et Nurcan, 2007-b], la notion de contexte vient compléter cette intentionnalité associée aux acteurs afin de définir les contextes d'utilisation des processus et de faciliter ainsi leur adaptation.

Enfin, d'autres travaux ont proposé d'utiliser les protocoles pour décrire l'interaction entre agents impliqués dans des systèmes multi-agent, appelés

protocoles d'interaction, et pour faciliter l'adaptation coopérative et le pilotage de l'exécution des processus de crise par une cellule de crise [Faure et al., 2010]. Ces travaux sont regroupés dans la catégorie **approche par intégration de protocoles**.

La section suivante détaille les principales contributions de quelques directions les plus convaincantes ou les plus fournies empruntées par les chercheurs du domaine pour traiter le problème d'adaptation des processus d'entreprise. Elle présente notamment l'approche évolution en décrivant les travaux menés autour d'ADEPT [Reichert et al., 2005] et VerFlexFlow [Châabane et al., 2010]. Elle présente également les principes de l'approche par cas (ou instance de processus) en prenant comme exemple Flower [Aalst et al., 2005], les principes de l'approche par conception en présentant les travaux de [Adams et al., 2006], les principes de l'approche par contrainte en prenant comme exemple Declare [Aalst et al., 2009], les principes de l'approche par intention et contexte présentée dans les travaux de [Nurcan et al., 2005], [Nurcan et Edme, 2005], [Saidani et Nurcan, 2007-a], [Saidani et Nurcan, 2007-b], [Saidani et Nurcan, 2009], et l'adaptation des processus de crise par intégration de protocoles d'interaction décrit dans les travaux de [Faure et al., 2010].

## **2.2 Principales contributions**

Cinq principales directions sont présentées dans cette section et quelques contributions illustrent, avec plus ou moins de détails, les travaux menés dans ces principales directions.

### **2.2.1 L'approche par évolution**

Les travaux menés selon cette approche ont abordé les problèmes d'évolution des schémas de processus, de versionnement des schémas de processus, et de migration des instances de processus.

#### ***2.2.1.1 Évolution des schémas de processus: les patrons d'évolution***

Les premiers travaux portant sur l'évolution de schéma ont défini les opérations de base (ou primitives) permettant la modification des schémas de processus. On peut citer par exemple [Casati et al., 1996-a], [Kradolfer et Geppert, 1999], [Reichert et Dadam, 1998], [Reichert et al., 2003-a], [Reichert et al., 2003-b], [Reichert et al., 2005]. Mais la principale contribution dans cette direction est celle de [Weber et al., 2008] qui fait une synthèse et une extension des travaux précédents en définissant des

opérations de haut niveau pour la modification de schémas de processus. Ces opérations sont appelées patrons d'évolution. D'autre part, [Weber et al., 2008] mesure également la capacité des Systèmes de Gestion de Processus (SGP) à supporter l'évolution des schémas de processus. Plus précisément, [Weber et al., 2008] a défini 18 patrons d'évolution (ou patrons de changement), ainsi que 7 caractéristiques que les SGP doivent avoir pour supporter les patrons en question. [Weber et al., 2008] évalue également plusieurs SGP à la lumière de ces patrons et caractéristiques.

### **Concepts de base**

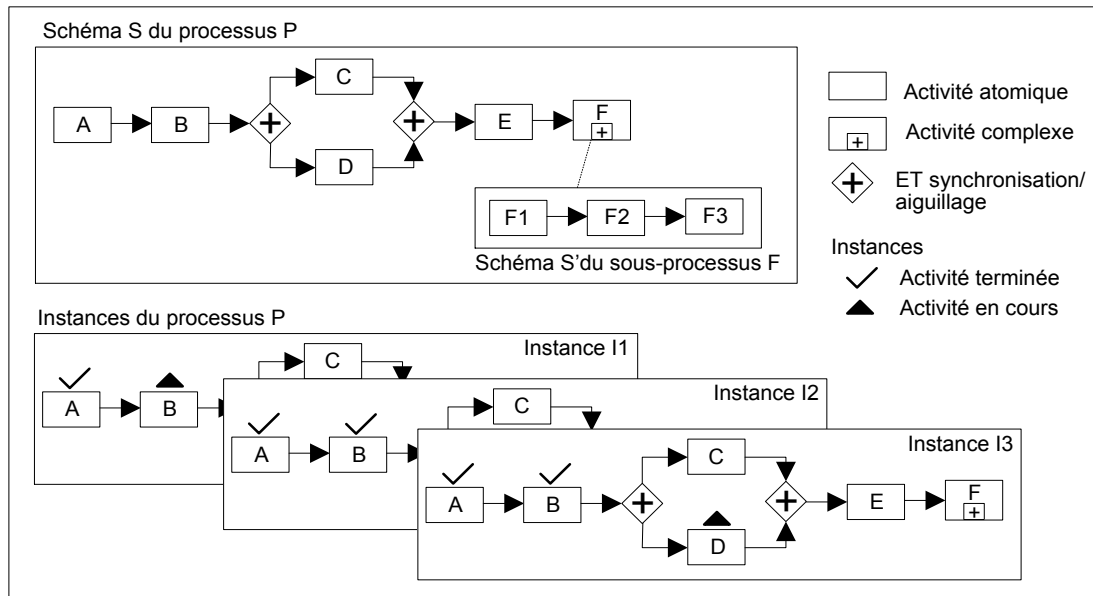
Comme dans ce chapitre, [Weber et al., 2008] distingue le niveau schéma et le niveau instance des processus, et considère qu'un processus peut avoir plusieurs schémas, chacun représentant une version du schéma du processus en question.

Un schéma de processus est, comme visualisé en figure 3 ci-dessous, décrit par un graphe dirigé regroupant des nœuds, représentant les activités du processus ou des patrons d'aiguillage ou de synchronisation d'activités (par exemple, AND/OR-Split, AND/OR-Join) et un ensemble de connecteurs entre ces nœuds exprimant le séquençement de ces nœuds. Les activités peuvent être atomiques ou complexes. Une activité complexe correspond à un sous-processus et son schéma est le schéma du sous-processus en question.

Une instance de processus s'exécute conformément au schéma dont elle est l'instance. Les activités atomiques du processus peuvent être, pour une instance donnée en cours d'exécution, terminées ou inactives.

### **Les patrons d'évolution**

Les patrons d'évolution sont applicables au niveau schéma ou au niveau instance. Ils définissent des opérations de modification portant sur les activités atomiques ou complexes, mais également sur des parties d'un processus regroupant plusieurs activités (i.e. un sous-graphe). Plus précisément, les patrons d'évolution couvrent à la fois les modifications qui peuvent être apportées aux schémas des processus et les précisions qui peuvent être apportées par les acteurs lors de l'exécution des instances (notamment lorsque les activités du processus ne sont modélisées ou choisies que lors de l'exécution –notion de *late modeling* et de *late binding* [Adams et al., 2006]). L'application des patrons d'évolution va éventuellement se propager aux instances en cours d'exécution ce qui va amener à traiter la migration d'une instance d'un ancien schéma de processus vers un nouveau schéma de processus. Ce point-là sera abordé en section 2.2.1.2.

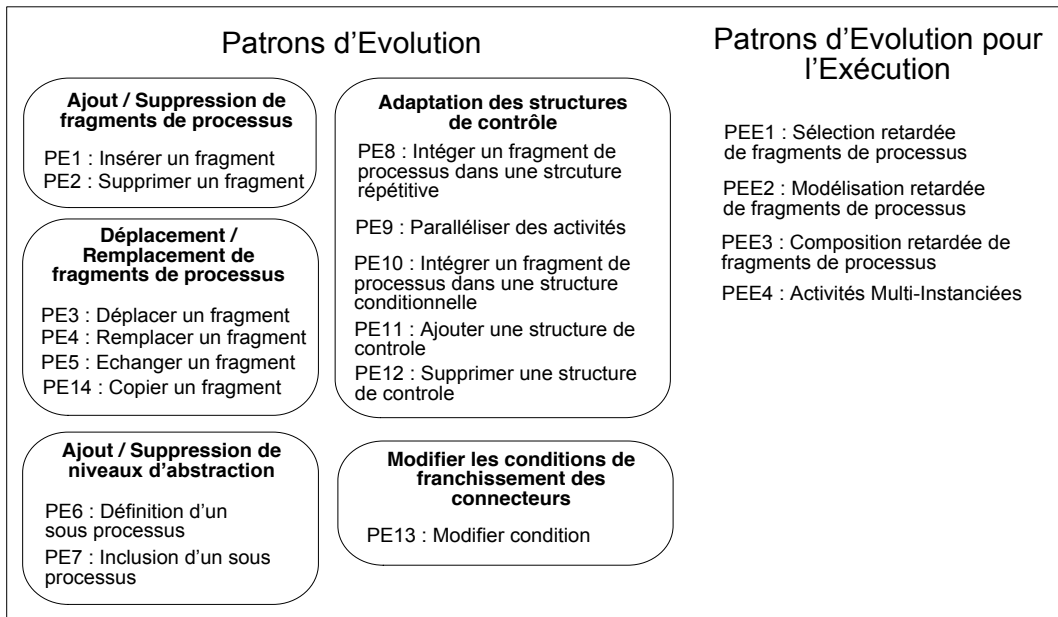


**Figure 3.** Un exemple pour la représentation des concepts de base

Parmi les 18 patrons d'évolution proposés dans [Weber et al., 2008], les 14 premiers concernent les modifications qui peuvent être apportées aux schémas des processus et s'appuient sur les opérations élémentaires telles que *ajouterNoeud*, *supprimerNoeud*, *ajouterLien*, *supprimerLien* et *déplacerLien*. Les 4 derniers patrons concernent les précisions apportées aux instances des processus lors de leur exécution. La Figure 4 récapitule l'ensemble de ces patrons.

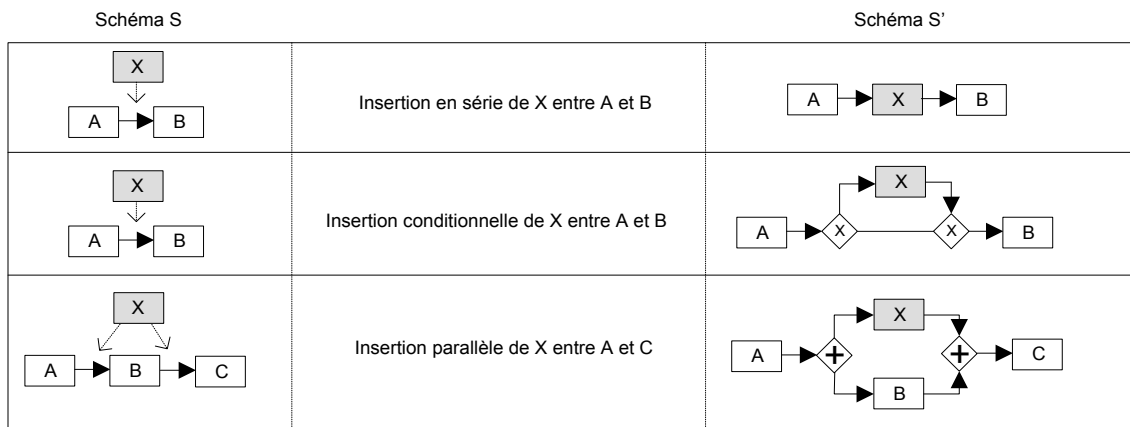
D'une part, les patrons PE1–PE5 et PE14 assurent la gestion des fragments des schémas de processus, les patrons PE6 et PE7 permettent de changer le niveau de visibilité/d'abstraction des sous-processus, les patrons PE8–PE12 assurent la gestion des liens entre parties d'un processus tandis que le patron PE13 permet de modifier les conditions de franchissement qui peuvent être associées aux connecteurs. D'autre part, les patrons PEE1–PEE4 permettent de préciser la modélisation, la composition, la sélection retardées des fragments lors de l'exécution des instances de processus, ou les conditions d'exécution des activités multi-instanciées.

À titre d'exemple, nous présentons plus particulièrement un patron d'évolution à savoir le patron PE1 : Insertion Fragment. Le lecteur intéressé pourra se reporter à [Weber et al., 2008] pour plus de détails.



**Figure 4.** Les patrons d'évolution

Le patron d'insertion d'un fragment de processus supporte l'ajout d'un nouveau fragment dans un schéma de processus. Cet ajout de fragment peut se faire en séquence, en parallèle ou de manière conditionnelle. La Figure 5 ci-dessous illustre ces différents types d'ajout.



**Figure 5.** Le patron Insertion Fragment

### Caractéristiques des SGP pour supporter l'évolution de schémas

[Weber et al., 2008] définit également sept caractéristiques que les SGP doivent avoir pour mettre en œuvre les patrons d'évolution : C1-Evolution de schéma, Gestion de Version, et Migration d'instances, C2-Evolution d'instances spécifiques, C3-Validation des évolutions, C4-Traçabilité et analyse des évolutions, C5-Contrôle d'accès pour les évolutions, C6-

Réutilisation des évolutions, et C7-Contrôle de concurrence pour les évolutions.

Ces caractéristiques sont importantes pour supporter la mise en œuvre des patrons d'évolution. Par exemple, si un SGP implémente des versions de schémas de processus, l'évolution d'un schéma de processus et l'impact sur les instances en cours d'exécution pourront être plus facilement gérés que si le SGP n'a pas de mécanisme de versionnement. Par exemple dans la figure 5, le schéma S du processus a été modifié en rajoutant les activités X et Y. Le nouveau schéma produit est S'. Les modifications apportées sur S ont des répercussions différentes sur ses instances I1 et I2 qui sont en cours d'exécution. I1 n'est pas dans un état critique puisque l'avancement de son exécution est compatible avec les modifications apportées ; la migration de I1 vers son nouveau schéma S' est donc possible. A l'opposé, I2 est dans un état critique puisque l'activité X a été ajoutée avant l'activité B du processus, qui a déjà été réalisée par les acteurs de I2 ; la migration de I2 vers son nouveau schéma S' est donc impossible. Si la notion de version de schémas de processus existe, il est alors plus facile de gérer le problème puisqu'il suffit que I2 reste instance du processus ayant pour schéma S tandis que I1 migre pour devenir instance du schéma S' du même processus.

Nous ne nous étendons pas davantage sur ces caractéristiques qui permettent de mesurer si un SGP peut être qualifié d'adaptatif ; le lecteur intéressé peut consulter [Weber et al., 2008] pour plus de détails sur ces caractéristiques.

### ***2.2.1.2 Migration d'instances***

La migration d'instances de processus est un point crucial dans les travaux portant sur l'évolution des processus et plusieurs travaux l'ont abordé parmi lesquels on peut encore citer [Casati et al., 1996-a], [Reichert et Dadam, 1998] et [Kradolfer et Geppert, 1999]. Ce travaux ont défini les stratégies de migration d'instances en cours d'exécution d'un schéma S donné à un autre schéma S', résultant d'un ensemble de modifications apportées à S. Cette migration n'est pas toujours possible ; aussi il est intéressant pour un système de gestion de processus de disposer de mécanismes de versionnement de schémas de processus afin de permettre à différentes instances d'un même processus de s'exécuter conformément à des schémas différents.

Par exemple, dans [Casati et al., 1996-a], trois techniques de migration sont proposées :

- Stop : les instances conformes au schéma S et en cours d'exécution sont stoppées et relancées conformément au schéma S',



- Blocage : les instances conformes au schéma S et en cours d'exécution se poursuivent jusqu'à la fin et les nouvelles instances sont créées à ce moment-là conformément au nouveau schéma S',
- Progressive : la migration se fait au cas par cas : chaque instance en cours d'exécution est analysée et en fonction de la progression de l'instance et des modifications apportées, la migration peut se faire, ne pas se faire (on a recours alors à une des deux autres techniques pour l'instance concernée), ou se faire de manière progressive par migration dans des schémas intermédiaires.

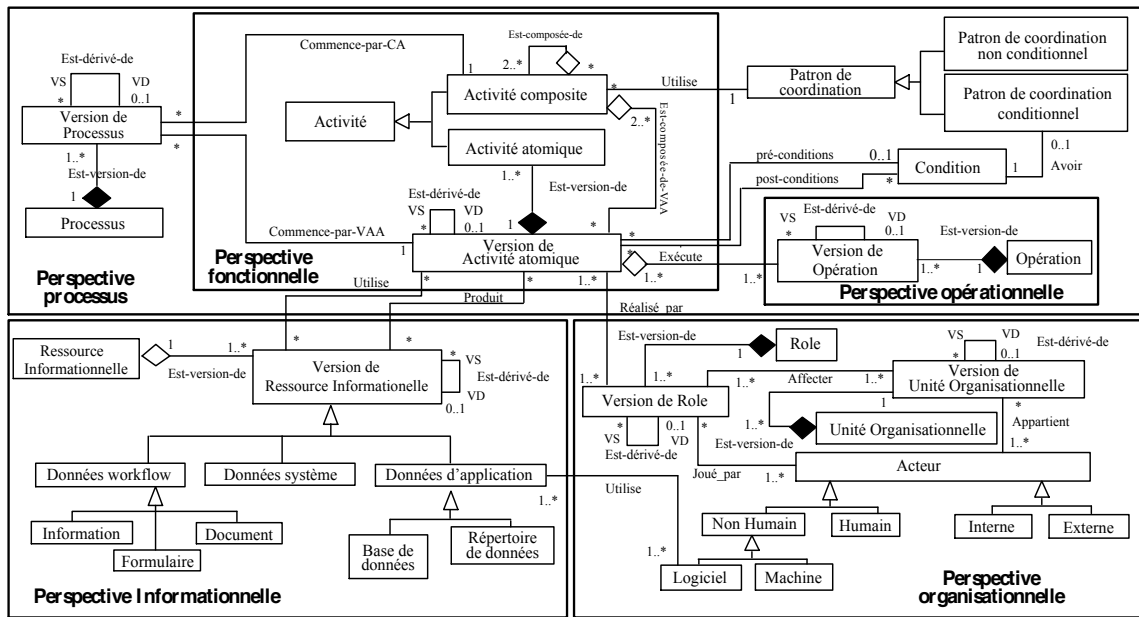
On peut noter que les instances d'un même processus concernées par une migration peuvent suivre des techniques de migration différentes.

### **2.2.1.3 Versions**

Dans un premier temps, les recherches autour des versions de processus n'ont considéré que la perspective comportementale des processus [Casati et al., 1996-a], [Reichert et Dadam, 1998], [Kammer et al., 1999], [Kradolfer et Geppert, 1999]. Ces contributions ont été étendues ces dernières années en intégrant la perspective informationnelle [Zhao et Liu, 2007], mais également la perspective organisationnelle [Châabane et al., 2010]. Dans cette dernière contribution, les auteurs introduisent également une autre perspective, appelée perspective intentionnelle, qui définit le contexte d'utilisation des versions de processus, i.e. les objectifs qui leur sont associés. Les sous-sections suivantes présentent rapidement cette dernière contribution.

### **Le méta modèle VBP-M**

L'idée est de définir les versions de processus comme instance d'un méta-modèle, appelé VBP-M (Version of Business Process Metamodel) qui combine la représentation des aspects comportementaux, informationnels, organisationnels et intentionnels des processus, et un patron de versionnement appliqué aux concepts pour lesquels on souhaite gérer de versions. La Figure 6 présente le méta-modèle VBP-M.



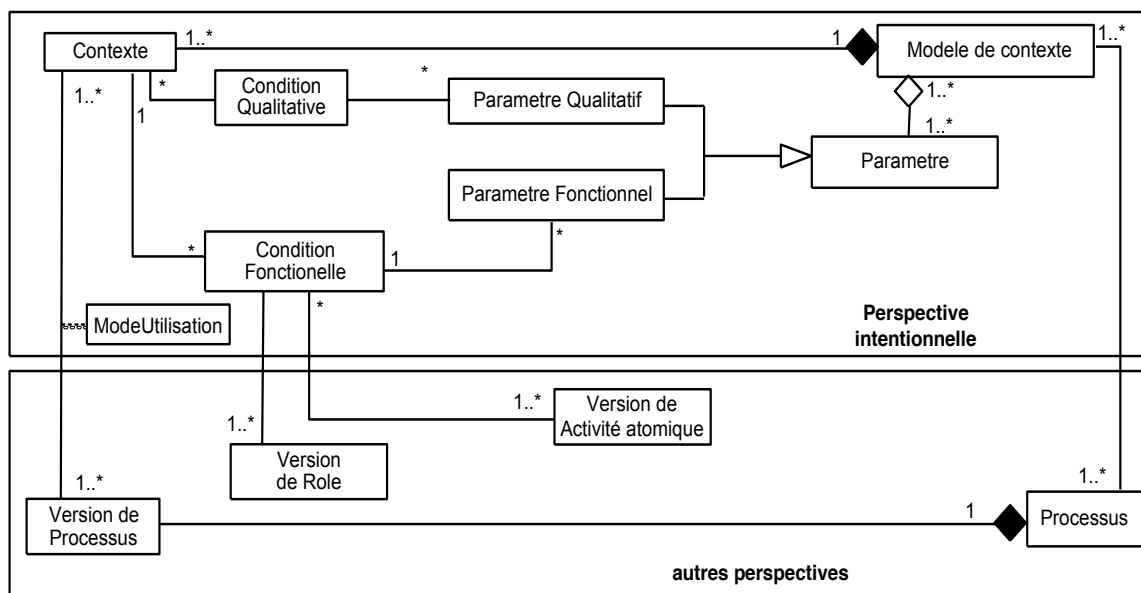
**Figure 6.** Le méta-modèle VBP-M

Le patron de versionnement est utilisé afin d'appliquer la notion de version à plusieurs concepts du méta-modèle appartenant aux perspectives comportementale, organisationnelle et informationnelle. En ce qui concerne la perspective comportementale, la notion de version est introduite pour les classes Processus, Activité atomique et Opération. En effet il est intéressant de garder trace des changements dans l'organisation du travail –c'est-à-dire les activités qui composent le processus et leur schéma d'enchaînement–, alors que les versions d'une activité atomique ou d'une opération décrivent les modifications dans la manière de réaliser l'activité. En ce qui concerne les autres perspectives, la notion de version est introduite pour les classes Opération, Ressource informationnelle (perspective informationnelle) et Rôle et Unité Organisationnelle (perspective organisationnelle) [Chaabane et al, 2008].

### Prise en compte de la perspective intentionnelle

Le méta-modèle VBP-M permet de modéliser différentes versions d'un même processus. Cependant, pour pouvoir choisir, parmi l'ensemble des versions d'un processus, celle qui convient à une situation donnée, [Châabane et al., 2010] indique qu'il faut permettre à un concepteur de décrire l'intention qu'il a lorsqu'il définit les différentes versions d'un même processus. Une nouvelle perspective, appelée perspective intentionnelle, est introduite dans la modélisation des versions de processus. Cette perspective

permet d'indiquer pourquoi une version de processus est plus appropriée qu'une autre dans un contexte donné. La Figure 7 détaille la modélisation de cette perspective en introduisant les concepts qu'elle manipule et en montrant comment ces concepts sont liés à ceux du méta-modèle déjà présenté.



**Figure 7.** La perspective intentionnelle

Les contextes définissent les conditions d'utilisation des versions de processus. Ils sont vus comme des entités à part entière et sont spécifiés à travers deux notions : la notion de Modèle de contexte et la notion de Contexte, qui correspond à l'occurrence d'un modèle de contexte. Un modèle de contexte définit un ensemble de paramètres représentant la connaissance nécessaire à la définition des contextes. [Châabane et al., 2010] distingue deux types de connaissances et par conséquent deux types de paramètres. Les paramètres quantitatifs correspondent à des connaissances générales du contexte. Les paramètres fonctionnels sont, quant à eux, des fonctions qui font référence à des éléments des autres perspectives. Lors de la définition d'un contexte, les paramètres du modèle de contexte correspondant sont utilisés pour définir les conditions du contexte. Enfin, le concepteur définit le mode d'utilisation des versions de processus en accord avec les contextes. Le concepteur peut soit indiquer qu'une version est adaptée à un contexte (le mode d'utilisation est Use), soit indiquer qu'une version n'est pas adaptée à un contexte (le mode d'utilisation est Avoid). Bien sûr, un concepteur peut à la fois indiquer qu'une version est adaptée à un contexte et qu'une autre (version) est à absolument éviter pour le contexte en question.

### 2.2.2 L'approche par cas

Il s'agit d'un paradigme formalisé par [Aalst et al., 2005] à partir de l'analyse de plusieurs systèmes tels que FLOWer [Pallas, 2002], Staffware Case Handler [Staffware, 2000], COSA Activity Manager [Software-Ley, 2002] qui ont été appliqués en entreprise dans divers domaines (assurance, banque, santé, télécommunications, etc). L'idée principale est de permettre une exécution spécifique pour chaque cas (instance de processus), en permettant aux acteurs de diriger et contrôler l'exécution des tâches à partir des données utilisées et produites par le cas. Ce paradigme introduit un équilibre entre les approches dirigées par les données et celles dirigées par le contrôle. Plus précisément, l'adaptation par cas se distingue des processus traditionnels en adoptant une nouvelle posture : se focaliser sur ce qui est possible plutôt que sur ce qui est prescrit. La différence réside dans les quatre points suivants :

- Une visibilité globale sur l'ensemble du cas (données, tâches, acteurs, rôles, autorisation... et leurs évolutions) est donnée aux acteurs plutôt qu'une visibilité locale, limitée aux tâches à réaliser ;
- La conduite du cas, c'est-à-dire le choix des activités à exécuter à chaque étape, est déterminée par l'ensemble des données déterminant l'état du cas dans sa globalité plutôt que déclenchée par la dernière activité réalisée. La séparation du contrôle et des données, fondement des systèmes de processus traditionnels, est remise en cause.
- La séparation des aspects autorisation et distribution des tâches est introduite pour permettre une visibilité sur toutes les tâches possibles versus autorisées. La relaxe de cette contrainte permet à des acteurs de pouvoir pro-activement demander à certains de leurs collaborateurs de réaliser des tâches que ces premiers jugent utiles à la réalisation de l'objectif global et qu'ils n'ont pas l'autorisation d'exécuter ;
- Deux nouveaux rôles : « skip » et « redo ». En plus des rôles conventionnels, éventuellement structurés en hiérarchie, qui autorisent l'exécution de tâches, il existe deux rôles supplémentaires qui permettent d'agir sur la structure de contrôle du processus : « skip » pour sauter des tâches ou « redo » pour défaire des activités réalisées et revenir à des états antérieurs.

[Aalst et al.,2005] a proposé un méta-modèle UML et une formalisation théorique de ce paradigme à l'aide de diagrammes états transitions et de règles actives. On y note le rôle central joué par les données qui peuvent être obligatoires (pré requis), restreintes (limitées à) vis à vis des activités les



modélisation des processus a une puissance d'expression adéquate : par exemple, il est difficilement envisageable de généraliser un processus sans savoir exprimer des alternatives [Aalst et al., 2003-b].

Nous distinguons trois principaux courants dans les différents travaux adoptant cette approche. Tout d'abord [Aalst, 2001] préconise d'utiliser la notion d'héritage pour définir des schémas génériques de processus. Ensuite, les travaux autour des variantes ([Lu et Sadiq, 2006], [Lu et al., 2006], [Lu et al., 2009]), ou des versions à travers la notion d'alternative ([Châabane et al., 2008], [Châabane et al., 2010]) permettent de représenter différents schémas d'un même processus chacun représentant un cas possible d'exécution du processus. Enfin, [Adams et al., 2006] propose d'utiliser des techniques particulières pour retarder la définition d'une partie d'un schéma de processus ou le choix de la manière dont l'exécution d'activités va se concrétiser. Pour cette dernière stratégie, on parle de liaison tardive. Les paragraphes ci-dessous vont détailler ce principe de liaison tardive ; il est à noter qu'il a été mis en œuvre sous la forme de worklets dans le système de gestion de Workflow Yawl [Aalst et Hofstede, 2004].

Les worklets permettent aux concepteurs de processus de modéliser la manière dont les activités des processus peuvent se concrétiser sans pour autant définir fermement cette concrétisation au moment de la conception. Ainsi, l'utilisateur, en phase d'exécution d'une instance pourra choisir, parmi l'ensemble des concrétisations possibles pour une activité celle qui convient au cas en question. La théorie des activités (Activity Theory [Nardi, 1996]) est sous-jacente à cette idée là. Elle indique que les actions entreprises dans le cadre des activités dépendent du contexte dans lequel les personnes les réalisant se trouvent. Ces personnes ont donc un degré de latitude certain pour la réalisation de leurs activités.

Les worklets correspondent donc à des répertoires de sous-processus associés à des activités. À l'exécution, lors de la réalisation d'une activité, l'utilisateur va alors choisir le sous-processus (qui peut se ramener à une activité) qui convient le mieux au contexte courant. Cependant, pour faciliter ce choix et le rendre plus automatique, les worklets sont couplés à un système de règles (RDR pour Ripple-Down Rules), qui induit en fonction du contexte d'exécution de l'instance considérée le sous-processus à choisir. C'est ce système qui guide l'utilisateur dans son choix du sous-processus concrétisant l'activité qu'il doit exécuter. Nous ne détaillerons pas ici le fonctionnement du système de règles. Le lecteur intéressé peut consulter [Adams et al., 2006] ou se reporter au site de Yawl [YAWL Foundation, 2004] pour de plus amples informations.

## 2.2.4 L'approche par contrainte

L'approche par contrainte remet en cause l'approche procédurale habituellement utilisée qui définit le schéma d'un processus comme un enchaînement d'activités, mais préconise plutôt de définir la liste des activités du processus et un ensemble de contraintes (règles) qui réglemente l'exécution de ces activités. Dans cette approche, le schéma d'un processus n'est pas explicitement défini. Plusieurs travaux préconisent d'adopter cette philosophie. On peut tout d'abord citer celui, précurseur, mené dans le cadre du projet WIDE [WIDE, 1999], qui modélise l'enchaînement de l'exécution des activités par des règles actives [Casati et al., 1996-b]. Cependant, même si les règles servent de support dans cette proposition, l'enchaînement des activités est tout de même plus ou moins défini à travers les règles. Les travaux qui ont suivi, quelques années plus tard, ont repris cette idée de règle comme support de la représentation des contraintes qui existent entre les activités et n'ont plus décrit de manière explicite l'enchaînement de ces activités. Parmi ces travaux, on peut citer BPAL [Lezoche et al., 2008], Escape/BP-Fama [Boukhbouze et al., 2009] et surtout Declare [Pesic et Aalst, 2006], [Pesic et al., 2007-a], [Pesic et al., 2007-b], [Aalst et al., 2009]. Declare va servir à illustrer l'approche par contrainte.

Cette approche est très intéressante pour le problème de l'adaptation des processus. En effet, adapter un processus ne consiste plus à modifier son schéma en ajoutant/supprimant des activités et en ajoutant/supprimant des liens entre ces activités, mais plutôt en ajoutant/supprimant des activités et des contraintes. Le schéma des processus n'étant plus explicitement décrit, son adaptation semble en effet plus simple à mettre en œuvre.

Les deux sections suivantes vont successivement présenter les solutions proposées dans Declare pour la conception et l'exécution de processus décrits selon cette approche et les opérations proposées pour l'adaptation des processus.

### 2.2.4.1 Conception et exécution de processus

Spécifier un processus dans Declare revient à définir l'ensemble des activités qui le composent ainsi qu'un ensemble de contraintes liant ces activités. Plus précisément, les contraintes indiquent de manière implicite le possible enchaînement de ces activités : tout enchaînement respectant les contraintes est valide.

Declare distingue les contraintes *obligatoires*, i.e. qui doivent être respectées par les instances du processus en cours d'exécution, des contraintes *optionnelles*, qui peuvent ne pas être respectées lors de

l'exécution de ces instances. Declare permet de définir des contraintes génériques (appelées *constraint templates*) exprimées entre activités génériques. Ces contraintes sont nommées et s'utilisent en spécialisant les activités génériques qu'elles contraignent. Elles ont un cycle de vie qui repose sur trois états : *vérifiée, non vérifiée, temporairement non vérifiée*.

Declare propose deux langages pour spécifier des contraintes : un langage graphique, plus intuitif, et donc plutôt destiné à des concepteurs de processus, et un langage s'appuyant sur une Logique Linéaire Temporelle (LLT) pour la formalisation des contraintes. LTL est une forme de logique qui combine les opérateurs de logique classique et des opérateurs de logique temporelle tels que *toujours, éventuellement, jusqu'à et prochaine-fois*. Utiliser des formules LTL pour exprimer les contraintes a pour avantage de pouvoir contrôler la sémantique des contraintes et s'assurer de la cohérence globale.

Les instances de processus s'exécutent en accord avec les contraintes du processus. Plus précisément, ces contraintes pilotent l'activation et l'exécution des activités possibles dans le processus : tout ce qui respecte les contraintes obligatoires est possible et toutes les contraintes obligatoires doivent être vérifiées à la fin de l'exécution des instances. Declare traduit les différentes contraintes (obligatoires) sous la forme d'automates d'états finis et spécifie également un automate par contrainte et instance. Exécuter une instance de processus revient à exécuter de manière conjointe ces automates. Le lecteur intéressé peut consulter [Pesic, 2008] pour de plus amples informations à ce sujet.

#### **2.2.4.2 Adaptation de processus**

La question qui se pose ici est relative à l'adaptation des processus : qu'est ce qu'adapter un processus défini en adoptant l'approche par contrainte ? Deux types de changement sont possibles dans Declare. Des changements ponctuels sur une instance en cours d'exécution et des changements durables traduisant des modifications plus profondes pour le processus considéré [Pesic et al., 2007-a]. Ces changements sont mis en œuvre par cinq opérations de base : ajout d'une contrainte, suppression d'une contrainte, modification de la nature d'une contrainte (une contrainte optionnelle peut devenir obligatoire et vice-versa), ajout d'une activité et suppression d'une activité. Les automates associés sont modifiés en accord avec les changements relatifs aux contraintes obligatoires. Enfin, Declare s'assure que les changements opérés soient possibles. En effet, une nouvelle contrainte peut être en contradiction avec l'état d'avancement de l'instance considérée et par conséquent l'ajout de la nouvelle contrainte peut être rejeté.



### 2.2.5 L'approche par intention et contexte

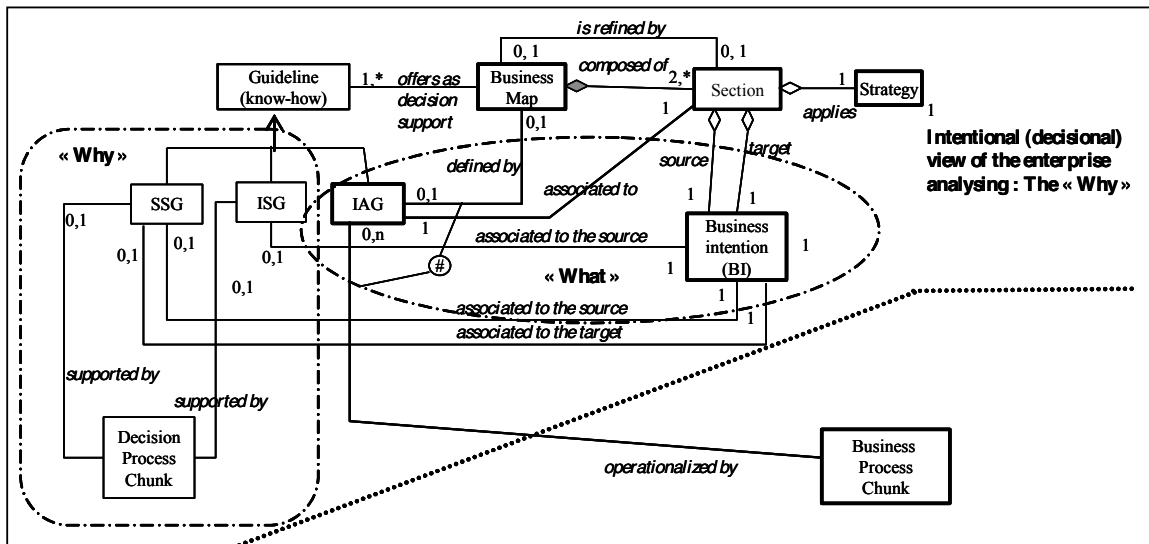
L'ingénierie des processus d'entreprise requiert des formalismes qui ont l'aptitude à représenter la nature variable des processus. Elle a besoin de représentations adaptables dont l'exécution puisse évoluer selon les situations que l'on ne peut pas toujours 'prescrire'. Or, la majorité des formalismes de représentation de processus offerts par la littérature, sont dirigés par une perspective fonctionnelle. Ceci a comme principale conséquence de figer ces processus dès la fin de leur modélisation tandis que les processus d'entreprise sont soumis à des facteurs situationnels dans leurs réalisations. L'approche par intention et contexte propose de traiter ces besoins d'adaptation en utilisant deux mécanismes complémentaires: variabilité et conscience du contexte.

Pour traiter de la variabilité des processus, il est nécessaire de disposer de modèles plus concis que les modèles classiques dirigés par les activités [Mendling et Nüttgens, 2003], [Ould 1995]. Pour ce faire, ces travaux se sont inspirés des modèles "acteurs/rôles" de EKD-CMM [Barrios et Nurcan, 2004] et du standard RBAC (Role Based Access Control) [Sandhu et al., 1999]. Ce dernier a été étendu à la modélisation des processus d'entreprise par l'expression des règles métier contextuelles relatives aux responsabilités [Saidani et Nurcan, 2006-a], [Saidani et Nurcan., 2006-b]. L'objectif est aussi de pouvoir lisser dynamiquement la charge de travail des acteurs impliqués dans l'ensemble des processus d'une organisation et de garantir ainsi la performance du réseau de processus qui représente la dynamique de l'organisation [Bessai et Nurcan, 2009].

L'aptitude à l'adaptation et à la variabilité nécessite que les environnements de modélisation et d'exécution des processus métier sachent intégrer la conscience de l'environnement externe des processus. Ces travaux proposent donc une formalisation de la notion de contexte qui intègre plusieurs aspects qui peuvent caractériser non seulement un processus métier, mais aussi, les acteurs du processus et l'environnement dans lequel le processus s'exécute [Saidani et Nurcan, 2009].

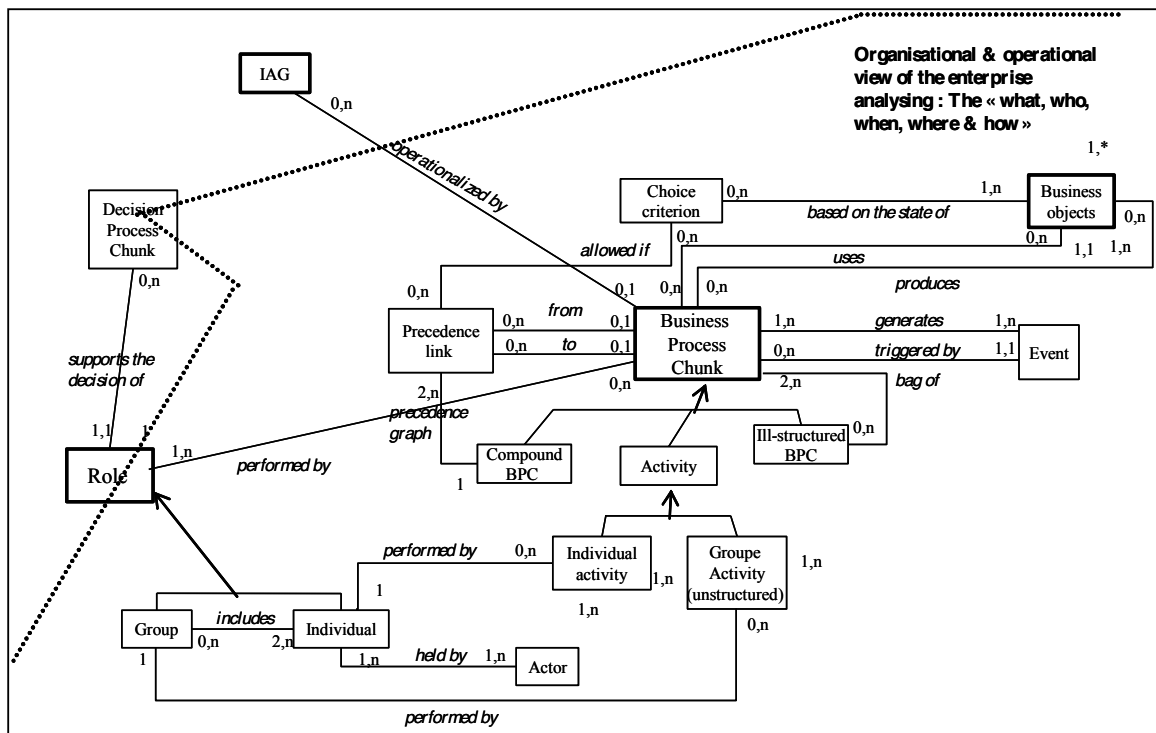
Bien que l'intérêt de l'adaptation soit largement reconnu, les techniques de sa mise en œuvre restent limitées. Les approches existantes qui traitent l'adaptation des processus métier sont axées sur les moyens intrinsèques de la modification des processus, après qu'un besoin de changement soit observé [Rosemann et Aalst, 2007], [Rosemann et al., 2008] ; il s'agit d'un changement a posteriori (par adaptation) des modèles et/ou des instances de processus. Ces approches ne tiennent compte que de la partie réactive de l'adaptabilité des processus et ignorent le stimulus pour le changement, c'est-

à-dire le contexte. [Nurcan et al., 2005] propose donc une approche pour la modélisation des processus d'entreprise avec une perspective intentionnelle permettant de représenter, de manière explicite, les objectifs et les stratégies de l'organisation. Cette représentation donne aux modèles de processus exprimés au niveau intentionnel la capacité d'incorporer l'adaptation par sélection ou a priori, i.e. en phase de modélisation des processus (parcours dynamique de la Carte et choix dynamique des intentions et des stratégies).



**Figure 9.** Concepts pour décrire la vue intentionnelle des processus

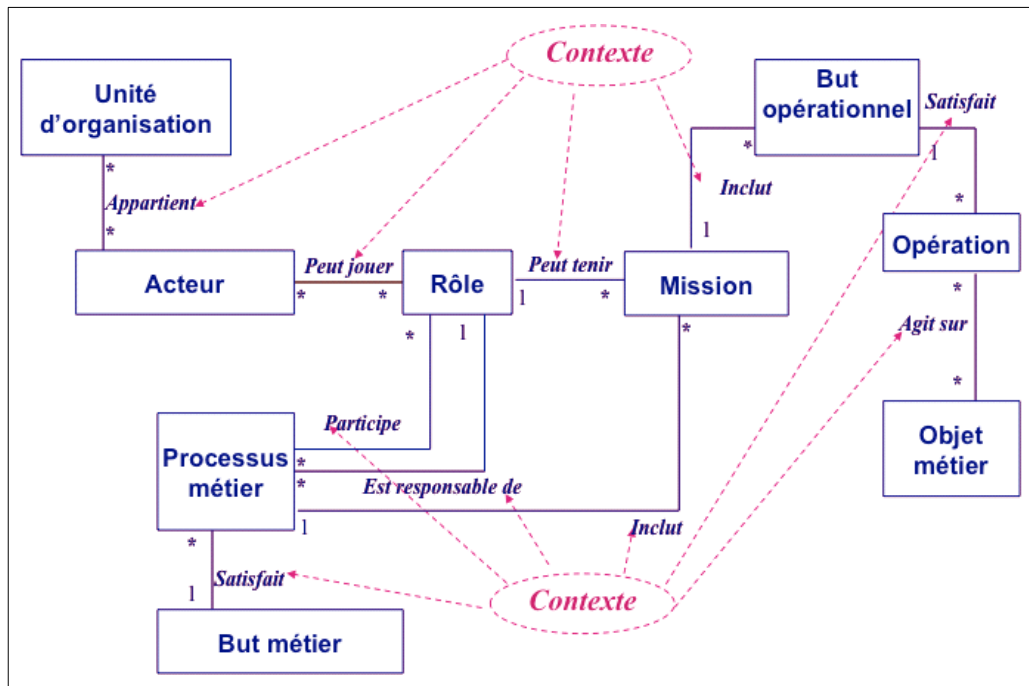
La Carte est ainsi une approche qui permet de considérer le réseau entier des processus de l'organisation comme une collection dynamique de processus coopératifs et coopérants qui doivent être dynamiquement orchestrés selon le contexte de réalisation [Nurcan et Edme 2005] ou comme une famille de processus dont l'adaptation sera réalisée pendant la conception ou l'exécution [Rolland et Nurcan, 2010]. Ces observations ont incité à apporter à la notion de carte les extensions nécessaires pour intégrer les rôles et les liens qu'ils ont avec les niveaux organisationnel et opérationnel [Nurcan, 2004], [Nurcan et Edme, 2005], [Nurcan et al., 2005-a]. Les processus bien définis et peu structurés coexistent et sont gérés et administrés dans un modèle d'entreprise global [Nurcan, 2008]. Le méta-modèle de processus multi-dimension présenté dans les Figures 9 et 10 offre l'adaptabilité exigée pour les processus peu structurés et créatifs et la rigueur nécessaire pour représenter les processus structurés.



**Figure 10.** Concepts pour décrire les vues organisationnelle et opérationnelle des processus

En outre, les travaux adoptant cette approche considèrent les acteurs comme des individus qui ont une capacité de décision. Le rôle n'est plus vu comme une entité exécutant une suite d'activités ; il incorpore les dimensions de savoir-faire et d'autonomie [Saidani et Nurcan, 2006-b], [Saidani et Nurcan, 2007-a], [Saidani et Nurcan, 2009]. Par conséquent, les responsabilités et les capacités d'un acteur peuvent varier selon les situations. Par exemple, dans une situation d'urgence, un acteur peut être autorisé à assumer une responsabilité, qui nécessiterait des qualifications supérieures dans un contexte normal. La notion de contexte est donc introduite pour capturer la variabilité du comportement de l'acteur. Le contexte a un impact sur toutes les relations d'assignation présentes dans le formalisme utilisé. Le méta-modèle CxRB2PM (Context and Role-Based Business Process Model) résultant est donné en figure 11 [Saidani et Nurcan, 2007-b]. La capacité d'un acteur à jouer un rôle et le degré d'autonomie qui lui est accordé varient selon le contexte. Par exemple, un acteur expérimenté doit avoir plus d'autonomie qu'un novice. Si la modélisation ne tient pas compte du contexte, en l'occurrence ici, l'expérience, une mission risque de ne pas être bien réalisée par un acteur novice auquel on aura accordé une autonomie totale. De même, la mission (i.e. l'ensemble des opérations sous la responsabilité d'exécution d'un acteur d'un processus) risque de consommer plus de ressources si elle

est réalisée par un acteur expérimenté contraint par une démarche préétablie que si cet acteur bénéficiait d'une plus grande marge de manœuvre. Le pilote du processus et le mécanisme d'assignation doivent déterminer quel acteur peut jouer quel rôle et dans quel contexte. De même, les missions les plus appropriées pour composer un processus donné sont sélectionnées selon le contexte. Il en va de même pour le reste des relations d'assignation entre les éléments qui composent un modèle de processus.



**Figure 11.** Le méta-modèle CxRB2PM

## 2.2.6 L'approche par intégration de protocoles d'interaction

L'idée de ce mode d'adaptation consiste à intégrer dans un cadre cohérent, d'une part un processus, et d'autre part des protocoles d'interaction associant des groupes humains au processus afin de leur permettre de le guider, lever l'indéterminisme qu'il comporte, délibérer sur les choix possibles ou décider de nouvelles actions à entreprendre. Il a été présenté dans [Faure et al., 2010] et appliqué aux processus de gestion de crises.

Les sous-sections suivantes introduisent les protocoles d'interaction en décrivant leur structure et montrent comment ces protocoles sont intégrés au sein des processus.

### **2.2.6.1 Les protocoles d'interaction**

#### **Notion de protocole d'interaction**

Les protocoles d'interaction [Hanachi et Sibertin, 2004] sont connus pour être une technique de coordination multi-agent explicite et fiable. Par exemple, la mise en place d'un protocole de vote peut permettre à des intervenants répartis géographiquement de prendre une décision concertée, conforme à des règles connues de tous et dont on peut garder trace.

Un protocole d'interaction est défini comme un ensemble de communications structurées, conforme à un certain schéma, et visant à coordonner les interventions (éventuellement synchrones) des différents acteurs dans la réalisation d'un objectif. Ces protocoles peuvent par exemple permettre la sélection de partenaires par appel d'offres, le vote d'une décision importante ou la négociation de critères de réalisation d'une tâche. L'idée soutenue dans cette approche est que l'issue de l'exécution d'un protocole a une influence sur la suite du processus et participe donc à son guidage dynamique.

Dans l'exemple de la gestion de crises, mise en œuvre par les auteurs, les processus de résolution de crises sont collaboratifs et constituent un objet partagé dont le mode d'exécution et les alternatives doivent faire l'objet de concertation et de choix formels en cours d'exécution, dans la mesure où ils engagent la responsabilité des intervenants (Préfet, responsables des pompiers, SAMU, police...). L'adaptation doit donc également prendre en compte cette dimension humaine, institutionnelle et collective et intégrer des techniques d'interaction pour faciliter la prise de décision. On peut noter que d'autres travaux ont traité aux protocoles dans le cas des crises. On peut par exemple citer [Hemaissia et al., 2008].

#### **Structure des protocoles d'interaction**

Les protocoles d'interaction sont un mécanisme de coordination de systèmes distribués, principalement utilisés dans le cadre des systèmes multi-agent [Hanachi et Sibertin, 2004]. Plus précisément, les protocoles d'interaction s'attachent à coordonner différents intervenants en contraignant leurs conversations, c'est-à-dire les séquences d'interactions qu'ils réalisent afin de réaliser un certain objectif. Les protocoles s'intéressent aux conversations qui se déroulent conformément à un schéma prédéfini qui détermine notamment :

- les pré-requis et le but de la conversation,

- les rôles des participants,
- le type des interventions, ou interactions élémentaires, qu'ils peuvent effectuer. Ils correspondent à des actes de communication ou performatifs (informer, interroger, confirmer, refuser, diffuser...).
- les règles d'attribution des rôles aux participants,
- les règles d'intervention qui précisent les circonstances dans lesquelles un participant jouant un certain rôle peut (ou doit) réaliser une intervention d'un certain type.

Ainsi, une conversation est considérée comme une occurrence d'un protocole d'interaction. Dans le cadre des contraintes imposées par les règles de ce protocole, les participants gardent la liberté de choisir quand et comment intervenir. Ces protocoles ne sont bien souvent que l'adaptation informatique de procédures largement utilisées dans la société et qui sont, de ce fait, optimisées pour atteindre un but déterminé.

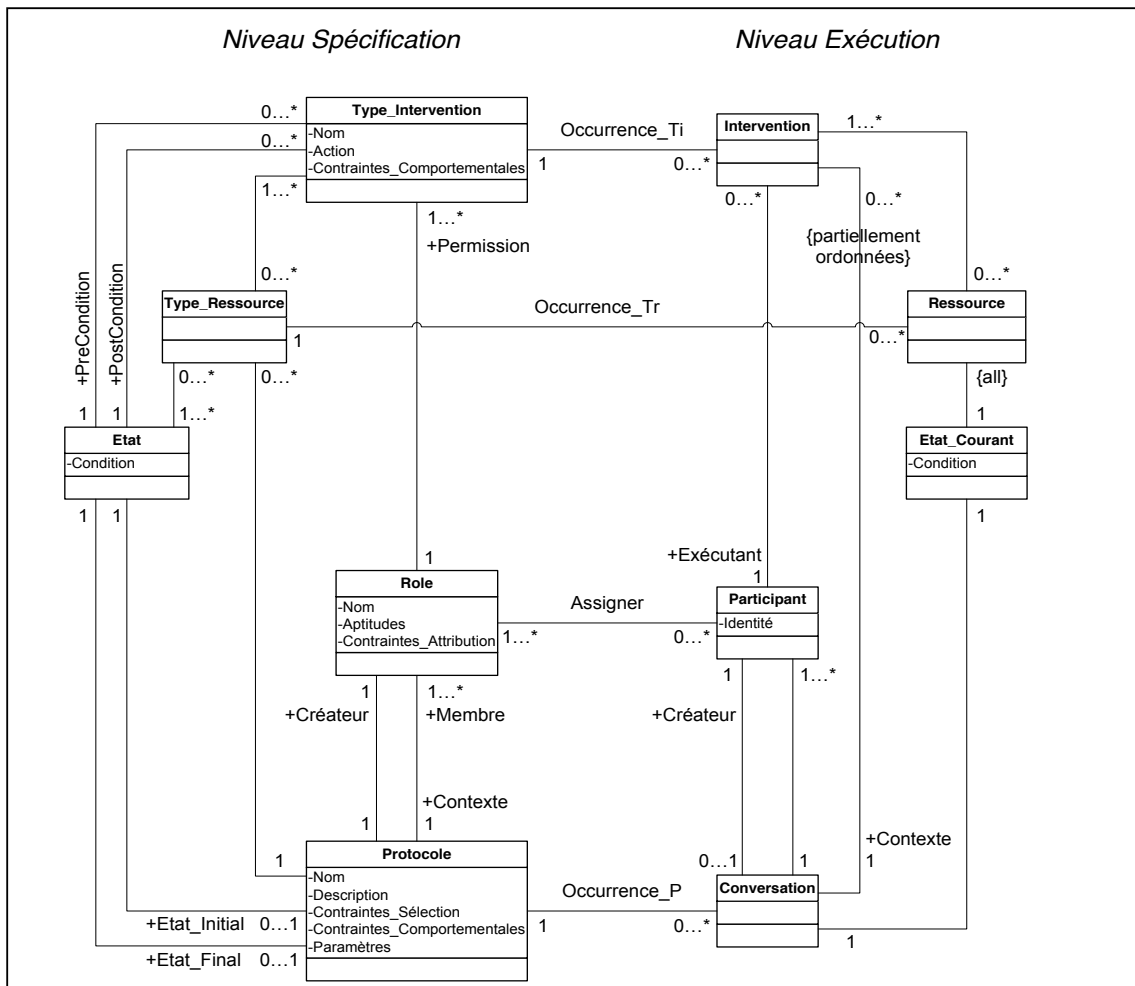
Un protocole peut être implanté de manière répartie, c'est-à-dire dissous dans chaque participant, ou au contraire défini comme un composant à part entière, qui peut être instancié et réutilisé, et que les participants peuvent partager et utiliser à la manière d'une ontologie [Hanachi et Sibertin, 2004], [Tamma et al., 2002]. C'est cette approche qui a été suivie dans [Faure et al., 2010]. La description d'un protocole est basée sur le méta-modèle de la Figure 12 qui est une adaptation de [Hanachi et Sibertin, 2004]. Ce méta-modèle comporte deux niveaux : le niveau spécification et le niveau exécution.

### ***2.2.6.2 Intégration des protocoles d'interaction***

Trois techniques ont été envisagées pour intégrer des protocoles d'interaction dans un processus :

- La première technique, implémentée dans [Faure et al., 2010], concerne l'ajout de protocoles lors de la phase d'adaptation du processus collaboratif avant toute exécution. Elle consiste à insérer une tâche dans un processus, cette tâche étant l'invocation d'un service de protocole.
- La deuxième technique est une variante de la précédente dans laquelle la tâche insérée dans le processus n'est pas un protocole, mais une tâche de sélection associée à une base de protocoles prédéfinis dont l'interface, en terme d'entrées et de sorties, est compatible avec celle de la tâche. Au moment de l'exécution de cette tâche, une action de choix est engagée

afin de sélectionner le protocole qui est le plus approprié, en accord avec le contexte d'exécution.



**Figure 12.** Méta modèle des Protocoles d'Interaction

- La troisième technique consiste à introduire un protocole dans le processus au cours de son exécution. Cette technique exige de pouvoir contrôler l'activité du moteur d'exécution, pour l'interrompre au moment adéquat afin de modifier l'instance et/ou le schéma du processus en cours pour insérer l'invocation d'un protocole et ensuite reprendre l'exécution. Le moteur doit donc répondre à de nouvelles exigences : acceptation de points de contrôle, points d'arrêt et d'observation permettant une interaction du moteur avec le processus de pilotage, et acceptation de directives, actions permettant de modifier l'état du processus (arrêt, suspension, reprise de l'exécution du processus collaboratif). Ce type de moteur a également été implémenté dans [Faure et al., 2010] dans un environnement SOA.

### **3. SYNTHÈSE DES APPROCHES EXISTANTES**

Cette section a pour objectif de faire une analyse des travaux existants à la lumière d'un certain nombre de critères.

#### **3.1 Grille d'analyse des travaux**

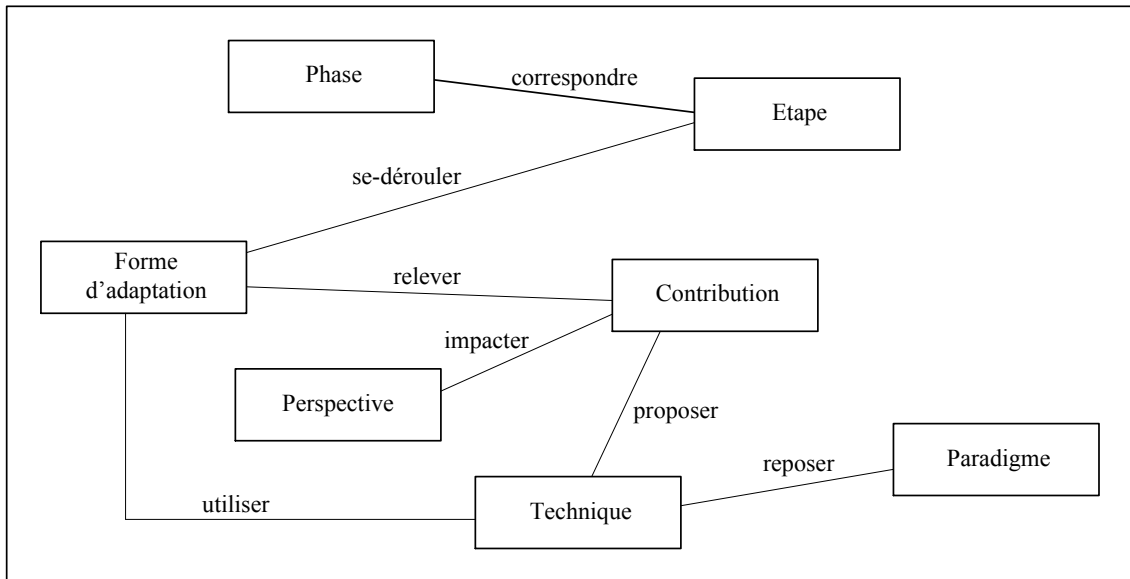
Conformément au cadre d'analyse introduit en section 1.2, nous tenons compte de l'étape du cycle de vie dans lequel l'adaptation est réalisée et la forme d'adaptation opérée. Concernant le cycle de vie des processus, nous distinguons d'une part la phase de modélisation de la phase d'exécution, et d'autre part les différentes étapes qui composent ces phases : analyse/diagnostic, conception, configuration/déploiement, exécution, contrôle/supervision. Les formes d'adaptation recouvrent l'adaptation par flexibilité, par déviation et par évolution. Rappelons que l'adaptation par flexibilité consiste à retarder des choix de conception en phase d'exécution. L'adaptation par déviation consiste à remettre en cause le processus prescrit pour face aux imprévus par l'ajout ou la suppression d'activités par exemple. Enfin, l'adaptation par évolution fait suite à une analyse d'exécutions passées et consiste à apporter des modifications aux processus en sur la base de cette analyse.

La Figure 13 ci-dessous présente la grille d'analyse des travaux existants sous la forme d'un graphe où les sommets sont les critères de différenciation des travaux et où les arêtes expriment des relations entre ces critères. En plus des critères relatifs à l'adaptation (cycle de vie et forme d'adaptation), ont été ajoutés des critères descriptifs des travaux : le paradigme sous-jacent (déclaratif, procédural), les techniques (liaison tardive, versions...) ou formalismes (Réseaux de Petri, règles logiques, contraintes...) utilisés pour la spécification des schémas de processus, et les contributions de ces travaux (modèle, système opérationnel, prototype).

#### **3.2 Analyse**

En se basant sur la grille d'analyse précédente, nous classifions les travaux les plus représentatifs des approches précédemment examinées et nous les positionnons dans le Tableau 1. Plus précisément, les critères utilisés pour cette classification sont les formes d'adaptation, la phase dans laquelle se déroule l'adaptation et le paradigme sous-jacent à l'approche ou au travail.





**Figure 13.** Grille d’analyse des travaux existants

La position de chacun de ces travaux est justifiée dans les sections ci-dessous.

Phase / Paradigme	Modélisation		Exécution	
	Déclaratif	Procédural	Déclaratif	Procédural
Flexibilité	<p>Approche par conception - Liaison tardive [Adams et al., 2006]</p> <p>Approche par intention et contexte [Nurcan et al., 2005] [Nurcan et Edme, 2005] [Saidani et Nurcan, 2007-a et -b] [Saidani et Nurcan, 2009] [Rolland et Nurcan, 2010]</p>	<p>Approche par conception - Versions : VerFlexFlow [Chaabane et al., 2010] - Variantes : [Lu et Sadiq, 2006] - Modélisation tardive [Adams et al., 2006]</p>	<p>Approche par intention et contexte [Saidani et Nurcan, 2007-a et -b] [Saidani et Nurcan, 2007-a et -b]</p> <p>Approche par cas [Aalst et al., 2005]</p>	<p>Approche par intégration de protocoles [Faure et al 2010]</p>
Déviation			<p>Approche par contraintes [Péic et Aalst, 2006], [Péic, 2008]</p>	
Evolution				<p>Approche par évolution - Evolution de schémas : Adept [Reichert et al., 2005] - Versions : VerFlexFlow [Chaabane et al., 2010]</p>

**Tableau 1.** Classification des approches et travaux

### **3.2.1 L'approche par évolution**

L'approche par évolution traite essentiellement de l'adaptation par évolution. Elle repose sur le paradigme procédural et il est possible de faire évoluer les schémas de processus à partir d'instances en cours d'exécution, mais également lors de la phase de modélisation lorsqu'aucune instance du processus considéré n'est en cours d'exécution.

Plus précisément, les différents travaux adoptant cette approche ont pour base une définition explicite du schéma des processus. Cette définition spécifie l'ensemble des activités du processus et leur enchaînement. Le paradigme sous-jacent à cette approche est donc le paradigme procédural. D'autre part, ces travaux ont essentiellement défini des opérations de base ou des patrons pour la modification des schémas de processus. C'est principalement la perspective comportementale qui est prise en compte comme par exemple dans WIDE [Casati et al., 1996-a] [Casati et al., 1996-b] ou dans les propositions faites dans [Kammer et al., 1999] ou [Kradolfer et Geppert, 1999]. Certains travaux ont cependant intégré d'autres perspectives, comme dans Adept [Reichert et al., 2005] et [Zhao et Liu, 2007] où la perspective informationnelle est prise en compte, mais c'est essentiellement dans VerFlexFlow [Chaabane et al., 2010] que la modification des schémas pour définir des versions de processus peut porter sur les perspectives comportementale, informationnelle, organisationnelle et intentionnelle (i.e. contexte) des processus. Enfin, il s'agit également d'indiquer que cette approche a fait l'objet de nombreux travaux de recherche et que la contribution opérationnelle phare est certainement Adept [Reichert et al., 2005], issue du milieu académique.

### **3.2.2 L'approche par cas**

L'approche par cas [Aalst et al., 2005] relève d'une combinaison de l'adaptation par flexibilité et par déviation. D'une part, un cas est initié conformément à un schéma de processus, mais il peut être ensuite adapté en cours d'exécution et peut ainsi dévier de son schéma. Bien que les mécanismes qui gouvernent les adaptations soient définis en phase de conception, les adaptations sont réalisées en phase d'exécution. Il y a essentiellement deux mécanismes d'adaptation : des règles actives qui permettent l'agencement dynamique des activités et la gestion événementielle de la saisie des données, et des rôles spécifiques (redo et undo) introduits pour agir sur la structure de contrôle du cas. Les adaptations impactent différentes perspectives : comportementale en agissant sur les activités et leurs relations, organisationnelle compte tenu des rôles spécifiques introduits,

et informationnelle en réaction aux modifications comportementales. [Aalst et al., 2005] fournit un méta modèle UML des concepts manipulés dans cette approche, mais la sémantique opérationnelle est décrite à l'aide de règles actives et donc selon un paradigme déclaratif. Il faut enfin signaler qu'il s'agit de l'une des approches ayant donné lieu au plus grand nombre de contributions opérationnelles, si l'on se réfère aux systèmes qui relèvent de cette approche (FLOWer [Pallas 2002], Staffware Case Handler [Staffware, 2000], COSA Activity Manager [Software-Ley, 2002]) et à la variété des domaines d'application traités (assurance, banque, santé, télécommunications, etc).

### 3.2.3 L'approche par conception

Les travaux de l'approche par conception ont pris le parti de traiter en amont l'adaptation des processus en proposant des solutions évitant d'avoir à adapter. L'approche par conception propose donc des solutions traitant de l'adaptation par flexibilité, en phase de modélisation. Certains travaux reposent sur un paradigme procédural. C'est notamment le cas des travaux autour des variantes [Lu et Sadiq, 2006], [Lu et al., 2006], [Lu et al., 2009] ou des versions alternatives de processus [Châabane et al., 2008] [Châabane et al., 2009]. Par exemple, les auteurs de [Châabane et al., 2010] défendent également l'idée que la notion d'alternative sous-jacente à la définition de versions de processus est une forme d'adaptation par flexibilité puisqu'il est possible de définir en phase de modélisation les changements prévisibles d'un processus, chacun étant exprimé par une des alternatives du processus. D'autres travaux reposent sur un paradigme déclaratif. C'est notamment le cas des techniques retardées (modélisation tardive et liaison tardive) proposées dans [Adams et al., 2006]. En effet, en ce qui concerne la modélisation tardive, le concepteur d'un processus indique lors de la modélisation du schéma du processus considéré, qu'il sait qu'une activité doit être modélisée, mais qu'il ne sait pas le faire pour l'instant. Il laisse donc cette modélisation à l'utilisateur qui sera chargé d'exécuter l'activité en question. La modélisation tardive est une solution pour l'adaptation par flexibilité. Elle se déroule en phase de modélisation et repose sur un paradigme procédural. En ce qui concerne la liaison tardive, le concepteur d'un processus ne précise par nécessairement la manière dont une activité va être réalisée (i.e. il ne définit pas les opérations associées). Il laisse plutôt à l'utilisateur qui sera chargé d'exécuter l'activité en question, le choix de la manière d'exécuter cette activité : ce dernier va doit choisir une solution d'exécution parmi plusieurs. Dans [Adams et al., 2006], un système de règles sert de support au choix de cette solution d'exécution, en accord avec le contexte dans lequel se trouve l'instance du processus. La liaison tardive est

une solution pour l'adaptation par flexibilité. Elle se déroule en phase de modélisation et combine les paradigmes déclaratif et procédural. On peut noter que les techniques retardées, et notamment la liaison tardive à travers la notion de *worklet*, sont mises en œuvre dans le système de gestion de workflow YAWL [Aalst et Hofstede, 2005].

### **3.2.4 L'approche par contrainte**

L'approche par contrainte [Pesic et Aalst, 2006], [Pesic et al., 2007-a], [Pesic et al., 2007-b], [Pesic,2008] relève d'une combinaison de l'adaptation par déviation et par évolution. En effet, l'utilisateur exécutant un processus ne suit pas un schéma prédéfini pour exécuter le processus considéré, mais exécute plutôt les activités qui lui semblent appropriées dans la mesure où elles sont autorisées, i.e. conformes aux règles associées au processus considéré. Il peut donc dévier de ce qu'il fait habituellement pour faire face aux imprévus du cas considéré ; il peut également personnaliser l'exécution du cas en rajoutant des activités. D'autre part, Declare, qui est l'outil phrase implémentant cette approche, supporte l'adaptation par évolution en s'assurant que l'ajout ou la suppression de contraintes ou d'activités est possible en accord avec le cas sur lequel la modification porte (évolution partielle du processus, i.e. ne concernant qu'une de ses instances), ou l'ensemble des cas en cours d'exécution (évolution totale) [Pesic et al., 2007-b]. Si une seule instance est concernée, la modification est acceptée si les contraintes obligatoires sont respectées ; elle est rejetée sinon. Si toutes les instances sont concernées, toutes les instances respectant le nouvel ensemble de contraintes obligatoires vont être modifiées et celles ne respectant ce nouvel ensemble de contraintes vont continuer à s'exécuter en accord avec leur ensemble d'activités et de contraintes précédent. On peut donc constater que le problème de migration d'instances étudié dans l'approche par évolution ne se pose pas dans l'approche par contrainte. D'autre part, même si dans Declare l'étude de l'adaptation porte essentiellement sur la perspective comportementale, les adaptations impactent également les perspectives organisationnelle et informationnelle [Pesic, 2008]. L'approche par contrainte repose sur un paradigme déclaratif puisque c'est à travers un ensemble de contraintes que le schéma des processus est implicitement défini. Enfin, on peut signaler que les travaux adoptant cette approche sont somme toute assez récents et peu nombreux.

### **3.2.5 L'approche par intention et contexte**

L'approche par intention et contexte [Saidani et Nurcan,2006-a], [Saidani et Nurcan,2006-b], [Saidani et Nurcan,2007-a], [Saidani et Nurcan,2007-b],

[Saidani et Nurcan, 2009] relève d'une combinaison de l'adaptation par flexibilité et par déviation. Les utilisateurs impliqués dans l'exécution d'un processus vont, en fonction du contexte et des objectifs à attendre, choisir à l'aide de la définition des variantes, spécifiées pendant la phase de conception, les activités les plus appropriées. Ils n'ont pas à leur disposition de plan pré établi, simplement des habitudes auxquelles ils peuvent déroger (dévier) suivant les cas. Le processus se construit donc de manière flexible. Par contre, les activités, qu'elles soient atomiques ou complexes (sous-processus) ont un schéma pré établi, que l'utilisateur va suivre lors de leur exécution. Cette approche mixe donc déclaratif et procédural, respectivement utilisés pour décrire les possibilités des utilisateurs et les opérations associées à des activités atomiques ou des sous-processus. Certaines perspectives des processus sont présentes dans cette approche : la perspective comportementale est spécifiée de manière déclarative et procédurale pour guider les actions de l'utilisateur tandis que la perspective organisationnelle est prise en compte à travers les notions de rôle, de groupe et d'acteur. La modélisation intentionnelle [Nurcan et al., 2005], [Nurcan et Edme, 2005], [Rolland et Nurcan, 2010] permet de capturer la variabilité des modèles de processus à un niveau intentionnel dès la phase de conception.

### **3.2.6 L'approche par intégration de protocoles d'interaction**

L'approche par intégration de protocoles, telle que présentée dans [Faure et al., 2010], relève de l'adaptation par flexibilité. En effet, d'une part l'intégration de protocoles permet de laisser une part d'indéterminisme dans le comportement des processus en phase de conception et l'aiguillage du processus se décide dynamiquement en phase d'exécution en fonction du résultat produit par l'exécution du protocole. D'autre part, les protocoles peuvent être intégrés dans les processus lors de leur conception ou faire l'objet d'une procédure de sélection en cours d'exécution (liaison tardive). Cette approche impacte essentiellement la perspective comportementale des processus puisqu'elle aide à diriger l'exécution. Néanmoins, les autres perspectives sont référencées. En effet, un protocole a une finalité qui doit être en conformité avec une part du processus dans lequel il s'intègre, il utilise et produit des informations manipulées par le processus et les acteurs du protocole sont susceptibles d'être ceux du processus. L'approche par intégration de protocoles intervient en phase de modélisation et plus précisément lors de l'étape de conception où il s'agit de définir les protocoles et de les intégrer dans les processus. Cette intégration est statique ou dynamique. L'intégration statique consiste à considérer le protocole comme une activité du processus tandis que l'intégration dynamique consiste à intégrer une activité de sélection de protocole à exécuter. D'un point de vue

conceptuel, les auteurs de cette approche ont produit un méta-modèle de protocole d'interaction indépendant de toute considération technique. Ils ont cependant adopté le paradigme de programmation procédural puisqu'ils utilisent les Réseaux de Petri à Objets pour spécifier les protocoles et le langage BPEL pour leur mise en œuvre.

## **4. BILAN ET PERSPECTIVES**

Cette section a pour objectif de faire le bilan des travaux présentés précédemment en se basant sur le tableau de la section précédente, puis d'indiquer un ensemble de perspectives pour l'adaptation des processus.

### **4.1 Bilan**

Deux positionnements sont examinés dans ce bilan : (i) le premier positionne les différentes approches étudiées relativement aux critères de différenciation de l'adaptation, et (ii) le second mesure le champ d'application des approches d'adaptation par rapport à la typologie introduite par Ellis [Ellis, 2000] qui classe les technologies collaboratives.

#### **4.1.1 Positionnement des différentes approches**

L'observation du Tableau 1 appelle les remarques suivantes. La première constatation est que les travaux d'adaptation proposent des solutions qui interviennent majoritairement en phase d'exécution. De plus, les solutions intervenant en phase de modélisation se limitent à l'adaptation par flexibilité. En ce qui concerne les deux paradigmes de modélisation, ils sont utilisés aussi bien en phase de modélisation que d'exécution, et sont parfois combinés entre eux. Cependant, la tendance actuelle est de privilégier le paradigme déclaratif qui apporte indéniablement davantage de souplesse pour traiter ce problème d'adaptation. En revanche, cette souplesse a un coût en termes de compréhension et de maintenabilité des modèles produits. Par exemple si l'on considère l'approche par contrainte, un concepteur de processus peut éprouver des difficultés à mesurer l'univers des processus possibles qui découlent de sa modélisation. Des travaux très récents se sont penchés sur ce problème et proposent d'utiliser une approche par test (TDM pour Test Driven Modeling [Zugal et al., 2011]) permettant d'exécuter des scénarios et distinguer les processus possibles des processus interdits.

Notre connaissance du domaine nous permet aussi d'affirmer que les travaux autour de l'adaptation par évolution sont complets : le problème de la modification des schémas de processus a été étudié en profondeur, celui de la migration a été traité à diverses reprises et l'adaptation par les versions largement explorée. Une des particularités fortes de ces travaux est que la majorité d'entre eux ont suivi un cycle classique allant de la recherche au développement d'outils à l'exception de l'approche par cas qui a donné lieu à de nombreux systèmes et applications avant d'être formalisée. Un autre élément à noter est que ces approches ont été proposées de manière séparée, mais la combinaison de certaines d'entre elles nous semble possible et pertinente. Par exemple, le schéma résultant d'une exécution par cas peut être matérialisé en une version de processus. De manière plus générale, on peut aussi constater que les techniques de découverte de processus [Aalst et al., 2011] sont sous exploitées pour l'adaptation. En effet, elles pourraient être utilisées pour servir à l'analyse de plusieurs cas et conduire à la découverte de processus génériques, et fermer ainsi le cycle de développement des processus (cf. Figure 2, Section 1).

#### **4.1.2 Impact de ces approches : du workflow au groupeware**

Nous avons conduit ce chapitre avec la perspective d'examiner la problématique de l'adaptation des processus afin qu'ils soient déployés avec des outils de type Workflow. Les différents travaux qui ont servi à illustrer cette problématique permettent de coordonner les activités d'un processus. Cependant, si l'on se réfère au domaine des collecticiels (CSCW ou Computer Supported Cooperative Work), on peut constater que ces propositions ont un usage plus large que la simple coordination. En effet, en considérant la typologie introduite par [Ellis, 2000], les technologies collaboratives peuvent être divisées en quatre classes :

- Gestionnaires d'artefacts partagés (keepers) qui sont des outils permettant l'accès et la gestion d'informations partagées offrant des fonctionnalités telles que la gestion des accès concurrents, le versionnement, la reprise après panne. Les outils d'édition partagée ou les agendas de groupe relèvent de cette classe.
- Coordinateurs (coordinators) qui désignent principalement les systèmes de gestion de processus.
- Facilitateurs de communication (communicators) qui désignent des gestionnaires de conversations depuis les gestionnaires de mails jusqu'au systèmes de vidéo-conférence.

- Agents personnels (team-agents) qui réalisent une tâche au service d'une ou plusieurs personnes comme des synchroniseurs de calendrier.

Si l'on considère les différentes approches d'adaptation présentées, on peut constater que la plupart d'entre elles couvrent plusieurs classes de cette typologie à l'exception des approches par évolution et contrainte qui se limitent à de la coordination. Comme le fait remarquer [Aalst et al., 2005], l'approche par cas supporte les trois premières classes de la typologie d'Ellis. En effet, les éléments des instances (cas) de processus, i.e. les activités qui le composent et les données en entrée et sortie de ces activités, sont des objets partagés que les acteurs du processus peuvent conjointement manipuler. D'autre part, les différents acteurs impliqués dans l'exécution de ces instances sont informés (notifiés) de l'avancement du cas dans un espace commun ; de plus, ces acteurs peuvent directement interagir entre eux. La coordination repose sur le fait que le cas s'exécute sur la base d'un schéma de référence. Enfin, l'approche par intégration de protocoles constitue à la fois un coordinateur et un facilitateur de communications. En effet, l'intégration d'un protocole dans un processus permet de guider le processus en question et participe donc à sa coordination. En même temps, le déploiement d'un protocole considéré individuellement est un support à la communication puisqu'il régleme des conversations structurées.

Ces différentes considérations montrent que les approches proposées pour pallier le problème de l'adaptation des processus ont un domaine d'application plus large que celui habituellement dévolu au workflow. Les approches proposées permettent également de considérer des processus ad hoc, i.e. des processus dont il n'est pas toujours possible de définir les règles d'enchaînement des activités à l'avance, et habituellement peu pris en compte par les systèmes de gestion de workflow. Enfin, la place prise par l'utilisateur évolue : dans les approches par cas, contrainte, et intention et contexte, le degré de liberté qui leur est accordé, individuellement ou collectivement, sur le contrôle du processus est plus important que celui qui leur est habituellement laissé.

## 4.2 Perspectives

Deux perspectives principales vont être discutées pour l'adaptation des processus. La première perspective concerne le passage d'une adaptation menée par un utilisateur ou un concepteur de processus, à une adaptation suggérée, voire menée de manière autonome par le système. La deuxième perspective concerne la portée de ce chapitre, qui a principalement considéré l'adaptation des processus dans le contexte d'une organisation. Mais cette



portée peut être étendue pour prendre en compte les processus entre organisations.

#### **4.2.1 Vers un guidage automatique de l'adaptation**

Les techniques proposées pour l'adaptation sont principalement à la charge des utilisateurs en phase d'exécution des processus. D'une part, ils doivent identifier le besoin d'adaptation et d'autre part procéder aux ajustements correspondants. Une des voies que nous pensons être prometteuse est le guidage de l'utilisateur dans cette activité. Il s'agit de pouvoir disposer d'un composant chargé de l'adaptation et ayant des capacités de perception, décision et d'action. La perception consiste à l'identification du besoin d'adaptation, par réaction à des événements ou par anticipation de ce qui peut se passer dans le futur. La décision consiste à choisir le type d'adaptation à appliquer. L'action peut, quant à elle, consister à recommander l'adaptation choisie à l'utilisateur pour le guider, ou bien à la mettre en œuvre de manière autonome. L'anticipation peut être basée sur des prédictions qui s'appuient sur données passées, sur le cas en cours, et sur des mesures (probabilités de succès ou d'échec du cas en cours, estimation du temps restant pour le traitement du cas en cours, ...) [Aalst, 2011].

D'un point de vue technologique ce composant peut être mise en œuvre par l'extension des outils de découverte de processus (process mining), pour en faire un usage temps réel plutôt que l'usage conventionnel, post-exécution, sous forme d'audit. D'autre part, la technologie agent est bien appropriée pour dérouler le cycle de vie Perception, Décision, Action, éventuellement de manière autonome pour tendre vers de l'auto adaptation [Di Marzo et al., 2011].

#### **4.2.2 Vers l'adaptation de processus inter organisationnels**

Cette étude a porté sur l'adaptation des processus dans un cadre uniquement intra organisationnel. Or, on constate aujourd'hui une évolution naturelle vers le Workflow Inter Organisationnel dont l'objectif est de gérer et automatiser des processus qui traversent les organisations [Aalst et Hofstede, 2002]. Dans ce contexte, le problème de l'adaptation devient plus complexe, car il s'agit de prendre en compte l'hétérogénéité des processus partenaires, leur distribution, leur autonomie, pour les coordonner de manière efficace et cohérente vers l'objectif global, c'est-à-dire la production d'un service commun [Andonoff et al., 2006]. L'adaptation d'un processus inter organisationnel porte essentiellement sur cette coordination, c'est à dire sur la synchronisation des différents processus partenaires.

On trouve deux formes de schéma de coordination entre processus impliqués dans un WIO :

- coordination syntaxique, limitée à de la structure de contrôle, comme par exemple, l'exécution chaînée, le partage de capacité, le transfert de cas [Aalst, 2000],
- coordination sémantique, qui tient compte de la sémantique des échanges entre partenaires, et qui est le plus souvent basé sur des protocoles comme par exemple la sous-traitance, la négociation, la délégation, l'appel d'offre [Bouaziz, 2010].

La problématique de l'adaptation va essentiellement consister à définir des mécanismes d'adaptation de protocoles ou mettre à disposition du WIO une base de protocoles dans laquelle on peut sélectionner le plus adapté de manière dynamique.

Un des verrous scientifiques concerne la consistance des adaptations opérées dans le WIO : quelles adaptations du schéma de coordination n'impactent pas les processus partenaires ? Inversement, quelles adaptations des processus partenaires sont sans conséquence sur le schéma de coordination ? Comment s'assurer que l'objectif global du WIO peut continuer à être atteint malgré des adaptations locales ou du schéma de coordination ?

Ce problème a été abordé dans le cadre des composants logiciels [Sibertin et al., 2007] et des services [Baldoni et al., 2005]. Il mérite d'être traité dans le cadre des processus inter organisationnels.

## RÉFÉRENCES

- [Aalst, 2000] W. van der Aalst, Loosely coupled interorganizational workflows: modeling and analyzing workflows crossing organizational boundaries. *International Journal on Information and Management*, Vol. 37, n°2, p. 67–75, 2000.
- [Aalst et Berrens, 2001] W. van der Aalst, P. Berrens, Beyond Workflow Management: Product-Driven Case Handling. *International Conference on Supporting Group Work*, Boulder, Colorado, USA, October 2001, p. 42–51.
- [Aalst, 2001] W. van der Aalst, How to Handle Dynamic Change and Capture Management Information: An Approach based on Generic Workflow Model. *International Journal on Computer Systems: Science and Engineering*, Vol. 16, n° 5, September 2001, p. 295–318.
- [Aalst et Hee, 2002] W. van der Aalst, K. van Hee, *Workflow Management: Models, Methods and Systems*. MIT Press 2002.
- [Aalst et al., 2003-a] W. van der Aalst, A. ter Hofstede, M. Weske, Business Process Management: A Survey. *International Conference on Business Process Management*, Eindhoven, The Netherlands, June 2003, p. 1–12.
- [Aalst et al., 2003-b] W. van der Aalst, A. ter Hofstede, B. Kiepuszewski, A. Barros, Workflow Patterns. *International Journal on Distributed and Parallel Databases*, Vol. 14, n°1, 2003, p. 5–51.
- [Aalst et Hofstede, 2005] W. van der Aalst, A. ter Hofstede, YAWL: Yet Another Workflow Language. *International Journal on Information Systems*, Vol. 30, n°4, 2005, p. 245–275.
- [Aalst et al., 2005] W. van der Aalst, M. Weske, D. Grünbauer, Case Handling: a New paradigm for Business Process Support. *International Journal on Data Knowledge Engineering*, Vol. 53, n°2, 2005, p.129–162.
- [Aalst et al., 2009] W. van der Aalst, M. Pesic, H. Schonenberg, Declaration Workflow: Balancing between Flexibility and Support. *International Journal on Computer Science – Research and Development*, Vol. 23, n°2, 2009, p. 99–113.
- [Aalst, 2011] W. van der Aalst, *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Edition Springer, 2011, p. 1–352.
- [Aalst et al., 2011] W. van der Aalst, A. Adriansyah, AK. Alves de Medeiros, F. Arcieri, T. Baier, T. Blickle, RP. Jagadeesh Chandra Bose, P. van den Brand, R. Brandtjen, J. Buijs, A. Burattin, J. Carmona, M. Castellanos, J. Claes, J. Cook, N. Costantini, F. Curbera, E. Damiani, M. de Leoni, P. Delias, B. van Dongen, M. Dumas, S. Dustdar, D. Fahland, D. Ferreira, W. Gaaloul, F. van Geffen, S. Goel, C. Günther, A. Guzzo, P. Harmon, A. ter Hofstede, J. Hoogland, JE. Ingvaldsen, K. Kato, R. Kuhn, A. Kumar, M. La Rosa, F. Maggi, D. Malerba, RS. Mans, A. Manuel, M. McCreesh, P. Mello, J. Mendling, M. Montali, H. Motahari Nezhad, M. zur Muehlen, J. Muñoz-Gama, L. Pontieri, J. Ribeiro, A. Rozinat, H. Seguel Pérez, R. Seguel Pérez, M. Sepúlveda, J. Sinur, P. Soffer, M. Song, A. Sperduti, G. Stilo,

C. Stoel, K. Swenson, M. Talamo, W. Tan, C. Turner, J. Vanthienen, G. Varvaressos, E. Verbeek, M. Verdonk, R. Vigo, J. Wang, B. Weber, M. Weidlich, T. Weijters, L. Wen, M.I Westergaard, M. Thandar Wynn, Process Mining Manifesto. International Workshop on Business Process Intelligence, held at International Conference on Business Process Management, Clermont-Ferrand, August 2011, p. 169–194.

[Adams et al., 2006] M. Adams, A. ter Hofstede, D. Edmond, W. van der Aalst, Worklets: A Service-Oriented Implementation of Dynamic Flexibility in Workflows. International Conference on Cooperative Information Systems, Montpellier, France, October 2006, p. 291–308.

[Andonoff et al., 2006] E. Andonoff, L. Bouzguenda, C. Hanachi, Coordination dans le Workflow inter organisationnel lâche : une approche basée sur les agents et le Web sémantique. Revue Ingénierie des Systèmes d'Information, Vol. 11, n°3, p. 127–150, 2006.

[Baldoni et al., 2005] M. Baldoni, C. Baroglio, A. Martelli, V. Patti, C. Schifanella, Verifying the Conformance of Web Services to Global Interaction Protocols: A First Step. International Workshop on Web Services and Formal Methods, Versailles, France, September 2005, p. 257–271.

[Barrios et Nurcan, 2004] J. Barrios, S. Surcan, Model Driven Architectures for Enterprise Information Systems. International Conference on Advanced Information Systems Engineering, June 2004, Riga, Latvia, p. 3–19.

[Bessai et Nurcan, 2009] K. Bessai, S. Nurcan. An actor-driven approach for business processes - How to take into account the environment of work? International Workshop on Business Process Modelling, Development, and Support, held at International Conference on Advanced information System Engineering, Amsterdam, The Netherlands, June 2009, p. 187–196.

[Bouaziz, 2010] W. Bouaziz, Coordination à base de Protocoles dans les Systèmes Multi-agent : Application au Workflow Inter-Organisationnel. Thèse de l'Université Toulouse 1, Institut de Recherche en Informatique de Toulouse, Toulouse, Septembre 2010.

[Boukhebouze et al., 2009] M. Boukhebouze, Y. Amghar, A. Benharkat, Z. Mamaar, A Rule-based Modelling for the Description of Flexible and Self-Healing Business Processes. International Conference on Advances Databases and Information Systems, Riga, Latvia, September 2009, p. 15–27.

[BPMN, 2011] Specifications of BPMN 2.0. Information available at <http://www.omg.org/spec/BPMN/2.0/>, January 2011.

[Casati et al., 1996-a] F. Casati, S. Ceri, B. Pernici, G. Pozzi, Workflow Évolution. International Conference on the Entity Relationship Approach, Cottbus, Germany, October 2006, p. 438–455.

[Casati et al., 1996-b] F. Casati, S. Ceri, B. Pernici, G. Pozzi, Deriving Active Rules for Workflow Enactment. International Conference on Database Application and Expert Systems, Zurich, Switzerland, September 1996, p.94–115.

[Châabane et al., 2008] MA. Châabane, E. Andonoff, L. Bouzguenda, R. Bouaziz, Dealing with Business Process Evolution using Versions. International Conference on E-Business, Porto, Portugal, July 2008, p. 267–278.

[Châabane et al., 2010] MA. Châabane, E. Andonoff, L. Bouzguenda, R. Bouaziz, Modélisation multidimensionnelle des versions de processus. Revue Ingénierie des Systèmes d'Information, Vol. 15, n° 5, Décembre 2010, p. 89–114.

[Charoy et al., 2006] F. Charoy, A. Guabtni, M. Valdes Faura, A Dynamic Workflow Management System for Coordination of Cooperative Activities. International Workshop on Dynamic Process Management, at International Conference on Business Process Management, Vienna, Austria, September 2006, p. 205–216.

[Chebbi et al., 2006] I. Chebbi, S. Dustdar, S. Tata, The view-based approach to dynamic inter-organizational workflow cooperation. International Journal on Data and Knowledge Engineering, Vol. 56, n°2, p. 139–173, 2006.

[Desel, 2002] J. Desel, Process Modelling using Petri Nets. Chapter 7 in [Dumas2002], 2002, p. 147–176.

[Di Marzo et al., 2011] G. Di Marzo Serugendo, MP. Gleizes, A. Karageorgos, Self-organizing Software - From Natural to Artificial Adaptation. Edition Springer, Natural Computing Series, October 2011.

[Dumas et al., 2002] M. Dumas, W. van der Aalst, A. ter Hofstede, Process-Aware Information Systems: Bridging People and Software through Process Technology. Wiley-Interscience 2005.

[Engels et al., 2002] G. Engels, A. Förster, R. Heckel, S. Thöne, Process Modelling using UML. Chapter 5 in [Dumas2002], p. 85–116, 2002.

[Ellis, 2000] C.A. Ellis, An Evaluation Framework for Collaborative Systems. Technical Report, CU-CS-901-00, Department of Computer Science, University of Colorado, Boulder, USA, 2000.

[Faure et al., 2010] C. Faure, E. Andonoff, C. Hanachi, C. Sibertin-Blanc, N. Salatgé, Flexibilité de processus de gestion de crises par intégration de protocoles d'interaction. Revue Ingénierie des Systèmes d'Information, Vol. 15, n° 3, 2010, p.37–60.

[Grigori et al., 2004] D. Grigori, F. Charoy, C. Godart, Coo-Flow: A Process Technology To Support Cooperative Processes. International Journal on Software Engineering and Knowledge Engineering, Vol. 14, n°1, p. 61–78, 2004.

[Hanachi et Sibertin, 2004] C. Hanachi, C. Sibertin-Blanc, Protocol Moderators as Active Middle-Agents in Multi-Agent Systems. International Journal on Autonomous Agents and Multi-Agent Systems, Vol. 8, n°3, April 2004, p. 131–164.

[Hemaïssia et al., 2008] M. Hemaïssia-Jeannin, A. El Fallah-Seghrouchni, C. Labreuche, A New Multilateral multi-issue Negotiation Protocol and its application to a Crisis Management Problem. International Journal on Multiagent and Grid Systems, Vol. 4, n°1, 2008, p. 103–123.

- [Kammer et al., 1999] P. Kammer, G. Bolcer, R. Taylor, R. Bergman, Techniques for supporting Dynamic and Adaptive Workflow. *International Journal on Computer Supported Cooperative Work*, Vol. 9, n° 3/4, 1999, p. 269–292.
- [Kradolfer et Geppert, 1999] M. Kradolfer, A. Geppert, Dynamic Workflow Schema Evolution based on Workflow Type Versioning and Workflow Migration. *International Conference on Cooperative Information Systems*, Edinburgh, Scotland, September 1999, p. 104–114.
- [Lezoche et al., 2008] M. Lezoche, M. Missikoff, L. Tininini, Business Process Evolution : a Rule-based Approach. *International Workshop on Business Process Modelling, Development and Support*, held at *International Conference on Advanced Information Systems*, Montpellier, France, September 2008.
- [Lu et Sadiq, 2006] R. Lu, S. Sadiq, Managing Process Variants as an Information Resource. *International Conference on Business Process Management*, Vienna, Austria, September 2006, p. 426–431
- [Lu et al., 2006] R. Lu, S. Sadiq, G. Governatori, Utilizing Successful Work Practice for Business Process Evolution. *International Conference on Business Information Systems*, Klagenfurt, Austria, May 2006, p. 58–76.
- [Lu et al., 2009] R. Lu, S. Sadiq, G. Governatori, X. Yan, Defining Adaptation Constraints for Business Process Variant. *International Conference on Business Information Systems*, Poznan, Poland, April 2009, p. 145–156.
- [Mendling et Nüttgens, 2003] J. Mendling, M. Nüttgens, EPC Syntax Validation with XML Schema Languages. *International Workshop on Business Process Management with Event-Driven Process Chains*, Bamberg, Germany, 2003.
- [Muller et al., 2004] R. Muller, U. Greiner, E. Rahm, AgentWork: a Workflow System supporting Rule-based Workflow Adaptation. *International Journal on Data Engineering*, Vol. 51, n°2, 2004, p. 223–256.
- [Muller et al., 2008] D. Muller, M Reichert, J. Herbst, A New Paradigm Enactment and Dynamic Adaptation of Data-Driven Process Structures. *International Conference on Advanced Information Systems*, Montpellier, France, June 2008, p. 48–63.
- [Nardi, 1996] BA. Nardi, *Context and Consciousness: Activity Theory and Human Computer Interaction*. MIT Press, Cambridge, Massachusetts, 1996.
- [Nii, 1986] H. Nii, Blackboard Systems, Part One: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. *Magazine on Artificial Intelligence*, Vol. 7, n°2, 1986, p. 38–53.
- [Nurcan, 1996] S. Nurcan. A method for Cooperative Information Systems Analysis and Design: CISAD. *International Conference on the Design of Cooperative Systems*, Juan-Les-Pins, June 1996, p.681–700.
- [Nurcan, 2004] S. Nurcan. Business Process Modeling for developing Process Oriented IT Systems. The "Business Process Management Tools and Technologies" track of the 2004 Information Resources Management Association International Conference, May 2004, New Orleans, USA.

[Nurcan, 2008] S. Nurcan, A Survey on the Flexibility Requirements related to Business Processes and Modelling Artifacts. International Conference on System Science, Waikoloa, Big Island, Hawaii, USA, January 2008, p. 378–387.

[Nurcan et Edme, 2005] S. Nurcan, M.-H. Edme, Intention Driven Modelling for Flexible Workflow Applications. International Journal on Business Process Management, Development and Support, Special issue on Software Process: Improvement and Practice, Vol. 10, n°4, 2005.

[Nurcan et Rolland, 1997] S. Nurcan, C. Rolland. Meta-modelling for cooperative processes. European-Japanese Conference on Information Modelling and Knowledge Bases, May 1997, Toulouse, p. 361–377.

[Nurcan et Rolland, 2003] S. Nurcan, C. Rolland. A multi-method for defining the organizational change. Journal of Information and Software Technology, Vol. 45, n°2, 2003, p. 61–82.

[Nurcan et al., 1999] S. Nurcan, J. Barrios, G. Grosz, C. Rolland. Change process modelling using the EKD - Change Management Method. European Conference on Information Systems, ECIS'99, Copenhagen, Denmark, June 1999, p.513–529.

[Nurcan et al., 2005] S. Nurcan, A. Etien, R. Kaabi, I. Zoukar, C. Rolland, A Strategy Driven Business Process Modelling Approach. International Journal on Business Process Management, Special issue on Goal-oriented business process modelling, Vol. 11, n°6, 2005.

[Ould, 1995] MA. Ould, Business Processes: Modelling and analysis for re-engineering and improvement, John Wiley & Sons Editon, 1995.

[Pallas 2002] Pallas Athena company, Flower User Manual. Pallas Athena BV, Apeldoorn, The Netherlands, 2002.

[Pesic et Aalst, 2006] M. Pesic, W. van der Aalst, A Declarative Approach for Flexible Business Processes. International Workshop on Dynamic Process Management, held at International Conference on Business Process Management, Vienna, Austria, September 2006, p. 169–180.

[Pesic et al., 2007-a] M. Pesic, H. Schonenberg, N. Sidorova, W. van der Aalst, DECLARE: Full Support for Loosely-Structured Processes. International Conference on Enterprise Distributed Object Computing, Annapolis, Maryland, USA, October 2007, p. 287–300.

[Pesic et al., 2007-b] M. Pesic, H. Schonenberg, W. van der Aalst, Constraint-Based Workflow Models: Change Made Easy. International Conference on Cooperative Information Systems, Vilamoura, Portugal, November 2007, p. 77–94.

[Pesic, 2008] M. Pesic, Constraint-based workflow management systems: shifting control to users. PhD thesis, Eindhoven University of Technology, Eindhoven, 2008.

[Reichert et Dadam, 1998] M. Reichert, P. Dadam, ADEPT-Flex – Supporting Dynamic Changes of Workflow without Loosing Control. International Journal on Intelligent Information Systems, Special Issue on Workflow Management Systems, Vol. 10, n°2, 1998, p. 93–129.

[Reichert et al., 2003-a] M. Reichert, S. Rinderle, P. Dadam, On the Common Support of Workflow Type and Instance Changes under Correctness Constraints. International Conference on Cooperative Information System, Catania, Sicily, Italy, November 2003, p. 407–425.

[Reichert et al., 2003-b] M. Reichert, S. Rinderle, P. Dadam, ADEPT Workflow Management System: Flexible Support for Enterprise-Wide Business Processes. International Conference on Business Process Management, Eindhoven, The Netherlands, June 2003, p. 370–379.

[Reichert et al., 2005] M. Reichert, S. Rinderle, U. Kreher, P. Dadam, Adaptive Process Management with ADEPT2. International Conference on Data Engineering, Demo Session, Tokyo, Japan, April 2005, p. 1113–1114.

[Reijers, 2006] Reijers, H., Workflow Flexibility: the Forlon Promise. International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Manchester, United Kingdom, June 2006, p. 271–272.

[Rinderle et Reichert, 2006] S. Rinderle, M. Reichert, Data-Driven Process Control and Exception Handling in Process Management Systems. International Conference on Advanced Information Systems, Luxembourg, Luxembourg, June 2006, p. 273–287.

[Rolland et Nurcan, 2010] C. Rolland, S. Nurcan, Business Process Lines to deal with the Variability. Hawaii International Conference on System Sciences, Kauai, Hawaii, USA, January 2010.

[Rolland et al., 1999] C. Rolland, N. Prakash, A. Benjamen, A Multi-Model View of Process Modelling. International Journal on Requirements Engineering, p. 169–187, 1999.

[Rosemann et Aalst, 2007] M. Rosemann, W. van der Aalst. A Configurable Reference Modelling Language. International Journal on Information Systems, Vol. 32, n°1, p.1–23, 2007.

[Rosemann et al., 2008] M. Rosemann, J. Recker, C. Flender, Contextualization of Business Processes. International Journal on Business Process Integration and Management, Vol. 3, n°1, p. 47–60, 2008.

[Russel et al., 2005] N. Russell, W. van der Aalst, A. ter Hofstede, Designing a Workflow System Using Coloured Petri Nets. International Journal on Transactions on Petri Nets and Other Models of Concurrency, Vol. 3, 2009, p. 1–24.

[Saidani et Nurcan, 2006-a] O. Saidani, S. Nurcan, FORBAC: A Flexible Organization and Role-Based Access Control Model for Secure Information Systems. International Conference on Advances in Information Systems, Izmir, Turkey, October 2006, p. 364–376.

[Saidani et Nurcan, 2006-b] O. Saidani, S. Nurcan, A Role-based Approach for Modelling Flexible Business Processes. International Workshop on Business Process Modelling, Development and Support, held at International Conference on Advanced Information Systems Engineering, Luxembourg, Luxembourg, June 2006.



[Saidani et Nurcan, 2007-a] O. Saidani, S. Nurcan. Multi-Level Delegation for Flexible Business Process Modeling. International Conference of Information Resources Management Association, Business Process Management track, Vancouver, British Columbia, Canada, May 2007.

[Saidani et Nurcan, 2007-b] O. Saidani, S. Nurcan, Towards Context Aware Business Process Modelling. International Workshop on Business Process Modelling, Development and Support, held at International Conference on Advanced Information Systems, Trondheim, Norway, June 2007.

[Saidani et Nurcan, 2009] O. Saidani, S. Nurcan, Context-Awareness for adequate business process modelling. International Conference on Research Challenges in Information Science, Fes, Morocco, April 2009, p. 177–186.

[Scheer et al., 2002] AW Scheer, O. Thomas, O. Adam, Process Modelling using Event-Driven Process Chain. Chapter 6 in [Dumas2002], 2002, p. 119–144.

[Schonenberg et al. 2008] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, W. van der Aalst, Process Flexibility: A Survey of Contemporary Approaches, International Workshop on CIAO/EOMAS, held at International Conference on Advanced Information Systems, Montpellier, France, June 2008, p. 16–30.

[Sandhu et al., 1999] R. Sandhu, V. Bhamidipati, Q. Munawer, The ARBAC97 Model for Role-Based Administration of Roles, ACM Transactions on Information and System Security, Vol. 2, n°1, 1999.

[Sibertin, 1985] C. Sibertin-Blanc, High Level Petri Nets with Data Structure. International Workshop on Petri Nets and Applications, Espoo, Finland, 1985.

[Sibertin et al., 2007] C. Sibertin-Blanc, P. Mauran, G. Padiou, Safe Adaptation of Component Coordination. Electronic Notes in Theoretical Computer Science, Vol. 189, 2007, p. 69–85.

[Smith et Fingar, 2003] H. Smith, P. Fingar, Business Process Management: the Third Wave. Megan-Kiffer Press, 2003.

[Software-Ley, 2002] Software-Ley company, COSA Activity Manager. Software-Ley GmbH, Pullheim, Germany, 2002.

[Staffware, 2000] Staffware company, Staffware Case Handler–White Paper. Staffware PLC, Berkshire, United Kingdom, 2000.

[Tamma et al., 2002] V. Tamma, M. Wooldridge, I. Blacoe, I. Dickinson, An Ontology Based Approach to Automated Negotiation. International Workshop on Agent-Mediated Electronic Commerce, Bologna, Italy, July 2002, p. 219–237.

[Weber et al., 2008] B. Weber, M. Reichert, S. Rinderle, Change Patterns and Change Support Features – Enhancing Flexibility in Process-Aware Information Systems. International Journal on Data and Knowledge Engineering, Vol. 66, n° 3, p. 438–466, 2008.

[Weber et al., 2009] B. Weber, S. Sadiq, M. Reichert, Beyond Rigidity – Dynamic Process Lifecycle Support: A Survey on Dynamic Changes in Process-aware

Information Systems. International Journal on Computer Science – Research and Development, Vol. 23, n°2, 2009, p. 47–65.

[Weske, 2007] M. Weske, Business Process Management: Concepts, Languages, Architectures. Springer-Verlag 2007.

[WIDE, 1999] ESPRIT 4 Project N. 20280, WIDE Workflow on Intelligent Distributed Database Environment, November 1995-April 1999, Information available at <ftp://ftp.elet.polimi.it/users/Barbara.Pernici/is/process.html>.

[Wohed et al., 2006] P. Wohed, W. van der Aalst, M. Dumas, A. ter Hofstede, N. Russell, On the Suitability of BPMN for Business Process Modelling. International Conference on Business Process Management, Vienna, Austria, p.161–176, September 2006.

[YAWL Foundation, 2004] YAWL Foundation, Yet Another Workflow Language.available at: <http://www.yawlfoundation.org/>

[Zhao et Liu, 2007] X. Zhao, C. Liu, Version Management in the Business Change Context. International Conference on Business Process Management, Brisbane, Australia, September 2007, p. 198–213.

[Zugal et al., 2011] S. Zugal, J. Pinggera, B. Weber, The Impact of Testcases on the Maintainability of Declarative Process Models. International Conference on Enterprise, Business-Process and Information Systems Modeling, BPMDS/EMMSAD, held at International Conference on Advanced Information System Engineering, London, UK, June 2011, p.163–177.