

Conception et mise en place d'un système Workflow pour l'environnement cloud

Institution



Workflow & Cloud

djamel

Zerrouki Djamel

26 mars 2019

Résumé

.....

Abstrait

.....

Déclaration

.....

Acknowledgements

I want to thank...

Table des figures

Liste des tableaux

Table des matières

1	Concepts fondamentaux de cloud et de workflow	9
2	Modélisations et Réseau de Petri	10
2.1	Introduction	10
2.2	Modélisation des processus workflow	10
2.2.1	Aspects à modéliser	10
2.2.1.1	L’aspect fonctionnel	10
2.2.1.2	L’aspect comportemental	10
2.2.1.3	L’aspect informationnel (données)	11
2.2.1.4	L’aspect organisationnel	11
2.2.2	Techniques et outils de modélisation de workflow	11
2.2.2.1	Réseaux de Petri et workflow	11
2.2.2.2	UML et workflow	12
2.2.2.3	YAWL	12

Chapitre 1

Concepts fondamentaux de cloud et de workflow

Chapitre 2

Modélisations et Réseau de Petri

2.1 Introduction

La modélisation par les processus dont les domaines d'application étaient relativement restreints à l'origine, n'a cessé de progresser. Elle s'impose, aujourd'hui, en tant qu'outil incontournable, placé au cœur de la gestion des organisations. La déclinaison des processus organisationnels au niveau des architectures logicielles a donné, entre autres, naissance au workflow afin de répondre à des besoins divers dont celui d'optimisation des processus opérationnels, support et de pilotage.

2.2 Modélisation des processus workflow

La modélisation est une activité qui précède toute décision ou formulation, elle permet de représenter la description du système réel. Tout comme un système informatique, le système workflow comporte un certain nombre d'aspects à modéliser. Nous présentons en premier lieu ces aspects, nous décrivons en second lieu les principales techniques de modélisation utilisées dans le domaine de workflow et nous terminons cette section par évoquer certains aspects temporels et organisationnels des workflows.

2.2.1 Aspects à modéliser

2.2.1.1 L'aspect fonctionnel

L'aspect fonctionnel concerne l'identification des activités des processus que l'on souhaite modéliser. Il est important de comprendre qu'il ne s'agit pas uniquement d'identifier les fonctions des différents départements d'une organisation mais aussi de distinguer les activités composant un processus. La modélisation fonctionnelle doit également permettre d'établir la hiérarchie des activités, i.e. d'exprimer de possibles décompositions en termes de sous-processus.

Enfin, le modèle fonctionnel doit aussi représenter le flux de données associées aux activités et les interdépendances de données entre les activités (data flow).

2.2.1.2 L'aspect comportemental

L'aspect comportemental est un aspect primordial du workflow puisqu'il correspond à la dynamique du processus. Le comportement s'exprime par la modélisation

d'un contrôle de flux entre les activités. Ce dernier permet d'indiquer la chronologie de l'exécution des activités, leur flux (séquentiel ou parallèle), les points de synchronisation entre activités ou au contraire, les points de disjonction. De plus, le modèle comportemental doit représenter les événements qui permettent de déclencher les activités. Nous soulignons l'importance de ce modèle, qui permet l'exécution du workflow. L'aspect comportemental est également appelé aspect de coordination.

2.2.1.3 L'aspect informationnel (données)

Cet aspect concerne l'ensemble des informations et des données qui sont associées aux activités. Le modèle informationnel, souvent négligé lors de l'implémentation d'un workflow (S. JOOSTEN 1996), décrit en détail les relations qui existent entre les données, leur type et leur structure.

2.2.1.4 L'aspect organisationnel

Comme son nom l'indique, la partie organisationnelle concerne la description de l'organisation des acteurs de l'entreprise. Le modèle organisationnel peut soit refléter fidèlement l'organigramme de l'entreprise, c'est à dire la décomposition hiérarchique de celle-ci en départements et services soit décrire des unités organisationnelles dans lesquelles on identifie des acteurs. Selon la méthode choisie, la description est plus ou moins détaillée et permet d'établir des liens hiérarchiques entre les acteurs ainsi que des relations entre unités organisationnelles ou départements. Toutefois, quelle que soit la méthode retenue, la description des rôles associés aux différentes activités reste invariante. Les rôles créent l'interface entre le modèle organisationnel et les modèles représentant les activités.

2.2.2 Techniques et outils de modélisation de workflow

Associés aux aspects à modéliser définis précédemment, un certain nombre d'outils de modélisation peuvent être employés pour décrire le comportement des flux de travail.

2.2.2.1 Réseaux de Petri et workflow

Les Réseaux de Petri (RdPs) (PETRI.NETS.WORLD 2004) constituent un formalisme majeur pour modéliser les processus workflows. Une des forces des RdPs est la base mathématique forte qu'ils offrent avec une représentation graphique. Dans cette section, nous résumons la projection topographique entre concepts de workflow et de RdP .

Un processus définit les tâches aussi bien que les conditions pour leur exécution. En utilisant les RdPs, un processus est représenté en conversant sa seule entrée (i.e., un nœud du début) dans une place sans arcs entrants, et sa seule sortie (i.e., nœud de la fin) dans une place sans arcs sortants. Les conditions sont représentées par des places, et les tâches par des transitions. Habituellement, un processus spécifié par RdP devrait accomplir deux exigences : (1) il doit être possible d'accéder à tout moment un état pour lequel il y a un jeton dans la place finale, et (2) quand il y a un jeton dans la place finale, tous les autres jetons auraient dû disparaître.

Dans un processus modélisé par un RdP, une transition active correspond à un workitem, et le tir d'une transition à une instance de l'activité. Certains work-items peuvent seulement être transformés dans une instance d'activité une fois ils sont déclenchés. Un déclencheur pourrait correspondre à une initiative du participant, à un événement externe ou à un signal du temps initié par l'environnement. A chaque transition correspondante à une tâche qui exige un déclencheur, une autre place d'entrée est ajoutée. Une occurrence du déclencheur apporte un jeton dans cette place supplémentaire. Le jeton est consommé une fois la transition appropriée est franchie. Un échec pendant l'exécution d'une tâche exige un 'rollback' (i.e., revenir à l'état antérieur au début de l'activité). Quand une activité sera complétée avec succès, des changements deviennent définitifs.

2.2.2.2 UML et workflow

Les diagrammes d'états/transitions sont un autre formalisme majeur pour la modélisation des processus workflows. Ils ont été inventés par Harel (HAREL 1987), et ont été incorporés dans UML (Unified Modelling Language) (*Unified Modeling Language*.), dans une forme légèrement différente. Weissenfels et al. (J. WEISSENFELS 1998) ont investi sur l'usage des diagrammes d'états/transitions pour modéliser les processus workflows (le projet Mentor WfMS). Nous pouvons aussi citer Blake (BLAKE. 2002) qui a présenté une approche systématique pour la modélisation des workflows en utilisant UML (le projet WARP).

Dans le projet Mentor WfMS (J. WEISSENFELS 1998), les activités reflètent la décomposition fonctionnelle d'un système et dénotent les composants actifs d'une spécification ; elles correspondent directement aux activités d'un workflow. Un diagramme d'activités spécifie le flot de données entre les activités, dans la forme d'un graphe orienté avec les éléments de données comme annotations des arcs. Les diagrammes d'états capturent le comportement d'un système en spécifiant le flot de contrôle entre les activités.

Pour le projet WARP (Workflow Automation through Agent-based Reflective Processes) (BLAKE. 2002), l'approche utilisée distingue entre les vues structurelles, fonctionnelles, non fonctionnelles, et opérationnelles. Les vues structurelles informe sur les activités, la définition des rôles, et la composition du workflow. Elles sont représentées dans UML par les diagrammes des classes. Les vues fonctionnelles montrent le flot de données et de contrôle du workflow en utilisant les diagrammes d'activités UML. Les vues non-fonctionnelles (traitement des erreurs, concurrence, atomicité, etc.) utilisent le flot de données et de contrôle et peuvent être modélisées aussi par les diagrammes d'activités. Finalement, les vues opérationnelles sont reliées à l'initiation des instances du workflow, et la coordination pour l'achèvement du workflow. Ces vues peuvent être modélisées par les diagrammes de séquence UML.

2.2.2.3 YAWL

YAWL (Yet Another Workflow Language) [2] est basé sur les RdPs avec des caractéristiques additionnelles pour faciliter la modélisation des workflows complexes. YAWL hérite de la classe des RdPs et s'étend par la modélisation de l'instance multiple, les tâches composites, le retrait des jetons et les transitions connectées directement.

La spécification du workflow dans YAWL est une sorte de réseaux de workflow étendu (EWF-nets) qui forme une hiérarchie. Une tâche est soit atomique soit composite. Chaque tâche composite fait référence à un unique EWF-net dans le niveau le plus bas de la hiérarchie. Les tâches atomiques forment les feuilles de l'arbre. Il y a un seul EWF-net non référencé par une tâche composite. Cet EWF-net est nommée le niveau le plus haut du workflow et forme la racine.

Chaque EWF-net consiste en des tâches (composites ou atomiques) et des conditions qui peuvent être interprétées comme des places. Chaque EWF-net a une unique condition d'entrée et une unique condition de sortie. Contrairement aux RdPs, il est possible de connecter les transitions. Sémantiquement, cette construction peut être interprétée comme une condition ignorée.

Chaque tâche peut avoir des instances multiples. Nous pouvons créer un seuil maximal et un seuil minimal pour le nombre d'instances après l'initiation d'une tâche. En plus, il est possible d'indiquer que la tâche se termine au moment où certaines instances seuils sont achevées

Bibliographie

- BLAKE., M.B. (2002). *An Agent-Based Cross-Organizational Workflow Architecture in Support of Web Services*. Proceedings of the 11th International Workshops on Enabling Technologies :Infrastructure for Collaborative Enterprises (WETICE'02).
- HAREL, D. (1987). "A Visual Formalism for Complex Systems". In : *Science of Computer Programming* 8, p. 231-274.
- J. WEISSENFELS, P. Muth et G. Weikum. (1998). "Flexible Worklist Management in a LightWeight Workflow Management System". In : *Proceedings of the Workshop on Workflow Management Systems at the Sixth International Conference on Extending Database Technology (EDBT'98)*, p. 29-38.
- OMG. *Unified Modeling Language*. URL : www.uml.org.
- PETRI.NETS.WORLD (2004). *Online Services for the International Petri Nets Community*. URL : www.daimi.au.dk/PetriNets.
- S. JOOSTEN, S. Brinkkemper (1996). *Fundamental Concepts for Workflow Automation in Practice*. International Conference on Information Systems Development.