

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

Workflow et Cloud

Thèse présentée par
Zerrouki Djamel

Encadre par : **M.Lahcen aid**

Une thèse présentée pour le diplôme de
Master 2 en Génie logiciel



Département Informatique
Université IBN KHALDOUN

Tiaret

2018-2019

Résumé

.....

Abstrait

.....

Déclaration

.....

Introduction Général

.....

Table des figures

1.1	Prévisions de la taille du marché du cloud computing public (Ried, 2011).	12
1.2	Les services XaaS du cloud computing	14
1.3	Modèles de déploiement du cloud computing	16
1.4	Modèle de référence des systèmes de gestion de workflow (WfMC, 95).	22
1.5	Différents standards adoptés dans les SGWf	22
2.1	Opération de franchissement	28
2.2	Réseau de Petri non borné	29
2.3	Réseau de Petri vivant	30
2.4	Exemple de réseau de Petri bloquant	30

Liste des tableaux

1.1	Comparaison entre types de Workflows.	21
-----	---	----

Table des matières

I	ch1	10
1	Concepts fondamentaux de cloud et de workflow	11
1.1	Introduction	11
1.2	cloud computing	11
1.2.1	Concept du cloud computing	11
1.2.1.1	Vers une définition du cloud computing	12
1.2.1.2	Caractéristiques principales du cloud computing	12
1.2.1.3	Technologies connexes	14
1.2.2	Modèles du cloud computing	14
1.2.2.1	Modèles de service du cloudcomputing	14
1.2.2.2	Modèles de déploiement	15
1.2.3	Cloud computing et sécurité	16
1.2.3.1	Avantages et défis du Cloud en terme de sécurité	16
1.2.3.2	Les composants sécurité d'un système de Cloud computing	17
1.3	Workflow et systèmes de gestion de Workflow	17
1.3.1	Concepts de base et définitions de Workflow	17
1.3.1.1	Définitions de base du workflow	18
1.3.1.2	Concepts et terminologie de workflow	19
1.3.1.3	Les types de workflow	20
1.3.2	Architecture des systèmes de gestion de workflows	21
1.3.2.1	Définition	21
1.3.2.2	Standards utilisés dans les SGWf :	22
1.3.3	Intérêt du cloud pour les workflows	23
1.3.3.1	L'approvisionnement de ressources	23
1.3.3.2	L'allocation dynamique de ressources à la demande	23
1.3.3.3	L'élasticité	23
1.3.3.4	La garantie des QoS via des SLA	24
1.3.3.5	Le faible Coût d'exploitation	24
1.4	Conclusion	24
2	Modélisations et Réseau de Petri	25
2.1	Introduction	25
2.2	Modélisation des processus workflow	25
2.2.1	Aspects à modéliser	25
2.2.1.1	L'aspect fonctionnel	25
2.2.1.2	L'aspect comportemental	25
2.2.1.3	L'aspect informationnel (données)	26

2.2.1.4	L'aspect organisationnel	26
2.2.2	Techniques et outils de modélisation de workflow	26
2.2.2.1	Réseaux de Petri et workflow	26
2.2.2.2	UML et workflow	27
2.2.2.3	YAWL	27
2.3	Les réseaux de Petri	28
2.3.1	Définition	29
2.3.1.1	Propriétés des réseaux de Petri	29
2.3.1.2	Les réseaux de Petri colorés	31
2.4	Modélisation Des Politiques De Sécurité	32

Première partie

ch1

Chapitre 1

Concepts fondamentaux de cloud et de workflow

1.1 Introduction

Le cloud computing, traduit le plus souvent en français par "informatique dans les nuages", "informatique dématérialisée" ou encore "infonuagique", est un domaine qui regroupe un ensemble de techniques et de pratiques consistant à accéder, en libre-service, à du matériel ou à des logiciels informatiques, à travers une infrastructure réseau (Internet). Ce concept rend possible la distribution des ressources informatiques sous forme de services pour lesquels l'utilisateur paie uniquement pour ce qu'il utilise. Ces services peuvent être utilisés pour exécuter des applications scientifiques et commerciales, souvent modélisées sous forme de workflows.

Ce chapitre présente une introduction au cloud computing et au workflow, nécessaire pour la compréhension générale de ce rapport.

Tout d'abord, nous présentons dans la section 1.2 une introduction au paradigme du cloud computing. Nous donnons un aperçu général du cloud computing, y compris sa définition, ses caractéristiques principales et une comparaison avec les technologies connexes. Nous présentons les différents modèles de service, les différents modèles de déploiement, ainsi que les différents acteurs du cloud computing. Nous résumons quelques challenges de recherche en cloud computing. Par la suite, nous présentons, dans la section 1.3, une introduction au workflow et systèmes de gestion de workflow. Nous donnons le concept du workflow, sa définition, et l'architecture de référence d'un système de gestion de workflows et, finalement, nous résumons l'intérêt du cloud pour les workflows.

1.2 cloud computing

1.2.1 Concept du cloud computing

L'idée principale du cloud est apparue dans les années 60, où le professeur John McCarthy avait imaginé que les ressources informatiques seront fournies comme des services d'utilité publique (Garfinkel, 1999). C'est ensuite, vers la fin des années 90, que ce concept a pris de l'importance avec l'avènement du grid computing (Foster, 1999). Le terme cloud est une métaphore exprimant la similarité avec le réseau électrique, dans lequel l'électricité est produite dans de grandes centrales,

puis disséminée à travers un réseau jusqu'aux utilisateurs finaux. Ici, les grandes centrales sont les Datacenter, le réseau est le plus souvent celui d'Internet et l'électricité correspond aux ressources informatiques. Le cloud computing n'est véritablement apparu qu'au cours de l'année 2006 (Vouk, 2008) avec l'apparition d'Amazon EC2 (Elastic Compute cloud). C'est en 2009 que la réelle explosion du cloud survint avec l'arrivée sur le marché de sociétés comme Google (Google App Engine), Microsoft (Microsoft Azure), IBM (IBM Smart Business Service), Sun (Sun cloud) et Canonical Ltd (Ubuntu Enterprise cloud). D'après une étude menée par Forrester (Ried, 2011), le marché du cloud computing s'élevait à environ 5,5 milliards de dollars en 2008, il devrait atteindre plus de 150 milliards d'ici 2020, comme l'illustre la figure 1.1.

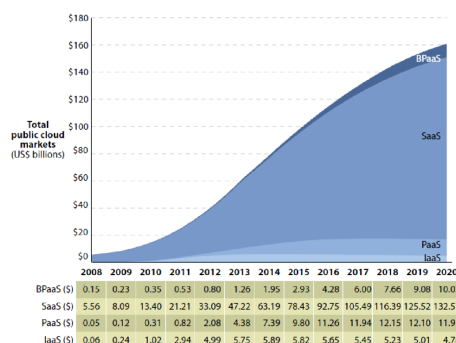


FIGURE 1.1 – Prévisions de la taille du marché du cloud computing public (Ried, 2011).

1.2.1.1 Vers une définition du cloud computing

Beaucoup de chercheurs ont tenté de définir le cloud computing (Geelan, 2008 ; McFedries, 2008 ; Buyya, 2009 ; Armbrust, 2010). La plupart des définitions attribuées à ce concept semblent se concentrer seulement sur certains aspects technologiques. L'absence d'une définition standard a généré non seulement des exagérations du marché, mais aussi des confusions. Pour cette raison, il y a eu récemment des travaux sur la normalisation de la définition du cloud computing, à l'exemple de Vaquero et coll (Vaquero, 2009) qui ont comparé plus de 20 définitions différentes et ont proposé une définition globale. En guise de synthèse des différentes propositions données dans la littérature, nous introduisons une définition mixte, qui correspond aux différents types de cloud considérés dans les travaux réalisés dans cette thèse.

Nous définissons le cloud comme un modèle informatique qui permet d'accéder, d'une façon transparente et à la demande, à un pool de ressources hétérogènes physiques ou virtualisées (serveurs, stockage, applications et services) à travers le réseau. Ces ressources sont délivrées sous forme de services reconfigurables et élastiques, à base d'un modèle de paiement à l'usage, dont les garanties sont offertes par le fournisseur via des contrats de niveau de service (SLA, Service Level Agreement).

1.2.1.2 Caractéristiques principales du cloud computing

Le cloud computing possède les caractéristiques suivantes :

- **Accès en libre-service à la demande.** Le cloud computing offre des ressources et services aux utilisateurs à la demande. Les services sont fournis de façon automatique, sans nécessiter d'interaction humaine (Mell, 2011).
- **Accès réseau universel.** Les services de cloud computing sont facilement accessibles au travers du réseau, par le biais de mécanismes standard, qui permettent une utilisation depuis de multiples types de terminaux (par exemple, les ordinateur portables, tablettes, smartphones) (Mell, 2011).
- **Mutualisation de ressources** (Pooling). Les ressources du cloud peuvent être regroupées pour servir des utilisateurs multiples, pour lesquels des ressources physiques et virtuelles sont automatiquement attribuées (Mell, 2011). En général, les utilisateurs n'ont aucun contrôle ou connaissance sur l'emplacement exact des ressources fournies. Toutefois, ils peuvent imposer de spécifier l'emplacement à un niveau d'abstraction plus haut.
- **Scalabilité et élasticité.** Des ressources supplémentaires peuvent être automatiquement mises à disposition des utilisateurs en cas d'accroissement de la demande (en réponse à l'augmentation des charges des applications) (Geelan, 2008), et peuvent être libérées lorsqu'elles ne sont plus nécessaires. L'utilisateur a l'illusion d'avoir accès à des ressources illimitées à n'importe quel moment, bien que le fournisseur en définisse généralement un seuil (par exemple : 20 instances par zone est le maximum possible pour Amazon EC2).
- **Autonome.** Le cloud computing est un système autonome et géré de façon transparente pour les utilisateurs. Le matériel, le logiciel et les données au sein du cloud peuvent être automatiquement reconfigurés, orchestrés et consolidés en une seule image qui sera fournie à l'utilisateur (Wang, 2008).
- **Paiement à l'usage.** La consommation des ressources dans le cloud s'adapte au plus près aux besoins de l'utilisateur. Le fournisseur est capable de mesurer de façon précise la consommation (en durée et en quantité) des différents services (CPU, stockage, bande passante, . . .) ; cela lui permettra de facturer l'utilisateur selon sa réelle consommation (Armbrust, 2009).
- **Fiabilité et tolérance aux pannes.** Les environnements cloud tirent parti de la redondance intégrée du grand nombre de serveurs qui les composent en permettant des niveaux élevés de disponibilité et de fiabilité pour les applications qui peuvent en bénéficier (Buyya, 2008).
- **Garantie QoS.** Les environnements de cloud peuvent garantir la qualité de service pour les utilisateurs, par exemple, la performance du matériel, comme la bande passante du processeur et la taille de la mémoire (Wang, 2008).
- **Basé-SLA.** Les clouds sont gérés dynamiquement en fonction des contrats d'accord de niveau de service (SLA) (Buyya, 2008) entre le fournisseur et l'utilisateur. Le SLA définit des politiques, telles que les paramètres de livraison, les niveaux de disponibilité, la maintenabilité, la performance, l'exploitation, ou autres attributs du service, comme la facturation, et même des sanctions en cas de violation du contrat. Le SLA permet de rassurer les utilisateurs dans leur idée de déplacer leurs activités vers le cloud, en fournissant des garanties de QoS. Après avoir présenté les caractéristiques essentielles d'un service cloud, nous présentons, brièvement, dans la section suivante, quelques technologies connexes aux clouds.

1.2.1.3 Technologies connexes

1.2.2 Modèles du cloud computing

1.2.2.1 Modèles de service du cloud computing

XaaS (X as a Service) représente la base du paradigme du cloud computing, où X représente un service tel qu'un logiciel, une plateforme, une infrastructure, un Business Process, etc. Nous présentons, dans cette section, quatre modèles de services (Rimal, 2009), à savoir : (1) Logiciel en tant que services SaaS (Software as a Service), où le matériel, l'hébergement, le framework d'application et le logiciel sont dématérialisés, (2) Plateforme en tant que service PaaS (Platform as a Service), où le matériel, l'hébergement et le framework d'application sont dématérialisés, (3) Infrastructure en tant que service IaaS (Infrastructure as a Service) et (4) Matériel en tant que service HaaS (Hardware as a Service), où seul le matériel (serveurs) est dématérialisé dans ces deux derniers cas. La figure 1.2 montre le modèle classique et les différents modèles de service de cloud

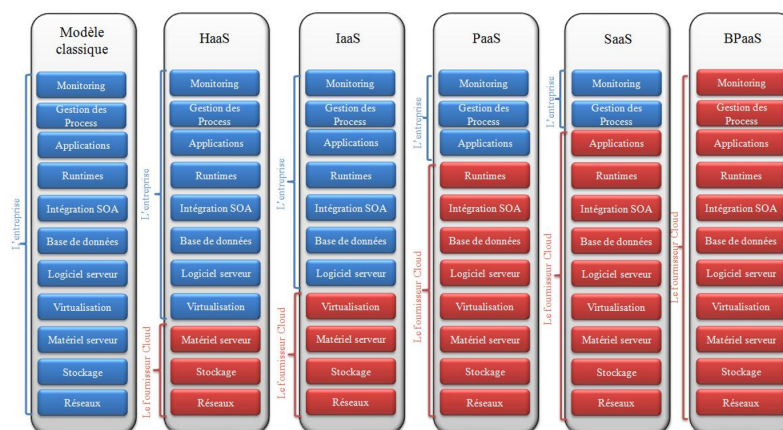


FIGURE 1.2 – Les services XaaS du cloud computing

1. Software as a Service (SaaS) :

Ce modèle de service est caractérisé par l'utilisation d'une application partagée qui fonctionne sur une infrastructure Cloud. L'utilisateur accède à l'application par le réseau au travers de divers types de terminaux (souvent via un navigateur web). L'administrateur de l'application ne gère pas et ne contrôle pas l'infrastructure sous-jacente (réseaux, serveurs, applications, stockage). Il ne contrôle pas les fonctions de l'application à l'exception d'un paramétrage de quelques fonctions utilisateurs limitées. On prend comme exemple les logiciels de messagerie au travers d'un navigateur comme Gmail ou Yahoo mail.

2. Platform as a Service (PaaS) :

L'utilisateur a la possibilité de créer et de déployer sur une infrastructure Cloud PaaS ses propres applications en utilisant les langages et les outils du fournisseur. L'utilisateur ne gère pas ou ne contrôle pas l'infrastructure Cloud sous-jacente (réseaux, serveurs, stockage) mais l'utilisateur contrôle l'application déployée et sa configuration. Comme exemple de PaaS, on peut citer un des plus anciens -IntuitQuickbase- qui permet de déployer ses applications

bases de données en ligne ou -Google Apps Engine (GAE)- pour déployer des services Web.

Dans ces deux cas l'utilisateur de ces services n'a pas à gérer des serveurs ou des systèmes pour déployer ses applications en ligne et dimensionner des ressources adaptées au trafic.

3. **Infrastructure as a Service (IaaS) :**

L'utilisateur loue des moyens de calcul et de stockage, des capacités réseau et d'autres ressources indispensables (partage de charge, pare-feu, cache). L'utilisateur a la possibilité de déployer n'importe quel type de logiciel incluant les systèmes d'exploitation. L'utilisateur ne gère pas ou ne contrôle pas l'infrastructure Cloud sous-jacente mais il a le contrôle sur les systèmes d'exploitation, le stockage et les applications. Il peut aussi choisir les caractéristiques principales des équipements réseau comme le partage de charge, les pare-feu, etc. L'exemple emblématique de ce type de service est Amazon Web Services qui fournit du calcul (EC2), du stockage (S3, EBS), des bases de données en ligne (SimpleDB) et quantité d'autres services de base. Il est maintenant imité par de très nombreux fournisseurs.

4. **Points forts et Points faibles des services cloud :**

1.2.2.2 Modèles de déploiement

Selon la définition du cloud computing donnée par le NIST (Mell, 2011), il existe quatre modèles de déploiement des services de cloud, à savoir : cloud privé, cloud communautaire, cloud public et cloud hybride, comme illustré dans la figure 1.3.

1. **Cloud privé :**

L'ensemble des ressources d'un cloud privé est exclusivement mis à disposition d'une entreprise ou organisation unique. Le cloud privé peut être géré par l'entreprise elle-même (cloud privé interne) ou par une tierce partie (cloud privé externe). Les ressources d'un cloud privé se trouvent généralement dans les locaux de l'entreprise ou bien chez un fournisseur de services. Dans ce dernier cas, l'infrastructure est entièrement dédiée à l'entreprise et y est accessible via un réseau sécurisé (de type VPN). L'utilisation d'un cloud privé permet de garantir, par exemple, que les ressources matérielles allouées ne seront jamais partagées par deux clients différents.

2. **Cloud communautaire :**

L'infrastructure d'un cloud communautaire est partagée par plusieurs organisations indépendantes ayant des intérêts communs. L'infrastructure peut être gérée par les organisations membres ou par un tiers. L'infrastructure peut être située, soit au sein des dites organisations, soit chez un fournisseur de services.

3. **Cloud public :**

L'infrastructure d'un cloud public est accessible à un large public et appartient à un fournisseur de services. Ce dernier facture les utilisateurs selon la consommation et garantit la disponibilité des services via des contrats SLA.

4. **Cloud hybride :**

L'infrastructure d'un cloud hybride est une composition de plusieurs clouds

(privé, communautaire ou public). Les différents clouds composant l'infrastructure restent des entités uniques, mais sont reliés par une technologie standard ou propriétaire permettant ainsi la portabilité des données ou des applications déployées sur les différents clouds.

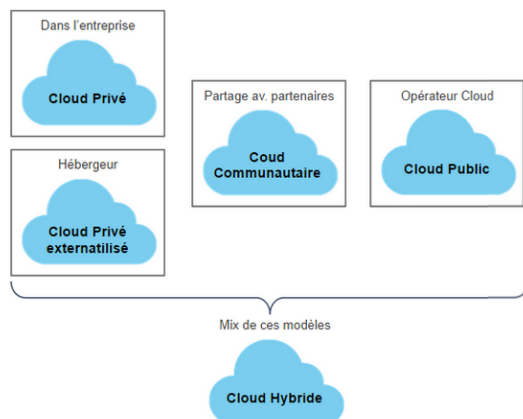


FIGURE 1.3 – Modèles de déploiement du cloud computing

1.2.3 Cloud computing et sécurité

Toutes les enquêtes montrent que la sécurité est la préoccupation majeure des organisations dans le processus d'adoption des technologies Cloud. Les questions sont nombreuses comme par exemple :

- Quelle confiance peut-on avoir dans le stockage des données à l'extérieur de l'entreprise ?
- Quels sont les risques associés à l'utilisation de services partagés ?
- Comment démontrer la conformité des systèmes à des normes d'exploitation ?

Les infrastructures Cloud sont de gigantesques systèmes complexes. Ils peuvent cependant être réduits à un petit nombre de primitives simples qui sont instanciées des milliers de fois et à quelques fonctions communes. La sécurité du Cloud est donc un problème gouvernable moins complexe qu'il n'y paraît.

1.2.3.1 Avantages et défis du Cloud en terme de sécurité

Le Cloud présente des avantages immédiats. D'une manière générale, le fait d'héberger des données publiques sur le Cloud réduit les risques pour les données internes sensibles. D'autre part, l'homogénéité dans la construction du Cloud en rend les tests et les audits plus simples. De même la conduite du système au travers de web services permet la mise en place de procédures automatiques accroissant notablement la sécurité.

En revanche les défis restent nombreux pour les fournisseurs. Il faut donner confiance dans le modèle de sécurité et dans les outils de gestion qui sont proposés. Les tâches de gestion sont réalisées de manière indirecte au travers d'une interface puisque l'utilisateur n'a pas de contrôle direct sur l'infrastructure physique. Ce partage des responsabilités complique un peu les audits de sécurité.

1.2.3.2 Les composants sécurité d'un système de Cloud computing

Les différents composants qui participent à la sécurité d'un système de Cloud computing présentent les caractéristiques suivantes :

- **Service de console de gestion (Provisioning) :**
La mise en route et la reconfiguration des composants des systèmes sont très rapides. Il est possible de mettre en service plusieurs instances dans plusieurs centres de traitement répartis dans le monde en quelques minutes. Les reconfigurations réseau sont facilitées. En revanche, la sécurité d'utilisation de la console de gestion devient impérative (authentification multi-facteurs, connexion chiffrée, etc..)
- **Service de stockage des données :**
Les avantages du stockage des données dans le Cloud dépendent des fournisseurs mais en général, ceux-ci fragmentent et répartissent les données. Celles ci sont aussi souvent recopiées dans des centres de traitement différents. Ces opérations améliorent considérablement la sécurité des données. Si leur contenu doit rester confidentiel, il convient de les chiffrer avant de les stocker.
- **Infrastructures de calcul :**
Un des gros avantages du Cloud pour le développement et l'exploitation des applications réside dans la virtualisation. Elle permet de préparer des configurations maîtres sûres qu'il suffit de dupliquer pour déployer. Les défis restent la sécurisation des données dans les applications partagées et la sécurité entre les instances garantie par les hyperviseurs.
- **Services de support :**
La principale caractéristique du Cloud est la mise en place a priori d'une sécurité renforcée et auditable (authentification, logs, pare-feux, etc..). Il reste à traiter les risques liés à l'intégration avec les applications des utilisateurs ainsi que les processus toujours délicats de mises à jour
- **Sécurité périmétrique du réseau Cloud :**
Ces grandes infrastructures partagées fournissent des moyens de protection au delà des capacités d'une entreprise normale comme par exemple la protection contre les attaques DDOS (Distributed Denial Of Service). Les mécanismes de sécurité périmétriques sont généralement bien conçus (fournisseur d'identité, authentification, pare-feux , etc..). En revanche, il reste à traiter les sujets liés à la mobilité.

1.3 Workflow et systèmes de gestion de Workflow

1.3.1 Concepts de base et définitions de Workflow

La notion de workflow (traduit en français par "flux de travail") est apparue dans l'industrie de l'image électronique et de la gestion de production assistée par ordinateur (GW, 1998). Ce concept a donc été créé dans le but d'automatiser les procédures de travail au sein des organisations. L'idée d'enchaîner différentes tâches pour réaliser un traitement complexe est pertinente. De plus, dans les infrastructures actuelles distribuées, gérant des ressources hétérogènes, telles que le cloud computing, bénéficier d'un environnement autorisant la définition et l'exécution des chaînes de traitement constitue une des fonctionnalités essentielles recherchée, à la fois par

les scientifiques et au-delà par le grand public.

Deux grandes catégories d'usages utilisent la notion de workflow : les protocoles expérimentaux, dans des domaines tels que la biologie, l'astronomie, la physique, la neuroscience, la chimie, etc. (workflows scientifiques) et les chaînes de traitement pratiquées dans des domaines commerciaux, financiers, pharmaceutiques (processus métiers). Elles donnent lieu à plusieurs pistes de recherche diverses, mais cependant connexes. Dans le cadre de cette thèse, nous traitons plus particulièrement les workflows scientifiques.

1.3.1.1 Définitions de base du workflow

Avant de définir le terme workflow, il est à noter qu'un problème de confusion persiste entre les termes : workflow (processus workflow), technologie workflow et système workflow. En ce qui suit, nous allons définir chacun de ces termes.

1. **Définition 1 :** Un **workflow** est la forme exécutable d'un processus d'une organisation, gérable par un système workflow. Il permet d'automatiser l'exécution du processus ou encore sa simulation.
2. **Définition 2 :** Un **système workflow** (ou **WfMS** pour Système de Gestion de Workflow) est un système informatique permettant la gestion des processus métiers. Les services proposés par un WfMS sont au minimum l'exécution d'un processus et sa gestion (contrôle et suivi) en plus de la mise à disposition des outils et des documents nécessaires à la réalisation des différentes étapes du processus.
3. **Définition 3 :** La **technologie workflow** est la technologie informatique du **TCAO** (Travail Coopératif Assisté par Ordinateur), qui s'intéresse à la gestion des processus de l'organisation. C'est l'ensemble des moyens utilisés pour automatiser et gérer un processus. Cette gestion est garantie vu qu'il est possible de présenter un modèle de processus sous une forme exécutable.

La relation entre ces trois termes est la suivante : Une entreprise peut introduire une technologie workflow dans son système d'information en installant un système de gestion de workflow qui gère ses processus, automatisés en workflow.

La WfMC (Workflow Management Coalition) (WfMC, 1999) a donné une définition qui généralise la notion de workflow indépendamment des domaines spécifiques :

"Workflow is the automation of business process, in whole or part during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules."

Nous traduisons cette définition par : " Un workflow est l'automatisation d'un processus métier, en tout ou en partie, au cours de laquelle des documents, des informations ou des tâches sont passées d'un participant à un autre pour l'action, selon un ensemble de règles procédurales".

En ce qui concerne le workflow scientifique, nous retiendrons la définition suivante.

Un workflow est composé d'un ensemble de tâches (traitements) organisées selon un ordre logique, afin de réaliser un traitement global, complexe et pertinent sur un ensemble de données sources. Ces données sont souvent complexes, tant au niveau de leurs structures que de leurs organisations. Elles sont souvent volumineuses.

La taille d'un workflow scientifique peut varier de quelques tâches à des millions de tâches, qui sont souvent de calcul intensif "Computation Intensive" (Ludäscher,

2009). Pour les grands workflows, il est souhaitable de répartir les tâches entre plusieurs ressources, afin d'optimiser les temps d'exécution. En tant que tel, les workflows impliquent souvent des calculs répartis sur des clusters, des grilles, et d'autres infrastructures informatiques. Récemment, les clouds computing sont évalués comme une plateforme d'exécution de workflows (Hoffa, 2008 ; Juve, 2008). L'exécution d'un workflow est gérée par un SGWf (Système de Gestion de Workflow) dont l'architecture de référence est décrite brièvement dans la section suivante. 1

1.3.1.2 Concepts et terminologie de workflow

La liste suivante présente les concepts de base de workflow et les structures de base pour la conception de workflow et le contrôle de processus comme le suggère la WfMC (WfMC 1999) :

- Une **activité** (tâche) : est une description d'une partie du travail qui constitue une étape logique dans un workflow. Elle peut être manuelle ou automatique [1]. Une activité manuelle est entièrement réalisée par une ou plusieurs personnes, sans aucune utilisation d'une application. En revanche, une activité automatique est effectuée par une application, sans aucune intervention des personnes, en se basant sur des données déjà enregistrées. Les activités sont classées en fonction des mutuelles dépendances imposées par des aspects structurels et de données (flot de contrôle et flot de données entre les activités). Différentes configurations permettent de couvrir les aspects structurels : séquence, sélection, itération, et concurrence. Pour la représentation des données, deux approches sont les plus utilisées : soit par le biais des flux de données entre les activités, soit par l'intermédiaire des services de fourniture des données des (ou vers les) activités .
- Une **instance** (instance de workflow (un cas) ou instance d'activité) : est la représentation d'une exécution unique d'un workflow ou d'une activité dans un workflow.
- Un **participant** (acteur, agent, utilisateur, entité de traitement, ressource) : est une entité qui exécute une instance d'activité. Cette entité peut être un être humain ou un système logiciel.
- Un **élément de travail** (work-item) : est la représentation du travail à traiter (par un participant) dans le cadre d'une activité d'une instance de workflow. Une liste des éléments de travail associée avec un participant de workflow donné (ou groupe de participants) est appelé une liste de travail (work-list).
- Un **état de workflow** (resp. d'activité) : est lié à des conditions internes définissant l'état d'une instance du workflow (resp. de l'activité) à un moment donné. Dans le cas d'un workflow, l'état pourrait être "initié", "en exécution", "actif", "suspendu", "achevé", "terminé" et "archivé". Dans le cas d'une activité, il pourrait être "inactive", "active", "en exécution", "suspendue", "sautée" et "terminée".

En résumé, nous distinguons dans un workflow des cas, des éléments de travail (workitems) et des ressources. Les work-items lient les cas et les tâches, les activités lient les cas, les tâches et les ressources. La figure [7] montre qu'un workflow comporte trois dimensions : (1) la dimension de cas, (2) la dimension du processus et (3) la dimension des ressources. La dimension de cas signifie le fait que tous les cas sont traités individuellement. Du point de vue workflow, les cas ne s'influencent pas des autres, mais ils s'influencent les uns des autres indirectement via le partage

des ressources et des données. Dans la dimension de processus, il est spécifié le processus de workflow, c'est à dire les tâches et l'acheminement de ces tâches. Dans la dimension des ressources, les ressources sont regroupées dans des classes particulières nommées les rôles et les unités organisationnelles. Une classe de ressource est un ensemble de ressources présentant des caractéristiques similaires. Si une classe de ressource est basée sur les capacités (exigences fonctionnelles) de ses membres, elle est appelée un rôle. Si le classement est basé sur la structure de l'organisation, une classe de ressource est appelée une unité organisationnelle (par exemple une équipe ou un département).

1.3.1.3 Les types de workflow

Regroupement des workflows : Une première manière de regrouper les workflows est la suivante :

1. les workflows documentaires (rédaction, validation, traduction de documents),
2. les workflows métiers, représentant un processus transverse à l'entreprise (concerne plusieurs entités organisationnelles). Il s'agit le plus souvent de procédures rattachées à une division opérationnelle de l'entreprise. Cette procédure peut être qualifiée de procédure métier, processus métier, procédure opérationnelle ou Business Process. Cela désigne un ensemble d'activités qui s'enchaînent de manière chronologique pour atteindre un objectif, généralement délivrer un produit ou un service, dans le contexte d'une organisation de travail (ex : une entreprise, administration, etc.).

Ainsi, on qualifie de Business Process (processus métier) l'ensemble des activités et de procédures qui permettent collectivement la réalisation d'un objectif métier. Le workflow correspond à l'automatisation de ce Business Process.

On peut aussi regrouper les workflows selon les catégories suivantes :

1. **le WorkFlow de production :** qui correspond à la gestion des processus de base de l'entreprise. Les procédures supportent peu de changements dans le temps, et les transactions sont répétitives (ex : contrats d'assurance, gestion de litiges, de réclamations clients, etc.),
2. **le WorkFlow Administratif :** gestion des procédures administratives par circulation de documents électroniques (ordres de mission, demandes de formations, ...) , automatise des processus variables (plusieurs cas) et mais bien définis (bien structurés).
3. **le WorkFlow Ad-Hoc :** généralement traitement de procédures secondaires (procédure d'exception) et dont la structuration est faible (circulation de notes d'information) et pour lesquelles il n'est pas toujours possible de définir des règles à l'avance, orientés vers le travail coopératif en groupe où l'initiative individuelle importante et problèmes à résoudre au cas par cas et en mode interactif.
4. **le WorkFlow collaboratifs :** Gestion d'un travail de groupe dont le processus est complexe, et souvent créé pour un travail de groupe particulier (travail collaboratif, équipe virtuelle,...) .

Comparaison entre types de workflows : 1.1

Critères	De production	Administratif	Ad-hoc	Collaboratif
Capacité de traitement	Haute capacité de traitement Temps de réponse rapide. Le but est la Productivité	Capacité de traitement inférieure (10 à 100) fois moins que pour un workflow de production	Facilite d'utilisation et d'apprentissage sont très importantes.	Capacité de changer dynamiquement la définition d'un processus est essentielle
Utilisation	Employés travaillant à plein temps sur des activités courtes.	Un grand nombre d'employés peuvent être impliqués	La modification dynamique et rapide des processus est essentielle.	Fournir une voie structurée pour travailler ensemble
Nature des processus	Processus formels avec peu de variation Les processus peuvent être très complexes.	Une variété de processus peut exister dans même système. Les processus peuvent être bien définis, mais requièrent moins d'exigence.	Facilité de déploiement.	Les processus sont moins rigides
Spécificités	Requiert une intégration serrée avec les systèmes de bases.	Utilise souvent des documents attaches.	Le but est de zéro coût d'administration .	La capacité de traitement est de moindre importance

TABLE 1.1 – Comparaison entre types de Workflows.

Exemples de workflows :

- Processus de déclaration de sinistre,
- Processus d'ouverture compte,
- Processus de création d'un dossier de prêt,
- Processus de gestion d'une succession,
- Processus de prise de congés.

1.3.2 Architecture des systèmes de gestion de workflows

1.3.2.1 Définition

La gestion du workflow est une technologie en évolution rapide, qui est de plus en plus exploitée par les entreprises. Un SGWf représente un système qui définit, implémente et gère l'exécution de workflows à l'aide d'un environnement logiciel fonctionnant avec un ou plusieurs moteurs de workflows et capable d'interpréter la définition d'un processus, de gérer la coordination des participants et d'invoquer des applications externes.

L'architecture de référence d'un SGWf proposée par la Workflow Management Coalition (WfMC, 95) en 1995 est présentée dans la figure 1.4. Ce modèle inclut un

service de déploiement, qui contrôle l'exécution des workflows et qui supporte cinq interfaces standardisées :

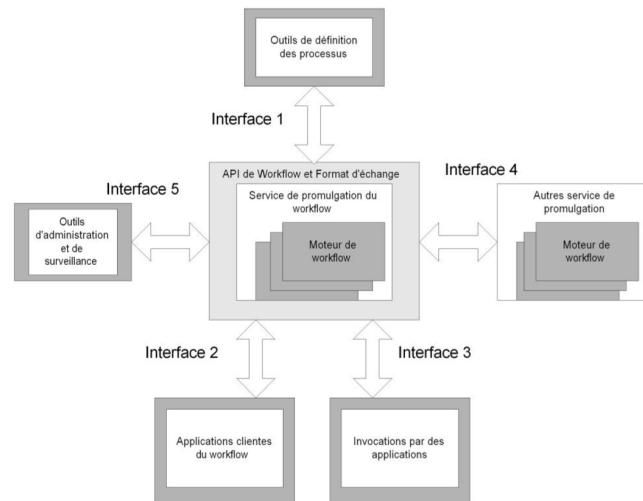


FIGURE 1.4 – Modèle de référence des systèmes de gestion de workflow (WfMC, 95).

1. Interface 1 : Correspond à l'échange des modèles entre les moteurs de workflows et les différents outils de modélisation de processus.
2. Interface 2 : Permet à des applications clientes de communiquer avec le moteur de workflows.
3. Interface3 : Permet au système de workflow d'invoquer des applications clientes.
4. Interface 4 : Permet l'interopérabilité entre les différents moteurs de workflows.
5. Interface 5 : Correspond à l'interaction entre les applications d'administration et de pilotage et le moteur de workflows

1.3.2.2 Standards utilisés dans les SGWf :

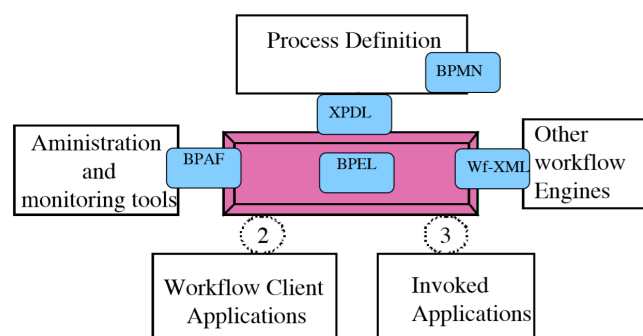


FIGURE 1.5 – Différents standards adoptés dans les SGWf

1. **BPNM (Business Process Model and Notation)** : est une représentation graphique permettant de spécifier les processus métier maintenus par le groupe de gestion d'objets (OMG).

2. **XPDL (Process Definition Language)** : est un format normalisé par la WfMC (Workflow Management Coalition) pour l'échange de définitions de partenaire entre différents produits de flux de travail, c.-à-d. entre différents outils de modélisation et suites de gestion. XPDL définit un schéma XML pour spécifier la partie déclarative du workflow / partenaire.
3. **BPAF (Business Process Analytics)** : fournit aux participants aux processus et aux décideurs des informations sur l'efficacité des processus organisationnels.
4. **BPEL (Business Process Execution Language)** : BPEL est un langage d'orchestre.
5. **Wf-XML** est un standard BPM développé par la Workflow Management Coalition. Wf-XML offre à un moteur BPM un moyen standard d'appeler un processus dans un autre moteur BPM et d'attendre qu'il se termine.

1.3.3 Intérêt du cloud pour les workflows

Les clouds offrent plusieurs avantages pour les applications à base de workflows. Ces avantages facilitent :

1.3.3.1 L'approvisionnement de ressources

Dans les grilles, l'ordonnancement est basé sur un modèle en best-effort, dans lequel l'utilisateur spécifie la quantité de temps nécessaire et délègue la responsabilité de l'allocation des ressources et d'ordonnancement de tâches à un ordonnanceur fonctionnant en mode batch utilisant des files d'attentes. Dans le cloud, au lieu de déléguer l'allocation au gestionnaire de ressources, l'utilisateur peut provisionner les ressources nécessaires et ordonnancer les tâches en utilisant un ordonnanceur contrôlé par l'utilisateur. Ce modèle d'approvisionnement est idéal pour les workflows, car il permet au système de gestion de workflow d'allouer une ressource une seule fois et de l'utiliser pour exécuter de nombreuses tâches.

1.3.3.2 L'allocation dynamique de ressources à la demande

Contrairement aux grilles, les clouds donnent l'illusion que les ressources informatiques disponibles sont illimitées. Cela signifie que les utilisateurs peuvent demander, et s'attendre à obtenir des ressources suffisantes pour leurs besoins, à tout moment. L'approvisionnement à la demande est idéal pour les workflows et d'autres applications faiblement couplées, car il réduit le surcoût (overheads) d'ordonnancement total et peut améliorer considérablement les performances du workflow (Singh, 2005 ; Juve, 2008)

1.3.3.3 L'élasticité

Outre l'approvisionnement des ressources à la demande, les clouds permettent aussi aux utilisateurs de libérer des ressources à la demande. La nature élastique de clouds facilite le changement des quantités et des caractéristiques de ressources lors de l'exécution, permettant ainsi d'augmenter le nombre de ressources, quand il y a un grand besoin, et d'en diminuer, lorsque la demande est faible. Cela permet aux

systèmes de gestion de workflow de répondre facilement aux exigences de qualité de service (QoS) des applications, contrairement à l'approche traditionnelle, qui nécessite de réserver à l'avance des ressources dans les environnements de grilles.

1.3.3.4 La garantie des QoS via des SLA

Avec l'arrivée des services de cloud computing provenant de grandes organisations commerciales, les accords de niveau de service (SLA) ont été une préoccupation importante pour les fournisseurs et les utilisateurs. En raison de compétitions entre les fournisseurs de services émergents, un grand soin est pris lors de la conception du SLA qui vise à offrir (i) de meilleures garanties de QoS aux utilisateurs, et (ii) des termes clairs pour l'indemnisation, en cas de violation du contrat. Cela permet aux systèmes de gestion de workflow de fournir de meilleures garanties de bout en bout en "mappant" les utilisateurs aux fournisseurs de services selon les caractéristiques des SLA.

1.3.3.5 Le faible Coût d'exploitation

Économiquement motivés, les fournisseurs de cloud commercial s'efforcent d'offrir de meilleures garanties de services par rapport aux fournisseurs de grille. Les fournisseurs de cloud profitent également des économies d'échelle, en fournissant des ressources de calcul, de stockage et de bande passante, à un coût très faible grâce, à la virtualisation. Ainsi l'utilisation des services de cloud public pourrait être économique et une alternative moins coûteuse, par rapport à l'utilisation de ressources dédiées, qui sont plus chères. Un des avantages de l'utilisation des ressources virtuelles pour l'exécution de workflow, plutôt que d'un accès direct à la machine physique, est le besoin réduit pour sécuriser les ressources physiques des codes malveillants. Cependant, l'effet à long terme de l'utilisation de ressources virtuelles dans les clouds qui partagent efficacement une "tranche" de la machine physique, plutôt que d'utiliser des ressources dédiées pour les workflows de calculs intensifs, est une question de recherche intéressante.

1.4 Conclusion

....lk

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 ou certaines instances seuils sont achevées

2.3 Les réseaux de Petri

Les réseaux de Petri(Petri,1962)sont une généralisation des automates a états. Ils offrent un contexte général pour modéliser la concurrence et la synchronisation dans les systèmes distribués. Un réseau de Petri est un graphe biparti alterné qui possède deux types de nœuds : les places(cercles)et les transitions(rectangles). Des arcs(flèches) relient les places aux transitions(figure 2.1). L'état du système,nommé marquage,est défini par la répartition des jetons dans les places. Une transition est franchissable sous certaines conditions, notamment lorsqu'il y a suffisamment de jetons dans ses places d'entrée. Le franchissement d'une transition se traduit par une modification du marquage consistant en la consommation des jetons indispensables au franchissement de la transition et la création éventuelle de nouveaux jetons dans les places en sortie de la transition.

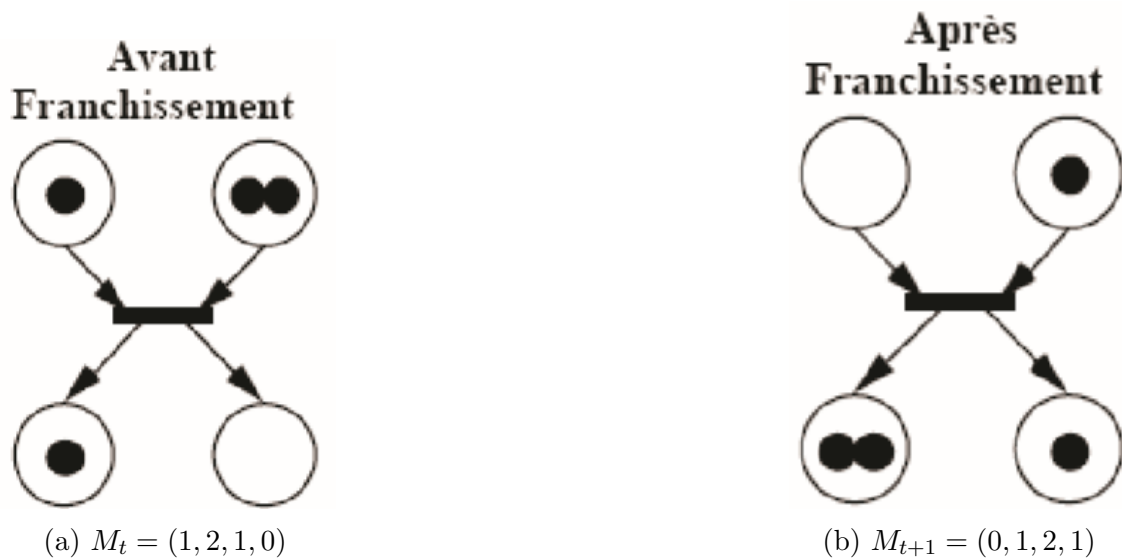


FIGURE 2.1 – Opération de franchissement

2.3.1 Définition

Définition 2.3.1. (Réseau de Petri simple) :

Un réseau de Petri simple est un tuple $(P, T, Pre, Post, M_0)$:

- P est un ensemble fini de places,
- T est un ensemble fini de transition ($P \cap T = \emptyset$),
- $Pre : P \times T \rightarrow \mathbb{N}$ est la fonction d'incidence avant,
- $Post : P \times T \rightarrow \mathbb{N}$ est la fonction d'incidence arrière,
- M_0 et le marquage initial $M_0 : P \rightarrow \mathbb{N}$.

Un réseau de Petri peut-être vu comme un système de transitions dont les états sont les marquages du réseau et les transitions entre états correspondent au franchissement des transitions du réseau.

2.3.1.1 Propriétés des réseaux de Petri

il existe un certain nombre de propriétés qui ont été définies pour les réseaux de Petri, à savoir, le caractère borné, la réinitialisation la vivacité, la conservation, la terminaison (Diaz, 2001). Certaines de ces propriétés sont dites propriétés dynamiques car elles dépendent du marquage initial et sont liées à l'évolution du réseau, alors que d'autres sont dites propriétés statiques du fait qu'elles soient liées à la typologie du réseau et indépendantes du marquage initial.

Définition 2.3.2. (Réseau de Petri borné) :

Une place P_i est bornée pour un marquage initial M_0 si pour tout marquage accessible à partir de M_0 , le nombre de jetons dans P_i reste borné. Elle est dite k -bornée si le nombre de jetons dans P_i est toujours inférieur ou égal à k . Un RdP est (k) borné si toutes les places sont (k) bornées.

Exemple 2.3.1. Un RdP peut ne pas être borné. Sur l'exemple représenté à la (figure 2.2), la transition T_1 admet la place P_1 comme unique place d'entrée. La place P_1 a un jeton : la transition T_1

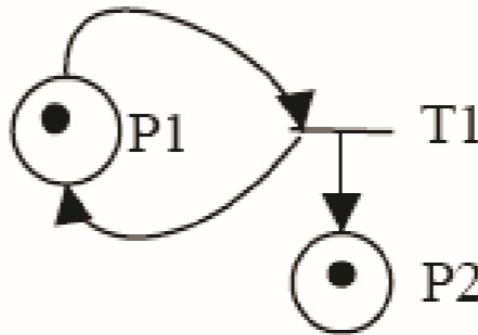


FIGURE 2.2 – Réseau de Petri non borné

est franchissable. Comme P_1 est aussi place de sortie de T_1 , le franchissement de T_1 ne change pas le marquage de P_1 . La transition T_1 est donc franchissable en permanence et peut donc être franchie un nombre de fois infini. Chaque franchissement de T_1 ajoute un jeton dans P_2 dont le marquage va donc tendre vers l'infini.

Définition 2.3.3. Réseau de Petri sauf :

Un RdP est sauf ou binaire pour un marquage initial M_0 s'il est un borné.

Définition 2.3.4. La vivacité :

Une transition T_j est vivante pour un marquage initial M_0 si pour tout marquage accessible M_k , il existe une séquence de franchissement S à partir de M_k contenant T_j : $M_k \in^* M_0, \exists S, M_k | S >$ et $S = \dots T_j \dots$

Si une transition T_j est vivante alors, à tout instant, on sait que T_j peut être franchie dans le futur. Dans le cas d'un réseau de Petri modélisant un système fonctionnant en permanence, si une transition n'est pas vivante et si une fonction du système est associée au franchissement de cette transition, ce la veut dire qu'à partir d'un certain instant, cette fonction ne sera plus disponible dans le futur, ce qui peut traduire une erreur ou une panne.

Définition 2.3.5. Blocage :

Un blocage (ou état puits) est un marquage pour lequel aucune transition n'est validée.

Un réseau de Petri est dit sans blocage pour un marquage initial M_0 si aucun marquage accessible n'est un blocage.

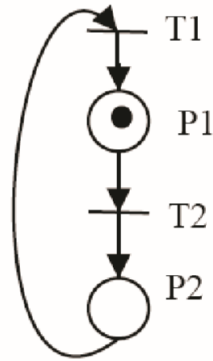


FIGURE 2.3 – Réseau de Petri vivant

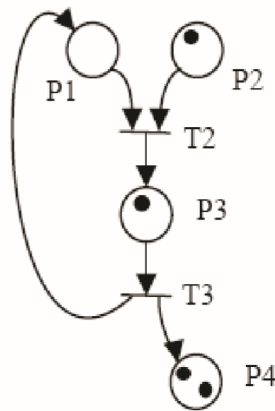


FIGURE 2.4 – Exemple de réseau de Petri bloquant

Le réseau de Petri de la (Figure 3.7), par exemple, a pour blocage le marquage : $M = (1,0,0,1)$.

Définition 2.3.6. (états d'accueil et Réseau de Petri réinitialisable)

Un RdP a un état d'accueil M_a pour un marquage initial M_0 si pour tout marquage accessible M_k à partir de M_0 , il existe une séquence de franchissement permettant d'atteindre le marquage $M_a : \forall M_k \in^* M_0, \exists S_j, M_k | S_j > M_a$

Un RdP est réinitialisable pour un marquage initial M_0 si M_0 est un état d'accueil.

Si un réseau de Petri présente un état d'accueil, il est facile de vérifier s'il est sans blocage et d'étudier sa vivacité.

2.3.1.2 Les réseaux de Petri colorés

Les réseaux colorés (Jensen, 1997) ont été introduits afin de modéliser des systèmes complexes tout en gardant les possibilités de vérification. Lorsque le nombre d'entités du système à modéliser est important, la taille du réseau de Petri devient rapidement énorme, et si les entités présentent des comportements similaires, l'usage des réseaux colorés permet de condenser le modèle. En effet, une couleur est une information attachée à un jeton. Cette information permet de distinguer des jetons entre eux et peut être de type quelconque. Par conséquent, une place peut contenir des jetons de différentes couleurs et une transition peut être franchie de différentes manières, selon la couleur. Ceci est réalisé en attachant un domaine de couleur à chaque place et à chaque transition. Ainsi, les arcs ne sont pas seulement étiquetés par le nombre de jetons mais aussi par leurs couleurs. Le franchissement d'une transition est alors conditionné par la présence dans les places en entrée du nombre de jetons nécessaires, qui en plus satisfont les couleurs qui étiquettent les arcs. Après le franchissement d'une transition, les jetons qui étiquettent les arcs d'entrée sont retirés des places en entrée tandis que ceux qui étiquettent les arcs de sortie sont ajoutés aux places en sortie de cette transition.

Ainsi, pour un même système, le nombre de comportements qui peuvent être exprimés par un réseau coloré est nettement plus élevé qu'avec un réseau simple. Ce sont des réseaux très adaptés aux architectures distribuées. D'autant plus qu'à tout réseau coloré correspond un réseau de Petri simple qui lui est isomorphe. Ce ci permet donc d'exploiter les mêmes techniques d'analyse que celles développées pour les réseaux simples en plus d'autres qui ont été complétées et adaptées aux réseaux colorés telle que le support de la hiérarchisation.

Définition 2.3.7. Réseau de Petri Colorés :

Un réseau de Petri Coloré (CPN) et un tuple $(\Delta, P, T, Arc, Noeud, Couleur, Grade, E, M_0)$ où :

- Δ : est un ensemble de domaines de couleurs (chaque domaine est un ensemble fini et non vide).
- Arc : est un ensemble fini d'arcs tel que $P \cap Arc = T \cap Arc = \emptyset$
- $Noeud$: est la fonction $Noeud, Noeud : Arc \longrightarrow P \times T \cup T \times P$.
- $Couleur$: $P \longrightarrow PowerSet(\Delta)$. $Couleur(p)$ est la fonction couleur qui associe à chaque place un domaine de couleur.
- $Garde$: est une garde, qui fait correspondre à chaque transition une expression booléenne. Les variables de la garde appartiennent à Δ .

- E : est l'application qui associe à chaque arc, un élément de $\text{Couleur}(p)\text{MS}$ où p est une place appartenant à l'arc. E indique le nombre de jetons colorés à recevoir de la place qui se trouve en entrée de la transition, et le nombre de ceux à produire dans la place qui se trouve en sortie.
- M_0 : est l'application qui associe à chaque place p , un élément de $\text{Couleur}(p)\text{MS}$. $M_0(p)$ indique la distribution initiale des jetons colorés dans la place p .

De manière générale, un marquage M d'un réseau coloré est une application qui associe à chaque place p , un élément de $\text{Couleur}(p)\text{MS}$. $M(p)$ est un multi-ensemble sur $\text{Couleur}(p)$ qui indique les marques colorées présentes dans la place p au marquage M . L'état du modèle est défini par un marquage coloré.

2.4 Modélisation Des Politiques De Sécurité

Durant les dernières années, l'utilisation des réseaux de Petri s'est répandue grâce au nombre de travaux qui a été développé pour enrichir les réseaux de Petri ainsi que la disponibilité des outils. Selon de nombreux chercheurs (Osborn, 2002), les réseaux de Petri sont le seul formalisme qui permet de modéliser la structure du système et de faire des analyses qualitative et quantitative. Les réseaux de Petri ont été utilisés pour la vérification de la sécurité (Ahmed and Tripathi, 2003), pour la spécification des autorisations dans les workflows (ATLURI et HUANG 1996), l'analyse des politiques de sécurité y compris les politiques de contrôle d'accès discrétionnaires (Kumaretal., 2002; Knorr, 2000), mandataires (Knorr, 2001; Varadharajan, 1991; Juopperi, 1995; Juszczyszyn, 2003; Jiangetal., 2004; Rakkay and Boucheneb, 2006; Zhangetal., 2006b; Zhangetal., 2006a) et à base de règles (Koch et al., 2002; Rakkay and Boucheneb, 2007; Shafiq et al., 2005).

Bibliographie

- ATLURI, V. et HUANG (1996). “An authorization model for workflows :Proceedings of the 4th European Symposium on Research in Computer Security”. In : *Springer-Verlag*. P. 44-64.
- 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.
- WFMC (1999). *Workflow management coalition terminology and glossary*.