

# 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 matières

<b>1</b>	<b>Concepts fondamentaux de cloud et de workflow</b>	<b>11</b>
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 . . . . .	13
1.2.1.3	Technologies connexes . . . . .	14
1.2.2	Modèles du cloud computing . . . . .	14
1.2.2.1	Modèles de service du cloud computing . . . . .	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
1.2.4	Les principaux grands fournisseurs cloud pour 2019 . . . . .	17
1.2.4.1	Amazon Web Services . . . . .	17
1.2.4.2	Microsoft Azure . . . . .	18
1.2.4.3	Google Cloud Platform . . . . .	18
1.2.4.4	IBM . . . . .	18
1.3	Workflow . . . . .	18
1.3.1	Introduction au Workflow . . . . .	18
1.3.2	Origines . . . . .	18
1.3.3	Définitions et terminologies . . . . .	19
1.3.4	Définitions de base du Workflow . . . . .	20
1.3.4.1	Définition d'un Workflow . . . . .	20
1.3.4.2	Méta Modèle basique . . . . .	20
1.3.4.3	Dimensions . . . . .	21
1.3.5	Description des Systèmes Workflow . . . . .	22
1.3.5.1	Build time . . . . .	22
1.3.5.2	Run time . . . . .	22
1.3.5.3	Utilisateurs, outils et applications . . . . .	23
1.3.6	Concepts et Terminologie Workflow fondamentaux . . . . .	23
1.3.6.1	Procédure Workflow (Workflow Process) . . . . .	23
1.3.6.2	Activité (Process Activity) . . . . .	23
1.3.6.3	Acteur, Ressource (Workflow Participant) . . . . .	23
1.3.6.4	Rôle (Role) . . . . .	24
1.3.6.5	Données (Workflow Relevant Data) . . . . .	24
1.3.6.6	Application externe (Invoked Application) . . . . .	24

1.4	Classification des systèmes Workflow . . . . .	25
1.4.1	Processus collaboratifs . . . . .	25
1.4.2	Workflow administratif . . . . .	26
1.4.3	Workflow de production . . . . .	26
1.4.4	Workflow adaptable ou Workflow ad hoc . . . . .	27
1.4.4.1	Le changement individuel (ad hoc) . . . . .	27
1.4.4.2	Le changement structurel (évolution) . . . . .	28
1.4.5	Modèle de référence des systèmes Workflow . . . . .	29
1.4.5.1	Interface avec les Outils de définition de procédures . . . . .	29
1.4.5.2	Interface avec les applications clientes Workflow . . . . .	29
1.4.5.3	Interface avec les applications invoquées . . . . .	30
1.4.5.4	Interface avec les autres Workflow . . . . .	30
1.4.5.5	Interface avec les outils de contrôle et d'administra- tion . . . . .	30
1.5	Architecture des systèmes de gestion de workflows . . . . .	31
1.5.1	Définition . . . . .	31
1.5.2	Standards utilisés dans les SGWf : . . . . .	31
1.5.2.1	BPNM (Business Process Model and Notation) . . . . .	31
1.5.2.2	XPDL (Process Definition Language) : . . . . .	32
1.5.2.3	BPAF (Business Process Analytics) . . . . .	32
1.5.2.4	BPEL (Business Process Execution Language) . . . . .	32
1.5.2.5	Wf-XML . . . . .	32
1.6	Intérêt du cloud pour les workflows . . . . .	32
1.6.1	L'approvisionnement de ressources . . . . .	32
1.6.2	L'allocation dynamique de ressources à la demande . . . . .	32
1.6.3	L'élasticité . . . . .	33
1.6.4	La garantie des QoS via des SLA . . . . .	33
1.6.5	Le faible Coût d'exploitation . . . . .	33
1.7	Conclusion . . . . .	33
<b>2</b>	<b>Modélisations et Réseau de Petri</b>	<b>34</b>
2.1	Introduction . . . . .	34
2.2	Modélisation des processus workflow . . . . .	34
2.2.1	Aspects à modéliser . . . . .	34
2.2.1.1	L'aspect fonctionnel . . . . .	34
2.2.1.2	L'aspect comportemental . . . . .	34
2.2.1.3	L'aspect informationnel (données) . . . . .	35
2.2.1.4	L'aspect organisationnel . . . . .	35
2.2.2	Modélisation de Workflow adaptable . . . . .	35
2.2.2.1	Approche Meta modèle . . . . .	35
2.2.2.2	L'approche de point ouvert Open-point . . . . .	35
2.2.2.3	Approche synthétisée . . . . .	36
2.2.3	Techniques et outils de modélisation de workflow . . . . .	36
2.2.3.1	Réseaux de Petri et workflow . . . . .	37
2.2.3.2	UML et workflow . . . . .	37
2.2.3.3	YAWL . . . . .	38
2.3	Les réseaux de Petri . . . . .	38
2.3.1	Définition . . . . .	38

2.3.1.1	Propriétés des réseaux de Petri . . . . .	39
2.3.1.2	Les réseaux de Petri colorés . . . . .	41
2.4	Modélisation Des Politiques De Sécurité . . . . .	42



# 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	Les systèmes de gestion de Workflow dans une perspective historique	19
1.5	Méta modèle Workflow pour la définition de Processus [WFMC11 99]	21
1.6	Dimensions des systèmes Workflow . . . . .	21
1.7	Différentes classes des systèmes Workflow . . . . .	25
1.8	Modèle de référence des systèmes de gestion de workflow (WfMC, 95).	29
1.9	Différents standards adoptés dans les SGWf . . . . .	31
2.1	Opération de franchissement . . . . .	39
2.2	Réseau de Petri non borné . . . . .	40
2.3	Réseau de Petri vivant . . . . .	40
2.4	Exemple de réseau de Petri bloquant . . . . .	41

# Liste des tableaux

# 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. Nous présentons les différents modèles de service, les différents modèles de déploiement, ainsi la sécurité dans le cloud computing. Nous présentons Les principaux grands fournisseurs cloud. 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. Puis la classification de système de workflow dans la section 1.4 et par la suite, nous présentons, dans la section 1.5 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.

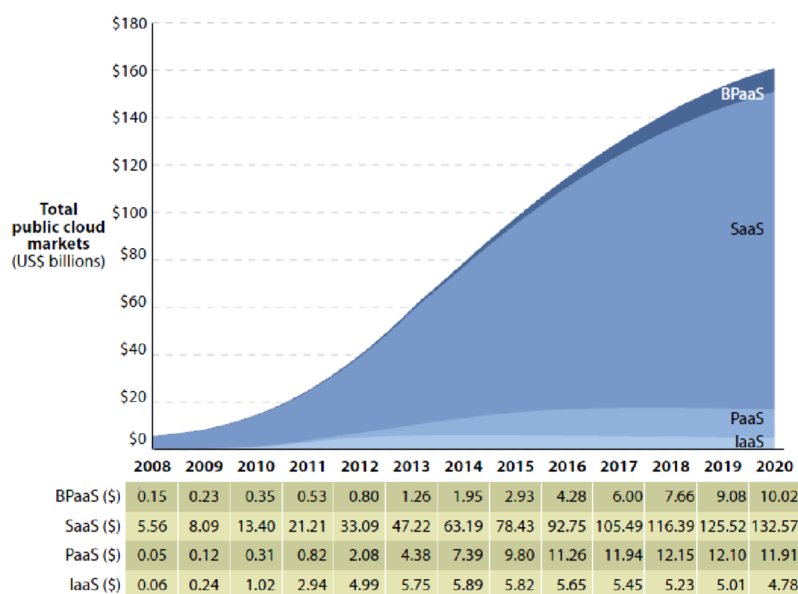


FIGURE 1.1 – Prévisions de la taille du marché du cloud computing public (Ried, 2011).

### 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 ordinateurs 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

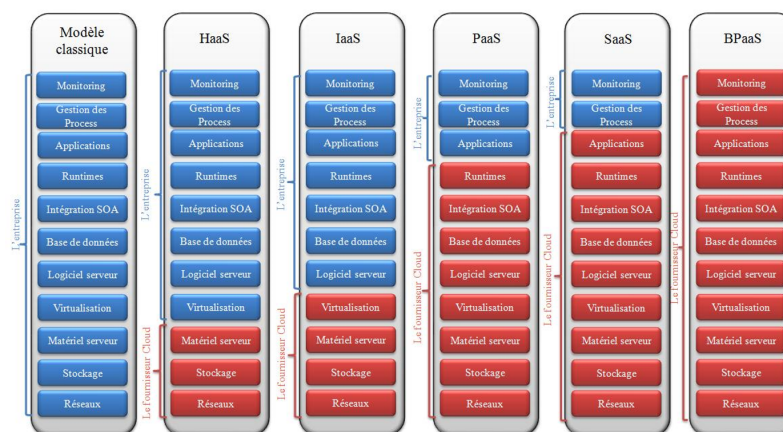


FIGURE 1.2 – Les services XaaS du cloud computing

#### 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 fortset 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.

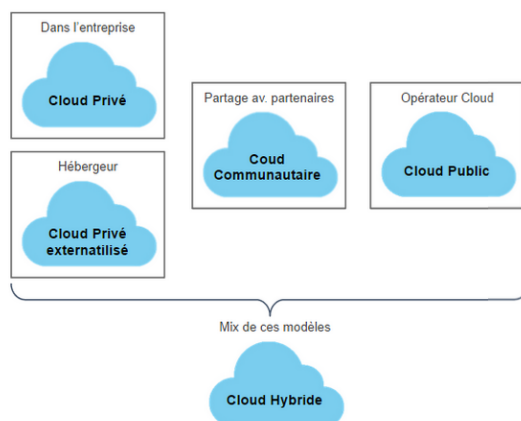


FIGURE 1.3 – Modèles de déploiement du cloud computing

### 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.

### 1.2.4 Les principaux grands fournisseurs cloud pour 2019

Comment choisir la solution de cloud la plus adaptée à votre entreprise, quelles sont les différentes offres et les tarifs pour stocker les données informatiques sur un serveur à distance. Comparatif des principales solutions du marché.

#### 1.2.4.1 Amazon Web Services

Avec un chiffre d'affaires de 25.65 milliards de dollar ([Zdnet.fr](http://Zdnet.fr)) réalisé en 2018, le géant de l'IaaS considère 2019 comme une année d'investissement afin d'accélérer le développement de sa technologie et renforcer sa force de vente. En ce début de 2019, AWS est le leader incontesté dans le domaine de l'IaaS (EPSI 2019).

#### 1.2.4.2 Microsoft Azure

Le service cloud, Microsoft Azure a réalisé un chiffre d'affaires annuel de 11 milliards de dollar. En 2019, le service cloud Microsoft Azure vient en deuxième position derrière AWS (IaaS et PaaS). A la différence d'AWS, Microsoft opère plutôt dans le commercial cloud et propose un service très diversifié grâce à une plateforme familiarisée auprès des utilisateurs (EPSI 2019).

#### 1.2.4.3 Google Cloud Platform

L'année 2018 a été particulièrement fructueuse chez le service cloud de Google. Avec un chiffre d'affaires de près 4 milliards de dollar, l'entreprise a décroché en 2018 les contrats les plus importants dans le secteur. Pour 2019, elle va constituer un véritable contrepoint à AWS et à Microsoft même si l'entreprise a toujours du mal à publier convenablement ses rapports financiers jusqu'à présent (EPSI 2019).

#### 1.2.4.4 IBM

Très présent dans le monde du multcloud et du cloud hybride, il dispose actuellement d'un chiffre d'affaires de 12.2 milliards de dollar. Même s'il reste un peu derrière AWS et Microsoft, il est en train de se focaliser sur le développement d'une plateforme capable de gérer plusieurs outils d'intelligence artificielle pour les principaux fournisseurs cloud. En plus, l'acquisition de Red Hat est un signal fort pour le cloud hybride (EPSI 2019).

### 1.3 Workflow

#### 1.3.1 Introduction au Workflow

Un Workflow est la modélisation et la gestion assistée par ordinateur de l'accomplissement des tâches composant un processus administratif ou industriel, en interaction avec divers acteurs (humains, logiciels, ou matériels) invoqués [COURTOIS 96]. Outil informatique d'origine industrielle, le Workflow est l'adaptation de la GED<sup>24</sup> adjoint de la faculté à gérer l'échange de messages. Le Workflow propose des solutions d'optimisation et de rationalisation des flux d'informations ; que ces informations soient associées à des documents, des procédures ou des messages complétant les systèmes de gestion électronique de documents et d'informations.

A l'heure actuelle plus de 250 Systèmes de Gestion de Workflow (WFMS) sont utilisés ou en développement. Cela signifie que le terme « gestion de Workflow » n'est pas simplement une nouvelle expression à la mode. Ce phénomène de gestion de processus (Workflow) aura certainement un fort impact sur la génération suivante de systèmes informatiques [COURTOIS 96, HAYES 91, KOULOPOULOS 95, SCHAEEL 97].

#### 1.3.2 Origines

Il est intéressant de considérer l'évolution des systèmes informatiques au cours des quatre dernières décennies [VAN DER AALST 02] pour prendre conscience de la

pertinence d'une gestion électronique de processus (Workflow) et apprécier l'impact de la gestion de Workflow dans un avenir proche.

La Figure 1.4 présente le phénomène de gestion de Workflow dans une perspective historique. Cette figure décrit l'architecture d'un système informatique classique en termes de composants. Dans les années soixante, un système informatique était composé d'un certain nombre d'applications autonomes. Pour chacune de ces applications une interface utilisateur et un système de base de données spécifiques étaient développés, chaque application possédait donc ses propres routines pour interagir avec l'utilisateur, stocker et récupérer les données. Dans les années soixante-dix, le développement des systèmes de gestion de base de données (SGBD) a permis d'extraire les données des applications. En utilisant les SGBD, les applications ont ainsi été libérées du fardeau de la gestion de données. Dans les années quatre-vingts, l'apparition de systèmes de gestion d'interface utilisateur « User Interface Management Systems » (UIMS) a permis aux développeurs d'application d'extraire l'interaction avec les utilisateurs des applications. Enfin, les années quatre-vingt-dix sont marquées par l'apparition de logiciels de Workflow, permettant aux développeurs d'application d'extraire les procédures de travail des applications. La Figure 1.4 fait apparaître le système de gestion de Workflow comme une composante générique pour représenter et manipuler les processus d'entreprise<sup>25</sup>.

Ainsi, à l'heure actuelle, beaucoup d'organisations commencent à considérer l'utilité d'outils avancés pour soutenir la conception et l'exécution de leurs processus d'entreprise.

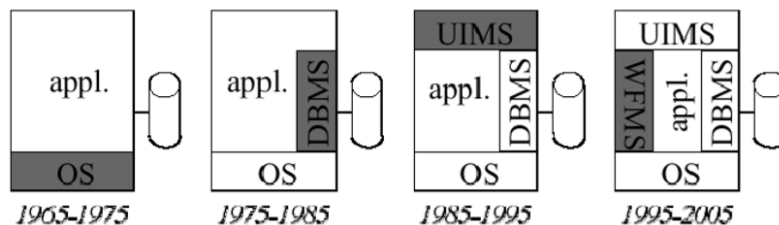


FIGURE 1.4 – Les systèmes de gestion de Workflow dans une perspective historique

### 1.3.3 Définitions et terminologies

Les définitions sont, pour la majorité, issues de la Coalition de Gestion de Workflow « Workflow Management Coalition » (WfMC). La WfMC a été fondée en 1993 par un regroupement d'industriels de l'informatique, de chercheurs et d'utilisateurs, associée à l'essor du développement des Workflows. Cette coalition a pour but de promouvoir les Workflow et d'établir des standards pour les « Workflow Management System » (WfMS). Elle a en particulier publié un glossaire de référence contenant les terminologies employées dans ce domaine [WFMC03 95; WFMC11 99]. Ces standards servent notamment à résoudre les problèmes d'interopérabilité entre systèmes Workflow mais également à définir les caractéristiques fondamentales de ces systèmes. Les documents publiés par la WfMC, qui couvrent plusieurs aspects, peuvent être considérés comme des références en la matière.

### 1.3.4 Définitions de base du Workflow

Le sens du mot Workflow peut varier en fonction du contexte. Pour plus de clarté, les définitions les plus communément admises sur les concepts et les termes du Workflow sont rappelées ci dessous. Ces définitions sont principalement issues du « Workflow Management Coalition Terminology and Glossary » WfMC-TC-1011 [WfMC11 99], dont il existe une traduction à usage francophone [WfMC03f 98]. L'idée première du Workflow est donc de séparer les processus, les ressources et les applications, afin de se recentrer sur la logistique des processus travail et non pas sur le contenu des tâches individuelles. Un Workflow est donc le lien entre ces trois domaines comme précise la Figure 13.

#### 1.3.4.1 Définition d'un Workflow

Le Workflow est une technologie informatique ayant pour objectif la gestion des processus d'organisations ou d'entreprises : les termes suivants sont également employés pour qualifier cette technologie « Système de Gestion Electronique de Processus », « Gestion de Workflow » ou « Gestion de processus » [COURTOIS 96].

Le Workflow est l'ensemble des moyens mis en œuvre pour automatiser et gérer les processus d'une organisation. Cette gestion est rendue possible par la représentation sous forme d'un modèle, de tout ou partie des processus considérés. Le Workflow doit ensuite transcrire les modèles obtenus en une forme exécutable. Enfin, ces modèles sont exécutés et gérés. Il est ainsi possible de suivre l'évolution de leur état au fil du temps. La gestion de processus inclut également, au cours de l'exécution, la coordination et la synchronisation des différents acteurs des processus en fonction de l'état actuel des modèles.

Pour résumer, la Gestion de Processus permet donc d'attribuer à chacun et au bon moment, les tâches dont il a la responsabilité et de mettre à disposition les applications, les outils et les informations nécessaires pour leurs réalisations. Dans un contexte d'acteurs humains, le Workflow permet de décharger les acteurs de certaines tâches de gestion administrative, en leur laissant la possibilité de se concentrer sur les contenus des tâches techniques en rapport avec leurs compétences. De plus, le Workflow donne la possibilité d'effectuer une activité de monitoring sur le déroulement des Workflow de l'entreprise, permettant en particulier de connaître, en fonction de la date, l'état des activités, des acteurs, des applications et quelles sont les prochaines activités planifiées.

En synthèse, La WfMC présente le Workflow comme l'automatisation d'un processus d'entreprise, en intégralité ou en partie, pendant laquelle on définit les transmissions des documents, de l'information ou des tâches d'un participant à un autre pour agir, selon un jeu de règles procédurales [WfMC11 99]. Un Système Workflow définit, gère et exécute des procédures en exécutant des programmes dont l'ordre d'exécution est prédéfini dans une représentation informatique de la logique de ces procédures - les Workflow [WfMC11 99].

#### 1.3.4.2 Méta Modèle basique

Le Workflow est basé sur un ensemble de concepts. La WfMC [WfMC11 99] a proposé un méta modèle de Définition de Procédures, qui identifie les concepts de

haut niveau dans la Définition de Processus. Ce modèle permet de mieux appréhender les concepts et leurs interrelations.

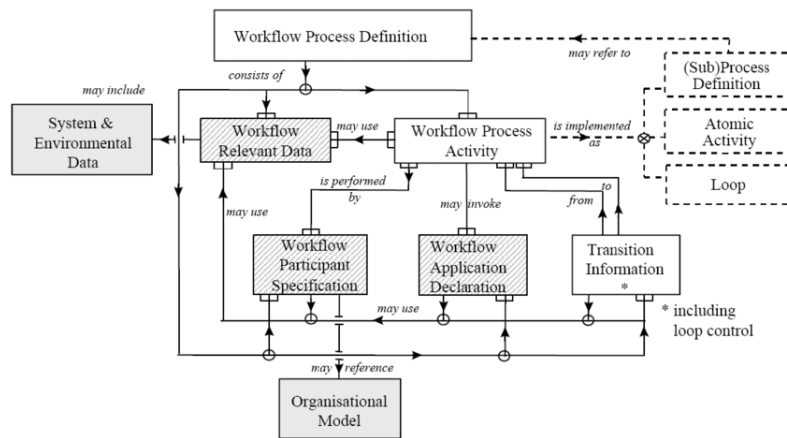


FIGURE 1.5 – Méta modèle Workflow pour la définition de Processus [WFMC11 99]

Le méta modèle, présenté Figure 1.5, identifie un ensemble d'objets fondamentaux qui entrent dans la définition d'un processus géré par un système Workflow, et que nous allons définir et commenter dans les paragraphes suivants. Remarquons que le méta modèle peut être enrichi par les développeurs de systèmes, il peut également être utilisé à des fins d'échanges entre différents systèmes Workflow.

### 1.3.4.3 Dimensions

Associé aux définitions précédentes, il a été défini une représentation tridimensionnelle des systèmes Workflow [VAN DER AALST 98a] (Figure 1.6).

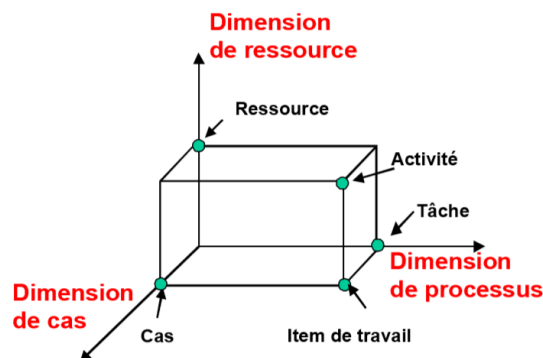


FIGURE 1.6 – Dimensions des systèmes Workflow

La Figure 1.6 donne une représentation tridimensionnelle d'un Workflow : avec une dimension de cas, une dimension de processus et une dimension de ressource. La dimension de cas représente l'instanciation du Workflow, chaque cas est un modèle à simuler et traiter individuellement. Les cas ne s'influencent donc pas directement. Ils peuvent éventuellement s'influencer indirectement via les ressources et les données. Le processus de Workflow est spécifié dans la dimension de processus, c'est-à-dire, la définition des tâches et de leur enchaînement. Enfin, dans la dimension de ressource, les ressources sont regroupées en rôles et en unités organisationnelles. En résumé,

nous pouvons donc visualiser un certain nombre de termes Workflow dans la vue tridimensionnelle de la Figure 16. En particulier, une entité de travail est définie par la coordonnée cas + tâche et une activité par le cas + la tâche + la ressource. Cette représentation met en relief la gestion de Workflow comme le lien entre les cas, les tâches et l'organisation.

### 1.3.5 Description des Systèmes Workflow

Après avoir présenté la terminologie Workflow, il est maintenant possible de présenter le principe de fonctionnement des systèmes Workflow. A un haut niveau, ce type de système est composé de trois parties fonctionnelles [WPMC03 95]. La Figure 17 illustre un Système de gestion de Workflow et les relations entre ses différentes fonctions.

#### 1.3.5.1 Build time

Cette première phase permet la définition et la modélisation des procédures Workflow, elle est nommée build time. Elle est principalement composée d'un outil permettant la modélisation des Workflow dans un formalisme existant ou propriétaire à partir d'une analyse de procédures d'entreprises, il est à noter que les outils actuels préfèrent une description graphique. Un deuxième composant transcrit les modèles obtenus dans une définition des procédures « exécutable », c'est à dire compréhensibles par la partie chargée de l'exécution (simulateur du run time). Certains systèmes permettent en retour de la partie run time des modifications dynamiques de la structure de définition de procédures comme indiqué sur la Figure 17.

#### 1.3.5.2 Run time

L'environnement d'exécution (ou run time) a l'entière gestion des modèles de Workflow établis dans la première partie. Cette gestion comprend l'exécution des Workflow avec un simulateur mais aussi la distribution des tâches aux rôles appropriés en cours d'exécution, la mise à disposition de l'ensemble des données et outils nécessaires, la supervision et le contrôle, etc.

Plus en détails, cette partie se décline en sous composants, introduits ci dessous : Le Moteur de Workflow (Workflow Engine) qui est chargé de la Gestion des Procédures à travers la simulation de leurs évolutions.

Le gestionnaire des listes de tâches (Worklist Handler), chargé de distribuer les activités dans les listes des acteurs en fonction de leurs rôles.

Les listes de tâches (Worklists), qui sont les listes associées à chaque rôle dans lesquelles le moteur place les tâches à réaliser. Les tâches sont classées par priorité ou par date avec les données et les outils à utiliser de façon appropriée.

Les outils d'administration et de contrôle (Workflow Process Monitoring) suivent le déroulement des procédures Workflow, elles peuvent fournir l'état actuel des composantes du Workflow et donner l'ordre de modifier le modèle de procédures.

### 1.3.5.3 Utilisateurs, outils et applications

La troisième phase concerne l'ensemble des outils et des fonctions d'API (Application Programming Interface) qui permettent au système Workflow de s'interfacer avec des ressources humaines, des applications informatiques extérieures et d'autres systèmes Workflow.

### 1.3.6 Concepts et Terminologie Workflow fondamentaux

Les principaux termes associés aux Workflow proposés par la WfMC [WfMC11 99] sont présentés dans le diagramme du méta modèle Workflow ci-dessus, ce diagramme permet également de mettre en évidence leurs interrelations. Les termes présentés ci-dessous en français avec la traduction anglaise originale associée, couvrent les notions plus importantes appartenant au Workflow et à son lexique [WfMC11 99].

#### 1.3.6.1 Procédure Workflow (Workflow Process)

Une procédure Workflow est une procédure contrôlée par un Workflow. Une procédure est composée de plusieurs activités enchaînées pour représenter un flux de travail. Une procédure possède une structure hiérarchique et modulaire, en l'occurrence une procédure peut donc être composée de sous procédures et d'activités. Les sous-procédures peuvent être composées elles mêmes de procédures manuelles ou de procédures Workflow.

#### 1.3.6.2 Activité (Process Activity)

Une activité est une étape d'un processus au cours de laquelle une action élémentaire est exécutée. On désigne par « action élémentaire » (ou tâche) une activité qui n'est plus décomposable en sous-procédures. La WfMC distingue une « activité manuelle », qui n'est pas contrôlée par le système Workflow, et une « activité Workflow » qui est sous le contrôle du Workflow. Un exemple d'une activité manuelle est l'ouverture d'un courrier. Une activité Workflow peut être le remplissage d'un formulaire électronique. Il existe donc des exemples d'activités manuelles intégrables dans un Workflow.

[VAN DER AALST 98a] présente l'activité Workflow comme l'intersection entre une ressource humaine ou matérielle et un bon de travail dans le cadre de l'exécution d'une tâche. Dans cette représentation, une ressource du modèle organisationnel est donc exigée pour qu'une tâche puisse être instanciée en activité et allouée à un participant de Workflow.

#### 1.3.6.3 Acteur, Ressource (Workflow Participant)

Un acteur est une entité du modèle organisationnel participant à l'accomplissement d'une procédure. L'acteur est chargé de réaliser les activités qui lui sont attribuées via le(s) rôle(s) qui lui sont définis dans le modèle organisationnel. Les autres dénominations courantes dans la littérature de cette entité sont « ressource », « agent », « participant » ou « utilisateur ». L'acteur peut être une ressource humaine ou matérielle (machine, périphérique informatique...).

Les ressources sont organisées en classes dans le modèle organisationnel. Ces classes sont des groupes de ressources possédant des propriétés communes. Une classe est basée sur :

Rôle : défini ci dans le § suivant.

Groupe : cette classification est basée sur l'organisation (département, équipe, unité).

#### **1.3.6.4 Rôle (Role)**

Un rôle décrit en général les compétences d'un acteur dans le processus ou sa position dans l'organisation. Un rôle est associé à la réalisation d'une ou de plusieurs activités. Plusieurs acteurs peuvent tenir un même rôle. La WfMC distingue deux types de rôles [WfMC11 99] :

Les rôles organisationnels définissent un ensemble de compétences qu'un acteur possède. Ce rôle définit la position de l'acteur dans une organisation. Les rôles procéduraux définissent une liste d'activités qu'un acteur est en capacité d'exécuter.

Il est à noter que certains travaux ne différencient pas les notions d'acteur et de rôle et ne parlent que d'acteur. Cette opinion semble restreindre la clarté et la flexibilité des modèles Workflow.

#### **1.3.6.5 Données (Workflow Relevant Data)**

Une donnée pertinente pour les procédures est une information en rapport avec la réalisation des activités (en définition de la tâche, en entrée ou en sortie). Elle peut constituer l'objectif d'une tâche (manipulation de la donnée et définition de l'état de la procédure), être un élément essentiel pour activer les transitions d'état d'une instance Workflow ou être généré par la tâche et ainsi intervenir dans la détermination de la prochaine activité à déclencher. Ces données sont en général des objets au sens purement informatique mais peuvent également être une représentation d'objets physiques.

Notons qu'il existe deux autres types de données utilisées hors de la gestion de procédures :

Donnée de contrôle (Control Data) : données gérées et utilisées par le système Workflow et les moteurs Workflow.

Données Applicatives (Applicative Data) : données propres aux applications, le système de gestion de Workflow n'y a pas accès.

#### **1.3.6.6 Application externe (Invoked Application)**

Une application externe est une application informatique dont l'invocation est nécessaire à la réalisation de la tâche ou à l'exploitation des résultats générés avant de déclencher la tâche suivante ou de recommencer cette première. On tiendra compte de l'allocation de ressources, si l'application n'est pas uniquement informatique. Il faut différencier les outils (Tools), qui sont eux directement interfacés par le système Workflow, sans l'intervention d'une ressource du Système Workflow.



## 1.4 Classification des systèmes Workflow

Il n'existe pas de classification commune des systèmes Workflow dans la littérature, reconnue par l'ensemble de la communauté Workflow [VAN DER AALST 02]. Ceci étant essentiellement dû au nombre important de critères de classification qu'il est possible de retenir. En effet, les spécialistes adoptent différents points de vue par rapport à la notion de Workflow, les critères qui en découlent varient donc en fonction de leurs perceptions des caractéristiques présentées par ces systèmes Workflow. Ainsi, il existe plusieurs classifications, permettant de sélectionner un outil de gestion de Workflow avec différents « éclairages » sur le sujet.

Malgré ce manque d'unité, la classification proposée par [McCREADY 92] est assez répandue dans la littérature, elle est reprise par bon nombre d'auteurs [VAN DER AALST 98a], [GEORGAKOPOULOS 95]. Elle propose de distinguer quatre catégories de Systèmes Workflow. La Figure fig :classification-de-workflow présente ces différentes classes selon deux axes : Approche et Structure.

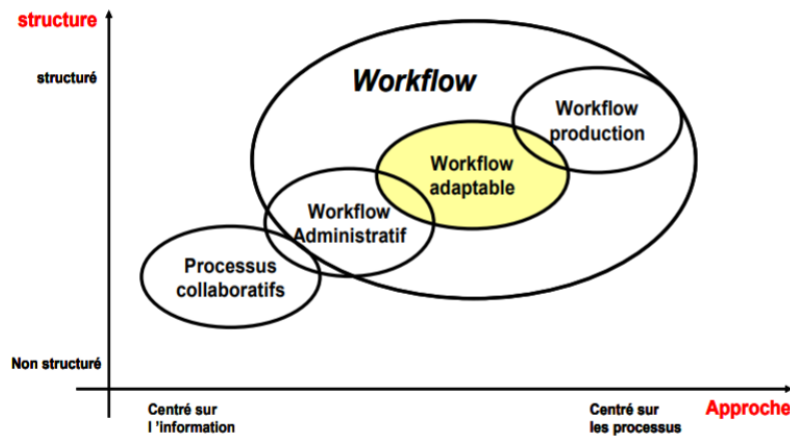


FIGURE 1.7 – Différentes classes des systèmes Workflow

### 1.4.1 Processus collaboratifs

Cette première classe est axée sur la communication et sur le partage d'information. Les systèmes collaboratifs sont définis pour supporter le travail en groupe, dans le cadre de la conception, de la gestion de projet ou de la résolution de problèmes faisant appel à plusieurs niveaux d'expertise. Ces systèmes permettent de réunir les intervenants d'un projet autour d'un objectif commun, les clients de la procédure y étant souvent eux-mêmes directement associés, les logiciels employés sont le plus souvent des groupwares [27]. Les tâches des procédures gérées sont le plus souvent complexes et leur réalisation implique l'intervention de ressources aux compétences très spécifiques pour une forte valeur ajoutée. D'un autre côté, l'enchaînement des activités des procédures à traiter est faiblement structuré et peu répétitif. De part la faible structure de ces processus, ils ne font pas partie ou se situe à la frontière de ce que l'on considère comme la « sphère » Workflow comme le précise la Figure 1.7.

### 1.4.2 Workflow administratif

Les systèmes Workflow administratifs (General Purpose Workflow Management Systems) ont pour objectif de décharger les ressources d'une entreprise des tâches administratives. En effet ces procédures sont répétitives, fortement prédictibles et les règles d'enchaînement des tâches sont très simples et clairement définis; ces procédures sont donc aisément automatisables, évitant ainsi un travail fastidieux où peuvent naître des erreurs souvent humaines. Les systèmes Workflow administratifs permettent de lier à une tâche administrative, les documents et les informations nécessaires à la réalisation de cette tâche par un acteur humain. Ces systèmes gèrent également le routage des documents et le remplissage de formulaires. La gestion par Workflow de procédures administratives permet un gain de l'ordre de 5% à 10% en termes de productivité et de 30 à 90% en termes de délais [ADER 99]. Enfin une dernière raison de l'automatisation de ce type de procédures en Workflow provient du fait que ces procédures possèdent une structure statique et ne sont donc pas souvent assujetties à modifications car elles possèdent une longue durée d'utilisation [SCHEER 97].

### 1.4.3 Workflow de production

Les systèmes Workflow de production impliquent des procédures prévisibles et assez répétitives. Leurs principales différences avec les Workflow administratifs résident dans la complexité des tâches et de la structure des procédures, dans leur capacité à faire appel à des informations provenant de systèmes d'information variés et dans l'enjeu que représente leur réussite. En effet, la procédure Workflow correspond directement au travail effectué par l'entreprise. En d'autres termes, la performance de l'entreprise est directement liée à l'exécution de la procédure managée par le Workflow. On dit dans ce cas qu'il est mission critical<sup>28</sup> [INCONCERT 97]. C'est par exemple le cas des organismes financiers, des compagnies d'assurances, des usines de production manufacturières. La réalisation des procédures est donc associée à une forte valeur ajoutée et un volume d'informations traitées important. La complexité des procédures traitées est également due à la répartition de leurs activités sur plusieurs sites. Dans ce cas, les tâches exécutées nécessitent souvent l'interrogation de plusieurs systèmes informatiques, hétérogènes et distribués. Il est donc nécessaire que les systèmes Workflow de production fournissent un ensemble d'outils ou de fonctions d'API permettant de se connecter à plusieurs systèmes. Enfin, même si les procédures traitées sont assez répétitives, elles sont susceptibles d'être modifiées plus souvent que les procédures administratives, car associées à la modification des objectifs du métier. Ces modifications peuvent par exemple avoir lieu dans le cadre d'une restructuration de BPR<sup>29</sup> ou d'un CPI<sup>30</sup>.

Les systèmes Workflow de production doivent donc pouvoir évoluer. Par ailleurs, l'exécution de certaines procédures ne peut pas toujours se poursuivre de manière automatisée, suite à l'occurrence d'un ou de plusieurs événements qui font aboutir le système dans un état particulier. Dans ce cas, il est nécessaire de faire intervenir des acteurs humains pour la prise de décision. Pour ce faire, le système Workflow de production peut faire appel à un autre système, de type collecticiel ou un autre système Workflow ad hoc, qui servira d'interface pour l'exécution dirigée par un acteur humain de la suite de la procédure. Ce type de Workflow est dit « composite » [EDER 96]. Enfin, dans la littérature, les systèmes Workflow de production sont

également appelés case-based [VAN DER AALST 98a].

#### 1.4.4 Workflow adaptable ou Workflow ad hoc

L'impossibilité pour les systèmes de gestion de Workflow traditionnels de traiter les différents changements dynamiques dans les flux de travail est une limite à dépasser. A ce titre, il a été introduit les concepts de Workflow adaptable (adaptive Workflow) [VAN DER AALST 98c] et de Workflow ad hoc [VOORHOEVE 97]. La nuance entre ces deux nouveaux termes provient du fait qu'ad hoc désigne un acte spécialement fait pour un objet déterminé alors qu'adaptable prévoit un changement définitif de la procédure.

Les Workflow ad hoc se situent à la frontière gauche de la représentation Figure 1.7 dans la « sphère » Workflow adaptable. Ils régissent des procédures dont la structure est déterminée pendant l'exécution en fonction des décisions humaines prises suite à la réalisation d'une tâche, plus concrètement, la structure se construit par pas en suivant le rythme de l'exécution. En effet, la réalisation d'une procédure non structurée peut impliquer à chaque fois l'exécution d'un nouvel enchaînement des tâches, voire la création de nouvelles tâches. Il n'y a pas a priori de persistance de l'enchaînement de ces tâches.

Les Workflow adaptables sont, quant à eux, des supports comparables aux Workflow de production classiques possédant une structure préétablie, mais pouvant traiter certains changements de structure « en ligne ». Ces changements peuvent aller des changements individuels/ad hoc (gestion d'exception), c'est à dire d'un aiguillage pour déterminer l'activité suivante, jusqu'à la reconception par BPR de processus [VAN DER AALST 98d]. En conclusion, ils ont une action globale pouvant inclure la définition du Workflow ad hoc

Il est intéressant de classer les différents changements possibles par un Workflow adaptables, dans le but de mieux les anticiper [SADIQ 99]. Les changements sont envisageables selon plusieurs perspectives : la ressource, le contrôle, la procédure, la tâche et le système [VAN DER AALST 98d]. Dans la suite de l'étude seul l'aspect procédure sera développé, c'est en effet la perspective dominante du management par Workflow et elle comporte un aspect important : les changements dynamiques. Nous présentons ci-dessous les différents types de changements envisageables.

##### 1.4.4.1 Le changement individuel (ad hoc)

Les systèmes Workflow ad hoc sont utilisés pour l'exécution de processus non structurés ou peu structurés (sujets à changement). Un processus peu ou non structuré est un processus dont l'ordre et le temps exact de réalisation des tâches ne sont pas établis au préalable et/ou peuvent être modifiés pendant l'exécution. Les choix de routage et la nature des tâches sont décidés au fur et à mesure de l'exécution. Par conséquent, un processus non structuré propose un objectif immuable, mais pouvant être atteint de différentes façons. Certaines situations rencontrées pendant le « run time » nécessitent donc des dérivations ad hoc dans la procédure, éventuellement planifiée, comme le proposent [HAN 98], telles que :

##### 1. Le raffinement Dynamique :

Dans certains cas, il est impossible ou peu pratique de définir une spécification complète du modèle de Workflow. En raison de l'indisponibilité d'une spécification

complète, le raffinement dynamique peut être nécessaire pendant le « run time », c'est-à-dire que certaines tâches ne seront complètement et définitivement spécifiées qu'en « run time ». Le même raisonnement peut être appliqué pour définir les ressources exigées pour l'exécution d'une tâche.

## **2. La participation d'Utilisateurs :**

Au lieu d'être des contrôleurs passifs, certains utilisateurs d'un système de Workflow doivent être traités comme des « propriétaires d'une tâche ou d'une procédure » [HAN 98]. L'approche de ces systèmes est souvent de type « pull », c'est-à-dire que leurs utilisateurs doivent les interroger pour connaître l'état du processus et en déduire leurs tâches ; par opposition, les autres types de systèmes, possède eux une approche plutôt de type « push », où les utilisateurs sont informés par le système des travaux qu'ils ont à traiter [GEORGAKOPOULOS 95]. Techniquement, la métaphore utilisée dans ce type de système est celle du « dossier » [WAINER 95]. Les utilisateurs font circuler un dossier virtuel dans lequel sont « placés » des documents et des données électroniques. Chaque utilisateur en possession du dossier décide du prochain destinataire. Le processus décisionnel de l'utilisateur doit être considéré dans l'exécution de processus de Workflow.

## **3. Adaptation aux événements externes :**

Des événements non pris en compte par le modèle de Workflow, y compris certains stimuli externes, l'intervention d'utilisateurs, les temps morts, etc., doivent être traités correctement pour résoudre les problèmes du monde réel et faciliter la communication entre des procédures Workflow différentes. En outre, une fois qu'une communication inter-Workflow a lieu, les utilisateurs ou les propriétaires de procédures doivent être capables de répondre à ces événements en raffinant dynamiquement leur procédure ou en modifiant la tâche actuelle et/ou les interdépendances de tâches.

## **4. Situation d'échec :**

Un défaut système, des conflits de ressource et des fausses opérations peuvent causer des erreurs et des difficultés dans l'exécution d'une procédure Workflow. Les mécanismes pour traiter les situations d'erreur sont donc très importants pour assurer l'amélioration des processus de Workflow.

### **1.4.4.2 Le changement structurel (évolution)**

Ces changements sont souvent une réaction pour s'adapter à un changement d'environnement dû au contexte concurrentiel très dynamique ou au besoin d'adaptation aux progrès technologiques, au travers la parution de nouveaux logiciels ou de nouvelles versions [HAN 98].

Ces changements sont souvent le fruit d'un travail de BPR.

Après une telle modification, il existe plusieurs possibilités [VAN DER AALST 98d] et [SADIQ 99] d'intégrer les cas existants dans la nouvelle procédure, contrairement aux changements ad hoc qui restent un traitement d'exceptions et sont gérés individuellement.

Première possibilité : Redémarrer « restart » « abort »

Les cas en cours de traitements dans l'ancien processus sont remis à zéro et redémarrés dans le nouveau processus au lancement du nouveau système.

Deuxième possibilité : Parallèle « proceed » « flush »

Le système conserve en parallèle l'ancien et le nouveau processus le temps de l'exécution des cas en cours sur l'ancien processus.

Troisième possibilité : Transférer « transfer » « migrate »

Ce changement n'affecte pas le traitement des cas qui sont transférés directement dans le nouveau processus dans l'état actuel de leur déroulement.

### 1.4.5 Modèle de référence des systèmes Workflow

Le modèle de référence, Figure 18, présente l'architecture générale de l'environnement proposée par la WfMC, il identifie les interfaces couvrant cinq domaines de fonctionnalités entre le système Workflow et son environnement.

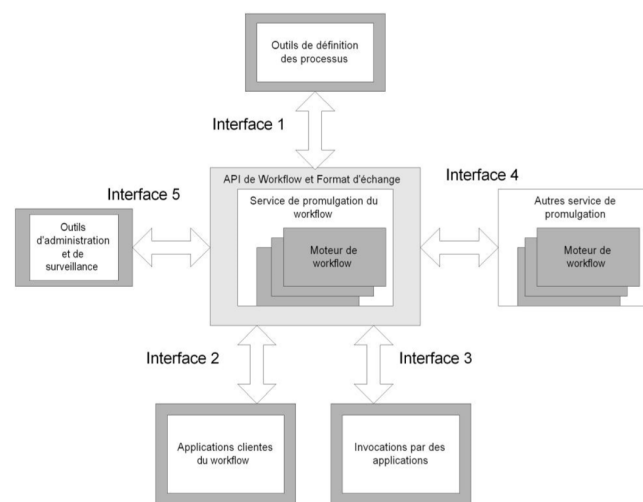


FIGURE 1.8 – Modèle de référence des systèmes de gestion de workflow (WfMC, 95).

#### 1.4.5.1 Interface avec les Outils de définition de procédures

Cette interface, située entre les outils de modélisation/définition et le logiciel de gestion du Workflow pendant l'exécution, est nommée interface d'import/export de définition de processus. Cette interface définit le format d'échange et d'appels des APIs, qui permettent l'échange d'informations de définition de procédures sur une variété de médias d'échange : physiques ou électroniques. Cette interface permet l'échange d'une définition de processus complète ou d'un sous-ensemble. Par exemple le changement de définition d'un ensemble de procédures ou plus simplement la modification des attributs d'une activité particulière dans une définition de procédures.

#### 1.4.5.2 Interface avec les applications clientes Workflow

La liste des tâches (Worklist) à exécuter par une ressource est généralement définie et gérée par le service d'exécution du Workflow. Cette liste doit pouvoir déclencher des appels à des applications clientes diverses et des ressources. La solution retenue pour respecter la susdite exigence, consiste à encapsuler la variété

d'application qui peut être utilisée derrière un jeu standard d'API (le WAPI Workflow Application Programming interface). Ce jeu permet ainsi d'utiliser une communication standardisée entre les applications clientes, le moteur de Workflow et les Worklist, indifféremment de la nature de l'implémentation réelle des produits clients.

#### 1.4.5.3 Interface avec les applications invoquées

Il est évident que le système Workflow ne peut pas intégrer l'invocation automatique de toutes les applications qu'il peut être amené à utiliser pendant l'exécution d'un Workflow. Par exemple les applications dont les données sont fortement typées. Dans ce cas un composant externe supplémentaire, nommé agent d'application, est ajouté, il est chargé de la traduction des informations dans un format compréhensible par le standard WAPI.

Dans le cas le plus simple, l'invocation d'application est traitée localement par un moteur de Workflow, mais les applications invoquées peuvent être utilisées par plusieurs moteurs de Workflow et peuvent se situer sur des machines distantes, il convient donc de définir un format commun d'utilisation des ces applications entre les Workflow dans le but de communiquer correctement et de synchroniser l'appel à ces applications.

#### 1.4.5.4 Interface avec les autres Workflow

Un des objectifs de la normalisation dans la définition de Workflow est de pouvoir transmettre des WorkItem entre deux systèmes Workflow conçus par des concepteurs de systèmes Workflow différents. Trois principaux types d'interopérabilité ont été identifiés :

**Workflow chaînés** : La dernière activité d'un Workflow A doit pouvoir fournir un item à la première activité d'un Workflow B.

**Workflow hiérarchiques** : une activité d'un Workflow A doit pouvoir être vu comme un Workflow B.

**Workflow Peer to Peer** : Une procédure globale est composée d'activités gérées en partie par un Workflow et en partie par un autre Workflow, sans système de supervision de la procédure complète.

**Workflow Synchronisés** : Deux Workflow s'exécutent en parallèle et doivent pouvoir se synchroniser sur certaines activités.

Pour résumer, il est possible d'identifier deux aspects principaux nécessaires à l'inter fonctionnement de Workflow :

- L'interprétation commune de la définition de procédures (ou d'un sous-ensemble).
- L'appui pendant l'exécution de l'échange des divers types d'information de contrôle et le transfert des données appropriées et/ou d'applications entre les services d'exécution Workflow différents.

#### 1.4.5.5 Interface avec les outils de contrôle et d'administration

L'objectif de cette interface est de permettre à un logiciel de Monitoring de Workflow de s'interfacer avec plusieurs Workflow différents et ainsi regrouper la supervision d'un ensemble de systèmes Workflow dans un logiciel.

L'interface 5 permet à une application de gestion indépendante d'interagir avec des Workflow de différents domaines. L'application de gestion peut aussi se charger d'autres fonctions de gestion, au-delà de celles-ci. Par exemple, elle peut aussi gérer des définitions de procédures de Workflow, agissant comme un dépôt d'information commun à plusieurs systèmes et distribuant des définitions de processus aux divers Workflow via des opérations au travers de leurs interfaces 1. Malgré cela, des scénarii d'implémentations moins modulaires sont aussi envisageables ; par exemple l'application de gestion peut être une partie intégrante du service d'exécution.

## 1.5 Architecture des systèmes de gestion de workflows

### 1.5.1 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 1.8. Ce modèle inclut un service de déploiement, qui contrôle l'exécution des workflows et qui supporte cinq interfaces standardisées :

### 1.5.2 Standards utilisés dans les SGWf :

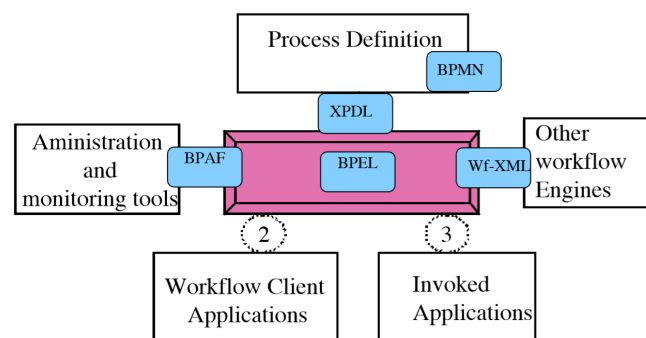


FIGURE 1.9 – Différents standards adoptés dans les SGWf

#### 1.5.2.1 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).

#### **1.5.2.2 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.

#### **1.5.2.3 BPAF (Business Process Analytics)**

fournit aux participants aux processus et aux décideurs des informations sur l'efficacité des processus organisationnels.

#### **1.5.2.4 BPEL (Business Process Execution Language)**

BPEL est un langage d'orchestre.

#### **1.5.2.5 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.

## **1.6 Intérêt du cloud pour les workflows**

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

### **1.6.1 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.

### **1.6.2 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)



### 1.6.3 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.

### 1.6.4 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.

### 1.6.5 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.

## 1.7 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 Modélisation de Workflow adaptable**

Dans le cas particulier de la modélisation des Workflow adaptables, plusieurs approches sont proposées [HAN 98].

#### **2.2.2.1 Approche Meta modèle**

Cette approche emploie un méta modèle pour déterminer la structure et les types de composants constitutifs ; elle définit un jeu de primitives pour transformer un modèle Workflow générique ou une instance de modèle.

#### **2.2.2.2 L'approche de point ouvert Open-point**

Elle définit des points spéciaux dans un modèle de Workflow, où l'adaptation peut être faite. Cette approche inclut la mise à disposition de choix multiples pour les utilisateurs, l'allocation dynamique de certaines ressources aux tâches en run time, ou une interface ouverte par laquelle une « dernière modélisation » peut être réalisée. La « dernière modélisation » signifie que, pendant l'exécution des modèles complets, certains sous modèles peuvent être définis dynamiquement pour une utilisation immédiate.

### 2.2.2.3 Approche synthétisée

Il semble que les deux approches puissent être employées complémentaires pour la définition de Workflow adaptable. Pour cela, les deux hypothèses suivantes doivent être prises en compte :

Tout d'abord, il paraît nécessaire de distinguer les évolutions de procédures Workflow et les changements ad hoc dynamique dans une procédure. De plus, une séparation claire est faite entre le build time et le run time en termes de Workflow. Il paraît impératif de faciliter le franchissement bilatéral de cette frontière pour rendre modélisables les Workflow adaptables.

En synthèse, il est souhaitable de définir la frontière entre les changements ad hoc et les évolutions du Workflow. Il semblerait que ces changements devraient être testés sur une instance de modèles puis si nécessaires propagés au niveau global pour une modification persistante.

### 2.2.3 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 comme le précise [VAN DER AALST 97a] [VAN DER AALST 97b].

Il existe un certain nombre de techniques de modélisation des procédures, tels que : Pour la représentation en terme de fonctions, les diagrammes de flots de données DFD31, ISAC32, SADT33 semblent tout à fait indiqués.

Pour l'aspect comportement, les diagrammes état-transition (par extensions : state charts) semblent convenir plus particulièrement. Mais aussi les organigrammes (Flowcharts) qui illustrent les pas dans une procédure. En visualisant les procédures, un organigramme peut rapidement aider à identifier des goulots d'étranglement ou des inefficacités dont la procédure peut être rationalisée ou améliorée.

Concernant l'aspect représentation des données, on pourra utiliser par exemple un Modèle Conceptuel de Données de la méthode MERISE [TARDIEU 85] ou un modèle de classes d'une méthode Objet UML34. Le modèle informationnel peut également représenter le flux des informations, c'est-à-dire leur circulation et les différents états qu'elles peuvent prendre ainsi que les dépendances hiérarchiques.

Il existe également des techniques de diagramme spécifiques aux concepteurs de logiciels utilisées dans les outils de simulation WfMS. Ces outils de modélisation et simulation spécifiques intègrent un outil modélisation, le plus souvent dans un langage graphique et un simulateur également spécifique se basant sur un formalisme particulier après transformation dans ce dit format simulable, par exemple un automate à états finis.

Mais ce sont certainement les réseaux de Petri de haut niveau qui sont les plus utilisés à l'heure actuelle pour représenter et simuler les comportements des procédures d'un Workflow autour des nombreux travaux de Van der Aalst [VAN DER AALST 98b].

Cependant, jusqu'alors pas, ou peu, de travaux de spécification et de modélisation de la partie processus d'un Workflow ont été réalisés en utilisant la clarté et la précision du formalisme DEVS. Ce formalisme offre néanmoins un aspect modulaire hiérarchique important dans la définition d'un Workflow. De plus leur simulation par événements discrets permet une exécution rapide. Nous reviendrons sur ce point dans le chapitre 4.

### 2.2.3.1 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 conservant 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.

### 2.2.3.2 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 informes 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.

### 2.2.3.3 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

## 2.3 Les réseaux de Petri

Les réseaux de Petri (Petri, 1962) sont une généralisation des automates à 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.

### 2.3.1 Définition

**Définition 2.3.1. (Réseau de Petri simple) :**

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

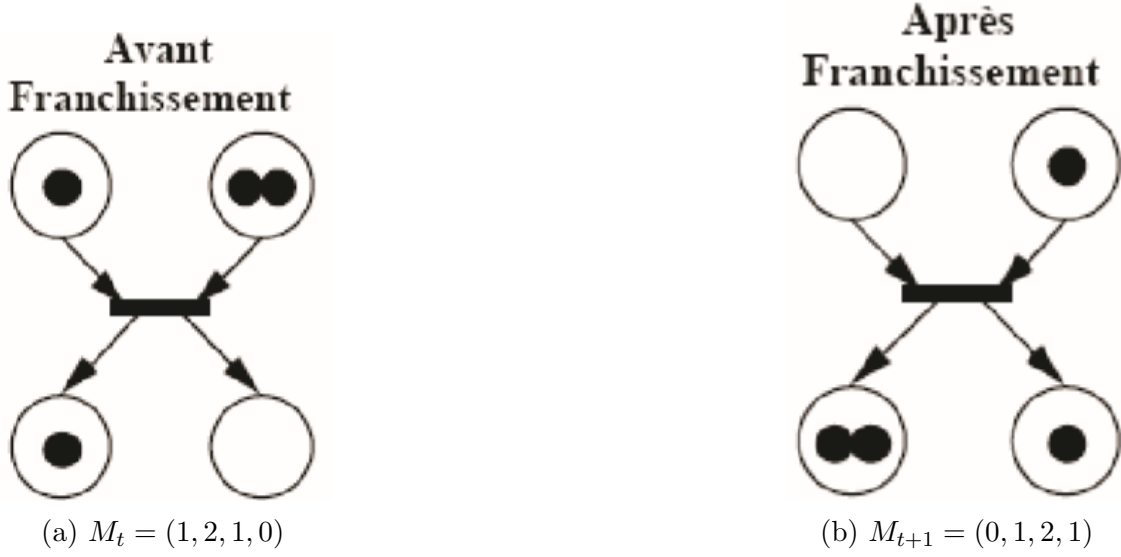


FIGURE 2.1 – Opération de franchissement

- $P$  est un ensemble final de places,
- $T$  est un ensemble final de transition ( $P \cap T = \emptyset$ ),
- $\text{Pre} : P \times T \rightarrow \mathbb{N}$  est la fonction d'incidence avant,
- $\text{Post} : P \times T \rightarrow \mathbb{N}$  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  $T_1$

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.

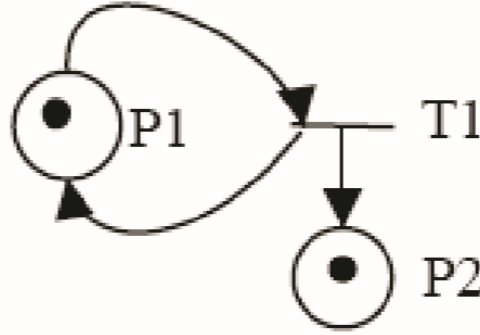


FIGURE 2.2 – Réseau de Petri non borné

**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.

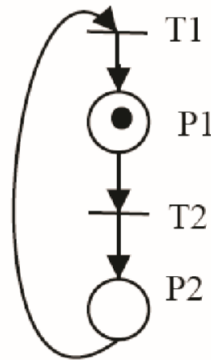


FIGURE 2.3 – Réseau de Petri vivant

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



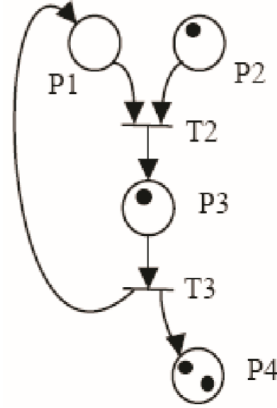


FIGURE 2.4 – Exemple de réseau de Petri bloquant

**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 :

- $\Delta$  : 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  $\Delta$ .
- $E$  : est l'application qui associe à chaque arc, un élément de  $Couleur(p)MS$  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  $Couleur(p)MS$ .  $M_0(p)$  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  $Couleur(p)MS$ .  $M(p)$  est un multi-ensemble sur  $Couleur(p)$  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).
- EPSI (2019). *PALMARÈS 2019 DES FOURNISSEURS DU CLOUD*. URL : <http://www.epsi.fr/fournisseurs-cloud-2019/>.
- 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](http://www.uml.org).
- PETRI.NETS.WORLD (2004). *Online Services for the International Petri Nets Community*. URL : [www.daimi.au.dk/PetriNets](http://www.daimi.au.dk/PetriNets).
- S. JOOSTEN, S. Brinkkemper (1996). *Fundamental Concepts for Workflow Automation in Practice*. International Conference on Information Systems Development.